

Metode Segmentasi Citra Foto Pada Biodata Menggunakan
Fuzzy C-Means Cluster



Tgl. Terbit	
Tgl. Pengantar	24-2-10
Kejurusan	MIPA
Kejurusan	1413
Marga	Arung
No. Inventaris	38
No. Klas	SKR-MPTD

AGUNG ARUNG LABI

H 111 05 030

LAB
a

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR

2010

**METODE SEGMENTASI CITRA FOTO PADA BIODATA
MENGUNAKAN *FUZZY C-MEAN CLUSTER***

S K R I P S I

*Diajukan sebagai syarat untuk memperoleh gelar Sarjana Sains pada Jurusan
Matematika Fakultas Matematika dan Ilmu pengetahuan Alam
Universitas Hasanuddin
Makassar*

OLEH :

**AGUNG ARUNG LABI
H 111 05 030**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2010**

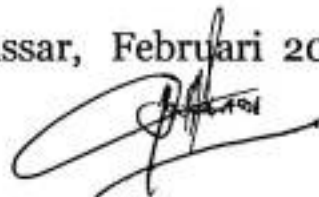
LEMBAR KEOTENTIKAN

*Saya yang bertanda tangan di bawah ini menyatakan dengan
sesungguhnya-sungguhnya bahwa skripsi yang saya buat dengan judul:*

METODE SEGMENTASI CITRA FOTO PADA BIODATA MENGGUNAKAN FUZZY C-MEAN CLUSTER

*Adalah benar hasil kerja saya sendiri, bukan hasil plagiat dan
belum pernah dipublikasikan dalam bentuk apapun.*

Makassar, Februari 2010



AGUNG ARUNG LABI
NIM: H 111 05 030

**METODE SEGMENTASI CITRA FOTO
PADA BIODATA MENGGUNAKAN
FUZZY C-MEAN CLUSTER**

Disetujui Oleh :

Pembimbing Utama



Dr. Armin Lawi, M.Eng
NIP. 19720423 199512 1 001

Pembimbing Pertama

Pembimbing Kedua



Hendra. S.Si, M.Kom
NIP. 19760102 200312 1 001

Drs. Diaraya, M.Ak

NIP. 19631231 198702 1 011

Pada tanggal: Februari 2010

Pada hari ini, Senin 15 Februari 2010, Panitia Ujian Sidang Skripsi menerima dengan baik skripsi yang berjudul :


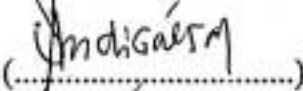


**METODE SEGMENTASI CITRA FOTO PADA BIODATA
MENGUNAKAN *FUZZY C-MEAN CLUSTER***

Yang diajukan untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Sains Jurusan Matematika Program Studi Statistika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

Makassar, 15 Februari 2010

Panitia Ujian Skripsi

Tanda Tangan

- | | | |
|---------------|-------------------------------------|---|
| 1. Ketua | : Dr. Loeky Haryanto, M.Sc., M.A.T. | () |
| 2. Sekretaris | : Andi Galsan Mahie, S.Si., M.Si. | () |
| 3. Anggota | : Dr. Armin Lawi, M.Eng. | () |
| 4. Anggota | : Hendra, S.Si., M.Kom. | () |
| 5. Anggota | : Drs. Diaraya, M.Ak. | (.....) |

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas limpahan rahmat dan kasih sayang-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul : **“Metode Segmentasi Citra Foto Pada Biodata Menggunakan Fuzzy C-Mean Cluster”** sebagai salah satu syarat untuk mendapatkan gelar Sarjana Sains.

Tulisan ini dibuat dengan segala daya upaya serta segala kemampuan yang penulis miliki. Selama penulisan dan penyusunan tugas akhir ini, penulis banyak melibatkan berbagai pihak untuk memberikan bantuan pemikiran, bimbingan, kritikan serta petunjuk-petunjuk yang sangat bermanfaat bagi penulis.

Oleh sebab itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada orang-orang yang paling penulis sayangi di muka bumi ini, Ayahanda **Kassa Arung Labi** dan Ibunda **Kadek Suparni** yang telah membesarkan dan mendidik penulis dengan penuh kesabaran dan dengan limpahan cinta dan kasih sayang serta dengan ketulusan hati memberikan dukungan dan doa yang tak ternilai harganya demi keberhasilan penulis selama menjalani proses pendidikan. Demikian juga buat kakakku tercinta **Marthen Tangke Salu** dan **Litha Arung Labi** serta ketiga ponakanku **Yoel, Ote’ dan Ima**, yang selalu memberikan semangat, bantuan materi, dan kehangatan dalam keluarga. Biarlah semua yang diberikan itu, Tuhan yang membalasnya. Buat adekku **Kasnia Arung Labi (Nia)** terimakasih atas motivasi, nasehat-nasehatnya dan dukungan lainnya yang tidak sempat penulis paparkan di sini. Biarlah semua itu Tuhan yang membalasnya, tetap semangat dalam menjalani masa-masa praktiknya, hingga akhirnya boleh selesai tepat pada waktunya.

Ucapan terima kasih yang tiada terhingga kepada:

1. **Bapak Dr. Armin Lawi, M.Eng** selaku pembimbing utama yang selalu memberikan rasa optimis pada penulis dalam menghadapi semua masalah. dan **Bapak Drs. Diaraya, M.Ak** selaku pembimbing kedua yang telah mengajar penulis arti dari sebuah kesabaran dalam menghadapi suatu masalah hingga setiap kesulitan yang penulis hadapi dapat teratasi.
2. Secara khusus buat **Bapak Hendra, S.Si, M.Kom** selaku pembimbing pertama yang telah memberikan banyak bantuan dari awal penyusunan skripsi ini hingga selesainya. Terima kasih buat semangat yang telah diberikan saat mental penulis dalam keadaan *down*, dan masih banyak lagi yang tidak dapat penulis paparkan satu per satu, semoga semuanya itu bernilai ibadah di sisi Tuhan, dan mendapat pahala yang setimpal di sisiNya.

3. Bapak **Drs. Khaeruddin, M.Sc** selaku Ketua Jurusan Matematika atas arahan dan nasehatnya selama masa perkuliahan penulis, **Bapak Andi Kresna Jaya, S.Si, M.Si**, selaku Sekretaris Jurusan Matematika sekaligus Penasehat Akademik bagi penulis, yang senantiasa memberikan dorongan dan semangat buat penulis, sebagai tempat menyampaikan segala masalah yang dihadapi selama masa kuliah dan masih banyak lagi, demikian juga para **Dosen Jurusan Matematika** yang telah memberikan bekal ilmu dan pengetahuan yang sangat berharga dan para Staf Akademik serta Staf Jurusan Matematika yang telah banyak membantu dalam mengurus segala keperluan yang mendukung studi penulis dan telah memberikan bantuan yang tak terhingga selama penulis menjalani perkuliahan sampai ujian akhir "*Terima kasih atas kerjasamanya Pak/Bu*".
4. **Bapak Dr. Loeky Haryanto, M.S., M.Sc. M.A.T** selaku ketua penguji, terima kasih atas masukan dan arahan selama penulis menyusun skripsi ini, dan **Bapak Andi Galsan Mahie, S.Si., M.Si.** selaku sekretaris penguji, terima kasih buat metode pengajaran SCL yang sangat baik, selama penulis mengikuti perkuliahan, penulis berharap semoga metode yang bapak gunakan dapat juga diterapkan oleh dosen-dosen lain di Fakultas MIPA.
5. Teman-teman "**Pondok Fortuna**", "*Athye, Anthi, Esther, Iche', Sherly dan kandax, Vhemy, Nitha, Tyson, Nanu, Kanda Ewink, Hasnah, Etha' Nike, MelQ, Riel, Kanda Olant, Nella, Mika, Robi, Leo-Teo, Forman, Heri, Urip, Marthen-Luther, dan Rhein* terima kasih, buat setiap canda-tawa kalian, setiap bantuan makan-minumnya (*Benalu Pondokan*) dan kebisingan musik fortuna memang tiada duanya. Kalau pondokan kita digusur, dan tinggal Batu Nissannya, ingatlah bahwa kita semua pernah tinggal di tempat itu...!!!
6. Teman-teman "**PKPK Filadelfia Makassar**", terima kasih buat kebersamaan di dalam pelayanan dan hal-hal lainnya yang tidak dapat penulis paparkan satu per satu, serta teman-teman "**GMKI Kom. MIPA**" terima kasih buat kesempatannya dalam tahun ini, untuk menjadi pengurus. Biarlah pelayanan kita di dalam Tuhan, boleh menumbuhkan Iman dan Kasih.
7. Spesial Thanks buat **Kak Afrain, S.Si** yang sudah memberikan penulis fasilitas Internetan Gratis di Warnet Bless, meminjamkan laptopnya buat asistensi, dan yang telah mencarikan penulis kerja sambilan... Semoga segala kebaikan kakak, Tuhan yang membalasnya.
8. Senior-seniorku, **Angk '01, Angk '02, Angk '03, Angk '04** dan Junior-juniorku **Angk '06, Angk '07, Angk '08 dan Angk '09** "*Terima Kasih atas dukungan dan kebersamaannya*"
9. Teman-temanku "**ORDINAT 05**", Ambang, Imin dan Alan (*semangat yah bro, thanks buat fasilitas Lab yang disediakan, spesial buat Imin: klo programnya gak bisa di RUN, lebih baik input manual saja supaya cepat S.Si, Saya duluan ke Baruga yah... Ntar kalian nyusul*), Alam, S.Si. (*Thanks buat arahanx dalam menghadapi pembimbing dan setiap snack di tempat kost ta')* Cholish & Chali (*Thanks sudah mau jadi guru privat renang yang baik*) Chundink (*Thanks sudah mw ngajak traveling ke Jeneponto dan sekitarnya liat-liat Kuda, semangat yah bro.. semoga cepat S.Si*), Octo, Ucup dan Erick (*Thanks buat setiap canda tawa yang selalu dihadiran, semangat, dan rasa persodaraan di antara kita, saya doakan semoga cepat S.Si yah...Tetap*

semangat dalam bangku kuliah, jangan mudah kendor), Hidar (Saya tunggu janjimu bro...!!), Alman (Tentor muda yg optimis, Lanjutkan perjuanganmu Bro.), Anchu (anak teknik sipil klo dapat proyek, ajak-ajak donks), Inna, S.Si (Lanjutkan perjuanganmu dalam mencari soulmate!!), Ijha, S.Si & Imma, S.Si (Thanks buat nasehatnya, arahannya dan masih banyak lagi.), Dewi Sagita, S.Si (Kapan undanganx neeh??), Amhe, S.Si (Tawwa penyiar radio... Btw Frekuensinya berapa Mbak??), Erlin, S.Si (Thanks buat semua bantuannya selama kuliah, ndak ada lagi yang temani ka' jalan kaki ke pintu I inie), Iznha, S.Si (Thanks buat semangat yang diberikan sobat, boncengan di MIO, dan arahan di setiap masalah-masalahQ, Zha ndak pernah lupa itw !!), Fith, S.Si (Thanks buat segala bantuannya selama kuliah pren, rasa solidaritas selama di Campus merah dan Printernya untuk skripsi ini) Eki, S.Si (Bunda.. Thanks buat setiap solusi-solusi yang diberikan atas masalah yg saya hadapi, walaupun kita berjauhan, namun bunda tetap di Hatiku...), Anni, S.Si., Via, S.Si., Ani, S.Si., Red S.Si., Idhonk, S.Si., dan Thira, S.Si. (selamat yah atas pekerjaan yang telah diperoleh!!), Titin, S.Si (Thanks sudah melupakanQ di skripsimu!! Btw mw Q lupakan ki' juga tapi ndak enak, soalnya Titin itw master pacalla, aslinya badak; gak mempan dgn setiap callaanQ, asyik buat jadi temen pelepas stress..wkwkwk), Widya, Rina dan Che-che (Thanks buat semangatnya, bantuan-bantuan selama kuliah, semoga cepat S.Si yah!!), Ingkuh, S.Si. (Kapan ki' OL lagi??), Adin, S.Si, Ulla, S.Si., Supri (Thanks buat kebersamaanx selama di campus), Lina, S.Si., Nana, S.Si., Hasni, S.Si., dan Eka, S.Si (Klo dapet gaji pertamanya hubungi no HPq yah, Ntar Z kasi No. RekQ, haha...), Fajar, S.Si (Thanks atas bantuannya dalam pengeditan SkripsiQ, dan segala kemudahanx selama saya menggunakan fasilitas Perpustakaan!! Moga cepat dapat kerja yah bro.). Jani, S.Si & Yana, S.Si (Jangan lupakan kebersamaan kita yah, meskipun kalian jauh di sana, sukses selalu sista...),
Semuanya...terima kasih banyak atas rasa persaudaraan yang telah kalian berikan. "Bila sudah Tua nanti, Kita akan hidup masing-masing, ingatlah akan Hari-hari ini, saat-saat dimana Kita berjuang bersama-sama dalam suka dan duka!!!".

Sekiranya segala bantuan dan partisipasinya bernilai ibadah dan mendapat pahala yang setimpal di sisi Tuhan Yang Maha Esa. Akhir kata, semoga tulisan ini dapat memberikan manfaat kepada semua pihak yang membutuhkan.

Makassar, Februari 2010

Penulis

ABSTRAK

Salah satu metode dalam segmentasi citra yang banyak dikembangkan saat ini adalah *fuzzy clustering*, yang di dalamnya terdapat algoritma *Fuzzy C-Mean Cluster*. Dengan *Fuzzy C-Mean Cluster*, pengelompokan dibagi atas beberapa *cluster* yang keanggotaan tiap-tiap titik data dalam salah satu *cluster* tersebut ditentukan oleh derajat keanggotaannya. Metode ini dapat diterapkan untuk memisahkan foto dalam biodata yang telah didigitalisasi.

Kata Kunci : Segmentasi Citra, Fuzzy Clustering, Fuzzy C-Mean Cluster, derajat keanggotaan.

ABSTRACT

One of image segmentation method has been developed today is fuzzy clustering, in which the algorithm Fuzzy C-Means Cluster. With Fuzzy C-Means cluster, grouping divided into several clusters and the membership of each data point in one cluster is determined by the degree of membership. This method can be applied to separate the images in the biographical data that have been digitalized.

Keywords: Image Segmentation, Fuzzy Clustering, Fuzzy C-Means Cluster, the degree of membership.

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR ISTILAH DAN SIMBOL.....	xi
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	3
I.3 Batasan Masalah	3
I.4 Tujuan Penulisan	4
BAB II LANDASAN TEORI	5
II.1 Citra.....	5
II.2 Pemrosesan Citra.....	7
II.3 Segmentasi Citra	15
II.4 <i>Data Clustering</i>	16
II.5 <i>Fuzzy Clustering</i>	20
II.6 <i>Fuzzy C-Mean (FCM)Cluster</i>	21
II.7 <i>Noise Filtering</i>	24

BAB III IMPLEMENTASI.....	26
III.1 Pengumpulan Data dengan Mencari Nilai <i>Mean</i> dan Standar Deviasi	27
III.2 Penerapan Algoritma <i>Fuzzy C-Mean Clustering</i> pada Segmentasi	29
III.3 Defuzzifikasi	34
III.4 Penerapan <i>Noise Filtering</i>	35
BAB IV HASIL DAN PEMBAHASAN	36
BAB V KESIMPULAN DAN SARAN	44
V.1 Kesimpulan	44
V.2 Saran	45
DAFTAR PUSTAKA	46
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1	<i>Mask Convolution</i>	15
Tabel 4.1	Hasil <i>Clustering</i> Formulir 1	42
Tabel 4.2	Hasil <i>Clustering</i> Formulir 2	42
Tabel 4.3	Hasil <i>Clustering</i> Formulir 3	43
Tabel 4.4	Hasil <i>Clustering</i> Formulir 4	43

DAFTAR GAMBAR

Gambar 1.1	Proses Pengolahan Data.....	2
Gambar 2.1	Citra Gedung Rektorat Unhas dan Citra Logo Unhas.....	6
Gambar 2.2	Kubus RGB.....	10
Gambar 2.3	Komposisi Utama Citra RGB.....	10
Gambar 2.4	<i>Gray-level</i> pada Citra <i>Grayscale</i>	11
Gambar 2.5	<i>Pixel</i> pada Citra RGB yang akan dikonversi ke Citra <i>Grayscale</i>	12
Gambar 2.6	Mode Penyimpanan Citra.....	13
Gambar 2.7	<i>Convolution Window</i>	14
Gambar 2.8	Data sebelum dilakukan proses <i>Clustering</i>	17
Gambar 2.9	<i>Clustering</i> berdasarkan <i>Similaritas</i> (kesamaan) Warna.....	17
Gambar 2.10	<i>Clustering</i> berdasarkan <i>Similaritas</i> (kesamaan) Jarak.....	18
Gambar 2.11	<i>Filter</i> Untuk Menghilangkan <i>Noise</i>	25
Gambar 3.1	Flowchart Segmentasi Citra.....	26
Gambar 3.2	Proses pengelompokkan <i>pixel</i> pada <i>Bitmap</i> Formulir.....	28
Gambar 3.3	Banyaknya <i>Cluster</i> yang Dibentuk pada <i>Bitmap</i> Formulir.....	30
Gambar 4.1	Contoh <i>Bitmap</i> Formulir Biodata.....	36
Gambar 4.2	Grafik Hasil Pengclusteran dengan <i>Fuzzy C-Mean cluster</i> pada Formulir Biodata 1.....	37
Gambar 4.3	Grafik Hasil Pengclusteran dengan <i>Fuzzy C-Mean cluster</i> pada Formulir Biodata 2.....	38



Gambar 4.4 Grafik Hasil Pengclusteran dengan *Fuzzy C-Mean cluster* pada
Formulir Biodata 3 38

Gambar 4.5 Grafik Hasil Pengclusteran dengan *Fuzzy C-Mean cluster* pada
Formulir Biodata 4 39

Gambar 4.6 Pencitraan *Cluster* yang Memiliki Nilai Standar Deviasi
Medium 40

Gambar 4.7 Hasil Pengekstrakkan Citra Foto Biodata yang Tersegmentasi .. 41

DAFTAR ISTILAH DAN SIMBOL

- Bit** : Binary digit, satuan terkecil ukuran pixel data computer yang terkode secara biner (0 dan 1).
- Bitmap** : Citra digital yang tersusun atas titik-titik warna yang akan terlihat saat gambar diperbesar. Tipe file bitmap antara lain BMP, GIF, JPEG, PCX, TIFF.
- Byte** : Kumpulan dari 8 bit.
- Cluster** : Kelas-kelas pengelompokan.
- Noise** : Partikel-partikel kecil yang nampak pada hasil scan.
- X** : Himpunan data yang akan dicluster.
- μ_{ik} : Derajat keanggotaan suatu titik data ke-k di cluster ke-i.
- v_{fi} : Pusat vector data pada cluster fuzzy ke-i.
- $\mu_f(c)$: Matriks Partisi.
- $P_t(c)$: Fungsi Obyektif Fuzzy C-Mean Cluster.
- N** : Banyaknya data yang akan dicluster.
- c** : Banyaknya cluster yang akan dibentuk.
- x_k : Data ke-k
- ε : Termination tolerance/Kriteria penghentian iterasi.
- t** : Iterasi ke-...
- MaxIter** : Iterasi maksimum yang ditetapkan.
- w** : Parameter control fuzzy/pembobot.

BAB I

PENDAHULUAN

I.1 Latar Belakang

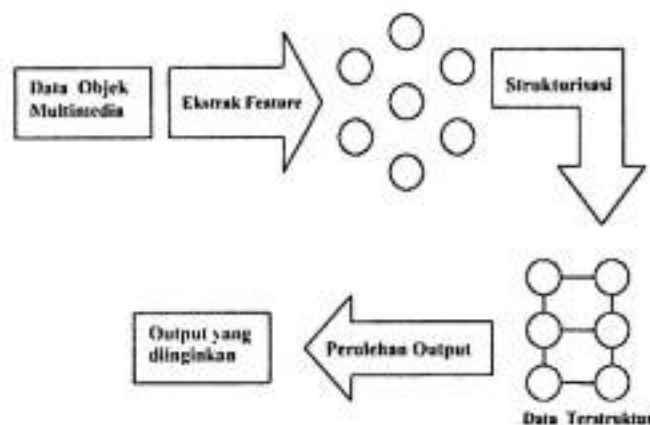
Masalah registrasi atau hal-hal yang berhubungan dengan biodata dalam sebuah institusi yang menyangkut banyak individu merupakan pekerjaan yang tidak mudah. Selama ini masalah utama dalam proses registrasi yaitu belum mengefisiensikan waktu, biaya, dan tenaga dalam pengolahannya. Hal inilah yang membuatnya nampak rumit, meskipun ditunjang dengan adanya kemajuan komputer saat ini. Adapun pekerjaan yang sulit dari pengolahan data pada sebuah formulir biodata adalah memisahkan foto, teks, dan *background* dari formulir tersebut.

Hal yang dapat dilakukan oleh manusia untuk membantunya dalam pengolahan data yaitu memanfaatkan kemajuan teknologi yang ada, dengan memanfaatkan teknologi tersebut diharapkan dapat mempermudah jalannya proses registrasi, sehingga waktu, tenaga, dan biaya dapat lebih diefisiensikan. Salah satu dari kemajuan teknologi yang dirasakan saat ini adalah *scanner*. Jadi lembaran formulir biodata dapat di-*scan* menggunakan *scanner* yang akan menghasilkan gambar, pada komputer akan dikenali sebagai titik-titik (*bitmap*), isi dalam *bitmap* ini terdapat data yang sangat penting, termasuk diantaranya teks, *image* (citra) yang berupa foto dan *background* dari lembaran itu sendiri.



Yang menjadi latar belakang penulisan tugas akhir ini adalah cara mengambil fotonya saja dari sebuah formulir biodata yang telah di-*scan*, itulah yang akan dibahas lebih lanjut.

Secara garis besar ada tiga tahap yang dilakukan dalam pengolahan data-data tersebut, yaitu: *feature extraction* (ekstra *feature*), strukturisasi, dan perolehan *output*. Pada gambar di bawah ini dapat dilihat arus dari ketiga tahapan tersebut dengan jelas.



Gambar 1.1 Proses Pengolahan Data

Ketika jumlah data yang terlibat sangat banyak, akan timbul kesulitan saat proses strukturisasi. Oleh sebab itu, saat ini telah banyak dikembangkannya metode-metode yang dapat melakukan pengolahan data secara otomatis. Beberapa jenis metode tersebut antara lain adalah algoritma genetika (*genetic algorithm*), jaringan saraf tiruan (*artificial neuron network*), dan *pattern recognition*.

Fuzzy C-Mean adalah bagian dari *pattern recognition*, yang paling banyak dikenal dan digunakan karena kesederhanaan dan keefektifannya. Dengan *Fuzzy C-Mean* jumlah data yang banyak akan dikelompokkan ke dalam beberapa *cluster* (kelompok) sehingga memudahkan proses pencarian *output*.

Berdasarkan uraian di atas, maka dalam skripsi ini akan dibahas suatu metode segmentasi citra foto pada biodata menggunakan *Fuzzy C-Mean* (FCM) *cluster*.

I.2 Rumusan Masalah

Pada penulisan tugas akhir ini, masalah yang dirumuskan adalah :

1. Bagaimana menentukan letak foto pada formulir biodata dengan metode *Fuzzy C-Mean cluster*?
2. Bagaimana menerapkan metode *Fuzzy C-Mean Cluster* dalam pengelompokkan *pixel* pada formulir biodata?
3. Bagaimana mengekstraksi foto dari suatu biodata?

I.3 Batasan Masalah

Dalam penulisan tugas akhir ini masalah dibatasi pada :

1. Menggunakan lembaran formulir biodata yang telah di-*scan*.
2. Citra yang digunakan adalah citra Red-Green-Blue (RGB).
3. Citra yang diambil selalu berbentuk persegi panjang.

I.4 Tujuan Penulisan

Tujuan dari penulisan tugas akhir ini adalah :

1. Menentukan letak foto pada formulir biodata melalui metode *Fuzzy C-Mean cluster*.
2. Menerapkan metode *Fuzzy C-Mean cluster* dalam pengelompokan *pixel* pada formulir biodata.
3. Mengekstraksi foto dari suatu formulir biodata.

BAB II

LANDASAN TEORI

II.1 Citra

Citra (*image*) istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*). Maksudnya tentu sebuah gambar dapat memberikan informasi yang lebih banyak bila dibandingkan informasi tersebut disajikan dalam bentuk kata-kata (tekstual). (Rinaldi M, 2005).

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Sedangkan definisi citra menurut kamus Webster adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Sebuah citra mengandung informasi tentang objek yang direpresentasikan. (Castleman, 1996).

Citra juga dapat dikelompokkan menjadi citra tampak (foto, gambar, lukisan, apa yang nampak di layar monitor/televisi, hologram, dll) dan citra tidak tampak (data foto/gambar dalam file, citra yang direpresentasikan dalam fungsi matematis).

Citra dari sudut pandang matematis, merupakan fungsi kontinu dari intensitas cahaya pada bidang 2 dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Sumber cahaya menerangi objek, objek

memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat-alat optik, misalnya mata manusia, kamera, *scanner*, sensor satelit dsb, kemudian direkam. Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat:

1. Optik berupa foto,
2. Analog berupa sinyal video seperti gambar pada monitor televisi,
3. Digital yang dapat langsung disimpan pada suatu pita magnetik.



a. Gedung Rektorat Unhas



b. Logo Unhas

Gambar 2.1 Citra Gedung Rektorat Unhas dan Citra Logo Unhas

Secara matematis fungsi intensitas cahaya pada bidang 2 dimensi dinotasikan sebagai $f(x,y)$, yang mana (x,y) merupakan koordinat pada bidang 2 dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi, dan citra yang dihasilkan merupakan citra digital. Citra digital dapat direpresentasikan dalam matriks ukuran $M \times N$, dan masing-masing elemen pada citra digital disebut *pixel* (*picture element*)

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad [1]$$

II.2 Pemrosesan Citra

Pemrosesan citra adalah ilmu untuk memanipulasi gambar, yang melingkupi teknik-teknik untuk memperbaiki atau mengurangi kualitas gambar, menampilkan bagian tertentu dari gambar, membuat sebuah gambar yang baru dari beberapa bagian gambar yang sudah ada, mengembalikan gambar yang rusak pada saat pengambilan, dan beberapa teknik manipulasi lainnya. Berdasarkan tujuannya, operasi pengolahan citra dikategorikan sebagai berikut :

1. Perbaiki Kualitas Citra (*image enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini, ciri-ciri khusus yang terdapat didalam citra lebih ditonjolkan.

Contoh-contoh operasi perbaiki citra:

- a. Perbaiki kontras gelap/terang citra.
- b. Perbaiki tepian objek (*edge enhancement*).
- c. Penajaman (*sharpening*).
- d. Pemberian warna semu (*pseudocoloring*).
- e. Penapisan derau (*noise filtering*).

2. Pemugaran Citra (*image restoration*)

Operasi ini bertujuan menghilangkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui.

Contoh-contoh operasi pemugaran citra :

- a. Penghilangan kesamaran (*deblurring*).
- b. Penghilangan derau (*noise*).

3. Pemampatan Citra (*image compression*)

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan citra adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus.

4. Segmentasi Citra (*image segmentation*)

Segmentasi citra merupakan suatu teknik pengelompokan (*clustering*) pada citra dengan tujuan menemukan karakteristik khusus yang dimiliki suatu citra. Jenis operasi ini berkaitan erat dengan pengenalan pola.

5. Pengorakan Citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan diskripsinya. Contoh-contoh operasi pengorakan citra :

- a. Pendeteksian tepian objek (*edge detection*).

- b. Ekstraksi batas (*boundary*).
 - c. Representasi Daerah (*region*).
6. Rekonstruksi Citra (*image reconstruction*)

Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis.

Untuk selanjutnya dalam penulisan tugas akhir ini hanya membahas tentang segmentasi citra (*image segmentation*). Operasi citra lainnya yang menjadi penunjang dari keenam operasi di atas yaitu mencakup representasi warna dan beberapa metode dasar pemrosesan citra.

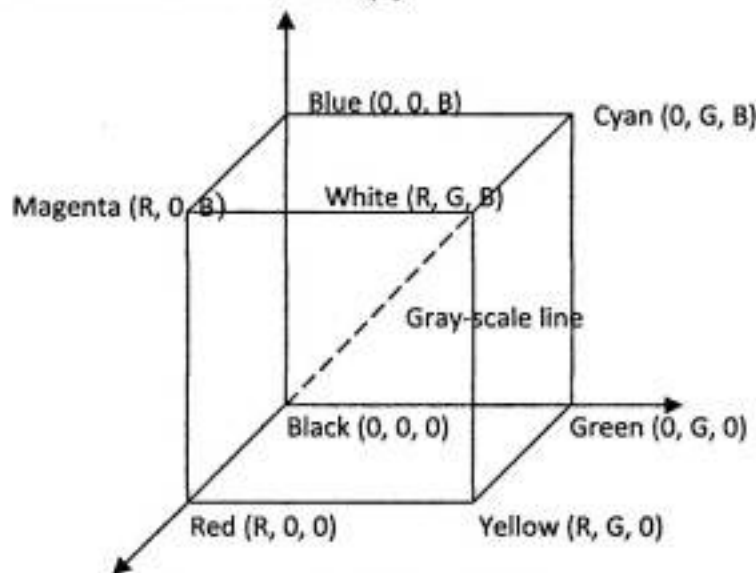
II.2.1 Representasi Warna (*Color Representation*)

Karena persepsi manusia akan warna berdasarkan respon dari tiga *cones* yang berada pada bola mata manusia, maka representasi warna pun didasarkan pada hal tersebut, yang biasanya disebut *tristimulus value*. Salah satu model representasi warna berdasarkan *tristimulus value* adalah representasi warna RGB yang nantinya dapat dijadikan representasi *grayscale* untuk memudahkan pemrosesan citra.

II.2.1.1 Representasi Warna Red-Green-Blue (RGB)

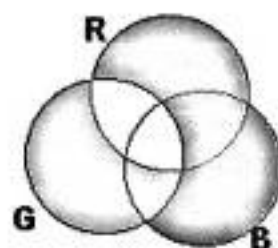
Representasi warna ini terdiri dari tiga unsur utama yaitu *red* (merah), *green* (hijau), *blue* (biru). Gabungan tiga warna ini membentuk warna-warna lainnya berdasarkan intensitas dari masing-masing warna tersebut, misalnya

warna putih adalah gabungan dari ketiga warna tersebut dengan intensitas maksimal satu (1) dan warna hitam merupakan gabungan dari ketiga warna tersebut dengan intensitas minimal nol (0).



Gambar 2.2 Kubus RGB

Dalam sistem RGB, suatu warna dapat didefinisikan berdasarkan tingkat intensitas dari komponen merah, hijau dan biru. Ketiga komponen tersebut dapat digambarkan sebagai koordinat 3 dimensi. Karena koordinat 3 dimensi dari model warna RGB mempunyai bentuk kubus, maka disebut dengan *RGB cube* (kubus RGB). Koordinat berdasarkan warna (merah, hijau, biru) atau (R, G, B). Koordinat (0, 0, 0) berarti hitam karena ketiga intensitas warna tersebut 0, sedangkan (R, G, B) berarti warna putih jika R, G, dan B mempunyai nilai (1,1,1). (Yoyokits, 2007).



Gambar 2.3 Komposisi Utama Citra RGB

Pada perhitungan dalam program-program komputer, model warna direpresentasi dengan nilai komponennya, seperti dalam RGB masing-masing nilai antara 0 hingga 255 sesuai dengan urutan masing-masing yaitu pertama *Red*, kedua *Green* dan ketiga adalah nilai *Blue* dengan demikian masing-masing komponen ada 256 tingkat. Apabila dikombinasikan maka ada $256 \times 256 \times 256$ atau 16.777.216 kombinasi warna RGB yang dapat dibentuk. (Herman P, 2008).

II.2.1.2 Representasi Warna *Grayscale*

Grayscale adalah citra yang memiliki *gray-level* sebagai nilai dari tiap *pixel*-nya. *Grayscale* adalah proses pemetaan citra RGB menjadi *gray-level*.

$$\text{Grayscale: } f(x,y,R,G,B) \rightarrow g(x,y) \quad [2]$$

Pada dasarnya proses ini dilakukan dengan meratakan nilai *pixel* dari 3 nilai RGB menjadi 1 nilai *grayscale*, sehingga dalam proses selanjutnya menjadi lebih mudah.



Gambar 2.4 *Gray-level* pada Citra *Grayscale*

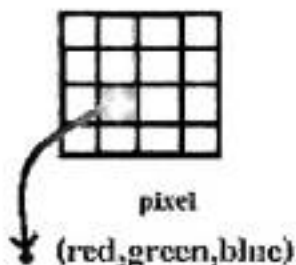
Pada gambar 2.4, terlihat bahwa skala *pixel* pada *grayscale* berada pada interval $[0,255]$, di mana pada skala 0 warna *pixel* hitam, sedangkan pada skala 255 menunjukkan warna putih. Dengan menggunakan representasi warna RGB

gambar yang berwarna dapat diubah menjadi gambar yang terdiri dari warna putih dan gradiasi warna hitam yang biasanya disebut gambar *grayscale*.

Untuk mengubah warna RGB menjadi *grayscale* dapat digunakan rumus sebagai berikut:

$$\text{grayscale} = 0.299R + 0.587G + 0.114B \quad [3]$$

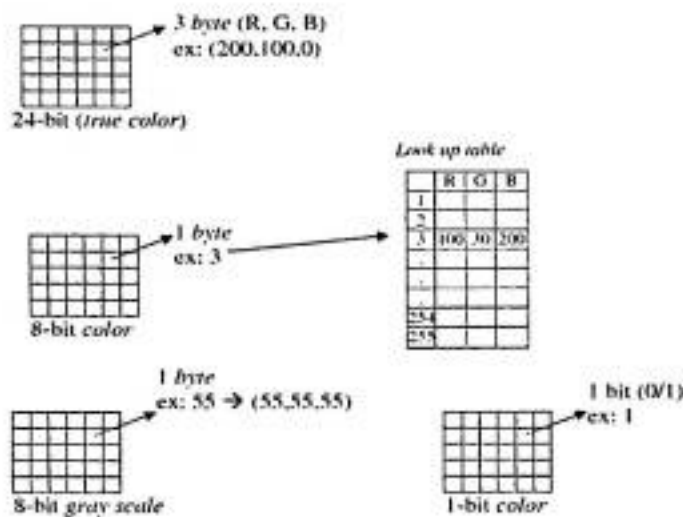
di mana R, G, dan B merupakan intensitas warna dari masing-masing komponen.



Gambar 2.5 *Pixel* pada Citra RGB yang akan dikonversi ke Citra *Grayscale*

II.2.2 Metode Dasar Pemrosesan Citra

Data citra (*image*) umumnya disimpan dalam bentuk array 2 dimensi dan berisi angka yang mengacu pada salah satu representasi warna. Berdasarkan kedalaman warna maka penyimpanan data pada citra umumnya terbagi atas empat, yaitu : 24-bit (*True Color*), 8-bit *color*, 8-bit *grayscale*, dan 1-bit (*white/black*). Pada 24-bit, data citra disimpan pada setiap titik (*pixel*) pada citra menjadi 3 *byte* pada saat penyimpanan dalam array yang berisi nilai R, G, dan B. Pada 8-bit *color* data disimpan pada 1 *byte* namun membutuhkan tabel *lookup* yang berisi data R, G, dan B untuk nilai pada setiap *pixel*. Pada 8-bit *grayscale* disimpan pada 1 *byte* yang mengacu pada intensitas warna *grayscale*. Dan pada 1-bit setiap 8 *pixel* disimpan pada 1 *byte* data.



Gambar 2.6 Mode Penyimpanan Citra

Ada empat klasifikasi dasar dari metode untuk pemrosesan citra yaitu *point process*, *area process*, *geometric process*, dan *frame process*

II.2.2.1 Point Process

Proses yang pertama ini merupakan algoritma yang paling sederhana dari keempat metode lainnya, yaitu operasi yang dilakukan dengan cara mengubah nilai *pixel* berdasarkan nilai *pixel* itu sendiri. Beberapa operasi yang dapat dilakukan pada *point process* antara lain:

1) Operasi aritmatika

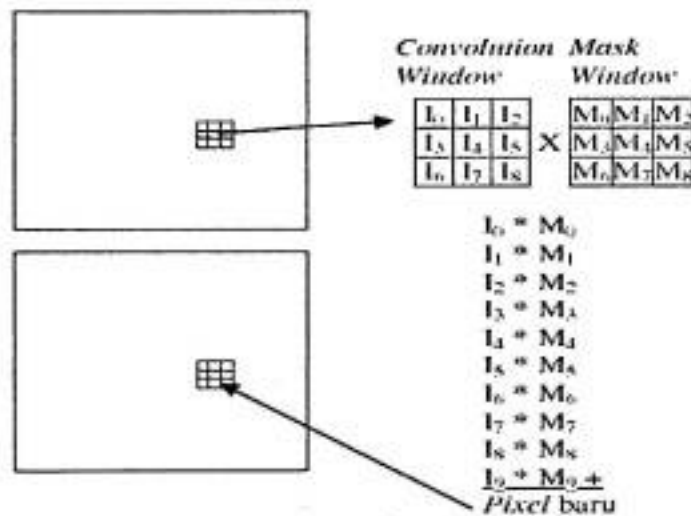
Operasi ini meliputi operasi tambah, kurang, kali, dan bagi untuk sebuah nilai pada suatu *pixel* dengan nilai konstan.

2) Operasi logika

Operasi ini meliputi operasi OR, AND, dan XOR untuk sebuah *pixel* pada suatu *pixel* dengan nilai konstan.

II.2.2.2 Area Process

Area memproses nilai *pixel* suatu gambar berdasarkan nilai *pixel* tersebut beserta nilai *pixel* sekelilingnya. Untuk itu diperlukan *convolution window* yang berupa matriks berukuran kecil untuk mendapatkan nilai disekitar *pixel* yang ingin diubah. Nilai pada *convolution window* ini akan dikalikan dengan *convolution mask* yang berukuran sama dengan *convolution window* dan hasil penjumlahan dari perkalian tersebut akan menjadi nilai baru bagi *pixel* tersebut.



Gambar 2.7 Convolution Window

Pemrosesan yang dapat dilakukan adalah dengan mengatur nilai pada *mask convolution*. Adapun beberapa mask yang umum digunakan adalah sebagai berikut:

Tabel 2.1 Mask Convolution

1	1	1	<i>LowPass Filtering</i>	1/9	1/9	1/9	Untuk membuat efek <i>bluring</i>
1	7	1		1/9	1/9	1/9	
1	1	1		1/9	1/9	1/9	
-1	-1	-1	Untuk membuat efek <i>sharpening</i>	0	1	2	Untuk <i>edge detection</i>
-1	9	-1		3	4	5	
-1	-1	-1		6	7	8	

Diadaptasi dari: Steven Jaowry, 2007

II.2.2.3 Geometric Process

Proses yang dilakukan oleh *geometric process* adalah mengubah bentuk dari gambar. Adapun bentuk perubahan tersebut antara lain : *scaling* (mengubah ukuran), *rotation* (mengubah arah), dan *translation* (mengubah tempat).

II.2.2.4 Frame Process

Frame memproses nilai *pixel* suatu gambar berdasarkan operasi dari 2 buah gambar atau lebih. Adapun operasi yang bisa dilakukan pada *frame process* adalah *addition* (penjumlahan), *subtraction* (pengurangan), dan *average* (rata-rata).

II.3 Segmentasi Citra

Segmentasi citra merupakan suatu teknik pengelompokan (*clustering*) pada citra. Dengan kata lain, merupakan suatu proses pembagian citra ke dalam wilayah (*region*) yang mempunyai kesamaan fitur antara lain : tingkat keabuan (*grayscale*), tekstur (*texture*) dan warna (*color*). Berdasarkan pengertiannya,

segmentasi memiliki tujuan menemukan karakteristik khusus yang dimiliki suatu citra. Oleh karena itulah, segmentasi sangat diperlukan pada proses pengenalan pola. Semakin baik kualitas segmentasi maka semakin baik pula kualitas pengenalan polanya.

II.4 *Data Clustering*

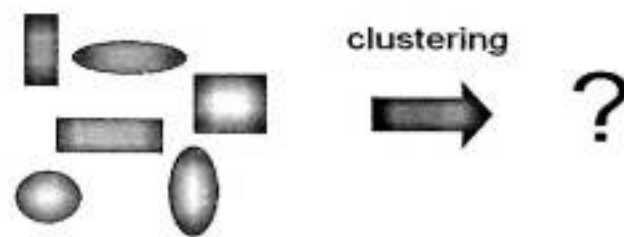
Data merupakan aspek penting dalam berbagai kehidupan, tidak semua data dapat dibaca dengan mudah karena keterbatasan kemampuan manusia dalam menangani data dalam jumlah besar. Seringkali data hanya dijadikan dokumentasi dan tidak diadakan pengolahan lebih lanjut menjadi suatu informasi yang lebih berguna. Maka pengolahan data merupakan hal penting dalam mendapatkan informasi. Salah satu contoh pengolahan data adalah dengan *pengclustering*.

Data clustering adalah suatu metode pemecahan masalah secara matematis dengan mencari bentuk (*structure*) menggunakan persamaan sifat yang ada pada data tersebut. Proses terpenting dalam *data clustering* ini adalah melihat persamaan yang terdapat pada data, sehingga data dapat digolongkan ke dalam kelas-kelas (*clusters*) yang ditentukan.

Cara kerja *data clustering* adalah mengkoordinasi data-data yang ada ke dalam beberapa kelas, yang anggota dari kelas tersebut memiliki kesamaan dalam hal-hal tertentu.

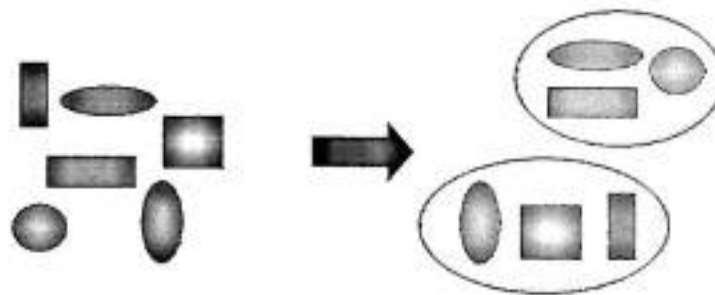
Sebuah *cluster* adalah kumpulan data-data yang memiliki kesamaan diantara data-data itu sendiri dan ketidaksamaan pada data-data yang terdapat pada kelas lain.

Jika ada n objek pengamatan dengan p variabel, maka sebelum dilakukan pengelompokan data atau objek, terlebih dahulu menentukan ukuran kedekatan sifat antar data. Gambar 2.8, berikut ini menunjukkan contoh data yang akan dilakukan *clustering*.



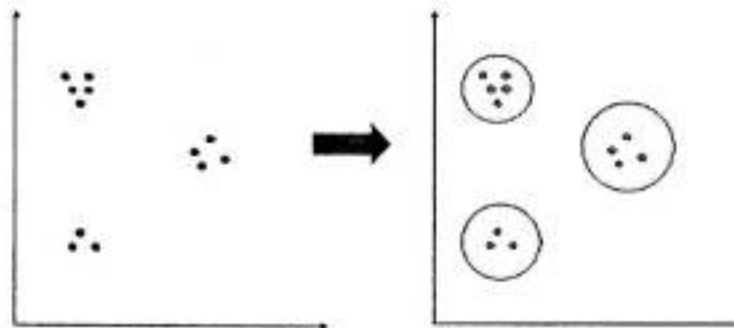
Gambar 2.8 Data sebelum dilakukan proses *Clustering*

Jika pada data dilakukan *clustering* (pengelompokan) berdasarkan warna, maka pengelompokannya seperti yang terlihat pada gambar 2.9,.



Gambar 2.9 *Clustering* berdasarkan *Similaritas* (kesamaan) Warna

Selain menggunakan *similaritas* (kesamaan) berdasarkan warna dan bentuk, *clustering* juga bisa dilakukan dengan menggunakan *similaritas* berdasarkan jarak, artinya data yang memiliki jarak berdekatan akan membentuk satu *cluster*, contohnya seperti yang diperlihatkan pada gambar 2.10,.



Gambar 2.10 *Clustering* berdasarkan *Similaritas* (kesamaan) Jarak

Teknik *clustering* bekerja dengan mencari kemiripan di antara objek dengan memperhatikan karakteristik objek, sekelompok objek yang mirip akan termasuk ke dalam satu *cluster*. (Jiawei, 2001). Jenis-jenis *clustering* berdasarkan kesamaan pada data:

II.4.1 *Distance-Based Clustering*

Dalam gambar 11, data terbagi atas 3 *cluster*. Kriteria kesamaan pada data tersebut adalah jarak (*distance*). Jika dua atau lebih data memiliki jarak yang berdekatan berdasarkan sebuah jarak yang ditentukan (dalam hal ini jarak geometris) maka hal ini disebut sebagai *distance-based clustering*.

II.4.2 *Conceptual Clustering*

Jika dua atau lebih data yang dimiliki oleh sebuah cluster memiliki kesamaan umum dalam konsep tertentu dan bukan berdasarkan kesamaan pengukuran, hal ini disebut sebagai *conceptual clustering*.

Data clustering yang secara umum digunakan untuk pengolahan data matematis adalah *distance-based clustering*, karena *clustering* jenis ini dilakukan

secara matematis, *fuzzy clustering* termasuk di dalamnya. *Distance-based clustering* memiliki beberapa jenis algoritma yang umum digunakan yaitu:

a. *Exclusive Clustering*

Clustering jenis ini biasa disebut juga *hard clustering*, karena algoritma *clustering* jenis ini menentukan apakah data tersebut masuk pada *cluster* tertentu dengan pernyataan ya atau tidak (0,1). Salah satu algoritma jenis ini adalah *Clustering K-Means, Residual Analysis*.

b. *Overlapping Clustering*

Algoritma *clustering* jenis ini menentukan seberapa besar suatu data menjadi milik *cluster* tertentu [0,1]. *Clustering Fuzzy C-Mean* termasuk dalam algoritma jenis ini dan *Gaussian Mixture*.

c. *Hierarchical Clustering*

Algoritma *clustering* ini menentukan setiap data harus termasuk dalam *cluster* tertentu dan data yang sudah masuk pada suatu *cluster* pada suatu tahapan proses, tidak dapat berpindah ke cluster lain. Contohnya *Single Linkage, Centroid Linkage, Complete Linkage, Average Centroid*.

d. *Hybrid Clustering*

Algoritma *clustering* jenis ini mencampurkan karakteristik dari *Exclusive Clustering, Overlapping Clustering, dan Hierarchical Clustering*.

Dalam algoritma *distance-based clustering*, yang memegang peranan terbesar adalah *distance measure* atau pengukuran jarak. *Distance* atau jarak

memiliki kemampuan yang akan menentukan sebuah data atau objek akan masuk ke dalam kelas yang mana.

II.5 *Fuzzy Clustering*

Salah satu proses paling mendasar dari *pattern recognition* adalah pengelompokan data, atau yang lebih umum disebut *clustering*. Masalah yang dihadapi pada proses ini adalah mencari beberapa pusat *cluster* (kelompok) yang dapat mewakili karakteristik kelompok-kelompok dari suatu *data set*.

Fuzzy clustering merupakan salah satu bagian dari *pattern recognition*. *Pattern recognition*, sendiri secara umum dapat didefinisikan sebagai proses pencarian struktur data dan mengelompokkannya berdasarkan kategori-kategori tertentu sedemikian, sehingga derajat kemiripan antara data yang berada dalam suatu kelompok adalah tinggi dan rendah untuk yang berlainan. Kategori yang dipakai biasanya berdasarkan pengalaman-pengalaman yang telah diperoleh sebelumnya.

Pada analisis *cluster* klasik, kelompok-kelompok yang terbentuk, setelah proses *clustering*, harus merupakan pembagian *data set* sedemikian rupa, sehingga untuk data yang berada dalam kelompok yang sama harus mirip dan berbeda dengan data yang berada pada kelompok lain. Walaupun demikian, syarat ini seringkali dirasakan terlalu kaku dan sulit dipenuhi pada aplikasi praktek, sehingga perlu diganti dengan syarat yang lebih ringan. Untuk itu digunakanlah kompromi dengan prinsip-prinsip *fuzzy* sehingga data dapat menjadi anggota dari semua kelas, atau *cluster* terbentuk dengan derajat atau tingkat keanggotaan yang

berbeda antara 0 hingga 1. Prinsip *fuzzy* ini yang kemudian dikenal sebagai *fuzzy clustering*. Jadi suatu cluster dikatakan *fuzzy* jika tiap-tiap objek dihubungkan dengan menggunakan derajat keanggotaan.

Ada beberapa algoritma clustering data dalam *fuzzy clustering*, salah satu diantaranya adalah *Fuzzy C-Means (FCM) cluster*.

II.6 *Fuzzy C-Mean (FCM) Cluster*

Fuzzy C-Mean (FCM) Cluster adalah suatu teknik pengclusteran data yang mana keberadaan tiap-tiap titik data dalam suatu *cluster* ditentukan oleh derajat keanggotaan. Teknik ini pertama kali diperkenalkan oleh Dunn (1973) dan dikembangkan oleh Jim Bezdek pada tahun 1981.

Konsep dasar FCM, pertama kali adalah menentukan pusat *cluster* yang akan menandai lokasi rata-rata (*mean*) untuk tiap-tiap *cluster*. Pada kondisi awal, pusat *cluster* ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan untuk tiap-tiap *cluster*. Dengan cara memperbaiki pusat *cluster* (*cluster center*) dan derajat keanggotaan tiap-tiap data secara berulang, maka dapat dilihat bahwa pusat *cluster* akan menuju lokasi yang tepat. Perulangan ini didasarkan pada minimasi fungsi obyektif yang menggambarkan jarak dari titik data yang diberikan ke pusat *cluster* yang terbobot oleh derajat keanggotaan titik data tersebut.

Apabila terdapat suatu himpunan data sebagai berikut:

$$X=(x_1, x_2, x_3, \dots, x_N) \quad [4]$$

Derajat keanggotaan suatu titik data ke-k di cluster ke-i adalah:

$$\mu_{ik}(x_k) \in [0,1] \text{ dengan } (1 \leq i \leq c; 1 \leq k \leq N) \quad [5]$$

Pada Metode *Fuzzy C-Mean cluster*, matriks partisi didefinisikan sebagai:

$$\mu_f(c) = \begin{bmatrix} \mu_{11}[x_1] & \mu_{12}[x_2] & \dots & \mu_{1N}[x_N] \\ \mu_{21}[x_1] & \mu_{22}[x_2] & \dots & \mu_{2N}[x_N] \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1}[x_1] & \mu_{c2}[x_2] & \dots & \mu_{cN}[x_N] \end{bmatrix} \quad [6]$$

dengan

$$\sum_{i=1}^c \mu_{ik} = 1 \quad [7]$$

yang berarti bahwa jumlah nilai keanggotaan suatu data pada semua *cluster* harus sama dengan 1.

Fungsi obyektif iterasi ke-t $P(c)$ pada matriks partisi adalah:

$$\bar{P}_t(c) = \sum_{k=1}^N \sum_{i=1}^c (\mu_{ik})^w |x_k - v_{fi}|^2 \quad [8]$$

dengan v_{fi} adalah pusat vektor data pada *cluster fuzzy* ke-i,

$$v_{fi} = \frac{\sum_{k=1}^N (\mu_{ik})^w x_k}{\sum_{k=1}^N (\mu_{ik})^w} \quad [9]$$

dan w adalah bobot pada nilai-nilai keanggotaan, $|x_k - v_{fi}|$ adalah bentuk norm Euclidean yang digunakan sebagai jarak antara x_k dan v_{fi} .

Algoritma *Fuzzy C-Mean*:

1. Bentuk matriks partisi awal $\mu_f(c)$ (derajat keanggotaan dalam *cluster*); matriks partisi awal biasanya dibuat secara acak

$$\mu_f(c) = \begin{bmatrix} \mu_{11}[x_1] & \mu_{12}[x_2] & \cdots & \mu_{1N}[x_N] \\ \mu_{21}[x_1] & \mu_{22}[x_2] & \cdots & \mu_{2N}[x_N] \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1}[x_1] & \mu_{c2}[x_2] & \cdots & \mu_{cN}[x_N] \end{bmatrix} \quad [10]$$

2. Tentukan:
 - a. Jumlah *cluster* yang akan dibentuk ($c \geq 2$);
 - b. Pangkat/pembobot ($w > 1$); w merupakan bilangan riil.
 - c. Maksimum iterasi (MaxIter);
 - d. Kriteria penghentian ($\epsilon > 0$ atau $\epsilon =$ nilai positif yang sangat kecil);
 - e. Iterasi awal ($t=1$);
3. Hitung fungsi obyektif awal : $P_t(c)$ dari matriks partisi awal;
4. Naikkan nomor iterasi: $t = t + 1$;
5. Hitung pusat vektor tiap-tiap cluster untuk matrik partisi tersebut sebagai berikut:

$$v_{\beta} = \frac{\sum_{k=1}^N (\mu_{ik})^w x_k}{\sum_{k=1}^N (\mu_{ik})^w} \quad [11]$$

6. Modifikasi tiap-tiap nilai keanggotaan sebagai berikut:

a. Jika $y_k \neq v_{fi}$,

$$\mu_{ik}(y_k) = \left[\sum_{g=1}^c \left(\frac{|x_k - v_{fg}|^2}{|x_k - v_{gi}|^2} \right)^{\frac{1}{(w-1)}} \right]^{-1} \quad [12]$$

b. Jika $y_k = v_{fi}$,

$$\mu_{ik}(y_k) = 1, \text{ jika } i=g; \quad [13]$$

$$\mu_{ik}(y_k) = 0, \text{ jika } i \neq g. \quad [14]$$

7. Modifikasi matriks partisi sebagai berikut:

$$\mu_f(c) = \begin{bmatrix} \mu_{11}[x_1] & \mu_{12}[x_2] & \cdots & \mu_{1N}[x_N] \\ \mu_{21}[x_1] & \mu_{22}[x_2] & \cdots & \mu_{2N}[x_N] \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1}[x_1] & \mu_{c2}[x_2] & \cdots & \mu_{cN}[x_N] \end{bmatrix} \quad [15]$$

8. Hitung fungsi obyektif:

$$\hat{P}_t(c) = \sum_{k=1}^N \sum_{i=1}^c (\mu_{ik})^w |y_k - v_{fi}|^2 \quad [16]$$

9. Cek kondisi berhenti:

a. Jika: $(|P_t(c) - P_{t-1}(c)| < \epsilon)$ atau $(t > \text{MaxIter})$ maka berhenti.

b. Jika tidak: $t = t+1$, ulangi langkah 4.

II.7 Noise Filtering

Noise filtering merupakan proses untuk menghilangkan *noise* yang mungkin terdapat pada citra hasil *scan*. *Noise* ini muncul karena kelemahan alat *scanner* yang mendeteksi debu/partikel-partikel kecil pada lembar formulir biodata saat formulir tersebut di-*scan*. Bisa juga karena debu yang terdapat pada

kaca scanner itu sendiri, sehingga pada hasil *scan*, *noise* ini dibaca sebagai suatu titik. *Noise* ini perlu dihilangkan untuk menghasilkan citra tersegmentasi dengan baik.

Jenis *filter* yang digunakan untuk menghilangkan *noise* pada tugas akhir ini adalah sebuah *spatial filter* 3x3 seperti gambar dibawah ini.

2	2	2
2	5	2
2	2	2

Gambar 2.11 *Filter Untuk Menghilangkan Noise*

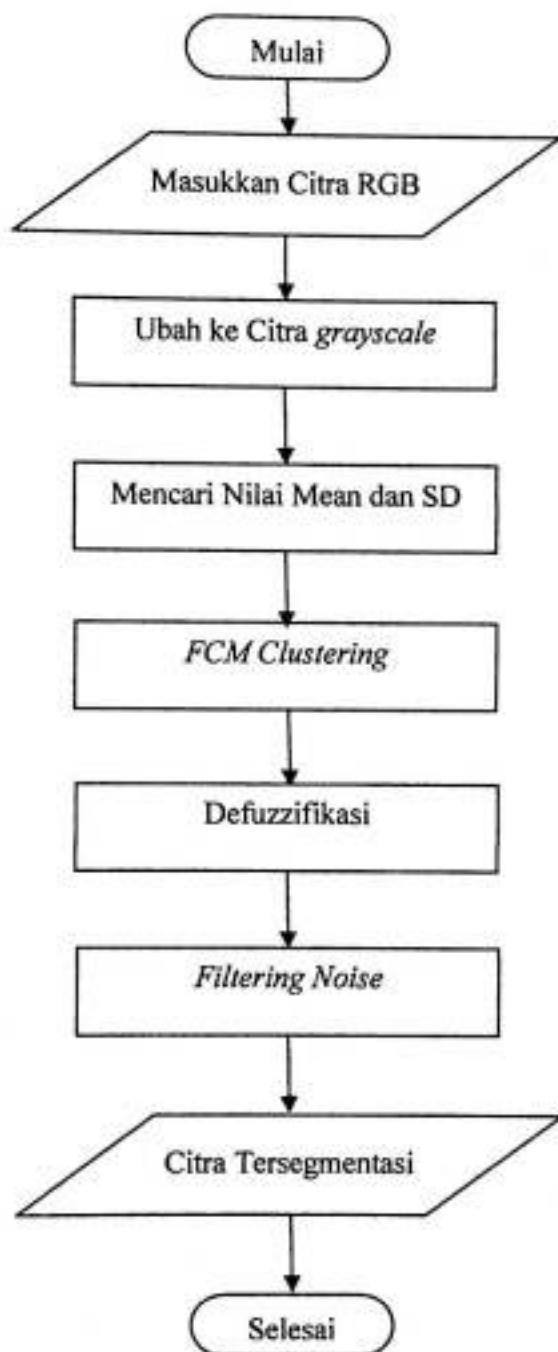
Untuk mengeliminasi *pixel noise* dilakukan dengan operasi konvolusi *filter* tersebut ke seluruh bagian citra. Dengan *filter* tersebut, pengelompokan *pixel* didasarkan pada nilai $f(x)$ berikut :

$$Pixel = \begin{cases} 1 & \text{jika } f(x) = 10 \\ 0 & \text{jika } f(x) < 10 \end{cases} \quad [17]$$

Jika $f(x)$ bernilai 10 maka akan dianggap sebagai *noise* dan selanjutnya dieliminasi sedangkan jika $f(x) < 10$ maka *pixel* tersebut dianggap bukan sebagai *noise*.

BAB III IMPLEMENTASI

Secara umum proses segmentasi ini dapat dilihat pada flowchart berikut:



Gambar 3.1 Flowchart Segmentasi Citra

Jadi proses kerja dari segmentasi citra ini dimulai dengan mengumpulkan data dari *pixel-pixel* citra, yang selanjutnya diproses dengan metode *Fuzzy C-Mean cluster*, lalu hasil dari *cluster* ini mengalami defuzzifikasi sehingga diperoleh citra foto seperti yang diinginkan.

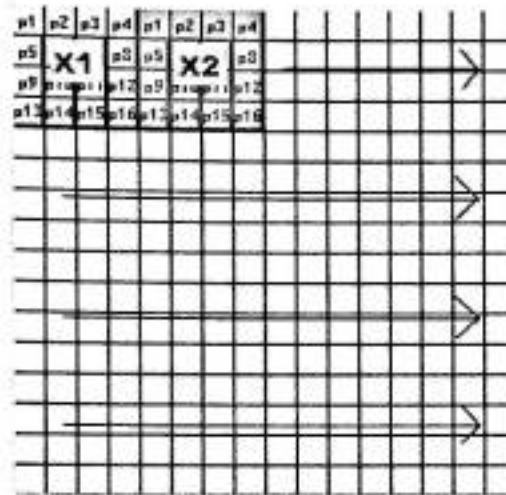
III.1 Pengumpulan Data dengan Mencari Nilai *Mean* dan Standar Deviasi

Dalam aplikasinya pada segmentasi ini, metode FCM menggunakan suatu fitur statistik yang sederhana dari sifat-sifat *pixel*. Di mana data-data yang akan diolah merupakan kumpulan *pixel-pixel* dalam satu kelompok. Untuk selanjutnya setiap *pixel-pixel* yang berada pada satu kelompok menjadi satu data yang akan diolah. Setiap data ini memiliki kriteria yang akan digunakan pada proses pengclusteran data yaitu nilai rata-rata (*mean*) dan standar deviasi dari pengukuran citra *grayscale*. Jadi citra RGB yang diperoleh dari hasil *scan* dikonversi dahulu ke citra *grayscale*.

Algoritma yang dilakukan untuk mendapatkan data-data *pixel* yang akan diolah adalah sebagai berikut:

1. Konversi Citra RGB ke citra *grayscale*. Hal ini dilakukan untuk memudahkan dalam pemrosesan citra karena nilai *pixel* dari 3 nilai RGB diubah menjadi 1 nilai *grayscale*.
2. Setelah citra RGB dikonversi ke citra *grayscale*, selanjutnya kelompokkan *pixel-pixel* tersebut sesuai dengan ketentuan yang diinginkan pada setiap formulir, di mana *pixel-pixel* ini berada dalam satu kelompok yang sama,

demikian seterusnya, hingga diperoleh data-data yang akan diproses. Hal tersebut dapat dilihat pada gambar berikut:



Gambar 3.2 Proses pengelompokkan *pixel* pada Bitmap Formulir

Sesudah *pixel* dikelompokkan, langkah selanjutnya adalah menghitung nilai *mean* dan standar deviasi dari masing-masing kelompok tersebut menggunakan persamaan berikut:

$$\bar{p} = \frac{1}{q} \sum_{i=1}^q p_i \quad [18]$$

$$s = \sqrt{\frac{1}{q-1} \sum_{i=1}^q (p_i - \bar{p})^2} \quad [19]$$

Keterangan:

p_i : nilai *pixel* tunggal.

q : banyaknya *pixel-pixel* dalam satu kelompok.

3. *Mean* dan standar deviasi yang diperoleh, selanjutnya menjadi nilai pada data x_i yang akan diolah lebih lanjut oleh *Fuzzy C-Mean cluster* untuk mendapatkan letak posisi foto dalam biodata ini.

Jadi secara umum diperoleh himpunan $R=\{\text{citra, teks, background}\}$ dengan subset dari R yaitu: $R_1=\{\text{citra}\}$, $R_2=\{\text{teks}\}$, $R_3=\{\text{background}\}$ sebagai himpunan fuzzy yang memiliki hubungan fungsi keanggotaan fuzzy sebagai berikut:

$$\mu_{ik} : X \rightarrow [0,1] \quad [20]$$

di mana X merupakan nilai dari kelompok *pixel* yang termasuk pada R , dan μ_{ik} merupakan derajat keanggotaan dari kelompok *pixel* k dalam daerah R_i , $i=1,2,3$

III.2 Penerapan Algoritma *Fuzzy C-Mean Clustering* pada Segmentasi

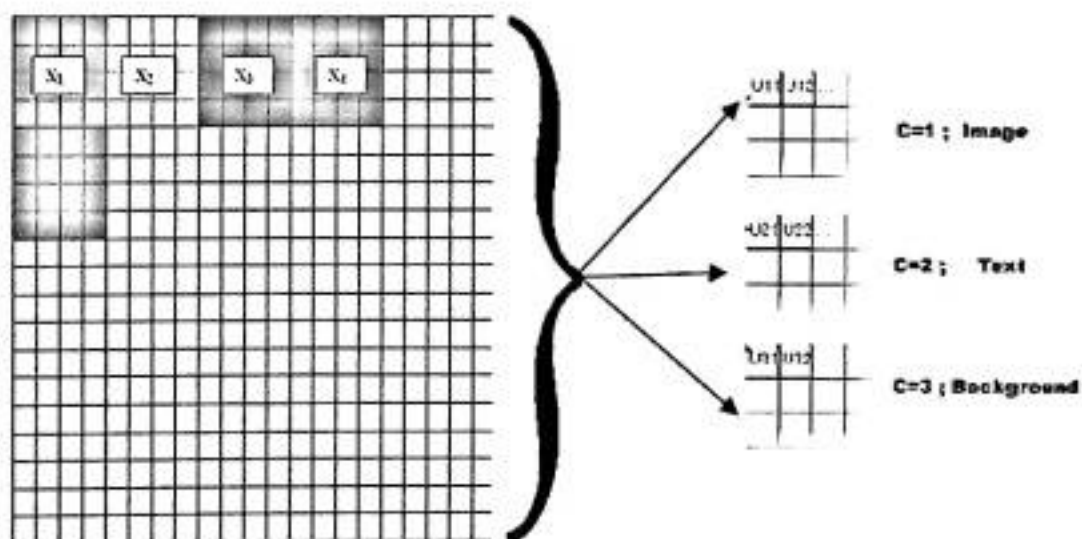
Metode *Fuzzy C-Mean (FCM) Cluster* adalah salah satu metode *clustering* yang memanfaatkan sekelompok data di mana data yang sama karakteristiknya dikelompokkan dalam satu kelompok, sehingga semua anggota dari setiap partisi mempunyai persamaan berdasarkan matrik tertentu. Ide dasar yang digunakan dalam metode ini adalah bagaimana menentukan ukuran kedekatan sifat antar data untuk dikelompokkan dalam satu *cluster*.

Secara umum proses dimulai dengan merumuskan masalah *pengclustering* dengan mendefinisikan variabel-variabel yang digunakan untuk dasar *pengclustering* atau pembentukan *cluster*. Konsep dasar pengukuran analisis *cluster* adalah pengukuran jarak pisah antar obyek, sedangkan *similarity* adalah ukuran kesamaan selain jarak. Konsep ini penting karena pengelompokkan pada analisis *cluster* didasarkan pada kedekatan jaraknya.

Algoritma *Fuzzy C-Mean (FCM)* adalah metode pengelompokkan yang bertujuan mengelompokkan data sedemikian hingga jarak setiap data ke pusat

cluster dalam satu *cluster* adalah minimum. Dalam metode ini sejumlah c *cluster* yang dikehendaki ditentukan terlebih dahulu. Setelah itu tentukan pusat *cluster* yang akan menandai lokasi rata-rata (*mean*) untuk tiap-tiap *cluster*. Pada kondisi awal, pusat *cluster* ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan untuk tiap-tiap *cluster*. Dengan cara memperbaiki pusat *cluster* (*cluster center*) dan derajat keanggotaan tiap-tiap data secara berulang, maka dapat dilihat bahwa pusat *cluster* akan menuju lokasi yang tepat. Perulangan ini didasarkan pada minimasi fungsi obyektif yang menggambarkan jarak dari titik data yang diberikan ke pusat *cluster* yang terbobot oleh derajat keanggotaan titik data tersebut.

Dalam skripsi ini, c sama dengan 3. Setiap baris dari matriks U mewakili derajat keanggotaan dari seluruh kelompok-kelompok yang termasuk pada daerah R_1 , R_2 , dan R_3



Gambar 3.3 Banyaknya Cluster yang Dibentuk pada Bitmap Formulir

Jadi setelah nilai *mean* dan standar deviasi diperoleh dari *pixel-pixel bitmap* formulir biodata maka data yang akan diolah adalah:

1. $X = \{x_1, x_2, x_3, \dots, x_n\}$. Di mana n merupakan banyaknya kelompok-kelompok *pixel* yang telah ditentukan.
2. Jika ukuran citra $L \times M$ maka $n = (L/p) \times (M/p)$. Di mana p adalah kelompok *pixel*.
3. Setiap data x_i terdiri dari nilai *mean* dan standar deviasi.

Banyaknya cluster yang akan dibentuk pada data di atas adalah 3, sehingga matriks partisi datanya didefinisikan sebagai berikut:

$$\mu_f(c) = \begin{bmatrix} \mu_{11}[x_1] & \mu_{12}[x_2] & \dots & \mu_{1N}[x_N] \\ \mu_{21}[x_1] & \mu_{22}[x_2] & \dots & \mu_{2N}[x_N] \\ \mu_{31}[x_1] & \mu_{32}[x_2] & \dots & \mu_{3N}[x_N] \end{bmatrix} \quad [21]$$

dengan

$$\sum_{i=1}^3 \mu_{ik} = 1 \quad [22]$$

yang berarti bahwa jumlah nilai keanggotaan suatu data pada semua *cluster* harus sama dengan 1.

Fungsi obyektif iterasi ke- t $P(c)$ pada matriks partisi adalah:

$$\bar{P}_t(c) = \sum_{k=1}^N \sum_{i=1}^3 (\mu_{ik})^{1.25} |x_k - v_{ij}|^2 \quad [23]$$



dengan v_{fi} adalah pusat vektor data pada *cluster fuzzy* ke- i ,

$$v_{fi} = \frac{\sum_{k=1}^N (\mu_{ik})^{1.25} x_k}{\sum_{k=1}^N (\mu_{ik})^{1.25}} \quad [24]$$

dan w adalah bobot pada nilai-nilai keanggotaan, $|x_k - v_{fi}|$ adalah bentuk norm Euclidean yang digunakan sebagai jarak antara x_k dan v_{fi} .

Algoritma Fuzzy C-Mean:

1. Bentuk matriks partisi awal $\mu_f(c)$, matriks partisi awal dibuat secara acak dengan ketentuan 3 *cluster* untuk kasus ini.

$$\mu_f(c) = \begin{bmatrix} \mu_{11}[x_1] & \mu_{12}[x_2] & \cdots & \mu_{1N}[x_N] \\ \mu_{21}[x_1] & \mu_{22}[x_2] & \cdots & \mu_{2N}[x_N] \\ \mu_{31}[x_1] & \mu_{32}[x_2] & \cdots & \mu_{3N}[x_N] \end{bmatrix} \quad [25]$$

dengan $\sum_{i=1}^3 \mu_{ik} = 1$

2. Tentukan:
 - a. Parameter kontrol fuzzy/pembobot (w);
 - i. $w = 1.25$ untuk percobaan pertama
 - ii. $w = 2$ untuk percobaan kedua.

Kedua nilai w diatas untuk melihat sifat kefuzzyannya dalam aplikasi.

- b. Maksimum iterasi (MaxIter = 100);
 - c. Kriteria penghentian ($\epsilon = 10^{-12}$);
 - d. Iterasi awal ($t=1$);
3. Hitung fungsi obyektif awal : $P_f(c)$ dari matriks partisi awal;

4. Naikkan nomor iterasi: $t = t + 1$;
5. Hitung pusat vektor tiap-tiap *cluster* untuk matrik partisi tersebut sebagai berikut:

$$v_{\beta} = \frac{\sum_{k=1}^N (\mu_{ik})^{1.25} x_k}{\sum_{k=1}^N (\mu_{ik})^{1.25}} \quad [26]$$

6. Modifikasi tiap-tiap nilai keanggotaan sebagai berikut:

- a. Jika $y_k \neq v_{\beta}$,

$$\mu_{ik}(y_k) = \left[\sum_{g=1}^c \left(\frac{|x_k - v_{\beta}|^2}{|x_k - v_{\beta}|^2} \right)^{\frac{1}{(1.25-1)}} \right]^{-1} \quad [27]$$

- b. Jika $y_k = v_{\beta}$,

$$\mu_{ik}(y_k) = 1, \text{ jika } i=g; \quad [28]$$

$$\mu_{ik}(y_k) = 0, \text{ jika } i \neq g. \quad [29]$$

7. Modifikasi matriks partisi sebagai berikut:

$$\mu_f(c) = \begin{bmatrix} \mu_{11}[x_1] & \mu_{12}[x_2] & \cdots & \mu_{1N}[x_N] \\ \mu_{21}[x_1] & \mu_{22}[x_2] & \cdots & \mu_{2N}[x_N] \\ \mu_{31}[x_1] & \mu_{32}[x_2] & \cdots & \mu_{3N}[x_N] \end{bmatrix} \quad [30]$$

8. Hitung fungsi obyektif:

$$P_t(c) = \sum_{k=1}^N \sum_{i=1}^3 (\mu_{ik})^{1.25} |y_k - v_{\beta}|^2 \quad [31]$$

9. Cek kondisi berhenti:

- a. Jika: $(|P_t(c) - P_{t-1}(c)| < \epsilon)$ atau $(t > \text{MaxIter})$ maka berhenti.
- b. Jika tidak: $t = t+1$, ulangi langkah 4.

III.3 Defuzzifikasi

Defuzzifikasi adalah proses pemetaan balik dari citra yang mengalami proses *fuzzy* ke citra sebenarnya (citra RGB). Proses-proses yang dilakukan pada defuzzifikasi antara lain:

1. Semua kumpulan *pixel* dianggap milik salah satu daerah yang telah ditetapkan, dalam hal ini *cluster* teks, citra, dan *background*.
2. Prosesnya dimulai dengan mengubah matriks $\mu_f(c)$ yang terakhir kembali ke citra dengan memutuskan x_k termasuk pada salah satu *cluster*.
3. Cara menetapkan x_k pada *cluster* i didasarkan pada seberapa tinggi nilai μ_{ik} dalam *cluster* tersebut.
4. x_k akan diwarnai dengan kode warna yang sama untuk *cluster* i , di mana citra yang diperoleh, diklasifikasikan berdasarkan daerah kode warnanya.
5. Langkah selanjutnya dari defuzzifikasi adalah memetakan daerah kode warna ke daerah citra asli.
6. Langkah akhir dari defuzzifikasi adalah menghubungkan tiap *cluster* pada kategori citra yang cocok, yaitu teks, citra, dan *background*.
 - a. Untuk *cluster* i yang memiliki v_{fi} dengan nilai standar deviasi *medium* menjadi citra.
 - b. Untuk *cluster* i yang memiliki v_{fi} dengan nilai standar deviasi terbesar menjadi teks.
 - c. Untuk *cluster* i yang memiliki v_{fi} dengan nilai standar deviasi terkecil menjadi *background*.

7. Pada proses defuzzifikasi ini hanya menggambarkan *cluster* yang memiliki nilai standar deviasi *medium*, untuk *cluster* yang memiliki nilai standar deviasi tinggi dan rendah, tidak digambarkan.

III.4 Penerapan *Noise Filtering*

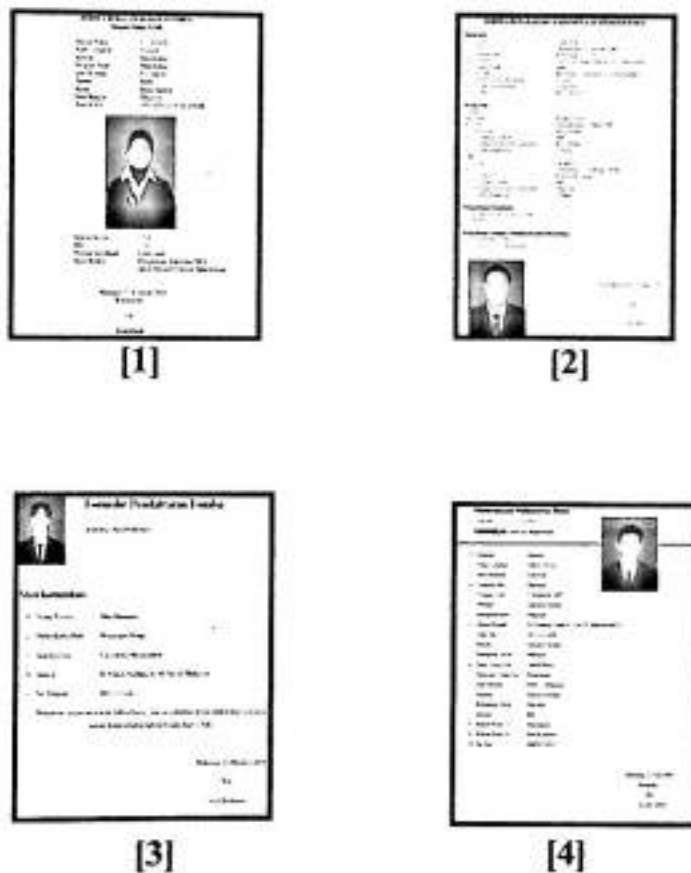
Pada proses ini yang dilakukan adalah menghilangkan *noise* pada citra hasil defuzzifikasi, sehingga diperoleh citra hasil ekstrak dengan kualitas yang baik.

BAB IV

HASIL DAN PEMBAHASAN

Hasil dari proses segmentasi citra ini diperoleh suatu foto yang disimpan dalam ekstensi JPG. Di mana prosesnya menggunakan 4 formulir biodata dengan letak foto yang berbeda-beda. Biodata ini setelah di-*scan* tersimpan dalam ekstensi JPG.

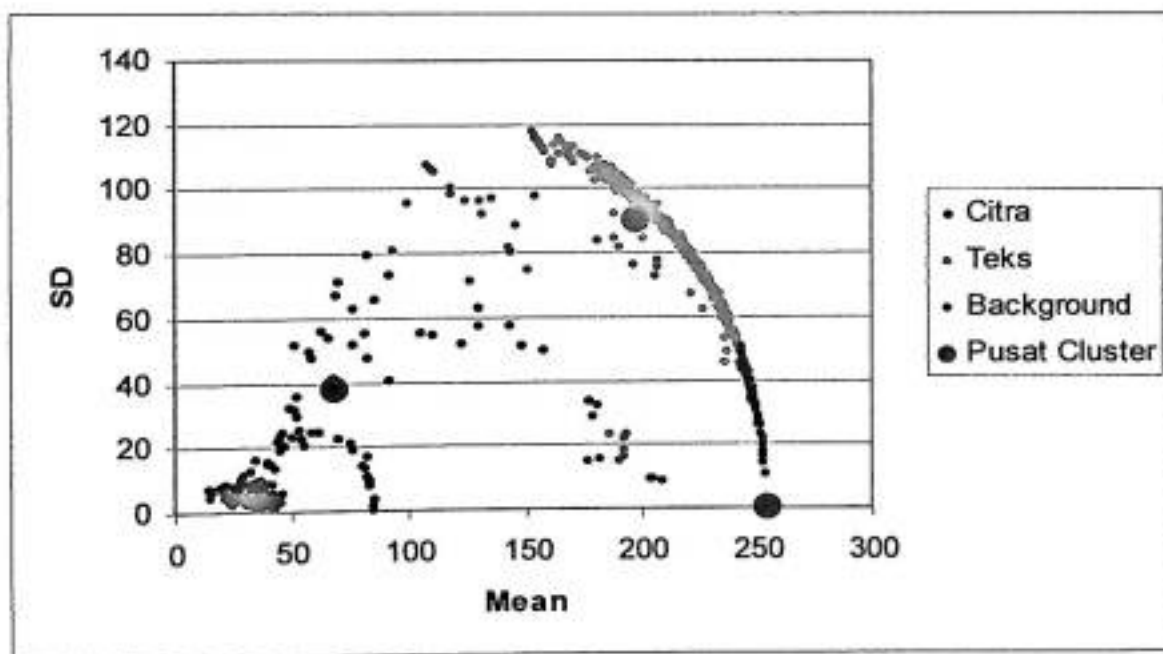
Adapun contoh biodata tersebut nampak pada gambar di bawah ini:



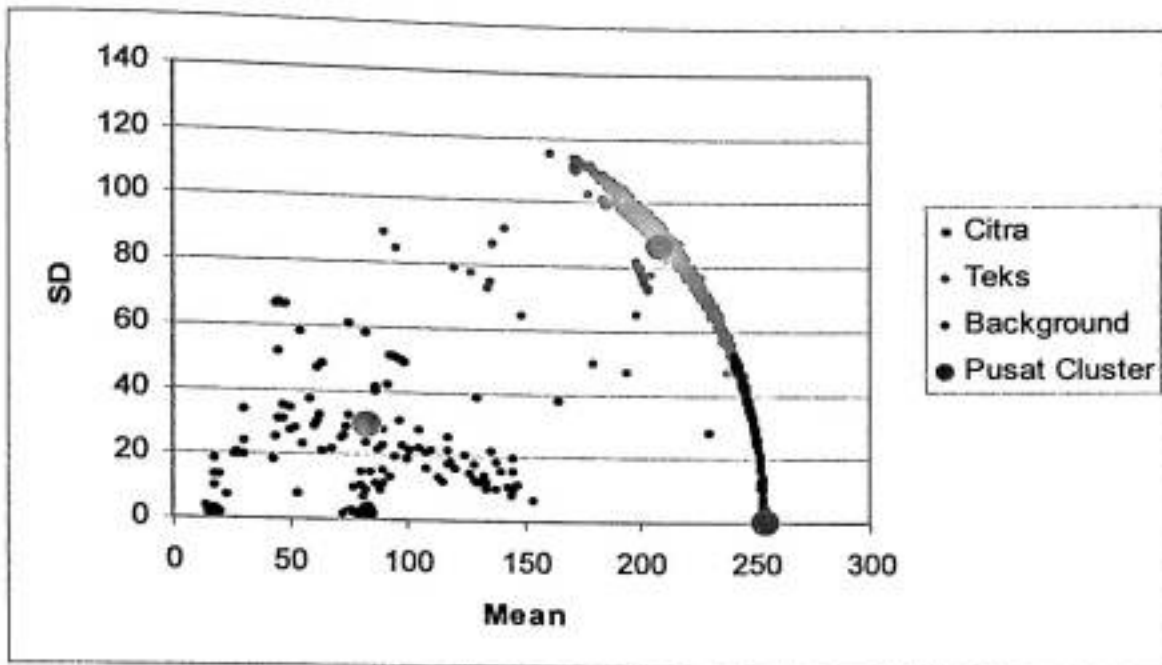
Gambar 4.1 Contoh *Bitmap* Formulir Biodata

Gambar 4.1, merupakan bentuk-bentuk formulir biodata yang digunakan dalam skripsi ini, pada bagian 1 adalah bentuk formulir biodata dengan posisi foto terletak di tengah, diapit antara teks. Sedangkan bagian 2, merupakan bentuk formulir biodata dengan posisi foto terletak pada sudut kanan bawah, Demikian juga pada bagian 3, yang merupakan bentuk formulir biodata dengan posisi foto terletak di sudut kanan atas, dan pada bagian 4 merupakan bentuk formulir biodata yang terletak pada sudut kiri atas. Letak foto dalam keempat biodata tersebut dianggap dapat mewakili jenis-jenis biodata yang secara umum berada dalam masyarakat.

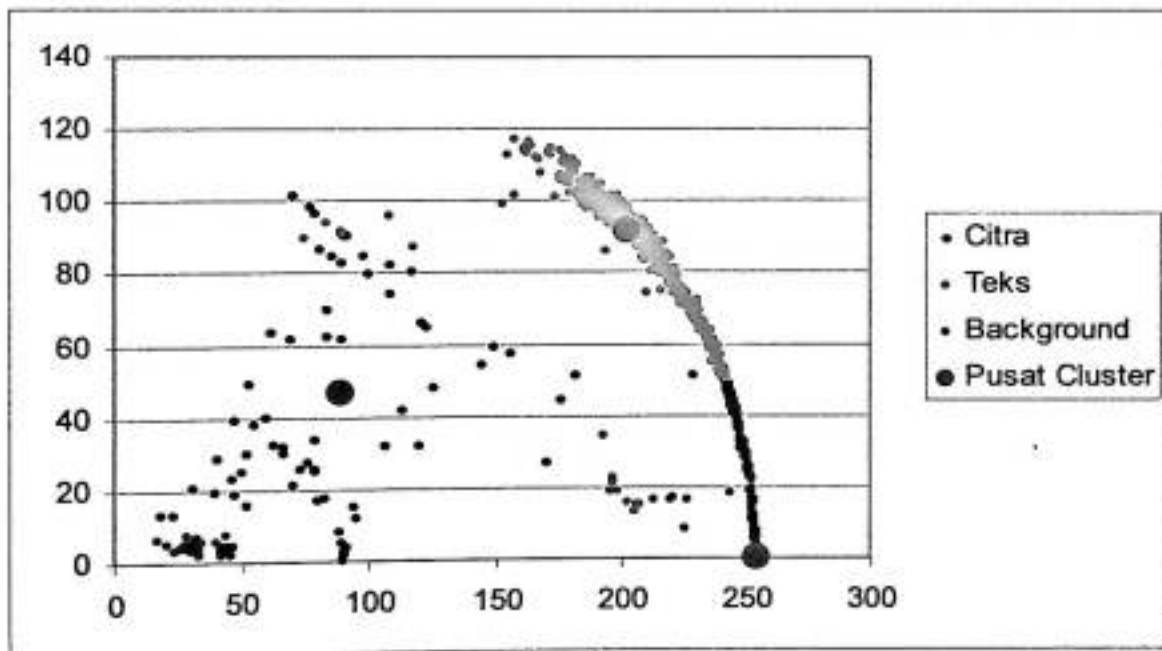
Setelah diproses dengan *Fuzzy C-Mean cluster* maka diperoleh hasil pengclusteran seperti pada gambar 4.2 sampai gambar 4.5, yaitu letak-letak *cluster* yang ditampakkan dalam bentuk grafik sebagai berikut:



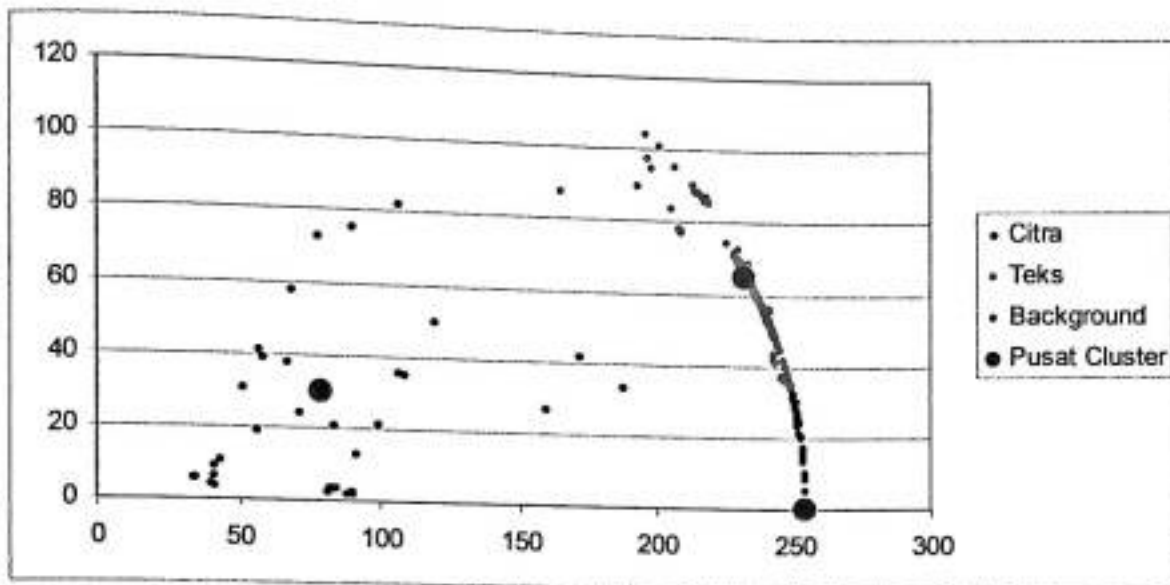
Gambar 4.2 Grafik Hasil Pengclusteran dengan *Fuzzy C-Mean cluster* pada Formulir Biodata 1



Gambar 4.3 Grafik Hasil Pengclusteringan dengan *Fuzzy C-Mean cluster* pada Formulir Biodata 2



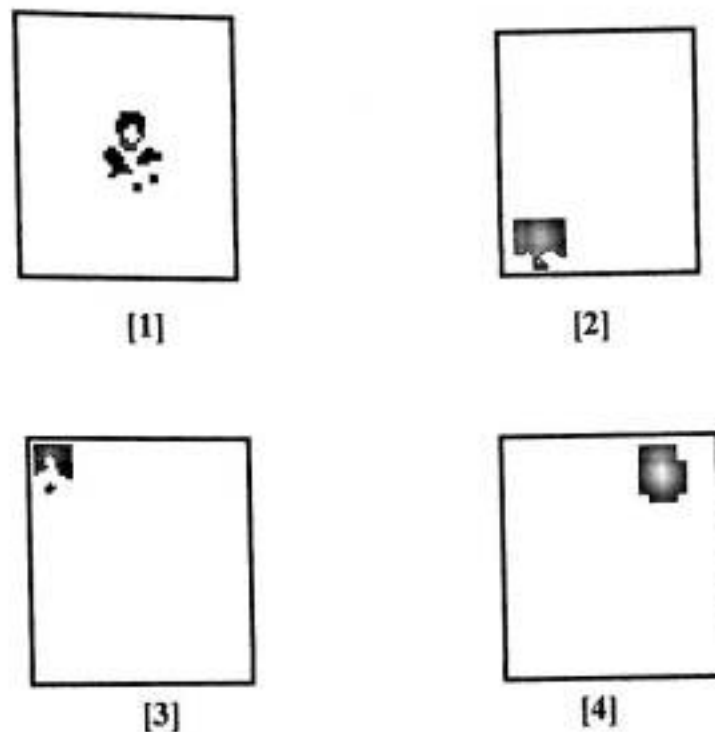
Gambar 4.4 Grafik Hasil Pengclusteringan dengan *Fuzzy C-Mean cluster* pada Formulir Biodata 3



Gambar 4.5 Grafik Hasil Pengclustering dengan *Fuzzy C-Mean cluster* pada **Formulir Biodata 4**

Jadi nampak dalam grafik letak *cluster* teks memiliki nilai standar deviasi yang tinggi, dan *cluster background* memiliki nilai standar deviasi yang rendah, sedangkan daerah *cluster* citra berada pada posisi *medium*.

Selanjutnya daerah yang digambarkan hanyalah *cluster* yang memiliki nilai standar deviasi *medium*, yaitu daerah *cluster* citra. Hasil dari pencitraan *cluster medium* ini nampak pada gambar berikut:



Gambar 4.6 Pencitraan *Cluster* yang Memiliki Nilai Standar Deviasi *Medium*

Demikianlah proses pendeteksian letak foto dalam formulir biodata yang dilakukan oleh *Fuzzy C-Mean Cluster*. Setelah letak foto tersebut diperoleh, kemudian didefuzzifikasi dan *filtering noise*.

Sudah dijelaskan pada bab sebelumnya bahwa proses defuzzifikasi adalah proses memetakan kembali citra hasil *Fuzzy C-Mean cluster* ke citra RGB, dan *filtering noise* ini memiliki tujuan menghilangkan *noise* yang ada pada sekeliling foto biodata, akibat debu atau partikel kecil lainnya saat formulir biodata tersebut di-*scan*. Setelah citra foto ini didefuzzifikasi dan *noise filtering*, maka diperoleh hasil citra tersegmentasi, yang selanjutnya dapat diekstrak seperti gambar 4.7 berikut:



[1]



[2]



[3]



[4]

Gambar 4.7 Hasil Pengekstrakkan Citra Foto Biodata yang Tersegmentasi

Jadi pada gambar 4.7, nampak citra foto hasil segmentasi dari masing-masing formulir biodata. Penjelasan mengenai hasil cluster dapat dilihat pada tabel berikut:

Tabel 4.1 Hasil Clustering Formulir 1

Ukuran pengelompokan <i>pixel</i>	40 x 40 <i>pixel</i>	40 x 40 <i>pixel</i>
Parameter <i>fuzzy</i> /pembobot	1.25	2
Kriteria penghentian iterasi (ϵ)	10^{-12}	10^{-12}
Banyaknya iterasi	24 iterasi	32 iterasi
Pusat <i>cluster</i> citra	[68.4908,37.3457]	[68.7441,37.9739]
Pusat <i>cluster</i> teks	[198.7812,89.9808]	[200.0611,89.8578]
Pusat <i>cluster background</i>	[254.997,0.0313]	[254.9392,0.1004]
Banyaknya <i>cluster</i>	3 <i>cluster</i>	3 <i>cluster</i>
Banyaknya percobaan	20 kali	20 kali
Banyaknya yang sukses	19 kali	20 kali
Banyaknya yang gagal	1 kali	-
Banyaknya iterasi yang sukses	19, 20, 23, 24, 25, 27, 28, 30, 31, 32, 33, 34 iterasi	32, 45, 46, 48, 49, 50, 51, 52, 53 iterasi
Banyaknya iterasi yang gagal	45 iterasi	-
Persentase kesuksesan aplikasi	95 %	100 %

Hasil penelitian: 2009

Tabel 4.2 Hasil Clustering Formulir 2

Ukuran pengelompokan <i>pixel</i>	38 x 38 <i>pixel</i>	38 x 38 <i>pixel</i>
Parameter <i>fuzzy</i> /pembobot	1.25	2
Kriteria penghentian iterasi (ϵ)	10^{-12}	10^{-12}
Banyaknya iterasi	19 iterasi	26 iterasi
Pusat <i>cluster</i> citra	[82.6312,29.5238]	[80.7697,30.0592]
Pusat <i>cluster</i> teks	[210.1603,86.1603]	[209.5759,86.5867]
Pusat <i>cluster background</i>	[254.984,0.1749]	[254.8346,0.329]
Banyaknya <i>cluster</i>	3 <i>cluster</i>	3 <i>cluster</i>
Banyaknya percobaan	20 kali	20 kali
Banyaknya yang sukses	19 kali	20 kali
Banyaknya yang gagal	1 kali	-
Banyaknya iterasi yang sukses	17, 19, 20, 21, 22, 23, 24 25, 26 iterasi	22, 23, 24, 25, 26, 27, 30 iterasi
Banyaknya iterasi yang gagal	34 iterasi	-
Persentase kesuksesan aplikasi	95%	100%

Hasil penelitian: 2009

Tabel 4.3 Hasil Clustering Formulir 3

Ukuran pengelompokan <i>pixel</i>	30 x 30 <i>pixel</i>	30 x 30 <i>pixel</i>
Parameter <i>fuzzy</i> /pembobot	1.25	2
Kriteria penghentian iterasi (ϵ)	10^{-12}	10^{-12}
Banyaknya iterasi	13 iterasi	61 iterasi
Pusat <i>cluster</i> citra	[89.5269,46.6712]	[83.1137,42.0154]
Pusat <i>cluster</i> teks	[203.105,91.7738]	[202.7987,91.9057]
Pusat <i>cluster background</i>	[254.9709,0.1856]	[254.94,0.1882]
Banyaknya <i>cluster</i>	3 <i>cluster</i>	3 <i>cluster</i>
Banyaknya percobaan	20 kali	20 kali
Banyaknya yang sukses	16 kali	19 kali
Banyaknya yang gagal	4 kali	1 kali
Banyaknya iterasi yang sukses	13, 22, 23, 24, 26, 27, 28, 29, 30 iterasi	56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66 iterasi
Banyaknya iterasi yang gagal	83, 66, 65, 80 iterasi	53 iterasi
Persentase kesuksesan aplikasi	80%	95%

Hasil penelitian: 2009

Tabel 4.4 Hasil Clustering Formulir 4

Ukuran pengelompokan <i>pixel</i>	100 x 100 <i>pixel</i>	100 x 100 <i>pixel</i>
Parameter <i>fuzzy</i> /pembobot	1.25	2
Kriteria penghentian iterasi (ϵ)	10^{-12}	10^{-12}
Banyaknya iterasi	15 iterasi	21 iterasi
Pusat <i>cluster</i> citra	[79.4162,30.1237]	[82.3892,31.568]
Pusat <i>cluster</i> teks	[232.2624,64.965]	[232.738,64.8918]
Pusat <i>cluster background</i>	[254.9913,0.1399]	[254.9305,0.2473]
Banyaknya <i>cluster</i>	3 <i>cluster</i>	3 <i>cluster</i>
Banyaknya percobaan	20 kali	20 kali
Banyaknya yang sukses	19 kali	20 kali
Banyaknya yang gagal	1 kali	-
Banyaknya iterasi yang sukses	12, 13, 14, 15, 16, 18, 26 iterasi	19, 20, 21, 22, 23, 24, 35 iterasi
Banyaknya iterasi yang gagal	38 iterasi	-
Persentase kesuksesan aplikasi	95%	100%

Hasil penelitian: 2009

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Selama penyusunan skripsi ini kesimpulan-kesimpulan yang diperoleh:

1. Metode *Fuzzy C-Mean cluster* memproses formulir biodata yang telah di-scan dengan menggunakan nilai *mean* dan standar deviasi dari kelompok *pixel* yang dibentuk. Di mana hasil *scan* formulir ini berupa citra yang terdiri dari *pixel-pixel*, dan *pixel-pixel* tersebut kemudian dikelompokkan sesuai dengan yang diinginkan. Dalam tiap kelompok ini terdapat nilai *mean* dan standar deviasinya, nilai-nilai inilah yang digunakan sebagai data untuk menentukan posisi foto pada formulir biodata.
2. Dalam proses segmentasi citra ini data-data *pixel* diproses dengan metode *Fuzzy C-Mean cluster* untuk menentukan *cluster-clusternya*, selanjutnya hasil *cluster* yang memiliki standar deviasi menengah (*medium*) di defuzzifikasi sehingga diperoleh hasil berupa foto dalam bentuk citra RGB. Sesudah foto diperoleh, proses selanjutnya adalah *filtering noise*, yaitu membersihkan *noise* yang terdapat di sekitar foto agar diperoleh hasil yang bagus.
3. Proses pengekstrakan foto ini dapat dilakukan setelah melalui defuzzifikasi dan *filtering noise*. File foto dapat disimpan dalam dokumen lain atau dapat langsung dimasukkan dalam sebuah *database*.
4. Aplikasi yang telah dirancang, dapat mengklasifikasikan foto dengan benar pada bentuk formulir yang letak fotonya berbeda-beda.

V.2 Saran

Adapun saran-saran setelah proses penyusunan skripsi ini adalah:

1. Pada penelitian selanjutnya, agar memperoleh data *pixel* yang lebih baik untuk diolah, dapat menggunakan fitur-fitur statistik tambahan lainnya selain *mean* dan standar deviasi dari *pixel* citra, sehingga proses pengelompokkan *pixel-pixel* dapat dilakukan dengan lebih baik lagi.
2. Untuk penelitian tingkat lanjut, sebaiknya data yang diekstrak tidak hanya berfokus pada bagian citranya saja, melainkan dengan mengambil bagian teksnya juga, sehingga proses registrasi dapat berlangsung dengan lebih baik.
3. Pada perancangan aplikasi tingkat tinggi (*High-Tech*), penelitian ini dapat dijadikan landasan pengembangan suatu proyek. Di mana formulir biodata yang di-*scan*, fotonya dapat langsung masuk dalam sebuah database di suatu instansi.

DAFTAR PUSTAKA

- Castleman, K.R., 1996. *Digital Image Processing*. Prentice Hall, Englewood Cliffs, New Jersey.
- Chuai-Aree, Somporn Cs., *Fuzzy C-Mean: A Statistical Feature Classification of Text and Image Segmentation Method* from URL : http://pioneer.chula.ac.th/~sperapho/pub/fuzzy_c_mean.pdf www.geocities.com/mathswu/seminar/present.pdf (Diakses terakhir tanggal 10 Juni 2009).
- Jaowry, Steven., 2007. *Pembuatan Perangkat Lunak Pengenalan Huruf Cetak Menggunakan Metode Jaringan Saraf Tiruan* from URL http://digilib.petra.ac.id/viewer.php?page=1&submit.x=0&submit.y=0&qual=high&fname=/jiunkpc/sl/elkt/2007/jiunkpe-ns-sl-2007-23403051-4669-c_means-references.pdf (Diakses terakhir tanggal 30 Juni 2009).
- Jiawei, H., Kamber, M., 2001. *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers
- Luthfi, Emha Taufiq., 2007, *Fuzzy C-Means untuk Clustering Data (Studi Kasus : Data Performance Mengajar Dosen)* from URL : <http://p3m.amikom.ac.id/p3m/26%20-%20FUZZY%20C-MEANS%20UNTUK%20CLUSTERING%20DATA%20-%20AMIKOM.pdf> (Diakses terakhir tanggal 20 September 2009).
- Munir, Rinaldi., 2005. *Pengantar Pengolahan citra* from URL: http://webmail.informatika.org/~rinaldi/Buku/Pengolahan%20Citra%20Digital/Bab-1_Pengantar%20Pengolahan%20Citra.pdf (Diakses terakhir tanggal 20 Agustus 2009).
- Pratomo, Herman. *Model Warna RGB* from URL: http://en.wikipedia.org/wiki/RGB_color_model; http://en.wikipedia.org/wiki/Hex_triplet#Hex_triplet (Diakses terakhir tanggal 1 Agustus 2009).
- Yoyokits. *Model Warna CIE-RGB* from URL : <http://www.cekli.com/en/content/model> warna cie rgb. (Diakses terakhir tanggal 17 Juli 2009).
- Zimmerman, H.J., 1991, *Fuzzy Set Theory and Its Applications*, 2nd Ed., Norwel, Massachusetts

L
A
M
P
I
R
A
N

**FORMULIR DATA WISUDAWAN FMIPA
Sarjana Sains (S.SI)**

Nomor Pokok : H11105038
Nama Lengkap : Hadijah
Jurusan : Matematika
Program Studi : Matematika
Jenis Kelamin : Perempuan
Agama : Islam
Status : Belum Kawin
Suku Bangsa : Makassar
Asal SLTA : SMA IRNAS MAKASSAR



Nomor Alumni : 2890
IPK : 3,56
Predikat Kelulusan : Cum Laude
Judul Skripsi : Penggunaan Algoritma DNA
untuk Masalah Lintasan Hamiltonian.

Makassar, 23 Februari 2010
Wisudawan

Ttd

HADIJAH

LAMPIRAN
Data Pada Formulir 1

Mean	SD	Cluster
255	0	2
194.3706	101.5271	1
185.8594	106.3911	1
163.3644	113.374	1
181.2888	109.641	1
193.7394	100.1885	1
188.0738	106.27	1
177.4413	109.6654	1
202.4431	96.9596	1
188.7056	105.4518	1
205.8138	94.2941	1
211.8219	89.5706	1
184.2956	107.4668	1
195.7419	99.6081	1
168.3519	112.7669	1
203.0994	94.4871	1
191.4444	103.2965	1
190.4425	104.208	1
198.5331	97.2377	1
189.6375	104.1602	1
207.1856	94.4411	1
180.8844	106.3372	1
170.9375	113.0575	1
194.5	102.3859	1
211.8094	89.5465	1
255	0	2
246.4431	43.8863	2
241.9894	51.3393	1
236.7688	62.8358	1
244.2019	48.4155	2
241.095	53.6098	1
234.5875	66.5961	1
228.9625	72.6897	1
236.4231	61.8407	1
236.0738	62.6407	1
244.2513	48.3563	2
242.0044	53.0182	1
237.095	59.4514	1
236.0575	62.3688	1
240.3119	55.1311	1
225.9188	74.928	1
231.5981	67.2933	1
221.7275	79.1218	1
242.46	50.4719	1
242.1756	52.6078	1
246.7888	42.4429	2
245.4194	44.9835	2
238.4681	60.1261	1

Mean	SD	Cluster
183.8963	102.8613	1
198.5875	96.4767	1
220.8338	80.9042	1
165.5063	110.8909	1
199.4844	96.4714	1
210.6394	89.0681	1
207.6806	90.2196	1
178.4169	105.5139	1
210.4019	89.0975	1
192.0244	100.0341	1
194.5813	99.0542	1
192.6206	99.1565	1
198.46	97.2021	1
192.4781	100.3884	1
254.9588	0.2511	2
254.99	0.0995	2
254.9675	0.3185	2
254.975	0.1561	2
254.9675	0.294	2
254.9606	0.354	2
254.99	0.0995	2
254.9275	0.4661	2
227.2681	61.9632	1
181.9881	83.4022	1
207.2194	75.484	1
190.575	81.7569	1
85	0	2
144.05	80.4096	1
190.575	81.7569	1
85	0	2
85	0	2
84.9363	1.0593	2
79.7744	13.6222	0
81.7775	10.6139	0
84.9788	0.1645	2
144.05	80.4096	1
190.575	81.7569	1
74.8569	20.5492	0
45.0931	18.5098	0
40.745	2.3369	2
43.3	3.4127	2
55.1888	20.1256	0
82.4575	10.6907	0
144.05	80.4096	1
190.575	81.7569	1
43.6563	21.4654	0
40.53	14.7347	0
75.3106	51.5382	0

Mean	SD	Cluster
22.8344	3.8571	2
16.4819	6.1018	2
25.02	5.1031	2
34.37	2.2095	2
35.8063	1.7223	2
34.3481	1.7394	2
33.7319	3.5343	2
30.6475	8.1809	0
27.3444	6.7343	0
30.2388	3.3441	2
30.0781	3.2478	2
21.3131	6.2233	2
24.8125	3.6841	2
31.0963	1.8188	2
110.0019	105.9605	1
169.0719	110.6624	1
24.1363	3.2369	2
15.2363	3.8598	2
20.0394	7.9775	0
24.6469	3.4838	2
30.9431	3.1157	2
33.86	6.4173	0
33.4031	2.8545	2
36.25	9.8563	0
26.01	6.7251	0
24.6019	1.7985	2
23.48	2.6064	2
14.1856	7.2329	0
24.53	4.6605	2
27.4569	3.2362	2
107.8569	107.586	1
255	0	2
217.7888	84.939	1
156.8688	113.7651	1
155.0744	115.8455	1
154.6681	116.3021	1
156.4694	114.1363	1
157.8725	112.5578	1
161.4619	108.5042	1
161.9313	107.9507	1
162.41	107.501	1
159.2119	111.126	1
158.3438	112.1069	1
156.9325	113.6822	1
153.4856	117.7031	1
155.6875	115.1007	1
156.6469	114.051	1
192.5581	100.5603	1

244.4538	47.8949	2
243.5956	50.1436	2
226.5038	75.5142	1
186.9813	104.1631	1
176.4844	110.2617	1
169.7181	112.5839	1
195.88	100.6542	1
196.4038	100.3779	1
174.3313	110.9238	1
182.8113	106.5759	1
206.4944	93.6112	1
205.0438	94.5387	1
192.3744	103.956	1
226.7013	76.1824	1
229.9925	69.7553	1
244.8494	44.4038	2
242.985	49.3651	1
245.3294	44.6281	2
226.0925	74.4612	1
237.7531	58.3944	1
237.0019	59.646	1
253.2919	17.083	2
249.8531	30.4515	2
230.1219	70.0028	1
233.9975	65.9335	1
234.1219	65.0285	1
226.9294	73.4549	1
223.965	77.0013	1
235.0406	62.4132	1
225.7019	75.9582	1
213.5031	86.9042	1
211.7419	89.0328	1
227.7156	73.3749	1
219.4369	81.7658	1
211.9588	87.9097	1
204.3613	93.761	1
251.5163	27.5483	2
248.145	34.7167	2
219.0188	83.911	1
226.7863	75.4725	1
227.0931	74.4406	1
224.7775	75.2539	1
216.5588	83.7064	1
225.4869	75.0558	1
200.6919	95.7247	1
200.6981	94.1732	1
192.6363	100.2963	1
214.9688	85.1289	1
203.445	93.1661	1
197.74	95.7336	1
182.885	104.1378	1
211.6444	87.3268	1

66.8594	40.6843	0
42.435	2.9057	2
58.6125	24.609	0
144.05	80.4096	1
190.575	81.7569	1
83.5169	7.8048	0
35.1006	8.4122	0
130.2169	56.8151	0
209.4606	8.4151	2
194.5094	23.1963	1
81.5894	47.6959	0
46.9563	20.2093	0
144.05	80.4096	1
190.575	81.7569	1
81.1363	13.1285	0
65.3669	53.7424	0
157.8894	49.6052	0
193.2856	15.8215	1
177.3238	33.2448	0
143.1506	56.8836	0
45.6263	24.5087	0
144.05	80.4096	1
190.575	81.7569	1
84.6419	3.4126	0
126.3844	70.9924	1
204.9144	9.3845	2
193.1644	18.2484	1
193.1544	21.3955	1
206.0238	9.1278	2
80.6513	54.892	0
144.05	80.4096	1
190.575	81.7569	1
84.9988	0.0353	2
122.4481	51.7603	0
186.55	23.0786	1
179.7356	28.6735	2
181.8944	32.117	2
190.745	14.3898	1
91.305	40.6342	0
144.05	80.4096	1
190.575	81.7569	1
105.0231	54.8095	0
148.3675	50.9226	0
182.9669	15.6604	2
177.7813	14.486	2
129.4906	62.769	0
82.3225	16.9368	0
144.05	80.4096	1
190.575	81.7569	1
109.55	54.536	0
197.1756	76.211	1
44.8663	23.2223	0

224.8506	75.6331	1
238.7775	56.7623	1
236.98	60.212	1
240.5394	54.6162	1
236.5844	60.6057	1
229.55	70.3811	1
231.9475	67.6543	1
238.9956	56.4647	1
240.2719	33.7209	2
216.6419	84.2371	1
218.6606	82.5642	1
242.2444	51.3355	1
230.5806	69.9898	1
219.7944	81.3766	1
217.9938	83.9534	1
232.6075	67.4173	1
227.475	73.2558	1
213.0669	87.6441	1
213.0031	89.1087	1
230.4731	69.6102	1
248.5213	36.4406	0
220.8338	79.3504	1
226.7988	72.8691	1
243.455	48.8602	1
190.6813	101.5129	1
181.8831	105.0029	1
252.9644	19.6041	2
243.9675	45.9978	1
214.3144	84.6919	1
188.5894	101.2533	1
251.1113	28.454	2
209.3931	90.3482	1
228.4663	70.2	1
206.91	90.8972	1
231.7381	65.3345	1
218.4813	82.5521	1
227.6369	70.9972	1
221.2088	79.2526	1
221.7131	78.8438	1
227.4125	71.4239	1
235.5781	62.9432	1
242.3506	52.0709	1
228.2425	71.0466	1
225.5488	74.8781	1
226.6038	74.3583	1
239.1994	55.0379	1
226.5456	72.6611	1
225.7094	75.0548	1
231.2606	68.9302	1
227.0388	73.0738	1
220.4875	80.1631	1
214.3913	85.2373	1

242.9725	45.4821	1
182.1913	106.0496	1
188.9363	99.9755	1
194.8106	97.3941	1
194.6913	99.4558	1
231.0975	68.2559	1
230.9244	68.4226	1
235.5313	60.3899	1
232.4944	66.6228	1
245.1175	44.9921	2
249.3925	33.7541	2
212.7356	87.7427	1
230.7081	68.5082	1
221.6875	77.987	1
234.82	62.5389	1
218.9575	81.759	1
238.91	55.2367	1
221.1075	79.3579	1
222.1531	79.7188	1
218.3094	81.674	1
220.4519	80.4426	1
252.78	21.521	2
250.205	31.5416	2
250.9906	28.7839	2
251.8819	25.5411	2
247.7519	36.4843	0
217.4288	83.4765	1
217.9331	82.809	1
222.7006	79.3836	1
221.6988	78.6072	1
213.5425	87.1194	1
222.9913	77.7416	1
199.5519	98.1778	1
210.7344	86.6654	1
191.0481	100.0055	1
186.0019	103.2196	1
220.0038	80.4571	1
194.1269	100.1173	1
193.0181	98.3726	1
236.9331	60.3562	1
244.1244	44.0579	2
189.1019	102.8077	1
193.6638	98.7552	1
195.1175	98.7776	1
201.5188	93.5209	1
191.6388	101.2819	1
206.9013	89.8042	1
222.3988	77.4187	1
219.0069	82.5627	1
233.9831	62.37	1
208.5963	89.3324	1
219.7744	79.5787	1

85.6069	65.6954	0
62.2069	55.5313	0
118.7875	98.0736	1
151.405	74.7486	1
84.9894	0.1025	2
144.05	80.4096	1
190.575	81.7569	1
83.4706	7.397	0
69.3156	22.1515	0
206.8075	72.4467	1
236.3181	53.1126	1
82.2019	79.2429	0
42.0869	3.4237	2
50.2975	23.2325	0
201.25	84.2168	1
237.1344	45.5804	1
67.47	41.44	0
75.9469	18.6779	0
84.9994	0.0433	2
144.05	80.4096	1
190.575	81.7569	1
74.7631	20.7779	0
42.3663	13.2183	0
51.0013	51.2478	0
99.3413	95.0768	1
93.9056	80.5119	1
207.7706	77.155	1
51.7969	29.0489	0
154.37	97.6016	1
124.6969	96.1934	1
68.045	66.9715	0
70.0313	71.2189	0
37.21	8.198	0
53.615	22.3522	0
83.1075	9.005	0
144.05	80.4096	1
189.0169	84.1721	1
39.8506	15.3713	0
35.6313	5.4658	0
36.8675	6.6821	0
34.0513	16.331	0
50.3488	31.3347	0
221.9869	66.7419	1
189.0288	91.6793	1
237.9619	48.8817	1
57.9444	47.635	0
32.26	12.6984	0
39.4081	8.6063	0
41.0606	5.3873	0
35.6763	5.9175	0
61.3344	24.6532	0
144.05	80.4096	1

234.1531	64.7443	1
223.5056	78.2832	1
224.0369	76.7084	1
227.4019	73.2543	1
224.0788	75.6853	1
237.4044	60.4198	1
247.5781	37.7666	0
224.6613	75.4622	1
227.3063	73.8029	1
235.21	63.9568	1
229.7981	69.5742	1
223.6556	76.7829	1
232.9688	66.6527	1
252.395	23.2325	2
250.3213	31.3482	2
251.6519	25.9575	2
251.5281	26.5467	2
241.9194	50.6611	1
249.2575	32.9624	2
254.3875	10.2833	2
211.145	88.7747	1
193.0275	98.3488	1
203.15	94.3301	1
202.535	95.2395	1
206.8519	90.3036	1
185.735	102.828	1
218.9531	80.8546	1
244.3763	45.5136	2
193.7263	100.6991	1
193.7763	99.0149	1
180.5869	102.3169	1
199.5219	96.2183	1
194.6163	97.9237	1
195.6463	97.614	1
218.7681	82.4428	1
196.4156	97.1014	1
189.9969	100.9547	1
206.3019	91.4036	1
194.6294	100.2846	1
213.7913	86.0149	1
199.8519	96.8929	1
212.5575	87.715	1
230.5131	71.1247	1
254.9488	0.4526	2
248.4594	37.4487	0
213.3163	87.4732	1
215.0206	85.8642	1
222.7056	76.7482	1
192.7394	99.7967	1
222.1769	76.7655	1
206.9894	90.3064	1
210.5606	88.3729	1

220.8213	80.3559	1
219.3013	81.4721	1
245.3925	45.3875	2
248.5263	33.119	2
204.2713	93.854	1
222.4788	77.9576	1
219.5206	81.7413	1
221.0725	80.7676	1
220.7513	79.4891	1
240.4156	54.6649	1
227.2075	73.3725	1
232.975	66.4031	1
241.4669	51.0215	1
229.6344	71.8701	1
230.8806	68.8665	1
231.7988	68.0526	1
232.3119	67.7299	1
247.7538	39.5108	2
248.9763	32.971	2
220.3956	81.4309	1
232.9013	65.7592	1
232.1919	67.9188	1
224.8944	75.642	1
225.5231	75.059	1
244.9906	45.7566	2
217.3438	83.4247	1
189.8431	98.6411	1
197.9456	97.6499	1
206.58	90.9422	1
247.0188	39.5003	2
192.6281	101.0954	1
200.6806	93.3813	1
217.0669	84.6079	1
200.5744	97.0829	1
211.4975	86.9165	1
216.1556	83.4184	1
246.5369	41.9393	2
247.91	33.6691	2
189.995	102.5241	1
211.2019	88.0165	1
209.9894	89.6224	1
207.3638	91.1738	1
217.3419	81.7848	1
216.6663	81.6445	1
211.0006	88.9882	1
224.2956	74.8501	1
224.7425	74.5521	1
235.7719	59.2064	1
227.3594	72.5478	1
245.0925	44.7742	2
246.895	39.295	2
249.7238	32.3101	2

171.455	107.781	1
35.2688	1.8294	2
37.0638	7.6922	0
29.4444	11.3612	0
27.9225	9.9249	0
45.1744	5.4702	0
131.2375	96.3375	1
254.5606	1.2042	2
136.3744	96.8303	1
45.2325	2.8937	2
37.0969	6.9612	0
35.3175	3.954	2
33.8814	4.4958	2
34.3506	2.8132	2
38.9669	14.3362	0
143.0419	81.4167	1
166.4956	114.028	1
31.605	3.5049	2
32.1038	4.9016	2
34.1256	3.1217	2
41.2006	8.3711	0
75.4063	62.9651	0
91.8244	72.8553	0
198.0081	87.7181	1
52.5606	25.1246	0
42.9594	2.068	2
34.1781	8.1621	0
31.9875	3.6993	2
37.0469	4.5915	2
32.7425	4.0988	2
32.6463	2.6802	2
131.3363	91.9128	1
165.1638	115.7707	1
28.9375	3.3738	2
25.4044	7.9464	0
33.7406	8.9828	0
42.5906	2.2443	2
57.37	49.3033	0
146.715	88.5113	1
48.3406	32.1458	0
43.1513	1.497	2
41.9019	2.1736	2
34.6131	7.3547	0
32.1181	3.9699	2
34.9313	4.2632	2
30.8369	4.457	2
30.9056	3.2347	2
118.8131	100.5919	1
169.6675	109.9406	1
22.0225	4.2023	2
21.38	8.293	0
35.1856	2.9633	2

221.4075	79.5993	1
224.7775	75.6944	1
211.5025	87.7612	1
220.2138	78.9018	1
218.6313	82.5067	1
204.5006	93.3161	1
219.105	80.3584	1
213.9463	86.4073	1
206.8919	91.7731	1
216.8694	84.0591	1
207.7244	90.9557	1
240.4881	53.6151	1
253.7944	15.9285	2
234.5431	64.3785	1
237.3281	59.5011	1
241.81	52.6833	1
241.2638	53.6205	1
238.1663	57.3919	1
234.4925	64.114	1
237.0575	60.3658	1
242.0106	53.1535	1
238.4713	58.7202	1
235.9869	61.8436	1
237.1838	59.0022	1
238.7463	57.9967	1
241.4519	53.9251	1
237.1806	60.0738	1
238.5331	58.9227	1
238.2494	58.4485	1
240.6506	54.5103	1
233.4619	66.2671	1
244.8619	45.7824	2
244.6438	45.8748	1
233.6781	65.3656	1
238.0794	57.6756	1
246.8775	39.4532	2
246.7013	39.5311	2
249.4363	34.3119	2
235.6063	62.1718	1
232.515	66.5332	1
243.9569	44.6021	2
235.1588	63.5098	1
246.2338	40.9995	2
243.3006	48.5357	1
234.3294	64.963	1
231.3806	68.3871	1
237.1775	60.1917	1
246.0281	41.9075	2
209.4563	90.739	1
204.5413	92.5834	1
206.1081	90.8136	1
203.075	92.4819	1

245.6788	41.8441	2
212.535	88.1756	1
225.4831	74.352	1
234.6588	61.9812	1
237.9413	57.7095	1
231.5381	68.1056	1
249.9313	31.7468	2
243.0469	50.7045	1
213.835	87.1342	1
207.8019	90.7497	1
223.7019	77.5144	1
204.8056	93.9134	1
205.5938	92.2694	1
197.1025	95.679	1
220.0681	79.8951	1
247.1081	37.802	0
210.8794	89.4043	1
199.0019	96.4335	1
211.6131	86.6593	1
210.1719	87.6338	1
222.6456	78.4153	1
210.7025	89.2121	1
196.9738	95.5638	1
209.4713	89.1129	1
193.7013	99.5719	1
212.3356	87.1023	1
210.7613	88.9817	1
239.995	53.7318	1

38.5381	4.9163	0
42.105	4.0896	2
51.4381	35.1816	0
42.0356	1.3419	2
40.0406	3.8577	2
39.495	4.6041	0
39.2431	2.5909	2
38.1538	1.6893	2
31.7744	5.3337	0
25.095	6.7638	0
28.57	3.6732	2
110.6356	105.4486	1
170.2094	109.1924	1
24.0406	4.1518	2
18.9906	6.6504	0
32.7488	4.0377	2
37.9588	1.318	2
38.2906	1.5099	2
39.7563	4.1958	2
40.6656	3.4149	2
38.1581	4.1326	2
36.65	2.861	2
37.21	1.7233	2
35.83	2.3598	2
27.9813	8.1743	0
25.1706	7.0924	0
29.5119	5.0017	2
111.2638	104.9942	1

229.9606	69.3712	1
231.6994	66.9076	1
221.9238	79.275	4
224.3894	75.072	1
209.2019	90.0903	1
205.005	92.5301	1
214.1088	86.5762	1
227.5831	73.4981	1
220.9331	80.2643	1
227.3069	73.8371	1
253.7463	14.2632	2
185.7325	102.7015	1
196.8381	96.412	1
189.4188	101.3264	1
200.8725	92.3988	1
202.495	92.3136	1
211.8569	87.607	1
244.9563	44.3615	2
251.6556	26.0597	2
213.0819	88.8874	1
203.3875	92.7431	1
232.5225	67.3676	1
218.7431	35.8258	2
234.3481	64.7358	1
232.2938	67.638	1
248.8	35.3186	2
231.9975	67.7368	1
168.92	110.8303	1

Pusat Cluster	
68.4908	37.3457
198.7812	89.9808
254.997	0.0313

BIODATA

1. Nama : Teuku Rafli
2. TTL : Lhouksemawe 4 Agustus 1980
3. Alamat Asal : Jl. Mangga 2 Lr I No 21
4. PTN/PTS : Universitas Negeri Nanggroe Aceh Darussalam
5. Tahun Masuk : 2000
6. Asal SLTA : Sma Negeri I Nanggroe Aceh Darussalam
7. Jumlah Saudara Kandung : 4 Orang
8. Sumber biaya Kuliah : Orang Tua
9. No Hp : 081234456123

Orang Tua

Ayah

- a) Nama : Teuku Ismail
- b) TTL : Sunda Kelapa 12 Maret 1945
- c) Pekerjaan : Buruh Pabrik
- d) Pendidikan terakhir : SMP
- e) Penghasilan Orang Tua perbulan : Rp. 750.000,-
- f) Jumlah tanggungan : 5 Orang

Ibu

- a) Nama : Cut Memi
- b) TTL : Lhouksemawe 12 Februari 1950
- c) Pekerjaan : Ibu Rumah Tangga
- d) Pendidikan terakhir : SMP
- e) Penghasilan Orang Tua perbulan : -tidak ada-
- f) Jumlah tanggungan : 5 Orang

Pengalaman Organisasi

1. Himpunan Mahasiswa Arsitek
2. HMI

Pengalaman Seminar/Pelatihan/Lomba/Workshop

1. Seminar Kesehatan
2. Seminar Sehari "Demokrasi"



Aceh Barat Daya, 21 Maret 2001

Ttd

Teuku Rafli

LAMPIRAN
Data Pada Formulir 2

Mean	SD	Cluster
255	0	1
207.1759	93.4466	0
210.7542	90.9443	0
177.2742	111.4812	0
189.1627	106.1438	0
191.1247	104.2558	0
200.0699	97.9422	0
215.2445	88.459	0
187.0263	106.1156	0
213.9896	89.2726	0
216.982	86.4457	0
216.2361	86.2144	0
206.6524	93.5742	0
227.428	76.4003	0
184.6087	108.0336	0
220.847	82.2292	0
193.7825	102.7232	0
213.8373	87.8449	0
199.0305	98.8219	0
202.7853	97.7782	0
204.5727	96.7057	0
210.7985	90.6696	0
192.9536	104.1596	0
203.6565	96.2649	0
182.0859	108.607	0
192.1489	101.8273	0
220.403	82.5056	0
248.4661	35.8782	1
193.9439	104.1029	0
206.3158	93.8625	0
200.5976	97.7291	0
215.4813	87.7037	0
184.6849	108.1497	0
220.8622	82.2129	0
184.4224	108.3752	0
220.9529	82.2153	0
205.9363	94.2804	0
210.8449	91.0931	0
197.6932	102.0009	0
212.0028	90.512	0
220.7659	80.3696	0
245.5956	46.054	1
207.2258	94.7047	0
193.8289	102.5285	0
176.8691	111.713	0
178.4134	110.6512	0
186.5409	107.5346	0
180.2763	109.7069	0
181.9363	109.2952	0

Mean	SD	Cluster
227.1323	70.2131	0
254.9751	0.292	1
245.973	42.688	1
250.1939	30.925	1
249.6745	32.9231	1
255	0	1
248.7368	35.692	1
255	0	1
246.8407	40.1328	1
254.9114	0.6186	1
206.5325	89.6296	0
254.6434	7.5169	1
206.7181	89.3604	0
202.5319	92.5225	0
211.3615	87.0352	0
255	0	1
229.3843	71.266	0
224.9273	72.4518	0
195.2126	98.8048	0
222.5831	77.7723	0
206.6711	89.3486	0
198.4827	95.9923	0
206.6614	90.1949	0
254.9266	0.5158	1
212.0783	87.7582	0
253.3435	16.3453	1
219.1904	79.6921	0
219.2632	81.7344	0
254.813	4.2205	1
232.4342	66.4889	0
224.0568	75.808	0
210.885	88.8652	0
237.0402	56.6922	0
211.16	87.9546	0
221.2258	76.2407	0
219.0533	81.4782	0
230.1738	70.0235	0
235.1253	60.5219	0
211.4792	87.9595	0
220.5769	78.1571	0
222.4778	75.6188	0
235.4882	61.9842	0
198.9647	96.4651	0
231.0042	69.0227	0
254.9813	0.2419	1
236.892	58.5622	0
255	0	1
245.0277	45.58	1
240.1807	56.2139	0
251.5596	26.0824	1

Mean	SD	Cluster
233.7168	66.3055	0
225.8123	75.3557	0
229.7355	71.2974	0
224.599	76.6442	0
232.268	67.684	0
232.5325	66.1376	0
208.8234	92.3658	0
220.3456	81.3882	0
222.7078	77.3505	0
220.3864	82.9364	0
234.8428	63.8751	0
227.4668	74.3523	0
228.5499	72.9116	0
221.5741	80.3833	0
234.1253	64.8353	0
205.259	94.3708	0
236.2639	59.5129	0
220.5471	81.2457	0
224.4217	76.5219	0
233.7971	65.4886	0
240.982	54.7319	0
235.6641	64.4373	0
222.8421	79.3546	0
197.0222	99.3798	0
201.2729	97.0184	0
208.9391	92.7417	0
206.2943	95.0391	0
209.4889	93.0544	0
224.759	76.331	0
207.759	94.2569	0
205.7985	96.0182	0
206.7438	93.4632	0
230.6766	70.2973	0
217.5949	85.5493	0
204.5713	94.8625	0
218.3878	83.1587	0
199.8989	99.3746	0
212.151	89.1832	0
223.8255	79.0877	0
202.7445	98.0424	0
212.0353	89.94	0
209.3019	91.3569	0
225.5748	77.262	0
222.9896	78.8434	0
210.2078	92.7609	0
212.5575	89.0207	0
202.7223	95.9645	0
218.8359	83.9953	0
229.5519	69.6336	0
254.7929	1.6918	1

178.0055	110.9471	0
188.4979	106.6314	0
188.6891	105.0925	0
190.6537	105.3963	0
185.0602	107.6418	0
192.6946	104.5225	0
174.3996	112.4925	0
182.635	108.6284	0
183.9079	106.867	0
191.8705	104.2428	0
191.6641	103.7866	0
186.081	107.0467	0
172.723	113.4965	0
184.6877	107.5658	0
181.162	108.5998	0
180.5457	109.3701	0
191.9488	103.1423	0
186.5325	107.0325	0
184.1032	108.1019	0
204.2285	95.8041	0
173.099	112.8558	0
187.7015	106.1174	0
184.1143	107.9931	0
195.0325	103.304	0
174.3837	112.4886	0
182.545	108.6536	0
173.0651	113.2078	0
181.9197	108.8317	0
195.1503	103.1901	0
188.1891	106.3318	0
186.8324	106.3215	0
179.482	111.1437	0
202.7084	96.8272	0
217.5409	86.9427	0
211.205	92.1431	0
161.8407	114.7807	0
172.0748	111.1605	0
189.437	103.2027	0
175.0859	111.2263	0
197.6468	96.6555	0
253.5665	16.8576	1
254.9827	0.2242	1
246.8026	40.9809	1
254.1787	10.5845	1
236.4785	59.5581	0
250.1614	29.5977	2
251.8989	23.8281	1
254.9882	0.2276	1
253.973	13.8073	1
237.0422	58.768	0
251.1032	27.2999	1
242.8864	49.0687	1
243.847	48.3086	1
242.6821	49.9766	1

245.1406	43.7757	1
236.6877	59.1295	0
231.6233	68.035	0
240.5076	52.7357	0
233.8248	66.1604	0
235.6849	60.9082	0
225.3726	73.8681	0
243.1434	49.3192	1
241.5713	52.0133	1
241.6191	52.1841	1
233.3961	64.942	0
237.7597	58.3296	0
248.3747	36.7745	1
245.5471	43.2579	1
247.7687	37.3007	1
254.9245	0.5122	1
210.2562	87.2064	0
254.1711	10.5154	1
195.3186	98.5899	0
195.9259	96.3074	0
201.1988	92.302	0
202.8885	91.6207	0
216.9855	82.3086	0
248.4619	36.4286	1
198.1004	97.6548	0
206.2348	90.8463	0
198.6427	95.6873	0
210.4827	89.4356	0
202.7271	92.5811	0
195.0783	98.1312	0
210.7098	86.9263	0
254.991	0.1554	1
232.4141	64.7163	0
254.5873	6.7509	1
227.3476	73.2846	0
241.5748	51.4128	1
230.2458	67.2704	0
225.1856	73.7945	0
243.2015	47.8297	1
242.1641	50.2079	1
241.7396	50.2734	0
236.0319	60.3342	0
232.4342	64.9154	0
247.4785	38.4666	1
251.7625	25.956	1
236.2846	59.9642	0
223.847	76.9078	0
230.0983	70.4831	0
254.9391	0.4393	1
214.6087	84.7878	0
254.5949	7.8721	1
223.0215	78.1915	0
214.2438	85.1437	0
216.6849	83.0815	0

196.126	95.9017	0
197.9813	96.1882	0
203.9917	91.7211	0
206.6835	89.2175	0
202.3338	94.9587	0
209.8781	85.8844	0
200.5048	92.9915	0
195.3539	97.2464	0
205.063	88.9803	0
239.4668	56.5359	0
254.9467	0.4816	1
241.858	52.0296	1
254.8961	1.5205	1
241.6018	50.2166	0
253.6801	11.1117	1
251.259	25.2602	1
253.6835	11.3418	1
242.0873	49.7423	1
245.9723	41.7004	1
253.705	11.2608	1
245.1988	43.0995	1
230.2251	69.285	0
253.644	11.7926	1
253.7874	10.7386	1
246.608	40.3672	1
253.4765	12.3228	1
239.3497	54.2708	0
254.9411	0.5143	1
254.9335	0.5114	1
233.6856	63.6795	0
254.9231	0.556	1
209.34	88.752	0
199.0194	96.0898	0
207.4536	90.0478	0
207.7098	88.9576	0
224.973	75.0788	0
200.5928	94.3909	0
208.1648	88.653	0
238.7078	56.8539	0
219.0242	82.7721	0
209.9993	86.9212	0
203.1101	94.1245	0
203.563	92.5957	0
209.4508	87.0732	0
232.4183	65.6739	0
254.9972	0.0832	1
239.1018	46.5341	0
201.3123	78.6688	0
201.2458	78.7874	0
201.4141	78.5273	0
201.5824	78.2761	0
201.3532	78.6108	0
201.259	78.7607	0
200.9384	79.2477	0

230.8975	67.2451	0
254.9404	0.4549	1
237.7715	58.7745	0
254.6454	7.5235	1
215.6579	82.888	0
207.2639	90.2374	0
214.4155	85.1612	0
249.4875	33.8779	1
238.8767	58.3175	0
215.4522	83.3238	0
201.7285	94.9896	0
222.2652	79.3645	0
210.7445	87.7778	0
211.8892	86.7173	0
254.7278	6.1112	1
254.9127	0.6018	1
228.6724	69.4732	0
253.3601	16.3129	1
214.0229	84.2772	0
212.5665	87.7455	0
252.9861	18.9083	1
250.1551	31.1308	1
229.7846	69.4872	0
207.0928	89.9588	0
210.1572	88.3266	0
206.9765	89.586	0
210.982	87.0596	0
212.964	84.4859	0
209.5942	88.0249	0
218.2119	81.3571	0
233.4578	63.631	0
212.7618	85.8705	0
210.8767	87.5969	0
210.741	87.5019	0
235.1046	60.6443	0
202.2992	93.8664	0
199.8449	94.8647	0
238.036	59.1992	0
254.9356	0.5309	1
240.4245	55.3114	0
255	0	1
244.4958	46.3132	1
246.732	42.3863	1
253.4591	18.6347	1
254.849	2.85	1
252.894	20.6484	1
251.0769	28.5681	1
253.1627	19.2074	1
239.0298	57.1802	0
242.1219	50.0305	1
244.106	46.4498	1
247.2043	39.3379	1
246.8096	41.309	1
246.9051	40.7128	1

210.4515	87.9483	0
216.5748	82.7027	0
230.9481	68.0435	0
220.946	78.2758	0
210.2452	88.1194	0
218.8234	81.3908	0
242.9453	49.8879	1
254.991	0.1364	1
254.9903	0.1112	1
251.6032	25.969	1
232.1108	66.5052	0
218.3899	82.2721	0
237.6233	60.1959	0
254.9314	0.4782	1
209.617	87.5518	0
254.1745	10.4868	1
195.5201	98.548	0
202.0083	93.1438	0
197.831	95.0295	0
204.1198	90.0495	0
195.0506	96.1023	0
213.5983	85.6291	0
204.5291	92.2516	0
204.9647	89.7749	0
196.0298	97.3824	0
222.144	77.9493	0
207.3608	89.3503	0
218.5637	81.4167	0
200.3262	93.842	0
199.1011	96.1415	0
202.3283	92.4811	0
208.6378	89.3315	0
248.4758	36.4308	1
224.0173	77.4231	0
198.2895	97.2047	0
237.8186	57.5494	0
198.5997	96.2349	0
208.7542	88.8752	0
189.5152	100.465	0
214.106	85.1577	0
250.7742	29.4364	2
242.0367	49.3174	1
247.9162	37.8847	1
249.313	34.0858	1
240.7936	51.8804	0
250.6385	29.6916	2
254.9834	0.1278	1
254.9958	0.0643	1
254.9612	0.2758	1
254.9799	0.1403	1
244.6579	44.9725	1
254.2022	12.2519	1
254.9654	0.1974	1
247.8137	37.9198	1

201.2999	78.6971	0
200.8366	79.4034	0
200.6343	79.7239	0
200.6766	79.6702	0
199.6918	81.0821	0
199.3539	81.5751	0
205.5582	77.5336	0
254.9709	0.2018	1
204.7632	72.1176	0
84.3193	1.5057	2
84.7029	1.4185	2
85.3463	1.5301	2
85.5776	1.5462	2
85.036	1.6107	2
82.3982	9.3801	2
71.6489	25.0377	2
79.9668	14.3709	2
84.3504	1.406	2
83.2098	1.6332	2
82.8213	1.2924	2
80.545	2.1026	2
78.8386	1.7959	2
98.6724	48.7767	2
254.9439	0.2301	1
204.8463	72.1004	0
84.8968	1.1908	2
84.7992	1.2493	2
84.9377	1.6673	2
85.4612	1.8073	2
62.153	32.113	2
17.4107	18.1271	2
19.7001	12.9529	2
17.9647	13.5739	2
47.0166	34.8663	2
84.2625	3.6068	2
83.5166	1.8138	2
81.3947	1.9803	2
78.383	1.5585	2
99.0242	48.7145	2
254.9411	0.2354	1
204.3871	72.815	0
83.7417	1.5619	2
84.3082	1.2702	2
85.1191	1.6204	2
72.9501	25.6038	2
27.4404	20.1814	2
57.9681	36.7505	2
82.277	26.4047	2
74.7922	31.8556	2
25.4834	19.5752	2
50.3186	33.9464	2
84.412	1.722	2
81.0893	1.979	2
77.7119	1.9191	2

242.8878	48.9294	1
248.045	37.6652	1
247.2071	38.9428	1
225.9169	72.6493	0
243.8615	46.2687	1
239.4785	55.2073	0
250.8643	27.6905	1
250.7791	28.4118	1
249.6676	31.0883	1
254.777	5.3373	1
254.9571	0.3678	1
229.7521	68.0681	0
254.3636	10.4323	1
202.8802	94.7634	0
206.9931	87.6254	0
202.117	93.3562	0
221.9993	77.3427	0
202.4744	92.7497	0
205.2673	90.0443	0
254.964	0.3922	1
248.3677	36.3375	1
216.9806	85.0174	0
232.8338	64.5173	0
201.0457	94.5778	0
201.6911	92.5047	0
178.0436	102.3406	0
222.8393	75.1058	0
226.8054	72.1449	0
212.6676	87.3516	0
233.4993	65.4303	0
207.9197	89.0548	0
226.6177	71.842	0
219.9211	80.1037	0
254.9619	0.4202	1
229.9425	70.4575	0
254.5789	6.8802	1
211.8553	87.2015	0
213.5055	86.5002	0
218.2334	79.1623	0
216.6842	83.1837	0
226.7936	73.124	0
251.6794	25.9617	1
231.6759	66.6129	0
222.4287	75.6518	0
234.2168	61.7346	0
226.3657	72.1629	0
224.6039	73.2507	0
233.5249	63.2714	0
220.0949	78.1323	0
226.3158	72.55	0
227.3934	69.8966	0
227.7742	70.9332	0
222.0104	75.9336	0
227.8262	71.041	0

254.9737	0.2122	1
254.9675	0.2852	1
254.9778	0.1472	1
254.9848	0.1433	1
246.7985	41.411	1
236.7078	58.8545	0
254.9439	0.367	1
254.9778	0.165	1
254.9633	0.231	1
254.9778	0.2159	1
253.0499	17.6512	1
254.9439	0.5051	1
216.7431	82.7338	0
253.5796	17.1069	1
205.7597	92.4666	0
205.4501	91.9031	0
203.8719	91.1978	0
210.8241	88.2904	0
207.0644	88.5544	0
184.9363	101.0006	0
196.5748	95.4935	0
191.6724	98.3085	0
200.3483	92.8956	0
238.1676	58.2082	0
248.455	36.358	1
232.9349	65.6262	0
219.4003	82.2188	0
203.876	90.9192	0
194.2555	96.7918	0
241.5492	51.6384	1
236.8352	59.7322	0
254.982	0.2226	1
221.5436	79.0917	0
228.6004	71.9964	0
242.0596	51.3637	1
251.6731	25.9201	1
227.4591	71.533	0
216.5305	81.1193	0
254.9252	0.4866	1
254.5471	5.0975	1
254.9855	0.3418	1
254.9494	0.402	1
254.9668	0.3781	1
254.9252	1.0034	1
254.385	10.4493	1
248.4785	36.4428	1
220.0194	79.7218	0
203.5187	92.1056	0
216.5506	83.6701	0
198.7999	94.8367	0
202.4543	92.9114	0
240.3795	55.755	0
254.9418	0.5028	1
228.205	71.227	0

97.5762	49.1822	2
254.9439	0.2301	1
204.365	72.8355	0
83.099	1.3742	2
84.0907	1.5184	2
84.9695	1.5623	2
46.4425	30.4487	2
87.5928	29.3539	2
134.2064	9.8981	2
145.1205	7.3301	1
138.4224	9.6683	2
99.777	19.0632	2
30.3837	23.8011	2
83.3407	4.1911	2
81.367	1.9808	2
78.5242	1.7281	2
97.9321	49.1363	2
254.9425	0.2328	1
204.1094	73.1883	0
82.259	1.2301	2
82.9294	1.4016	2
84.0956	1.4334	2
55.3089	22.7505	2
107.6364	16.1064	2
143.5284	10.6337	2
154.9024	6.1855	1
146.554	9.4949	1
114.9363	12.0739	2
52.1821	28.0318	2
79.669	10.7021	2
80.0194	2.1735	2
77.5173	1.9578	2
97.6267	49.2802	2
254.9432	0.2314	1
204.2008	73.1879	0
82.0976	1.5361	2
83.1357	1.4782	2
88.1434	9.274	2
81.7057	27.5373	2
87.4176	21.6142	2
96.1939	30.405	2
145.2015	15.39	2
104.4536	27.7948	2
82.2306	23.7753	2
74.2147	28.4335	2
86.3573	10.903	2
79.1434	2.0315	2
76.9889	2.0014	2
97.3428	49.4134	2
254.9432	0.2314	1
237.4086	58.3687	0
224.849	75.4232	0
211.5381	85.7528	0
210.982	89.3513	0

230.1039	67.7472	0
232.6267	65.1241	0
232.9654	65.4493	0
218.6066	79.8765	0
232.1025	67.5617	0
217.8276	81.856	0
227.732	72.3185	0
232.0069	63.9434	0
219.5727	79.9747	0
226.3109	72.7835	0
254.964	0.3723	1
254.9806	0.2571	1
242.3823	50.7079	1
254.5893	7.948	1
243.5042	49.5788	1
242.5616	50.665	1
237.286	59.5225	0
246.3241	43.3755	1
242.1378	50.1459	1
251.6711	25.957	1
241.8871	50.9581	1
233.6565	64.8399	0
239.0381	56.8763	0
238.0644	56.4982	0
229.2382	70.2452	0
236.3518	59.3505	0
242.2493	50.5316	1
220.4584	78.5591	0
232.3199	65.9955	0
242.6898	49.7415	1
232.7084	65.8013	0
221.8324	76.7153	0
233.0838	65.5601	0
246.2472	40.8161	1
231.9474	67.6366	0
230.4896	69.0566	0
240.8463	53.9882	0
229.0381	70.7012	0
234.973	61.4233	0
230.91	67.0714	0
227.5104	72.6114	0
231.5367	67.6997	0
254.8594	3.8834	1
254.9328	0.6197	1
221.5658	78.3595	0
253.3719	16.3225	1
208.4654	88.7185	0
191.5665	99.071	0
200.5873	94.5569	0
215.6108	84.1721	0
195.6662	96.0537	0
202.9391	92.5376	0
217.3193	82.2119	0
248.358	36.4046	1

253.3483	16.2976	1
226.9965	71.4159	0
229.2368	70.6414	0
251.7348	25.9646	1
241.7431	51.7515	1
233.4972	63.8348	0
236.2022	60.043	0
240.3386	52.5063	0
239.4654	54.7508	0
241.2251	52.1377	0
245.5499	42.8555	1
244.2071	46.829	1
235.606	61.9164	0
230.2971	68.3127	0
228.5201	71.1697	0
241.928	50.8493	1
237.2992	58.2536	0
244.2382	46.7735	1
227.4211	72.5181	0
224.3026	76.583	0
254.9578	0.3374	1
220.2867	79.8432	0
254.9986	0.0372	1
237.2112	60.157	0
230.3151	71.3766	0
251.4494	26.3014	1
251.6157	25.954	1
235.4737	63.1419	0
220.4086	79.8433	0
218.7992	81.1568	0
213.2992	85.6087	0
222.7964	77.3762	0
217.437	81.3247	0
234.8532	61.812	0
243.295	49.0115	1
236.2964	60.6854	0
224.2029	75.3306	0
223.1641	77.0951	0
220.3712	78.5603	0
222.3816	78.2727	0
243.5907	48.6744	1
234.8497	62.0413	0
227.0956	72.5226	0
254.9432	0.3897	1
212.2022	86.0753	0
254.1787	10.4469	1
195.3753	98.6818	0
195.7957	96.2496	0
204.4349	90.4075	0
202.9626	91.562	0
216.9889	82.3046	0
248.4931	36.3365	1
204.1281	92.7501	0
210.0956	88.9893	0

222.6524	75.9293	0
218.7708	79.7911	0
217.0152	84.7332	0
218.4751	80.9337	0
229.847	67.7435	0
235.4086	61.1549	0
237.2645	59.029	0
199.7216	96.0394	0
217.6544	81.1705	0
226.7057	71.3081	0
217.9328	81.6251	0
204.5699	92.9207	0
238.0187	58.422	0
203.6572	73.762	0
80.9051	1.7868	2
82.1724	1.2994	2
88.9785	10.9476	2
101.2029	21.7948	2
124.3968	19.8708	2
128.6087	17.5464	2
132.7147	14.0785	2
120.2202	15.702	2
116.8006	26.0062	2
97.6157	23.265	2
92.0956	13.423	2
79.4342	1.9876	2
75.9543	2.0803	2
95.6946	49.9959	2
254.9467	0.2247	1
251.1579	28.2593	1
240.2846	55.36	0
237.4363	59.9746	0
243.4917	49.2166	1
240.2015	54.5454	0
240.3262	54.0256	0
243.8643	48.3288	1
235.8719	60.7898	0
238.0319	57.1709	0
243.705	49.2794	1
249.4494	34.0945	1
243.5132	48.9076	1
240.0048	54.9147	0
241.9259	51.824	1
240.3442	54.8844	0
246.7846	39.8973	1
250.106	32.6675	1
202.8359	74.9293	0
79.4086	2.2547	2
81.8857	1.6768	2
84.7708	1.5365	2
89.635	15.1193	2
140.0173	15.1684	2
145.1801	19.3173	2
138.2964	17.8729	2

214.1524	85.4236	0
193.0706	98.2517	0
207.4868	90.6555	0
254.9861	0.235	1
245.7611	42.0097	1
255	0	1
242.9238	49.4035	1
245.6974	41.5381	1
236.7604	60.093	0
238.1309	56.237	0
237.3657	58.6588	0
246.2542	43.2204	1
254.9945	0.0742	1
253.1572	19.489	1
239.491	54.5336	0
246.7604	40.8677	1
241.34	51.9355	0
241.9945	49.2744	0
249.0166	32.9553	1
244.1572	46.0217	1
244.1655	46.7438	1
234.7957	61.669	0
246.6281	40.6771	1
249.0042	32.6554	1
245.8179	42.1691	1
251.0519	27.5267	1
244.0602	47.6476	1
248.3657	35.3647	1
242.144	50.4861	1
231.4993	68.3728	0
246.5152	41.0423	1
248.5589	34.5447	1
243.7271	47.0758	1
245.9301	41.3923	1
249.3026	34.6245	1
254.9432	0.427	1
225.0519	75.0737	0
254.376	10.556	1
218.8809	81.0646	0
214.1066	83.8464	0
226.9882	73.3956	0
227.4425	73.6688	0
229.6738	70.0739	0
236.9882	61.2217	0
250.3276	30.946	1
229.0602	70.5401	0
210.5042	88.8664	0
225.2348	73.6332	0
217.2576	82.0019	0
198.7943	94.1513	0
217.6233	81.4276	0
239.6032	56.4189	0
223.9245	74.9764	0
206.0499	90.7655	0

200.9086	96.218	0
205.5713	91.3447	0
195.9952	97.3862	0
215.0152	84.1027	0
207.4044	89.3784	0
206.4675	90.1255	0
194.9328	96.4045	0
225.0866	73.5063	0
242.187	48.9913	1
254.5873	6.7509	1
240.2943	55.1168	0
254.8733	0.6912	1
239.0111	55.2218	0
239.0547	55.7427	0
254.894	0.5637	1
253.3823	17.1083	1
254.8629	0.6461	1
248.3303	36.713	1
245.4605	43.3289	1
254.8747	0.7565	1
254.9571	0.3771	1
243.8996	45.8803	1
237.41	59.7573	0
239.9134	55.4382	0
252.9467	20.7973	1
236.8608	58.6742	0
254.4536	10.1179	1
255	0	1
254.9301	0.4648	1
202.9238	92.845	0
254.5949	7.8721	1
210.0935	89.4153	0
200.9398	93.8813	0
204.6073	91.4243	0
196.588	96.3364	0
204.9003	90.7337	0
219.7271	79.7237	0
207.8151	88.1224	0
197.9467	95.502	0
205.8033	90.914	0
235.554	61.5706	0
248.41	36.4218	1
224.4972	75.7256	0
204.8269	92.7181	0
227.8179	73.9858	0
254.9785	0.2299	1
226.9134	71.692	0
254.5907	6.7858	1
211.8657	87.4232	0
222.6343	77.2149	0
218.7791	80.2267	0
226.5457	71.5528	0
217.2071	80.9066	0
231.2112	67.6938	0

135.5388	21.726	2
147.2805	11.0165	2
94.2521	19.5957	2
83.4584	3.537	2
79.2073	2.0189	2
75.6461	2.2781	2
95.5388	50.1086	2
254.9404	0.2424	1
202.6177	75.1637	0
78.8643	1.9085	2
81.4425	1.8114	2
83.6988	1.7492	2
80.7008	6.7757	2
118.5658	17.9989	2
134.3102	12.1626	2
129.8248	12.1687	2
128.4051	12.8679	2
131.6011	11.5638	2
83.7438	14.6613	2
82.0471	1.4459	2
77.7569	2.2011	2
75.4017	1.9019	2
95.3483	50.0385	2
254.9391	0.2449	1
202.3788	75.5918	0
78.4584	1.9332	2
80.7971	1.965	2
82.7909	1.5306	2
83.527	2.6271	2
89.7306	23.0055	2
117.1724	17.6352	2
113.0886	13.5141	2
108.3954	16.0979	2
105.3179	22.1551	2
76.6115	9.7263	2
81.0997	2.04	2
77.1039	2.0517	2
74.2057	1.8745	2
93.5589	50.826	2
254.9259	0.2748	1
254.9868	0.114	1
254.9688	0.381	1
254.9972	0.1052	1
202.2701	75.8107	0
77.1226	2.1877	2
79.687	2.2891	2
81.3712	3.3687	2
59.7084	28.8931	2
91.0651	42.0747	2
110.5139	21.8899	2
127.162	14.8201	2
117.795	21.5448	2
89.6704	27.6503	2
44.1898	30.9623	2

195.1759	96.32	0
210.4938	86.9956	0
221.117	78.505	0
226.9522	73.012	0
212.2091	86.0235	0
227.5118	72.3961	0
213.313	87.4118	0
211.8615	86.7822	0
211.7327	85.8758	0
212.0907	86.1622	0
210.777	86.9201	0
227.7569	73.3978	0
254.9342	0.571	1
229.7569	68.8794	0
253.7486	16.2551	1
202.5145	94.8361	0
210.3359	88.2189	0
199.714	93.672	0
212.259	87.3968	0
204.7659	90.4444	0
196.7569	97.8673	0
205.9515	89.5825	0
224.7749	73.8211	0
195.0845	98.5454	0
209.7403	87.505	0
193.6281	99.1466	0
208.5637	89.0309	0
241.8636	50.6907	1
249.2929	34.0934	1
221.4072	79.7004	0
210.9861	88.7021	0
207.5395	88.671	0
208.1454	88.5948	0
246.1537	42.8742	1
254.9882	0.1412	1
247.2798	37.8519	1
254.7327	5.6891	1
243.1115	46.918	1
249.7895	33.0049	1
245.7521	43.111	1
251.1544	27.1863	1
243.5229	47.1313	1
250.142	29.4927	2
250.4301	28.8656	2
242.527	48.3798	1
242.3663	50.0983	1
245.8781	42.3473	1
250.0824	29.5967	2
243.2112	48.3147	1
247.6053	38.1311	1
254.0062	13.3194	1
246.2929	41.1044	1
252.2528	22.4128	1
251.0062	27.5625	1

217.7396	82.7608	0
225.9377	71.8772	0
220.3504	79.7872	0
230.3006	68.5297	0
228.6025	70.4759	0
233.5429	65.2064	0
221.8518	77.1381	0
216.6046	84.1885	0
223.6066	75.7834	0
227.9065	71.8022	0
251.6496	25.9047	1
235.4494	61.128	0
218.3213	80.704	0
216.7756	81.4329	0
239.3497	54.5476	0
214.8947	83.3311	0
248.2846	36.6405	1
254.9861	0.2101	1
231.8546	66.0719	0
254.5893	7.9953	1
236.7112	60.8903	0
228.4197	70.4602	0
226.1323	72.2808	0
231.4148	66.2707	0
232.572	65.9812	0
237.142	59.7166	0
241.4072	51.4504	0
233.6884	63.938	0
220.0956	79.4032	0
245.9391	43.6517	1
233.5166	64.2955	0
232.5949	65.2134	0
233.1094	65.3645	0
237.1939	58.9884	0
233.4446	64.3227	0
235.4391	62.2983	0
251.6579	25.996	1
246.687	39.8328	1
231.0125	67.4658	0
227.6198	72.4221	0
240.1295	54.0968	0
226.7299	73.1388	0
254.0416	11.6959	1
254.9439	0.5051	1
209.6295	88.1046	0
253.5796	17.1069	1
200.4197	95.9349	0
205.4134	91.8869	0
197.3234	94.9656	0
207.3539	90.58	0
207.0679	88.5429	0
185.7895	101.0875	0
197.2999	95.4318	0
192.4508	98.2852	0

67.8566	21.6961	2
76.1925	2.6145	2
72.8857	1.5529	1
92.9065	50.8957	2
254.9176	0.2968	1
223.5658	76.9239	0
193.0145	97.5563	0
254.8892	1.9893	1
201.1634	77.5787	0
63.6101	20.7409	2
49.801	27.1763	2
29.8393	19.2409	2
15.651	1.8757	1
136.7126	86.2414	0
129.8823	38.3981	2
108.3054	20.587	2
110.5526	21.7773	2
149.8788	64.3664	0
14.3636	3.1636	1
15.7922	1.7971	1
26.4979	19.8298	2
43.7555	25.4032	2
82.0815	57.6617	2
254.9307	0.2646	1
186.1904	99.1439	0
20.2999	1.6504	1
20.8428	1.537	1
19.1614	2.2826	1
17.3019	2.053	1
90.6676	89.2687	0
230.8186	28.2018	2
165.2715	37.6725	2
195.6115	46.4248	0
141.7943	91.1648	0
15.5229	1.7918	1
15.8532	1.9076	1
16.6136	1.4005	1
15.9564	1.3095	1
44.7078	67.0064	2
254.9861	0.1169	1
186.3767	98.9512	0
20.8179	1.3838	1
19.6274	1.6463	1
18.7846	2.0914	1
19.0727	2.1022	1
44.2812	51.6895	2
199.3566	64.9991	0
63.1807	47.8587	2
134.885	72.6946	0
95.8241	84.1381	0
17.3096	1.9094	1
16.4446	1.4934	1
16.8269	1.5142	1
16.8269	1.6125	1

243.4162	46.9398	1
236.5166	61.2778	0
247.5547	37.9628	1
239.5388	57.8588	0
230.759	70.2781	0
242.0686	51.0366	1
242.9162	50.3393	1
227.7763	72.2958	0
243.5457	47.5219	1
251.1891	29.0688	2
233.5132	67.4983	0
211.518	91.0873	0
199.3684	99.1787	0
187.5506	104.0002	0
231.5873	69.987	0
200.045	98.2961	0
235.1087	65.4163	0
232.9252	65.763	0
251.9882	25.4454	1
222.6932	77.3617	0
215.3199	83.685	0
199.3622	94.861	0
254.9723	0.4185	1
254.919	0.5846	1
254.7091	4.9319	1
240.2458	52.2142	0
231.9252	64.9451	0

207.5796	89.2877	0
197.2244	96.4778	0
213.2645	82.4146	0
203.2008	91.9547	0
220.4917	77.2228	0
222.937	76.9859	0
203.732	91.1792	0
205.714	89.4207	0
224.4169	75.5879	0
254.9169	0.6553	1
236.1399	60.548	0
254.59	6.8965	1
203.509	94.331	0
212.8075	86.6283	0
236.5499	60.8917	0
254.9494	0.4775	1
239.3961	55.9703	0
254.6018	7.8996	1
237.0672	60.9846	0
241.732	52.0094	1
248.1981	38.4562	1
231.9494	69.2315	0
222.865	79.1382	0
231.3352	69.6311	0
221.4017	80.0709	0
232.6717	66.8702	0
230.6281	70.301	0

74.6683	60.6251	2
53.0789	7.868	2
86.4889	39.4347	2
17.4037	9.7102	2
15.6614	2.3457	1
16.9619	2.0081	1
17.2091	1.7858	1
17.1586	1.7125	1
46.2105	66.1774	2
254.9744	0.158	1
186.1399	99.158	0
20.7479	1.3781	1
19.6981	1.5232	1
19.8677	1.5683	1
17.6461	2.896	1
16.5416	2.3432	1
41.2791	30.708	2
86.2057	41.173	2
42.3968	18.4433	2
15.1904	2.4704	1
15.9314	2.4185	1
15.8123	1.6826	1
16.7701	1.7111	1
16.5506	1.715	1
44.099	66.1188	2
254.9785	0.1449	1
255	0	1

Pusat Cluster	
210.1603	86.1603
254.984	0.1749
82.6312	29.5238



Formulir Pendaftaran Lomba

SUPER COMM FIDENCE

n Komunikasi

Nama Peserta : Alan Budianto

Judul Karya Tulis : Menggapai Mimpi

Asal Institusi : Universitas Hasanuddin

Alamat : Jl. Sungai Saddang Lr IV No. 35 Makassar

No. Telepon : 081255123342

Dengan ini saya menyatakan bahwa karya yang saya berikan di sini adalah karya origi
untuk diikutsertakan dalam Lomba Karya Tulis

Makassar, 23 Oktober 20

Ttd

Alan Budianto

LAMPIRAN

Data Pada Formulir 3

Mean	SD	Cluster
90.0133	0.3124	1
90.0122	0.3281	2
90.0133	0.3297	2
90.0111	0.3195	1
90.2611	1.7203	2
91.2833	3.5791	2
90.1622	1.1663	2
90.0044	0.3333	2
90.0133	0.3124	1
90.0122	0.3281	2
90.0133	0.3297	2
122.42	64.7146	2
254.9167	0.2764	1
255	0	1
242.8733	51.2339	0
220.4411	83.972	0
242.3011	52.8992	0
230.35	71.2944	0
239.9189	57.2032	0
255	0	1
251.4644	27.962	1
215.7222	87.7019	0
239.26	58.9068	0
252.8011	21.609	1
230.8678	71.9534	0
229.7544	70.6265	0
252.2522	22.8012	1
254.9622	0.4355	1
219.3022	84.6675	0
230.6044	71.4391	0
75.81	27.3428	2
40.3856	28.9226	2
47.2033	18.4884	2
52.0433	29.5924	2
62.19	32.0933	2
90.01	0.2184	1
90	0	2
122.4333	64.7527	2
254.9333	0.2494	1
232.1	69.1469	0
164.2433	116.2726	0
212.3322	89.2555	0
203.2533	96.2943	0
198.5478	98.5125	0
176.8844	110.9701	0
212.9378	88.2077	0
196.2822	101.1394	0
165.1278	115.3658	0
211.0244	92.1149	0

Mean	SD	Cluster
236.7833	60.3048	0
237.7056	59.0107	0
231.3833	69.2479	0
238.0122	59.6032	0
235.8222	62.3264	0
227.4167	73.9626	0
214.7911	87.2516	0
237.7144	58.9083	0
238.4256	58.026	0
233.5189	64.8911	0
187.2578	105.3595	0
184.1167	105.2004	0
166.6633	111.9697	0
196.2556	98.1282	0
163.3644	113.4717	0
210.6411	91.2577	0
162.3656	114.4875	0
182.5544	108.3502	0
181.7011	107.5901	0
188.5	104.8002	0
162.9422	114.1478	0
162.3656	113.8497	0
177.3467	110.8122	0
167.7989	111.189	0
194.3089	99.9766	0
154.9656	112.7937	0
196.9678	100.1424	0
254.9856	0.1193	1
254.9278	0.4299	1
254.9822	0.1321	1
254.9267	0.4609	1
254.9533	0.3232	1
254.9856	0.1193	1
254.9933	0.0814	1
254.9544	0.3545	1
254.9567	0.2694	1
254.9889	0.1048	1
254.9567	0.315	1
254.9133	0.5937	1
254.9811	0.1361	1
254.9844	0.1237	1
254.9244	0.4771	1
254.8678	0.6265	1
254.9944	0.0743	1
236.0856	63.2628	0
254.8811	0.6309	1
254.9233	0.5454	1
215.0967	80.3596	0
247.3478	37.3584	1

Mean	SD	Cluster
191.47	96.1273	0
192.1756	95.9798	0
201.5389	97.4429	0
193.2278	96.0574	0
191.7511	95.2965	0
185.3056	100.3236	0
205.2378	90.3418	0
254.04	9.6025	1
254.46	5.5367	1
254.9344	0.5468	1
254.4011	5.1457	1
254.1256	7.9253	1
254.4622	5.6429	1
254.6122	2.9234	1
254.1767	7.0647	1
237.8344	56.2292	0
254.1322	7.9254	1
254.6378	1.4483	1
254.5033	5.6003	1
253.29	12.8996	1
254.5944	1.4903	1
254.4878	4.0023	1
253.35	12.8699	1
254.6956	1.3616	1
254.5722	2.697	1
253.2011	13.3792	1
254.6689	1.2576	1
254.7644	1.1499	1
254.92	0.5014	1
254.9133	0.5649	1
254.9311	0.4964	1
254.9378	0.4927	1
254.9811	0.3378	1
255	0	1
254.9322	0.4306	1
254.9933	0.1413	1
254.9356	0.4694	1
255	0	1
254.9844	0.2206	1
254.9444	0.4017	1
254.9189	0.5707	1
254.9789	0.2508	1
254.9678	0.2496	1
254.9322	0.4254	1
254.9222	0.5401	1
254.9411	0.397	1
254.9844	0.2049	1
185.5044	103.5133	0
216.5533	80.8034	0

191.6167	103.0826	0
158.4678	117.1193	0
171.1511	113.0295	0
210.6622	87.4876	0
181.3567	109.0704	0
217.2589	85.4003	0
176.0678	113.5896	0
198.9	98.7214	0
178.9222	111.6072	0
188.3244	105.2024	0
195.3744	101.0041	0
207.1911	92.9694	0
214.5844	85.688	0
202.1844	96.4541	0
181.4678	109.0376	0
254.9356	0.4113	1
172.2	114.7532	0
240.4611	56.3374	0
196.5367	101.3269	0
204.2267	95.038	0
178.5678	110.1006	0
180.8989	109.5042	0
218.4044	84.6539	0
192.5933	101.9763	0
254.9756	0.1544	1
89.1656	5.2328	2
30.4667	20.6056	2
22.9344	13.3178	2
83.5756	62.509	2
46.5733	39.3502	2
39.0622	19.1572	2
87.8211	8.0898	2
122.4333	64.7527	2
254.9333	0.2494	1
237.5478	60.9418	0
217.3122	88.6452	0
218.8456	84.779	0
205.7667	95.165	0
218.4189	85.4867	0
180.7233	111.3303	0
209.6978	92.6047	0
196.5122	100.8559	0
188.8244	106.5898	0
209.3756	94.6013	0
211.6611	92.1418	0
251.4333	27.9707	1
203.6722	98.6112	0
222.08	81.1431	0
232.7667	64.0963	0
182.6567	110.1392	0
213.4189	88.6707	0
192.1733	104.5492	0
186.9544	106.2194	0

239.8833	52.7108	0
247.9567	36.5873	1
222.8533	77.6622	0
232.1656	67.1639	0
245.04	40.5999	1
234.3778	64.073	0
245.7533	40.2223	1
254.9989	0.0333	1
250.1522	31.8608	1
230.9978	67.899	0
229.2289	71.1056	0
244.5311	44.1308	2
215.3	83.8971	0
236.3467	60.5678	0
229.3844	68.432	0
238.0756	55.8384	0
243.9014	45.0684	2
236.92	59.1776	0
254.6333	7.679	1
236.4789	63.2365	0
246.6489	40.0952	1
254.9056	0.6725	1
225.9844	72.1854	0
219.3556	78.9063	0
199.2033	98.7691	0
217.5311	79.7498	0
229.97	71.4594	0
228.7733	69.3509	0
210.2156	85.9234	0
214.46	86.4598	0
215.7333	80.7567	0
254.9122	0.5657	1
243.3511	46.5521	2
222.1167	77.2171	0
201.8189	93.9867	0
216.4822	84.2534	0
222.62	78.9172	0
205.1733	93.6715	0
216.87	83.1646	0
201.1511	93.3942	0
216.3189	84.221	0
209.8689	89.7702	0
247.1889	41.7155	1
243.5656	45.8665	2
242.3378	52.4607	0
233.2578	64.1459	0
253.5178	11.09	1
236.9156	58.1874	0
238.3267	58.0183	0
226.0067	71.7835	0
252.8456	15.2718	1
253.1589	10.5956	1
253.5811	9.7329	1

207.6422	89.7549	0
208.3367	86.8736	0
217.2433	82.2362	0
220.3	79.3224	0
205.4233	90.8128	0
232.5033	63.4961	0
216.0956	82.2221	0
214.3333	81.9197	0
225.5944	75.1817	0
217.8911	82.0319	0
207.8889	88.9559	0
213.96	84.829	0
219.9422	77.4362	0
214.9889	83.7386	0
188.6989	100.3356	0
216.5256	81.1653	0
224.6389	76.0201	0
200.5878	94.2638	0
191.1233	100.3777	0
215.4211	83.3399	0
219.7644	76.8236	0
202.5444	93.8817	0
211.8156	85.8215	0
222.6389	75.1912	0
220.0689	76.4961	0
229.3811	70.4818	0
221.6911	74.2885	0
200.5956	95.123	0
226.2278	70.2044	0
228.2933	68.5427	0
208.6633	88.8813	0
218.3389	79.0413	0
217.5378	80.7031	0
210.5856	87.7539	0
206.8078	89.5104	0
198.2744	94.7218	0
210.7567	86.0408	0
222.1578	79.8818	0
193.87	97.0227	0
232.2567	63.7122	0
194.4333	97.3753	0
231.5956	66.8477	0
207.6778	89.7513	0
192.2189	97.8185	0
204.6689	92.7941	0
196.8033	96.4169	0
208.2422	90.5304	0
207.73	87.9914	0
219.0867	80.15	0
224.4756	71.0718	0
221.17	80.9003	0
212.2111	86.7014	0
209.0122	86.3352	0

210.3467	94.0309	0
211.2167	89.9404	0
178.0433	110.8806	0
202.9489	97.4125	0
212.0011	90.5516	0
198.4378	99.4966	0
182.7544	110.1216	0
252.02	24.7138	1
199.38	101.1438	0
212.7322	89.5858	0
198.9567	99.867	0
209.1467	93.9095	0
178.3233	112.1263	0
203.5833	95.38	0
217.56	84.9101	0
172.2611	112.9243	0
252.11	23.3382	1
90.0244	0.5005	2
90.0244	0.4916	2
78.5789	25.5114	2
18.2489	13.1564	2
68.8422	61.8977	2
207.2456	14.7548	0
149.3444	59.1118	2
45.5489	22.9286	2
66.3322	30.103	2
90.0111	0.5076	2
90.0267	0.507	2
122.4244	64.7073	2
254.89	0.3129	1
90.13	0.5393	2
90.1289	0.5628	2
78.3067	24.5954	2
61.3978	63.6533	2
144.33	54.0484	2
213.5967	16.029	1
176.3444	44.2583	2
100.0078	79.4805	2
78.5956	33.5882	2
90.1289	0.5628	2
90.13	0.5635	2
122.4889	64.6778	2
254.8689	0.3375	1
94.43	11.8293	2
125.2356	47.9067	2
207.6767	14.6821	0
196.64	18.8092	0
219.9689	16.1469	1
182.2933	50.8212	2
106.0767	32.0314	2
90.0133	0.2663	1
90.0122	0.2643	1
122.4178	64.7155	2

241.4689	49.9611	2
230.51	68.5802	0
236.0033	60.5397	0
250.4733	27.9128	1
250.6989	24.345	1
212.8878	85.8778	0
253.1067	13.827	1
253.1756	14.9773	1
252.8089	16.2814	1
253.7611	9.4367	1
253.3133	13.1421	1
248.8444	32.909	1
227.7489	68.3629	0
230.15	69.023	0
253.4189	11.1354	1
253.0689	13.5622	1
249.5433	32.3631	1
237.9122	57.5217	0
239.3289	55.5432	0
254.92	0.613	1
211.4289	88.0344	0
197.7544	97.5388	0
204.7622	90.9709	0
211.4644	91.3412	0
198.9344	99.7779	0
208.5344	87.9048	0
199.8267	95.0475	0
206.6044	89.0098	0
243.9033	44.8408	2
211.7256	90.0875	0
198.3844	98.4412	0
215.6044	81.3209	0
237.7267	56.451	0
200.9822	95.4125	0
201.5222	93.3667	0
196.0156	93.9929	0
174.5711	100.6557	0
198.7033	92.7941	0
191.2456	98.7199	0
208.3856	89.7307	0
221.68	75.444	0
203.29	94.5528	0
187.0633	104.4677	0
193.0456	97.4888	0
233.6067	65.9038	0
254.9267	0.4844	1
254.9556	0.3715	1
254.8967	0.4915	1
254.9489	0.3911	1
254.9322	0.4954	1
254.9867	0.1759	1
254.8989	0.6252	1
247.98	35.6368	1

205.4089	89.964	0
208.2211	89.6176	0
244.2556	46.1304	2
248.62	32.4282	1
229.5956	69.0722	0
230.2533	66.6842	0
242.59	49.6896	2
246.2622	42.6111	1
241.4767	52.351	0
250.1278	28.3324	1
239.0178	54.8628	0
241.8311	48.7996	2
245.4933	44.1377	2
244.0122	47.671	2
244.6656	43.5868	1
238.7967	54.9381	0
243.7867	46.6161	2
243.3433	47.159	2
238.5067	57.5933	0
241.2856	51.115	0
246.1033	43.1283	1
244.8933	43.1533	1
239.0122	55.8827	0
247.3333	39.3611	1
246.4433	40.5601	1
244.5033	46.073	2
239.6456	55.3115	0
242.7522	48.7182	2
242.4911	47.6547	2
245.0456	43.8655	1
242.8589	46.9219	2
239.7967	55.1282	0
223.6111	73.5052	0
248.5856	33.3754	1
236.0078	58.5122	0
244.6111	45.2735	2
246.5333	39.6045	1
246.9289	37.1239	1
243.8611	47.8765	2
240.7589	54.1746	0
240.6922	52.5631	0
247.6167	39.0973	1
241.6111	50.4103	0
249.8944	28.8512	1
239.4633	56.2968	0
248.5567	36.4811	1
241.4622	50.9092	0
239.6178	53.5885	0
241.4889	51.9898	0
241.6667	50.772	0
245.8033	44.4141	2
238.3567	57.2671	0
242.7056	48.565	2

254.8689	0.3375	1
89.93	0.3102	2
119.3656	31.8137	2
203.0744	15.8034	0
197.8822	22.5468	0
225.9222	7.8942	1
156.5011	57.2466	2
90.0722	0.8167	2
90	0	2
90	0	2
122.4133	64.7117	2
254.8733	0.3326	1
254.9011	0.6273	1
232.9989	63.5576	0
242.3633	47.303	2
227.4344	72.9529	0
233.3278	63.7102	0
243.3656	48.5595	2
237.3744	57.3695	0
237.4611	57.67	0
242.5822	50.5281	0
230.4556	69.0664	0
242.73	50.1273	0
239.2022	55.4574	0
229.7411	68.645	0
232.4567	65.6452	0
234.2567	62.6854	0
243.7578	45.5187	2
234.8322	61.2052	0
243.2967	47.0073	2
94.2522	15.0214	2
170.13	26.6043	2
197.0822	21.1526	0
205.7178	12.9766	0
113.1267	41.3814	2
122.4	64.6847	2
254.9011	0.2985	1
254.8211	0.7632	1
192.7189	98.9508	0
220.9778	76.0132	0
189.3611	104.4098	0
196.3033	96.509	0
219.3189	81.1658	0
214.75	84.1958	0
194.2167	99.8744	0
203.2867	92.1583	0
194.42	98.5545	0
197.3733	96.3967	0
215.5311	84.9067	0
193.7789	100.8358	0
190.0656	102.0589	0
215.6733	79.8238	0
191.3278	100.3089	0

250.5078	31.616	1
255	0	1
248.4156	37.1371	1
255	0	1
245.9989	42.844	1
242.4667	49.6579	2
254.9967	0.0576	1
254.2656	9.8899	1
250.5411	30.2409	1
255	0	1
249.84	31.7583	1
253.2489	14.3803	1
239.5544	53.4646	0
249.9678	31.7882	1
254.8544	0.8511	1
253.1478	19.1986	1
251.7356	25.5447	1
243.0756	48.5089	2
242.0844	49.0165	2
252.4844	18.6265	1
238.4867	57.2588	0
249.9967	31.7565	1
239.2622	51.5147	0
226.0344	74.8191	0
254.9044	0.7286	1
204.1511	91.7631	0
209.2556	83.7905	0
201.1611	95.8256	0
215.9522	87.1368	0
189.2556	98.745	0
209.1944	86.8105	0
191.3922	100.8832	0
202.6833	90.9186	0
222.4822	75.8225	0
255	0	1
255	0	1
255	0	1
255	0	1
255	0	1
255	0	1
240.4067	51.1754	0
211.5111	88.2833	0
183.4811	102.4627	0
224.62	72.8082	0
198.4589	93.6264	0
210.5789	83.4953	0
204.0467	92.4301	0
195.3278	93.519	0
213.2022	87.4351	0
199.2689	95.397	0
200.4722	89.7329	0
202.9044	90.1294	0
194.02	98.4868	0

243.3689	46.375	2
247.2344	38.5483	1
245.8911	41.7773	1
225.0433	73.0758	0
237.8722	57.2361	0
240.1478	54.2486	0
250.8911	27.3698	1
254.9644	0.4794	1
249.1889	31.7071	1
249.8411	30.7524	1
250.8467	27.4056	1
250.5589	28.3452	1
253.2744	16.6922	1
252.58	22.3629	1
250.6467	28.4275	1
252.7456	18.9528	1
250.9022	27.2728	1
250.7111	28.2569	1
248.6456	30.7158	1
241.5989	50.0774	2
253.8356	11.826	1
250.6856	29.0828	1
209.44	89.8021	0
204.1989	91.0148	0
200.4811	94.4415	0
178.4867	105.1709	0
229.3844	69.5456	0
183.1144	101.3007	0
176.32	105.6364	0
206.9289	90.3733	0
196.6656	94.1402	0
198.4744	93.6341	0
206.9711	90.1653	0
180.1922	102.2181	0
198.0589	93.7118	0
205.1656	90.7002	0
197.7011	96.6388	0
203.1356	91.0232	0
186.46	101.5685	0
201.9189	92.6289	0
201.0156	99.5505	0
216.9289	83.45	0
202.8122	91.7614	0
177.1511	106.5458	0
199.4944	93.2749	0
213.3856	89.278	0
197.4889	96.8642	0
194.6144	95.2672	0
220.4589	76.9534	0
212.43	84.381	0
203.6911	96.1142	0
205.3156	90.7686	0
178.6156	105.2034	0

222.1989	77.3339	0
226.0078	70.4448	0
90.0389	0.3795	2
82.45	17.5384	2
54.9633	37.745	2
193.5822	34.1404	0
199.1467	18.6113	0
221.2644	16.8754	1
118.1744	87.3038	0
79.7089	16.7785	2
89.81	2.2412	2
90.0011	0.0333	2
122.4333	64.7527	2
254.9333	0.2494	1
72.6789	25.4814	2
49.8189	25.1037	2
32.3889	7.0136	1
28.9244	7.7076	1
229.51	50.8618	2
227.1744	16.1202	1
244.48	18.1774	1
97.6133	84.3883	2
46.4478	2.0018	1
52.12	15.2529	2
70.0122	21.0367	2
120.8978	65.9207	2
254.9067	0.2909	1
28.4678	3.9101	1
29.4578	5.0477	1
34.0522	5.7722	1
30.9067	5.2819	1
194.4878	85.7225	0
108.4722	74.0004	2
211.1578	73.8401	0
59.5422	39.8441	2
45.2233	4.2536	1
43.1011	4.1923	1
44.1733	3.629	1
88.6978	83.0302	2
254.8511	0.411	1
30.2733	3.2329	1
30.2656	3.6745	1
31.3444	5.4493	1
33.6833	3.8845	1
157.7667	101.6969	0
116.9478	80.503	2
216.4856	74.9506	0
46.6844	4.2854	1
42.1	3.4492	1
45.9422	3.6798	1
43.9178	7.4343	1
85.1967	84.3216	2
254.9333	0.2494	1

201.7389	91.7618	0
176.1144	107.0136	0
198.9367	93.9422	0
238.3311	59.6004	0
254.9778	0.2529	1
241.8656	48.8689	2
254.2989	10.9262	1
248.08	35.3635	1
254.6944	6.8624	1
249.63	31.7087	1
250.6967	28.6149	1
249.3389	32.8024	1
249.3089	32.8287	1
247.2967	38.0622	1
254.7033	6.0105	1
247.0489	38.4912	1
244.2878	42.9984	1
245.5822	41.1112	1
250.9133	24.3268	1
250.29	28.7118	1
246.6844	42.0325	1
244.9078	43.3211	1
250.6656	28.9488	1
249.6933	31.6401	1
225.5233	73.1197	0
231.3544	65.3775	0
254.8944	0.7141	1
199.9856	94.5146	0
187.5589	102.0756	0
196.0367	96.3719	0
202.87	92.0441	0
209.7478	88.9764	0
234.7156	61.7354	0
180.2222	107.4396	0
236.1478	59.7052	0
195.7511	96.8007	0
202.3311	92.8774	0
184.7622	100.1149	0
206.6456	87.1919	0
207.7189	90.2407	0
193.58	97.2383	0
191.9733	97.5136	0
186.4344	101.6217	0
198.9356	92.5611	0
185.5078	99.9549	0
212.9733	90.4936	0
214.4867	83.4823	0
213.4367	90.2551	0
212.6889	86.4732	0
236.4289	54.8787	0
197.3144	97.6315	0
227.0756	72.0776	0
210.6967	86.3287	0

242.64	47.1038	2
254.8778	0.7458	1
254.7967	1.0131	1
254.8289	0.8176	1
254.8256	0.8496	1
254.9	0.5991	1
254.8489	0.8038	1
254.8478	0.8434	1
254.84	0.8144	1
254.8244	0.9299	1
254.8811	0.7426	1
254.8533	0.7398	1
254.8444	0.8655	1
254.8256	0.8728	1
254.8478	0.8139	1
254.9156	0.4662	1
254.85	0.7504	1
254.85	0.7751	1
254.84	0.7838	1
254.8756	0.7718	1
254.9256	0.5944	1
254.8122	0.9317	1
254.84	0.8737	1
254.8733	0.6876	1
254.9089	0.6536	1
254.8922	0.5721	1
254.8189	0.856	1
248.1189	35.4672	1
254.8644	0.7684	1
254.8989	0.6581	1
254.85	0.8156	1
254.83	0.8347	1
254.9378	0.5083	1
254.8789	0.7526	1
168.2867	107.3692	0
224.03	72.3767	0
199.6	93.7996	0
214.6167	81.6792	0
222.6833	74.416	0
212.2556	85.2156	0
249.0467	31.3393	1
210.11	83.5227	0
198.7822	98.2015	0
197.1311	97.8425	0
204.2856	91.7233	0
216.8244	81.8237	0
194.0333	99.8561	0
221.4067	77.1661	0
222.2078	74.7827	0
180.3656	104.6815	0
193.9144	97.2948	0
254.96	0.363	1
241.3956	52.3839	0

24.2211	3.3328	1
25.89	3.4477	1
33.9044	3.8274	1
33.1189	2.1598	1
108.3811	95.8202	0
89.1889	61.4619	2
153.5144	98.6963	0
45.0278	2.9725	1
42.4944	1.9093	1
44.1733	3.3713	1
40.3167	5.5917	1
80.06	86.799	2
254.9467	0.2247	1
17.4511	5.9635	1
21.2211	5.2857	1
27.7678	4.7919	1
33.1644	2.6427	1
52.9044	48.8714	2
66.2	31.501	2
83.2756	69.6393	2
43.2278	2.2066	1
42.0078	3.1887	1
43.0522	2.8726	1
32.1511	6.9376	1
75.1511	89.3243	0
254.9911	0.0939	1
70.0622	101.4691	2
70.1433	101.4707	2
78.7367	96.6365	2
83.3111	93.9967	0
89.2256	90.8723	2
108.3033	82.4101	2
89.3811	91.7504	0
90.6067	90.1522	2
89.6789	90.6758	2
89.9778	90.4168	2
76.7733	98.0031	2
115.9122	109.5131	0
254.9733	0.1611	1
191.2589	102.6454	0
223.7944	78.7435	0
217.5711	85.4576	0
237.6478	58.8809	0
234.1044	65.0303	0
247.9689	39.6914	1
199.1611	96.282	0

216.9056	79.8156	0
189.5278	97.7334	0
189.0922	101.3931	0
195.8278	96.3194	0
196.5311	93.4324	0
213.0967	80.4499	0
191.2278	97.3862	0
251.1044	28.1843	1
254.9411	0.5054	1
254.9067	0.6075	1
244.3233	42.309	1
254.9622	0.3632	1
244.3567	42.3316	1
254.95	0.5761	1
254.9611	0.3615	1
254.9656	0.4189	1
254.9533	0.4571	1
254.76	1.1226	1
254.9789	0.272	1
254.9111	0.6573	1
254.8156	0.9806	1
254.8844	0.7468	1
254.9722	0.3168	1
254.8567	0.799	1
254.8467	0.7852	1
254.8133	0.9229	1
254.8433	0.7781	1
254.8078	0.9567	1
254.8878	0.6233	1
254.8278	0.8422	1
254.8978	0.682	1
254.8767	0.7883	1
228.7356	67.2882	0
228.9522	71.4211	0
254.9344	0.5366	1
186.5844	97.3865	0
214.7478	85.1275	0
238.5	55.3575	0
180.3756	106.6131	0
185.7422	102.7891	0
205.35	88.9426	0
206.3356	92.0305	0
206.3411	91.0198	0
217.12	83.1407	0
185.1878	98.913	0
190.8778	103.1907	0

239.4744	54.3946	0
235.8911	62.2861	0
240.13	51.0418	0
243.2956	46.218	2
239.4244	54.7563	0
241.5633	50.8711	0
236.2967	60.6959	0
249.9911	32.0116	1
244.5011	43.7592	2
238.4878	57.329	0
244.7111	42.5562	1
240.9256	51.4345	0
245.5044	42.4028	1
237.75	60.5919	0
247.4689	36.0317	1
244.8578	43.8684	2
212.2144	88.8817	0
224.3667	75.1085	0
211.7744	87.0176	0
242.6533	52.1788	0
242.9456	48.7362	2
235.0822	62.0556	0
232.0911	67.2464	0
228.84	70.9901	0
243.7856	45.5406	2
226.8811	73.1876	0
227.6189	71.4251	0
226.8489	71.7761	0
239.2089	53.6385	0
243.0222	47.2387	2
239.4278	55.6199	0
251.4167	25.4659	1
254.9856	0.1193	1
220.4233	78.6994	0
208.4833	90.1016	0
208.6789	89.5364	0
230.7022	70.4636	0
210.4344	88.3209	0
204.3633	94.1399	0
212.1556	85.6142	0
204.9756	92.8697	0
218.8244	82.3227	0
237.8667	59.4894	0
255	0	1
255	0	1
255	0	1

Pusat Cluster	
203.105	91.7738
254.9709	0.1856
89.5269	46.6712

FORMULIR Identitas Pendaftaran



Program : Reguler
Nama Lengkap : Stanley Miller
Jenis Kelamin : Laki-Laki
Tempat Lahir : Makassar
Tanggal Lahir : 9 Nopember 1987
Propinsi : Sulawesi Selatan
Kabupaten/Kota : Makassar
Alamat Rumah : Jl. Pampang Utama Lr I No. 32 Makassar 90231
Telp / Hp : 081234123456
Propinsi : Sulawesi Selatan
Kabupaten / Kota : Makassar
Nama Orang Tua : Adolf Miller
Pekerjaan Orang Tua : Wiraswasta
Asal Sekolah : SMA 17 Makassar
Propinsi : Sulawesi Selatan
Kabupaten / Kota : Makassar
Jurusan : IPA
Pilihan Prodi (1) : Matematika
Pilihan Prodi (2) : Ilmu Komputer
No. Tes : 009876754321

Bandung, 23 Juni 2004

Pendaftar,

Ttd

Stanley Miller

AMPIRAN
Data Pada Formulir 4

Mean	SD	Cluster
255	0	2
241.6713	55.0702	0
217.2204	86.6856	0
215.3991	87.9211	0
213.8321	89.9228	0
219.1775	85.0712	0
207.4953	95.1432	0
214.5104	88.6282	0
218.26	85.57	0
216.171	87.8474	0
230.2163	72.6411	0
255	0	2
255	0	2
251.7779	22.9619	2
242.9469	43.2056	0
243.3699	41.6426	0
243.5776	40.8374	0
243.5866	40.8177	0
243.3922	41.5505	0
254.3752	9.7778	2
255	0	2
250.5176	31.3733	2
235.1009	62.6146	0
229.5496	70.326	0
237.5638	59.1381	0
236.0826	61.5436	0
232.1751	66.7199	0
234.2339	64.2416	0
252.2961	24.0501	2
209.3237	76.732	0
83.4562	3.1254	1
83.7348	20.1873	1
66.5548	37.6923	1
91.6326	12.4687	1
90.5565	1.7114	2
246.4842	36.2023	0
241.5779	55.8039	0
201.7793	100.4477	0
196.2537	104.147	0
218.6336	86.9477	0
230.8569	69.207	0
232.4972	66.7555	0
230.9481	68.3719	0
229.7506	69.9342	0
248.9881	36.1274	0
208.591	77.9572	0
82.0334	3.1975	1
56.6192	40.6767	1

Mean	SD	Cluster
71.1121	23.5959	1
246.3743	36.6962	0
250.4236	30.7038	1
237.9743	57.9804	0
235.0207	62.8039	0
240.1296	54.68	0
250.4427	31.4106	2
236.8257	59.8343	0
241.7388	51.5794	0
198.7532	94.581	0
42.0682	3.3144	2
77.9265	72.4873	1
165.2972	87.461	0
43.1662	10.3285	1
40.2988	3.6826	2
243.9245	47.6933	0
249.5509	34.2266	0
238.8383	56.7251	0
236.5905	60.4365	0
240.5909	53.6546	0
251.491	27.171	2
232.5176	65.7832	0
242.7191	49.198	0
197.294	97.0667	0
34.6551	5.4403	1
41.1567	8.8051	1
51.3861	30.3596	1
41.049	5.9627	1
33.7936	5.7529	1
243.9138	47.7014	0
237.4353	58.8476	0
234.7281	62.8235	0
239.5183	55.4692	0
251.8859	25.372	2
235.0784	62.3716	0
234.6328	63.2771	0
238.0425	57.991	0
251.8678	25.7141	2
253.2006	19.8375	2
248.1856	38.1157	0
248.383	36.9624	0
248.5292	36.1464	0
248.3801	36.9678	0
248.2165	37.8822	0
254.6604	8.325	2
236.1247	61.5841	0
241.8431	51.3685	0
251.7283	26.2514	2

Mean	SD	Cluster
253.6698	16.2804	2
233.5751	65.088	0
229.3446	69.6151	0
237.9655	58.3475	0
246.6399	41.3712	0
252.2475	24.0872	2
228.7101	70.5775	0
234.5435	62.4953	0
253.8944	14.991	2
250.6709	30.411	1
233.1477	65.5223	0
234.4633	63.1888	0
233.387	64.9372	0
242.1142	50.9595	0
252.2132	24.4483	2
231.4643	67.3393	0
236.2057	60.6812	0
249.7687	33.2058	0
252.002	25.0601	2
233.5959	64.578	0
228.4457	70.6709	0
235.058	62.6017	0
242.6491	50.2915	0
252.1179	24.3398	2
228.4973	71.2376	0
228.6787	70.568	0
241.2824	52.031	0
250.9006	29.0033	2
254.0879	13.4969	2
234.0997	64.4204	0
238.0356	57.7569	0
251.041	29.0035	2
255	0	2
251.1431	28.3327	2
232.355	66.198	0
236.2176	60.4158	0
229.887	68.6145	0
252.9928	20.5404	2
232.3994	66.5631	0
232.1271	66.3697	0
240.0582	54.9383	0
247.2688	39.8505	0
250.968	29.1185	2
230.1693	68.6649	0
240.4439	53.2667	0
251.8714	25.6895	2
238.2972	58.1521	0
243.4356	48.259	0

89.8374	75.4043	1
58.0586	38.5456	1
88.7072	1.7007	2
246.3538	36.7896	0
233.6	68.5577	0
193.2647	89.7737	0
83.9813	3.5911	1
119.1626	49.9841	1
188.078	33.0304	1
108.7512	34.7171	1
89.7622	1.9627	1
226.1097	73.935	0
255	0	2
251.3151	27.9399	2
235.9312	61.4811	0
240.8223	53.5242	0
250.4562	31.6927	2
233.1617	65.0533	0
248.3643	36.9529	0
209.0265	77.217	0
81.2938	2.3988	1
106.5652	35.2379	1
159.8214	26.5602	1
99.0638	21.0022	1
90.4201	1.9511	2
246.4248	36.479	0
250.0396	32.2919	2
237.1673	59.3936	0
238.8026	56.474	0
231.8603	66.8679	0
251.7188	26.3058	2
232.7999	65.3917	0
235.2748	62.3226	0
245.1043	44.9443	0
205.7073	83.3251	0
55.8549	18.8898	1
106.5332	81.7994	1
172.1242	41.6207	1
68.6318	57.4844	1
252.2155	24.343	2

234.4525	63.3463	0
237.3828	58.5953	0
234.151	63.4331	0
235.4501	62.7053	0
233.51	64.5046	0
234.9862	62.7599	0
252.3001	23.6542	2
251.5393	27.1719	2
232.6288	65.8074	0
242.7948	49.203	0
250.1225	31.977	2
237.32	59.2756	0
235.5669	62.2783	0
235.3714	62.1813	0
255	0	2
250.3114	31.7187	2
238.4611	57.5283	0
232.443	66.2206	0
236.9388	58.8956	0
238.3872	57.7344	0
239.8578	54.7497	0
240.6279	53.7516	0
233.0599	64.7386	0
235.9581	60.7882	0
241.4613	52.3707	0
237.1244	59.4837	0
239.3278	56.3053	0
251.1999	28.3706	2
234.59	62.8464	0
237.382	59.0488	0
241.1387	53.201	0
233.4106	65.1883	0
241.7969	51.3539	0
252.8184	20.904	2
254.8449	5.1016	2
250.7244	30.0107	1
232.3852	66.0568	0
237.4559	58.4687	0
234.1171	63.3644	0
255	0	2

250.506	31.0745	2
243.0398	49.9971	0
249.01	35.2565	0
234.7912	63.3975	0
237.2423	59.6519	0
236.1817	60.9228	0
251.8076	25.8773	2
251.524	27.1604	2
232.9865	65.4052	0
233.7901	64.0396	0
253.6768	16.9676	2
249.2495	34.6574	0
234.7928	63.3832	0
237.1449	59.6611	0
235.1386	62.4411	0
251.5201	26.7715	2
250.4758	31.0541	2
235.2424	62.9048	0
234.6196	63.3643	0
241.0338	52.8758	0
245.5506	44.3312	0
243.0344	48.7419	0
243.4201	48.3002	0
251.2212	28.3583	2
232.1406	66.455	0
236.5536	60.3253	0
243.7273	47.9967	0
240.1762	54.9971	0
232.0756	66.7521	0
241.4269	52.1772	0
236.3263	61.0914	0
240.9855	53.5686	0
244.9096	45.8342	0
231.7643	66.6165	0
249.0571	35.0723	0
241.4138	52.3766	0
244.4793	46.1937	0
236.8071	59.6411	0
232.2844	66.4834	0
255	0	2

Pusat Cluster	
232.2624	64.965
79.4162	30.1237
254.9913	0.1399


```
LogPal = record                               umain.~pas
  lpal: TLogPalette;
  entry: array [0..255] of TPaletteEntry;
end;
```

```
Twarna = record
  R : byte;
  G : byte;
  B : byte;
end;
```

```
Tmaxmin = record
  nilai : real;
  idx : longint;
end;
```

```
var
  frmmain: Tfrmmain;
  PaletKeabuan: LogPal;
  warna : array [0..2] of Twarna;
  wbaru : array of byte;
  pix, bpix : array of array of byte;
  x : array of array of real;
  v : array of array of real;
  u, ulama : array of array of real;
  ukr : integer;
  wimg2, himg2, wimg, himg : integer;
  ukuran : longint;
  jciri, jcluster : integer;
```

Implementation

```
$R *.dfm}
procedure Tfrmmain.simulasi;
var i,j : longint;
    nmax : real;
    idmax : byte;
    PH: PByteArray;
    wmax : Tmaxmin;
    wmin : Tmaxmin;
    orwarna : array[0..2] of byte;
    tota : byte;
    k,l : byte;
    idx : integer;
```

```
begin
  GetLength(wbaru,ukuran);
```

```
max.nilai := 0;
min.nilai := 10000;
```

```
for j:= 0 to ukuran-1 do
begin
  nmax := u[0,j];
  idmax := 0;
  for i := 1 to 2 do
    if nmax < u[i,j] then
      begin
        nmax := u[i,j];
```

```

    idmax := i;
end;

if x[j,1] > wmax.nilai then
begin
    wmax.nilai := x[j,1];
    wmax.idx := idmax;
end;
if x[j,1] < wmin.nilai then
begin
    wmin.nilai := x[j,1];
    wmin.idx := idmax;
end;

wbaru[j] := idmax;

//memo1.Lines.Add(inttostr(j)+'|'+inttostr(wbaru[j]));

end;
wimg2 := floor(wimg/ukr);
himg2 := floor(himg/ukr);
setlength(bpix,himg2,wimg2);
image3.Picture.Bitmap.PixelFormat := pf8bit;
image3.Picture.Bitmap.Width := wimg2;
image3.Picture.Bitmap.Height := himg2;

orwarna[wmax.idx] := 2;
orwarna[wmin.idx] := 0;
tota := wmax.idx+wmin.idx;
if (tota = 1) then orwarna[2] := 1;
if (tota = 2) then orwarna[1] := 1;
if (tota = 3) then orwarna[0] := 1;
idx := 0;
for i := 0 to floor(himg/ukr)-1 do
begin
    PH := image3.Picture.Bitmap.ScanLine[i];
    for j := 0 to floor(wimg/ukr)-1 do
    begin
        {PH[3*j] := warna[orwarna[wbaru[idx]]].R;
        PH[3*j+1] := warna[orwarna[wbaru[idx]]].G;
        PH[3*j+2] := warna[orwarna[wbaru[idx]]].B;}
        if orwarna[wbaru[idx]] = 1 then
        begin
            bpix[i,j] := 1;
            PH[j] := 0;

        end else
        begin
            PH[j] := 255;
            bpix[i,j] := 0;
        end;
        inc(idx);
    end;
end;
image3.Repaint;

id;

inction Power2(Base, Exponent : Double) : Double;
raises the base to the exponent }
CONST
    cTiny = 1e-15;

```

```

VAR
  Power : Double; { Value before sign correction }
BEGIN
  Power := 0;
  { Deal with the near zero special cases }
  IF (Abs(Base) < cTiny) THEN BEGIN
    Base := 0.0;
  END; { IF }
  IF (Abs(Exponent) < cTiny) THEN BEGIN
    Exponent := 0.0;
  END; { IF }

  { Deal with the exactly zero cases }
  IF (Base = 0.0) THEN BEGIN
    Power := 0.0;
  END; { IF }
  IF (Exponent = 0.0) THEN BEGIN
    Power := 1.0;
  END; { IF }

  { Cover everything else }
  IF ((Base < 0) AND (Exponent < 0)) THEN
    Power := 1/Exp(-Exponent*Ln(-Base))
  ELSE IF ((Base < 0) AND (Exponent >= 0)) THEN
    Power := Exp(Exponent*Ln(-Base))
  ELSE IF ((Base > 0) AND (Exponent < 0)) THEN
    Power := 1/Exp(-Exponent*Ln(Base))
  ELSE IF ((Base > 0) AND (Exponent >= 0)) THEN
    Power := Exp(Exponent*Ln(Base));

  { Correct the sign }
  IF ((Base < 0) AND (Frac(Exponent/2.0) <> 0.0)) THEN
    Result := -Power
  ELSE
    Result := Power;
END; { FUNCTION Pow }

```

```

procedure TfrmMain.grayscale(imgA, imgB: Tpicture);

```

```

var
  i, j: integer;
  PC, PH: PByteArray;
begin
  imgB.bitmap.PixelFormat := pf8bit;
  imgB.bitmap.Palette :=
    CreatePalette(PaletKeabuan.lpal);
  imgB.bitmap.Width := imgA.Width;
  imgB.bitmap.Height := imgA.Height;

  for i:= 0 to imgA.Height-1 do
    begin
      PC := imgA.Bitmap.ScanLine[i];
      PH := imgB.Bitmap.ScanLine[i];

      for j:= 0 to imgA.Width-1 do
        //PH[j] := round(0.299*PC[3*j]+0.587*PC[3*j+1]+0.114*PC[3*j+2]);
        PH[j] := round((PC[3*j]+PC[3*j+1]+PC[3*j+2])/3);
      end;
    end;
end;

```

```

procedure TfrmMain.BitBtn2Click(Sender: TObject);
begin

```

```

close;
end;

procedure Tfrmmain.FormCreate(Sender: TObject);
var
  i: integer;
begin
  ImageDLLLoader.Default.FindDLLs(ProgramPath);

  paletKeabuan.lPal.palVersion := $300;
  paletKeabuan.lPal.palNumEntries := 256;
  for i := 0 to 255 do
  begin
    PaletKeabuan.entry[i].peRed := i;
    PaletKeabuan.entry[i].peGreen := i;
    PaletKeabuan.entry[i].peBlue := i;
  end;

  warna[2].R := 255;
  warna[2].G := 0;
  warna[2].B := 0;

  warna[1].R := 0;
  warna[1].G := 0;
  warna[1].B := 255;

  warna[0].R := 255;
  warna[0].G := 255;
  warna[0].B := 255;

end;

procedure Tfrmmain.BitBtn1Click(Sender: TObject);
var nfile : string;
    PC : PByteArray;
    i,j : integer;
begin
  OpenPictureDialog1.Filter:='All images ('+BitmapLoaders.GetLoadFilter;
  if openpicturedialog1.Execute then
  begin
    nfile := openpicturedialog1.FileName;
    with TLinearBitmap.Create do
      try
        Image1.Picture:=nil;
        LoadFromFile(nfile);
        AssignTo(Image1.Picture.Bitmap);
      finally
        Free;
      end;
    if (image1.Picture.Bitmap.PixelFormat = pf24bit) then
    begin
      grayscale(image1.picture,image2.Picture);
      wimg := image2.Picture.Width;
      himg := image2.Picture.Height;
      setlength(pix,himg,wimg);
      for i := 0 to himg-1 do
      begin
        PC := image2.Picture.Bitmap.ScanLine[i];
        for j := 0 to wimg-1 do
        begin
          pix[i,j] := PC[j];
        end;
      end;
    end;
  end;
end;

```

end;

umain.~pas

end else

messagedlg('Format citra bukan RGB, konversi citra terlebih dahulu.', mtError, [mbOk], 0);

end;

end;

procedure Tfrmmain.FCM_cluster;

var i,j,k,l : longint;

tot1, tot2, n2, n1 : real;

nd, nmax : real;

iter : integer;

begin

setlength(u, 3, ukuran);

setlength(ulama, 3, ukuran);

setlength(v, 3, 2);

for i := 0 to ukuran-1 do

begin

j := i mod 3;

randomize;

u[j, i] := random;

u[(i+1) mod 3, i] := (1-u[j, i])/3;

u[(i+2) mod 3, i] := (2-2*u[j, i])/3;

//mem1.Lines.Add(inttostr(j)+' '+format('%d', [(i+1) mod 3]))+' '+format('%d', [(i+2) mod 3]));

//mem1.Lines.Add(inttostr(j)+' '+format('%1.4f', [u[0, i]])+' '+format('%1.4f', [u[1, i]])+' '+format('%1.4f', [u[2, i]]));

end;

end;

iter := 0;

repeat

for i := 0 to 2 do

for j := 0 to ukuran-1 do

ulama[i, j] := u[i, j];

for i := 0 to 2 do

for j := 0 to 1 do

begin

tot1 := 0;

tot2 := 0;

for k := 0 to ukuran-1 do

begin

tot1 := tot1+power2(u[i, k], w)*x[k, j];

tot2 := tot2+power2(u[i, k], w);

end;

v[i, j] := tot1/tot2;

//mem1.Lines.Add(format('%1.4f', [v[i]]));

end;

for i := 0 to 2 do

begin

for k := 0 to ukuran-1 do

begin

tot1 := 0;

n1 := 0;

n2 := 0;

for j := 0 to 2 do

begin

for l := 0 to 1 do

begin

```

        n1 := power2((x[k,l]-v[i,l])2/(w-1));
        n2 := power2((x[k,l]-v[j,l])2/(w-1));
        tot1 := tot1 + n1/n2;
    end;
end;
u[i,k] := 1/tot1;
end;

end;

    edit1.Text := inttostr(iter);
    application.ProcessMessages;
nmax := 0;
for i := 0 to 2 do
begin
    tot1 := 0;
    for j := 0 to ukuran-1 do
        tot1 := tot1 +sqr(u[i,j]-ulama[i,j]);
    if nmax < tot1 then nmax := tot1;
    end;
inc(iter);
until (nmax < eps) or (iter = maxiter);

end;
procedure Tfrmmain.getdata;
var tota : real;
    i,j,k,l : integer;
    jdata : integer;
    idx : integer;
    rata,stdv : real;
begin
    ukr := strtoint(edit2.Text);
    ukuran := floor(wimg/ukr)*floor(himg/ukr);
    setlength(x,ukuran,2);
    idx := 0;
    for i := 0 to floor((himg)/ukr)-1 do
        for j := 0 to floor((wimg)/ukr)-1 do
            begin
                tota := 0;
                for k := 0 to ukr-1 do
                    for l := 0 to ukr-1 do
                        tota := tota + pix[ukr*i+k,ukr*j+l];
                    end;
                rata := tota/(ukr*ukr);
                tota := 0;
                for k := 0 to ukr-1 do
                    for l := 0 to ukr-1 do
                        tota := tota + sqr(pix[ukr*i+k,ukr*j+l]-rata);
                    end;
                stdv := sqrt(tota/(ukr*ukr));

                x[idx,0] := rata;
                x[idx,1] := stdv;
                //mem1.Lines.Add(inttostr(idx)+'|'+format('%3.3f',[x[idx,0]])+'
'+format('%3.3f',[x[idx,1]]));
                inc(idx);
            end;
        end;
    end;

end;

procedure Tfrmmain.smoothing;
var i,j : integer;
    tota : integer;

```

```

    psta : array of array of byte;      umain.~pas
    pH : PByteArray;
begin
    setlength(psta,himg2, wimg2);
    for i := 0 to himg2-1 do
        for j := 0 to wimg2-1 do
            begin
                tota := 0;
                if (i >= 1) and (i <= himg2-2) and (j >= 1) and (j <= wimg2-2) then
                    begin
                        tota :=
2*bpix[i-1,j-1]+2*bpix[i-1,j]+2*bpix[i-1,j+1]+2*bpix[i,j-1]+5*bpix[i,j]+2*bpix[i,j+1]
+2*bpix[i+1,j-1]+2*bpix[i+1,j]+2*bpix[i+1,j+1];
                        end;
                        if tota >= 10 then psta[i,j] := 1 else psta[i,j] := 0;
                    end;
            end;
        end;
    for i := 0 to himg2-1 do
        begin
            PH := image3.Picture.Bitmap.ScanLine[i];
            for j := 0 to wimg2-1 do
                begin
                    if psta[i,j] = 1 then
                        pH[j] := 0 else pH[j] := 255;
                    end;
                end;
            image3.Repaint;
        end;

    procedure Tfrmmain.getimage;
    var i,j : integer;
        atas, bawah, kiri, kanan : integer;
        pH,pC : PByteArray;
        status : boolean;
        wimg3, himg3 : integer;
    begin
        status := false;
        kiri := wimg2-1;
        for i := 0 to himg2-1 do
            begin
                //if status then break;
                PH := image3.Picture.Bitmap.ScanLine[i];
                for j := 0 to wimg2-1 do
                    begin
                        if pH[j] < 150 then
                            begin
                                if status = false then
                                    atas := i;
                                if j < kiri then kiri := j;
                                status := true;
                                break;
                            end;
                        end;
                    end;
                end;
            end;

        end;

        kanan := 0;
        status := false;
        for i := himg2-1 downto 0 do
            begin
                //if status then break;
                PH := image3.Picture.Bitmap.ScanLine[i];
                for j := wimg2-1 downto 0 do

```

umain.~pas

```
begin
  if pH[j] < 150 then
    begin
      if status = false then
        bawah := i;
      if j > kanan then kanan := j;
      status := true;
      break;
    end;
  end;

end;

wimg3 := ukr*(kanan-kiri)+3;
himg3 := ukr*(bawah-atas)+3;
image4.Picture.Bitmap.PixelFormat := pf24bit;
image4.Picture.bitmap.Width := wimg3;
image4.Picture.Bitmap.Height := himg3;

for i := 0 to himg3-1 do
begin
  pC := image1.Picture.Bitmap.ScanLine[ukr*atas+i];
  pH := image4.Picture.Bitmap.ScanLine[i];
  for j := 0 to wimg3-1 do
    begin
      pH[3*j] := pC[ukr*3*kiri+3*j];
      pH[3*j+1] := pC[ukr*3*kiri+3*j+1];
      pH[3*j+2] := pC[ukr*3*kiri+3*j+2];
    end;
  end;
  //memo1.Lines.Add(inttostr(atas)+' '+inttostr(kiri)+' '+inttostr(bawah)+'
'+inttostr(kanan));
end;

procedure Tfrmmain.BitBtn3Click(Sender: TObject);
begin
  getdata;
  fcm_cluster;
  simulasi;
  smooting;
  getimage;
end;

procedure Tfrmmain.Button1Click(Sender: TObject);
var namafile : string;
begin
  savepicturdialog1.DefaultExt := 'jpg';
  if savepicturdialog1.Execute then
    begin
      namafile := savepicturdialog1.FileName;
      image4.Picture.Bitmap.SaveToFile(namafile);
    end;
end;

procedure Tfrmmain.Button2Click(Sender: TObject);
var namafile : string;
begin
  savepicturdialog1.DefaultExt := 'jpg';
  if savepicturdialog1.Execute then
    begin
      namafile := savepicturdialog1.FileName;
      image3.Picture.Bitmap.SaveToFile(namafile);
    end;
end;
```



```

end;
end;
procedure Tfrmmain.BitBtn4Click(Sender: TObject);
var T : textfile;
    namafile : string;
    i,j : integer;
begin
savedialog1.Filter := '*.txt';
savedialog1.DefaultExt := 'txt';
if savedialog1.Execute then
begin
namafile := savedialog1.FileName;
assignfile(T,namafile);
rewrite(T);

write(T,'Data');
writeln(T);
for i := 0 to ukuran-1 do
begin
for j := 0 to 1 do
begin
write(T,format('%3.4f',[x[i,j]]));write(T,char(VK_TAB));
end;
write(T,format('%d',[wbaru[i]]));
writeln(T);
end;
write(T,'Pusat Kluster');
writeln(T);
for i := 0 to 2 do
begin
for j := 0 to 1 do
begin
write(T,format('%3.4f',[v[i,j]]));write(T,char(VK_TAB));
end;
writeln(T);
end;
write(T,'Derajat Keanggotaan');
writeln(T);
for i := 0 to ukuran-1 do
begin
for j := 0 to 2 do
begin
write(T,format('%3.4f',[u[j,i]]));write(T,char(VK_TAB));
end;
writeln(T);
end;
closefile(T);
end;
end;
end.

```

umain.~pas

