

**RANCANG BANGUN PENGENALAN PENYAKIT
DARAH MENGGUNAKAN *HIDDEN MARKOV
MODEL***

SKRIPSI

Oleh

AFITA PUTRI LESTARI
04 04 03 001 6



**PROGRAM STUDI TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

**RANCANG BANGUN PENGENALAN PENYAKIT
DARAH MENGGUNAKAN *HIDDEN MARKOV*
*MODEL***

SKRIPSI

Oleh

AFITA PUTRI LESTARI
04 04 03 001 6



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**PROGRAM STUDI TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP 2007/2008**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

RANCANG BANGUN PENGENALAN PENYAKIT DARAH MENGUNAKAN *HIDDEN MARKOV MODEL*

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang merupakan sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 11 Agustus 2008

Afita Putri Lestari

NPM 04 04 03 001 6

PENGESAHAN

Skripsi dengan judul:

RANCANG BANGUN PENGENALAN PENYAKIT DARAH MENGUNAKAN *HIDDEN MARKOV MODEL*

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Skripsi ini telah diujikan pada sidang ujian skripsi pada tanggal 7 Juli 2008 dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 11 Agustus 2008

Dosen Pembimbing

Dr. Ir. Dodi Sudiana, M.Eng

NIP 131 944 413

UCAPAN TERIMA KASIH

Puji syukur kehadiran Tuhan Yang Maha Esa atas berkat dan rahmat-Nya skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

Dr. Ir. Dodi Suidiana, M.Eng

Dr. Ir. Arman Djohan Diponegoro, M.Eng

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan saran, diskusi, pengarahan, dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.

Afita Putri Lestari
NPM 04 04 03 001 6
Departemen Teknik Elektro

Dosen Pembimbing
Dr. Ir. Dodi Sudiana, M.Eng

RANCANG BANGUN PENGENALAN PENYAKIT DARAH MENGUNAKAN *HIDDEN MARKOV MODEL*

ABSTRAK

Darah merupakan unsur dalam tubuh manusia yang memiliki peran penting dalam mekanisme kerja tubuh. Banyak informasi penting yang terkandung dalam darah, termasuk informasi penyakit yang diderita seseorang. Pentingnya informasi tersebut ditambah kebutuhan diagnosis dini untuk mempercepat penanganan suatu penyakit, maka citra darah sangat vital sebagai media dalam proses pengenalan penyakit. Dengan menggunakan citra darah, proses pengenalan penyakit menjadi lebih mudah dan cepat karena tidak diperlukan proses reaksi kimia dengan darah.

Dalam skripsi ini dilakukan perancangan proses pengenalan penyakit leukemia dari citra darah dengan menggunakan metode *Hidden Markov Model* (HMM). Prosesnya melibatkan dua tahap proses utama yaitu proses pembentukan *database* dan proses pengenalan. Pada tahap pembentukan *database*, citra darah diubah menjadi vektor sebagai titik *sample* dan titik-titik yang terdekat akan dikuantisasi menjadi *centroid* atau *codeword*. Kumpulan *codeword* akan disimpan berupa *codebook* di dalam *database*. Pengenalan dilakukan dengan membandingkan besaran *log of probability* HMM yang dihitung berdasarkan titik *sample* dari setiap *sample* citra darah.

Dengan menggunakan *codebook* berukuran 32, 64 dan 128 dengan jumlah repetisi 5 dan 10 kali, diperoleh tingkat akurasi pengenalan penyakit darah antara 60% sampai 82,76%.

Kata kunci: Penyakit Leukemia, Citra Darah, *Hidden Markov Model*, *Codebook*, *Database*

Afita Putri Lestari
NPM 04 04 03 001 6
Electrical Engineering Department

Supervisor
Dr. Ir. Dodi Sudiana, M.Eng

BLOOD DISEASE IDENTIFICATION USING HIDDEN MARKOV MODEL

ABSTRACT

Blood is a part of human body which plays an important role in the body mechanism. Important informations could be achieved from blood, including information of diseases. This kind of information is very essential in order to diagnose the disease as early as possible. Blood cells in digital format will be easier to analyze using computers and the process itself could be performed faster than conventional methods, since it needs no chemical reactions in the process.

In this research, the disease identification for leukemia is performed from blood imageries analyzed using Hidden Markov Model (HMM). The whole process consists of two main processes: database construction and recognition. In the first process, blood image will be transformed to vectors as sample points and the nearest points will be quantized as centroids or codewords. The collection of codewords is built in codebook database. Recognition process is performed by taking the largest value of HMM's log of probability from sample points of several blood images.

Based on the simulation results, using codebook 32, 64 and 128 with repetition 5 and 10 times, the accuration levels of the recognition results are between 60% and 82.76%.

Keywords: Leukemia Disease, Blood Image, Hidden Markov Model, Codebook, Database.

DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN SKRIPSI	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
DAFTAR SINGKATAN	xii
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN PENULISAN	2
1.3 BATASAN MASALAH	2
1.4 METODE PENULISAN	2
1.5 SISTEMATIKA PENULISAN	3
BAB II LEUKEMIA DAN PENGENALAN CITRA	4
2.1 DARAH	4
2.1.1 Biologi Darah	4
2.1.1.1 <i>Sel Darah Merah (Eritrosit)</i>	4
2.1.1.2 <i>Sel Darah Putih (Leukosit)</i>	4
2.1.1.3 <i>Keping Darah (Trombosit)</i>	5
2.1.2 Penyakit Kelainan Darah	5
2.1.2.1 <i>Penyakit Sel Darah Merah</i>	5
2.1.2.2 <i>Penyakit Sel Darah Putih</i>	7
2.1.2.3 <i>Penyakit Keping Darah</i>	7
2.1.2.4 <i>Penyakit Sistem Pembekuan Darah</i>	7
2.2 LEUKEMIA	8
2.2.1 Leukemia Limfositik Akut	8
2.2.2 Leukemia Mielositik Akut	9
2.2.3 Leukemia Limfositik Kronik	10
2.2.4 Leukemia Mielositik Kronik	10
2.3 PENGENALAN CITRA	11
2.4 PROSES PENGENALAN CITRA DARAH	12

2.4.1	<i>Sampling</i> dan Kuantisasi	13
2.4.2	Pengolahan Citra Digital	15
	2.4.2.1 <i>Grayscale</i>	15
	2.4.2.2 <i>Leveling, Cropping dan Reshaping</i>	15
2.4.3	Ekstraksi	16
	2.4.3.1 <i>Mel Frequency Cepstrum Coefficient (MFCC)</i>	16
	2.4.3.2 <i>Fast Fourier Transform (FFT)</i>	18
2.4.4	Kuantisasi Vektor	20
2.5	HIDDEN MARKOV MODEL	22
2.5.1	Model Markov	22
2.5.2	Elemen HMM	23
BAB III RANCANG BANGUN PENGENALAN PENYAKIT DARAH DENGAN HMM		28
3.1	PEMBENTUKAN <i>DATABASE</i>	28
3.1.1	Proses Pelabelan	30
3.1.2	Proses Pembuatan <i>Codebook</i>	32
3.1.3	Proses Pembentukan HMM	34
	3.1.3.1 <i>State dari Markov Chain</i>	37
3.2	PROSES PENGENALAN	37
BAB IV UJI COBA PERANGKAT LUNAK PENGENALAN PENYAKIT DARAH DENGAN HMM		41
4.1	HASIL UJI COBA PENGENALAN	42
4.2	ANALISIS HASIL UJI COBA PENGENALAN	44
4.2.1	Analisis Citra Darah Sebagai Data <i>input</i>	45
4.2.2	Analisis Pengaruh Perubahan <i>Codebook</i>	47
4.2.3	Analisis Pengaruh Perubahan Repetisi	49
BAB V KESIMPULAN		51
DAFTAR ACUAN		52
DAFTAR PUSTAKA		54
LAMPIRAN		55

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Berbagai jenis sel-sel penyusun darah 5
Gambar 2.2	Perbandingan citra darah normal dengan darah anemia 6
Gambar 2.3	Citra darah Leukemia Limfositik Akut 9
Gambar 2.4	Citra darah Leukemia Mielositik Akut 9
Gambar 2.5	Citra darah Leukemia Limfositik Kronik 10
Gambar 2.6	Citra darah Leukemia Mielositik Kronik 11
Gambar 2.7	Komponen sistem pengenalan pola 12
Gambar 2.8	Blok diagram proses pengenalan penyakit dari citra darah dengan menggunakan metode <i>Hidden Markov Model</i> 13
Gambar 2.9	Proses digitisasi citra 14
Gambar 2.10	Blok diagram MFCC 16
Gambar 2.11	Mel scale 17
Gambar 2.12	Contoh <i>codeword</i> pada ruang dua dimensi 21
Gambar 2.13	Model <i>Markov</i> 23
Gambar 3.1	Diagram alir pembentukan <i>database</i> 29
Gambar 3.2	Proses Pelabelan 31
Gambar 3.3	Tampilan proses pelabelan 32
Gambar 3.4	Tampilan proses pemuatan <i>codebook</i> 33
Gambar 3.5	Plot titik-titik vektor pada proses kuantisasi vektor 34
Gambar 3.6	Tampilan proses pembentukan HMM 35
Gambar 3.7	Tampilan saat perhitungan LoP tiap label 36
Gambar 3.8	Grafik nilai <i>log of probability</i> 36
Gambar 3.9	Diagram alir proses pengenalan 38
Gambar 3.10	Tampilan proses pengenalan 40

DAFTAR TABEL

	Halaman
Tabel 2.1 Jenis-jenis leukemia	8
Tabel 4.1 Nama file dan label tiap jenis leukemia	41
Tabel 4.2 Hasil uji coba untuk penyakit ALL L2	42
Tabel 4.3 Hasil uji coba untuk penyakit AML M1	42
Tabel 4.4 Hasil uji coba untuk penyakit AML M2	43
Tabel 4.5 Hasil uji coba untuk penyakit AML M4	43
Tabel 4.6 Hasil uji coba untuk penyakit AML M5	44
Tabel 4.7 Tabel persentase akurasi uji coba tiap-tiap label	45
Tabel 4.8 Tabel persentase akurasi uji coba keseluruhan	45

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Citra Darah Leukemia Yang Digunakan	55



DAFTAR SINGKATAN

AIDS	Acquired Immuno Deficiency Syndrome
ALL	Acute Lymphoblastic Leukemia
AML	Acute Myelocytic Leukemia
CLL	Chronic Lymphoblastic Leukemia
CML	Chronic Myelocytic Leukemia
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
GLA	General Lloyd Algorithm
HIV	Human Immunodeficiency Virus
HMM	Hidden Markov Model
ITP	Idiopathic Thrombocytopenia Purpura
LoP	Log of Probability
LPC	Linear Prediction Coding
MFCC	Mel Frequency Cepstrum Coefficient
VQ	Vector Quantization

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Darah merupakan unsur dalam tubuh manusia yang memiliki peran dalam mekanisme kerja tubuh. Seluruh organ tubuh dihubungkan oleh darah melalui pembuluh-pembuluh darah. Oleh karena itu, darah dapat menjadi cerminan keadaan tubuh, baik dalam keadaan sehat maupun sakit. Walaupun telah banyak upaya lain yang dilakukan dalam mendeteksi penyakit, misalnya dengan air ludah atau iris mata, darah masih menjadi sumber diagnosa medis yang paling dapat diandalkan. Hal ini dikarenakan banyaknya informasi penting yang dikandung oleh darah. Sebagai perbandingan, kadar zat tertentu yang terkandung di ludah hanya seperseratus sampai seperseribu dari apa yang dapat ditemukan di dalam darah[1].

Pemantauan atau diagnosis penyakit bergantung pada hasil berbagai pemeriksaan laboratorium dari contoh darah. Pemeriksaan laboratorium sebagai pemeriksaan penunjang ditambah dengan pemeriksaan klinis yang baik dan lengkap dapat membantu bila gejala klinis yang muncul kurang memadai untuk mendiagnosa penyakit. Keterlambatan diagnosis akan berakibat keterlambatan penanganan yang berpotensi meningkatkan resiko kematian. Oleh karena itu, tuntutan akan teknologi yang dapat membantu dalam pelaksanaan diagnosis medis secara cepat, mudah dan teliti semakin besar.

Berdasarkan kemampuan darah dalam merepresentasikan suatu penyakit ditambah kebutuhan akan diagnosis dini tersebut, maka dapat dilakukan proses pengenalan penyakit dengan menggunakan darah, dalam hal ini dengan bantuan citra darah. Pengenalan penyakit dengan citra darah berbeda dengan pengenalan penyakit konvensional yang perlu melalui tahapan pemeriksaan laboratorium. Dengan kata lain, pada pengenalan penyakit dengan citra darah ini tidak diperlukan adanya reaksi dengan bahan-bahan kimia terhadap contoh darah yang diuji. Dengan demikian proses pengenalan penyakit relatif lebih mudah dan

singkat dibandingkan dengan cara konvensional karena proses yang diperlukan hanya proses pengolahan serta pengenalan citra darah.

Proses pengenalan citra dapat dilakukan dengan berbagai metode. Metode yang umumnya digunakan antara lain *Fuzzy Logic*, Jaringan Syaraf Tiruan (*Neural Network*) dan Kecerdasan Tiruan (*Artificial Intelligent*). Dalam skripsi ini, metode pengenalan penyakit dari citra darah yang digunakan adalah *Hidden Markov Model* (HMM). HMM merupakan metode pengenalan yang berbasis pada konsep Kecerdasan Tiruan.

1.2 TUJUAN PENULISAN

Tujuan dari penulisan skripsi ini adalah untuk menghasilkan rancangan perangkat lunak yang mampu mengenali penyakit darah dari citra darah dengan menggunakan HMM.

1.3 BATASAN MASALAH

Proses pengenalan penyakit melalui citra darah ini dibatasi hanya pada citra mikroskopis darah dengan penyakit darah atau kelainan darah. Penyakit darah yang diambil dalam skripsi ini dibatasi hanya penyakit leukemia atau kanker darah. Adapun jenis penyakit leukemia yang dipakai hanya 2 jenis namun terdiri atas 5 tipe, yaitu Leukemia Limfositik Akut (ALL) tipe L2 serta Leukemia Mielositik Akut (AML) tipe M1, tipe M2, tipe M4, dan tipe M5.

1.4 METODE PENULISAN

Metode-metode yang digunakan dalam penulisan skripsi ini antara lain:

(1) Studi kepustakaan

Mempelajari semua informasi mengenai darah dan kelainannya termasuk leukemia serta mengenai pengolahan citra dan HMM dari buku, jurnal, artikel dan literatur lain.

(2) Pencarian data

Mencari data-data yang diperlukan, dalam hal ini berupa citra darah mikroskopis dalam bentuk digital dari berbagai sumber, mulai dari institusi-institusi tertentu sampai data bebas yang dapat diperoleh melalui internet.

(3) Perancangan algoritma

Membuat algoritma proses pengenalan dengan menggunakan metode HMM. Keseluruhan algoritma terdiri atas dua algoritma proses yaitu algoritma proses pembentukan *database* dan algoritma proses pengenalan.

(4) Uji coba

Membuat program serta melakukan uji coba program dengan menggunakan perangkat lunak komputasi numerik berdasarkan algoritma yang telah dibuat.

(5) Analisis dan kesimpulan

Menganalisis serta membuat kesimpulan hasil uji coba.

1.5 SISTEMATIKA PENULISAN

Laporan skripsi ini disusun secara sistematis. Secara garis besar, sistematika penulisan laporan ini terdiri atas lima bab.

Bab 1 berisi Pendahuluan yang menjelaskan secara singkat mengenai latar belakang, tujuan penulisan, batasan masalah, metode penulisan dan sistematika penulisan untuk memberi gambaran tentang isi skripsi.

Pada bab 2 yang berjudul Leukemia dan Pengenalan Citra akan dijelaskan teori mengenai darah beserta berbagai macam penyakit darah termasuk leukemia, pengenalan citra serta mengenai *Hidden Markov Model*.

Bab 3 yaitu Rancang Bangun Pengenalan Penyakit Darah dengan HMM menjelaskan mengenai algoritma proses pengenalan penyakit melalui citra darah serta simulasi program yang dilakukan dengan menggunakan perangkat lunak komputasi numerik.

Sedangkan Uji Coba Perangkat Lunak Pengenalan Penyakit Darah dengan HMM akan dijelaskan pada bab 4 yang berisi hasil uji coba perangkat lunak dalam mengenali penyakit leukemia serta analisisnya.

Bab terakhir adalah Penutup yang berisi kesimpulan dari proses rancang bangun yang telah dibuat dan diuji coba.

BAB II

LEUKEMIA DAN PENGENALAN CITRA

2.1 DARAH

2.1.1 Biologi Darah[2]

Darah merupakan gabungan dari cairan, sel-sel dan partikel yang menyerupai sel yang mengalir melalui pembuluh-pembuluh darah. Terdapat dua jenis pembuluh darah, yaitu arteri dan vena. Arteri adalah pembuluh yang membawa darah yang mengandung oksigen dari jantung dan paru-paru menuju ke seluruh tubuh. Sedangkan vena adalah pembuluh yang membawa darah mengalir kembali ke jantung dan paru-paru.

Darah terdiri atas dua jenis komponen utama, yaitu cairan dan sel. Komponen cairan sebagian besar mengandung garam-garam terlarut dan protein seperti albumin, antibodi (*imunoglobulin*) dan protein pembekuan. Sedangkan komponen sel terdiri atas tiga jenis sel, yaitu sel darah merah, sel darah putih, dan keping darah. Berbagai jenis bentuk komponen tersebut dapat dilihat pada Gambar 2.1.

2.1.1.1 Sel Darah Merah (Eritrosit)

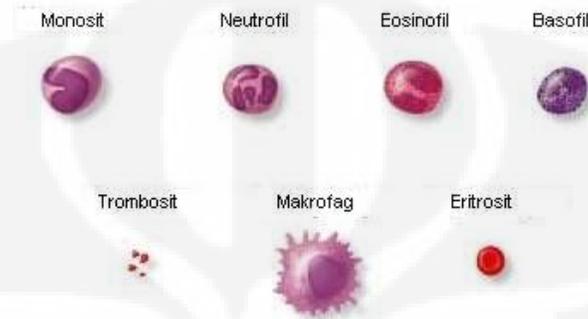
Sel darah merah merupakan sel yang paling banyak dengan jumlahnya mencapai hampir separuh dari volume darah pada keadaan normal. Sel darah merah mengandung *hemoglobin*, yang akan mengikat oksigen untuk dibawa dari paru-paru ke seluruh jaringan tubuh.

2.1.1.2 Sel Darah Putih (Leukosit)

Jumlah sel darah putih lebih sedikit dibandingkan sel darah merah dengan perbandingan sekitar 1:660. Terdapat beberapa jenis sel darah putih yang bekerja sama membangun mekanisme pertahanan tubuh, antara lain *Neutrofil*, *Limfosit*, *Monosit*, *Eosinofil*, *Basofil*.

2.1.1.3 Keping Darah (Trombosit)

Keping darah merupakan partikel yang menyerupai sel dengan ukuran lebih kecil daripada sel darah merah atau sel darah putih. Keping darah merupakan bagian dari mekanisme pembekuan darah.



Gambar 2.1 Berbagai jenis sel-sel penyusun darah[2]

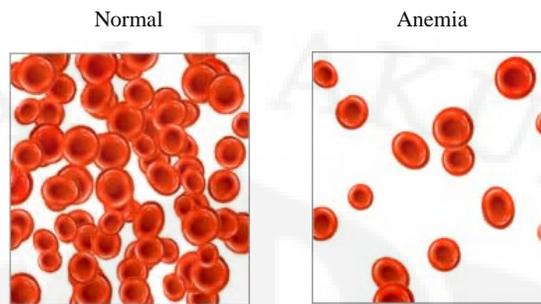
2.1.2 Penyakit Kelainan Darah

Penyakit kelainan darah seringkali kurang diperhatikan pada awalnya karena gejala-gejala awal yang timbul dianggap tidak terlalu membahayakan. Misalnya seperti mimisan yang merupakan salah satu gejala yang pemicunya mungkin diakibatkan karena adanya kelainan darah yang dimiliki penderita[3]. Padahal dampak yang ditimbulkan akibat kelainan darah bisa menjadi sangat fatal. Contohnya saja pada wanita hamil dapat menyebabkan keguguran berulang[4].

Beberapa jenis penyakit kelainan darah dapat diklasifikasikan berdasarkan jenis sel darahnya[5].

2.1.2.1 Penyakit Sel Darah Merah

Kondisi yang paling umum yang mempengaruhi sel darah merah adalah anemia, yaitu penurunan jumlah sel darah merah dari jumlah normalnya pada darah. Perbandingan jumlah sel darah merah normal dengan sel darah merah pada anemia dapat dilihat pada Gambar 2.2. Anemia diikuti dengan penurunan jumlah hemoglobin pada darah. Gejala-gejala anemia seperti kulit pucat, lemas, dan detak jantung cepat, terjadi karena adanya penurunan kapasitas darah untuk membawa oksigen.



Gambar 2.2 Perbandingan citra darah normal dengan darah anemia [6]

Dalam tes laboratorium, fungsi sel darah merah dapat ditunjukkan melalui tiga macam tes, yaitu perhitungan sel darah merah atau *Red Blood Cell Count*, perhitungan Hemoglobin dan perhitungan Hematokrit. Harga normal dari Hemoglobin dalam sel darah merah adalah 14-16 gram/dL untuk pria dan 12,5-15 gram/dL untuk wanita. Hematokrit perbandingan sel darah merah terhadap volume darah total adalah 42-50% untuk pria dan 38-47% untuk wanita. Sedangkan volume korpuskuler rata-rata perkiraan volume sel darah merah adalah 86-98 mikrometer³. Adanya anemia dapat dipastikan jika hasil tes laboratorium dari ketiga tes fungsi sel darah merah memiliki nilai yang kurang dari harga-harga normal di atas.

Penyebab-penyebab anemia dapat dibagi menjadi dua kategori, yaitu anemia yang disebabkan produksi sel darah merah yang tidak memenuhi dan anemia yang disebabkan oleh adanya penghancuran sel darah merah yang terlalu dini.

(1) Anemia disebabkan produksi sel darah merah yang tidak mencukupi

Beberapa kondisi dapat menyebabkan pengurangan produksi sel darah merah, di antaranya kekurangan zat besi, karena penyakit kronis atau karena penyakit ginjal.

(2) Anemia disebabkan perusakan sel darah merah yang luar biasa cepat

Saat sel darah merah dihancurkan lebih cepat dari seharusnya akibat penyakit (proses ini disebut hemolisis), sumsum tulang akan menyeimbangkannya dengan menambah produksi dari sel-sel darah merah yang baru untuk menggantikannya. Tetapi jika sel darah merah dihancurkan jauh lebih cepat daripada pengantiannya, orang tersebut akan mengidap anemia.

Beberapa penyebab dari peningkatan penghancuran sel darah merah yang dapat menyerang remaja adalah karena kekurangan G6PD, Sferositosis turunan, anemia hemolitik akibat sistem kekebalan tubuh, dan anemia *sickle-cell*.

2.1.2.2 Penyakit Sel Darah Putih

Beberapa jenis penyakit yang disebabkan adanya kelainan pada sel darah putih antara lain:

- (1) Neutropenia terjadi saat tidak terdapat sel darah putih jenis tertentu yang cukup untuk melindungi tubuh melawan infeksi bakteri.
- (2) Human Immunodeficiency Virus (HIV) menyerang sel-sel darah putih jenis tertentu (limfosit) yang bekerja melawan infeksi. Infeksi karena virus dapat menyebabkan AIDS (Acquired Immunodeficiency Syndrome), menyebabkan tubuh mudah terserang infeksi dan beberapa penyakit lain.
- (3) Leukemia yang merupakan kanker darah.

2.1.2.3 Penyakit Keping Darah

Salah satu jenis penyakit yang disebabkan oleh adanya kelainan pada keping darah adalah *Thrombocytopenia* atau jumlah keping darah yang lebih rendah dari seharusnya. Biasanya didiagnosa karena seseorang mengalami pendarahan atau pembengkakan yang tidak normal. *Idiopathic Thrombocytopenia Purpura* (ITP) adalah kondisi di mana sistem kekebalan tubuh menyerang dan menghancurkan keping darah.

2.1.2.4 Penyakit Sistem Pembekuan Darah

Sistem pembekuan darah bergantung pada keping darah dan juga seberapa banyak jumlah faktor pembeku dan komponen darah lain. Jika cacat turunan mempengaruhi salah satu komponen ini, seseorang dapat mengalami kelainan pendarahan. Kelainan pendarahan yang umum terdiri dari:

(1) Hemofilia

Kondisi turunan yang khususnya hampir hanya dialami oleh pria, menyebabkan kekurangan faktor-faktor pembeku tertentu dalam darah.

(2) Penyakit Von Willeberd

Kelainan pendarahan turunan yang paling umum, juga melibatkan penurunan faktor pembeku. Ini terjadi baik pada pria maupun wanita.

2.2 LEUKEMIA

Leukemia adalah kanker yang terjadi pada sel darah manusia. Leukemia dapat terjadi di salah satu jenis sel darah dan yang terbanyak adalah sel darah putih. Sel darah putih diproduksi pada sel stem di sumsum tulang. Leukemia terjadi jika proses pematangan dari stem sel menjadi sel darah putih mengalami gangguan dan menghasilkan perubahan ke arah keganasan.

Sel-sel darah yang terkena leukemia tidak dapat berfungsi seperti layaknya sel darah normal. Pada akhirnya sel-sel tersebut akan menguasai sumsum tulang dan menggantikan tempat dari sel-sel yang menghasilkan sel-sel darah yang normal sehingga sel darah yang abnormal tersebut akan mendesak pertumbuhan sel darah putih yang normal maupun sel darah merah dan keping darah.

Terdapat 4 jenis utama leukemia yang diberi nama berdasarkan kecepatan perkembangan penyakit dan jenis sel darah putih yang terkena seperti yang dapat dilihat pada Tabel 2.1.

Tabel II.1 Jenis-jenis leukemia

Jenis	Perkembangan Penyakit	Sel Darah Putih Yang Terkena
Leukemia Limfositik (Limfoblastik) Akut	Cepat	Limfosit
Leukemia Mieloid (Mielositik, Mielogenous, Mieloblastik, Mielomonositik) Akut	Cepat	Mielosit
Leukemia Limfositik Kronik (termasuk <i>sindroma Sézary</i> dan leukemia sel berambut)	Lambat	Limfosit
Leukemia Mielositik (Mieloid, Mielogenous, granulositik) Kronik	Lambat	Mielosit

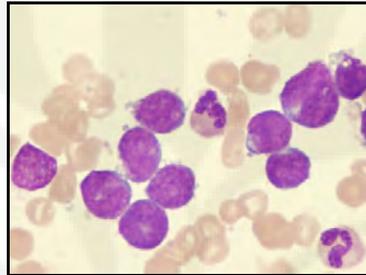
Sumber: Medicastore. (http://www.medicastore.com/med/detail_pyk.php?idktg=12&judul=Leukemia&iddtl=45&UID=2007123019215061.5.16.31)

2.2.1 Leukemia Limfositik Akut

Leukemia Limfositik (Limfoblastik) Akut atau *Acute Lymphoblastic Leukemia (ALL)* adalah leukemia di mana sel-sel yang dalam keadaan normal berkembang

menjadi limfosit berubah menjadi ganas dan dengan segera menggantikan sel-sel normal di dalam sumsum tulang.

ALL dapat diketahui dari pemeriksaan darah rutin seperti penghitungan darah komplit. Jumlah total sel darah putih dapat berkurang, normal ataupun bertambah, tetapi jumlah sel darah merah dan trombosit hampir selalu berkurang. Sel darah putih yang belum matang (sel *blast*) akan terlihat dalam contoh darah yang diperiksa dengan mikroskop. Contoh citra darah dari ALL dapat dilihat pada Gambar 2.3.

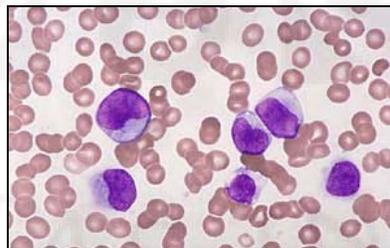


Gambar 2.3 Citra darah Leukemia Limfositik Akut[8]

2.2.2 Leukemia Mielositik Akut

Leukemia Mielositik (Mielositik, Mielogenous, Mieloblastik, Mielomonositik) Akut atau *Acute Myeloblastic Leukemia (AML)* adalah leukemia dimana mielosit (yang dalam keadaan normal berkembang menjadi granulosit) berubah menjadi ganas dan dengan segera menggantikan sel-sel normal di sumsum tulang.

Hasil perhitungan jenis darah merupakan langkah awal diagnosa untuk mengetahui bahwa seseorang menderita leukemia. Sel darah putih muda akan terlihat dalam darah yang diperiksa dengan mikroskop. Contoh citra darah AML dapat dilihat pada Gambar 2.4.

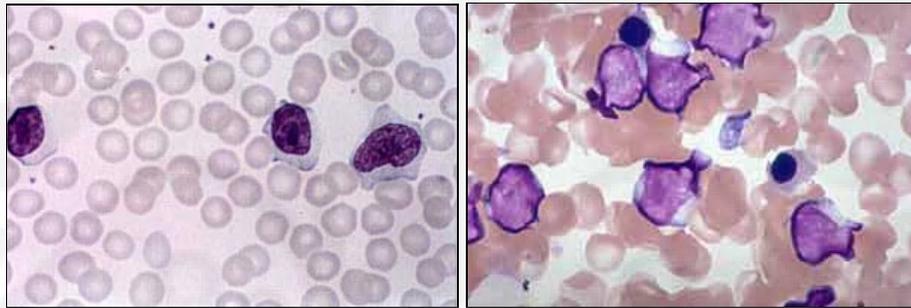


Gambar 2.4 Citra darah Leukemia Mielositik Akut[7]

2.2.3 Leukemia Limfositik Kronik

Leukemia Limfositik Kronik atau *Chronic Lymphoblastic Leukemia (CLL)* adalah leukemia yang ditandai dengan adanya sejumlah besar limfosit matang yang bersifat ganas dan terdapat pembesaran kelenjar getah bening.

CLL terkadang ditemukan secara tidak sengaja pada pemeriksaan hitung jenis darah. Jumlah limfosit meningkat sampai lebih dari 5.000 sel/mikroliter. Biasanya dilakukan biopsi sumsum tulang dan hasilnya akan menunjukkan adanya sejumlah besar limfosit di dalam sumsum tulang. Selain itu pemeriksaan darah juga dapat menunjukkan adanya anemia, berkurangnya jumlah trombosit dan berkurangnya kadar antibodi. Contoh citra darah CLL dapat dilihat pada Gambar 2.5.

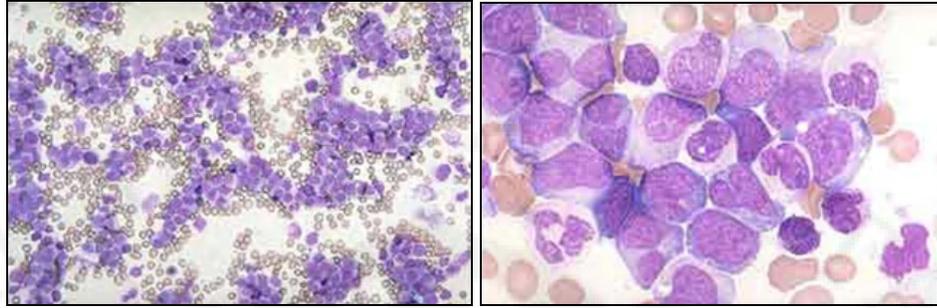


Gambar 2.5 Citra darah Leukemia Limfositik Kronik[7]

2.2.4 Leukemia Mielositik Kronik

Leukemia Mielositik (Mieloid, Mielogenous, granulositik) Kronik atau *Chronic Myeloid Leukemia (CML)* adalah Leukemia di mana sel darah putih di dalam sumsum tulang berubah menjadi ganas dan menghasilkan sejumlah besar granulosit yang abnormal.

CML sering terdiagnosis pada pemeriksaan darah rutin. Jumlah sel darah putih sangat tinggi, dapat mencapai 5.000-1.000.000 sel/mikroliter darah (normal kurang dari 11.000). Pada pemeriksaan darah, akan terlihat sel darah putih muda yang dalam keadaan normal hanya ditemukan di dalam sumsum tulang. Jumlah sel darah putih lainnya (eosinofil dan basofil) juga meningkat dan ditemukan bentuk sel darah merah yang belum matang. Contoh citra darah CML dapat dilihat pada Gambar 2.6.



Gambar 2.6 Citra darah Leukemia Mielositik Kronik[7]

2.3 PENGENALAN CITRA

Citra digital adalah gambar dua dimensi yang dihasilkan dari perubahan gambar analog dua dimensi yang kontinu menjadi gambar diskrit[9]. Pengolahan citra dapat didefinisikan sebagai proses memperbaiki kualitas citra dengan menggunakan berbagai teknik pengolahan citra yang akan mentransformasikan citra menjadi citra lain agar mudah diinterpretasikan oleh manusia[9]. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Pengolahan citra merupakan suatu proses awal atau *pre-processing*.

Sedangkan pengenalan pola (*pattern recognition*) adalah pengelompokan data numerik dan simbolik (termasuk citra) secara otomatis oleh komputer agar suatu objek dalam citra dapat dikenali dan diinterpretasi[9]. Pengenalan pola adalah tahapan selanjutnya atau analisis dari pengolahan citra.

Suatu sistem pengenalan pola terdiri dari komponen-komponen sebagai berikut[10]:

(1) Sensor

Berfungsi untuk menangkap objek dari dunia nyata menjadi sinyal-sinyal listrik dan selanjutnya menjadi bilangan-bilangan setelah proses digitisasi.

(2) Mekanisme *pre-processing*

Berfungsi untuk menonjolkan sinyal informasi dan menekan derau dalam sinyal. Secara keseluruhan merupakan proses pengolahan citra.

(3) Mekanisme Penyari fitur

Mengambil besaran komponen tertentu dari sinyal yang mewakili sifat utama sinyal, sekaligus mengurangi dimensi sinyal menjadi sekumpulan bilangan yang lebih sedikit tetapi representatif.

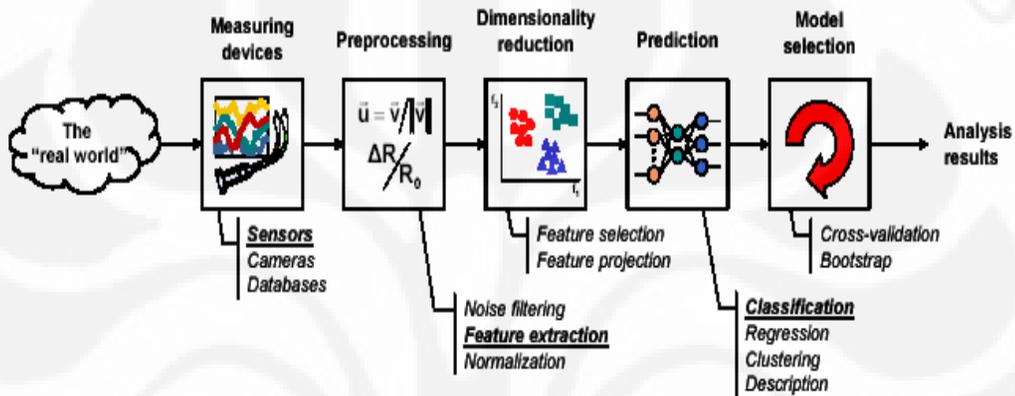
(4) Algoritma pemilah

Melakukan *assignment* fitur ke kelas yang sesuai.

(5) Sekumpulan contoh pelatihan yang telah dipilah

Dipakai untuk mendapatkan representasi kelas.

Diagram blok sistem pengenalan pola dapat dilihat pada Gambar 2.7.

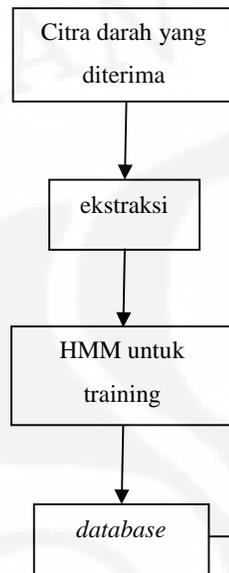


Gambar 2.7 Komponen sistem pengenalan pola[10]

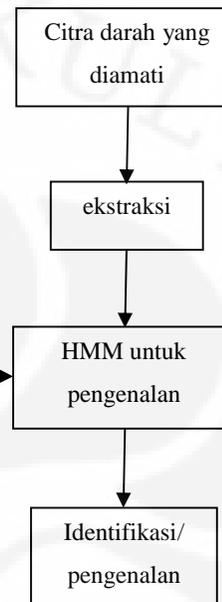
2.4 PROSES PENGENALAN CITRA DARAH

Proses pengenalan ini merupakan contoh aplikasi dalam bidang kedokteran di mana citra mikroskopis darah digunakan untuk mengenali jenis penyakit darah. Diagram blok proses pengenalan dapat dilihat pada Gambar 2.8. Pengenalan citra ini diawali dengan melakukan proses pengolahan citra yang meliputi proses ekstraksi sehingga diperoleh kumpulan data vektor yang akan menjadi data acuan dalam membuat *codebook*.

Proses Pembentukan Database



Proses Pengenalan



Gambar 2.8 Blok diagram proses pengenalan penyakit dari citra darah dengan menggunakan metode *Hidden Markov Model*

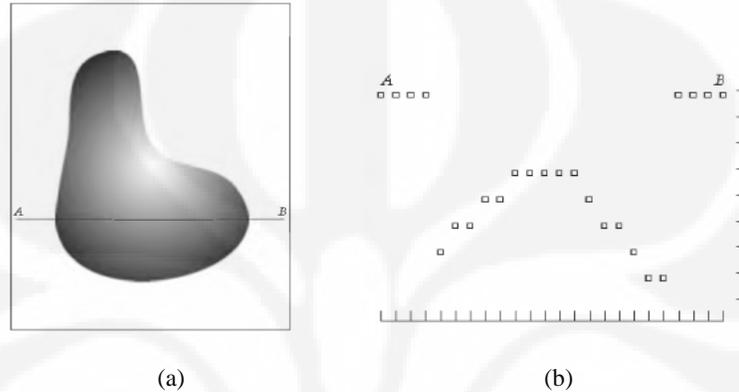
2.4.1 *Sampling* dan Kuantisasi

Citra darah yang digunakan pada pengenalan penyakit ini berasal dari citra darah analog yang telah didigitisasi. Gambar analog yang kontinu diubah menjadi gambar diskrit melalui proses *sampling*. *Sampling* adalah proses untuk menentukan warna pada piksel tertentu pada citra dari sebuah gambar yang kontinu. Dengan proses *sampling* akan diperoleh hasil pembulatan warna rata-rata dari gambar analog. Proses *sampling* sering juga disebut proses digitisasi.

Dalam proses *sampling*, warna rata-rata yang diperoleh dapat juga dihubungkan ke level warna tertentu. Kuantisasi merupakan proses mengasosiasikan warna rata-rata dengan tingkatan warna tertentu. Tujuan kuantisasi adalah untuk memberi nilai digital pada setiap *sampling* sehingga terjadi konversi dari data analog/kontinu menjadi data digital.

Proses *sampling* dan kuantisasi dapat diilustrasikan melalui Gambar 2.9. Gambar 2.9(a) menunjukkan suatu citra yang kontinyu, $f(x, y)$ yang akan diubah ke dalam bentuk diskrit atau digital. Untuk mengubahnya menjadi bentuk digital,

fungsi kontinyu tersebut harus diubah baik koordinatnya maupun *amplitude*-nya. Proses digitisasi koordinat dapat disebut sebagai proses *sampling* sedangkan proses digitisasi *amplitude* disebut sebagai proses kuantisasi. Citra hasil *sampling* dan kuantisasi sepanjang segmen garis AB dari Gambar 2.9(a) dapat dilihat pada Gambar 2.9(b).



Gambar 2.9 Proses digitisasi citra (a) citra kontinyu (b) hasil *sampling* dan kuantisasi [11]

Citra yang dihasilkan selanjutnya disebut sebagai citra digital. Secara umum citra digital merupakan hasil *sampling* dan kuantisasi yang disusun dalam bentuk matriks $M \times N$.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \dots \dots \dots (2.1)$$

Nilai-nilai elemen penyusun matriks disebut dengan piksel sedangkan posisi elemen dalam baris dan kolom menyatakan koordinat titik-titik (x,y) dalam citra. Tiap piksel mempunyai nilai digital yang dapat merepresentasikan citra sebenarnya.

2.4.2 Pengolahan Citra Digital

Proses pengolahan citra darah bertujuan untuk mengubah masukan citra darah digital menjadi bentuk yang dapat diolah menjadi database yang diperlukan dalam sistem pengenalan penyakit ini.

2.4.2.1 Grayscale

Citra digital sesungguhnya merupakan suatu fungsi intensitas dalam bidang dua dimensi. Intensitas yang dimaksud berasal dari sumber cahaya dan cahaya merupakan suatu bentuk energi. Pada dasarnya citra yang dilihat merupakan hasil pantulan berkas-berkas cahaya terhadap benda-benda. Jadi fungsi intensitas cahaya merupakan fungsi sumber cahaya yang menerangi obyek serta jumlah cahaya yang dipantulkan obyek, sehingga dapat ditulis sebagai berikut:

$$f(x,y)=i(x,y).r(x,y) \dots\dots\dots (2.2)$$

di mana: $i(x,y)$ adalah iluminasi sumber cahaya ($0 < i(x,y) < \infty$)

$r(x,y)$ adalah koefisien pantul obyek ($0 < r(x,y) < 1$)

Fungsi intensitas f pada suatu titik (x,y) disebut derajat keabuan atau *graylevel* (l) dengan l terletak antara L_{min} dan L_{max} ($L_{min} \leq l \leq L_{max}$).

Selang antara L_{min} dengan L_{max} sering disebut sebagai skala keabuan dengan L_{min} menyatakan hitam dan L_{max} menyatakan putih. Semua bilangan yang terletak antara L_{min} dan L_{max} merupakan derajat keabuan.

Citra *grayscale* merupakan citra yang hanya terdiri dari derajat keabuan antara warna hitam dan putih. Citra jenis ini berbeda dengan citra warna dalam bentuk RGB (*Red Green Blue*). Citra RGB dibentuk dari tiga lapisan warna, yaitu lapisan warna merah, hijau dan biru.

2.4.2.2 Leveling, Cropping dan Reshaping

Leveling merupakan pemberian level pada nilai *graylevel* suatu citra dari suatu array tertentu ke suatu *range* nilai diskrit tertentu.

Cropping merupakan proses pemotongan citra pada elemen-elemen tertentu dari citra. Proses ini bertujuan untuk mengambil elemen-elemen yang diinginkan dari citra digital. Dalam sistem pengenalan penyakit ini, *cropping* dilakukan

hanya pada bagian sel-sel darah putih dari citra darah. Hasil pemotongan tiap-tiap sel tersebut akan menjadi *sample-sample* citra yang selanjutnya akan dilatih dan diuji pada sistem pengenalan penyakit.

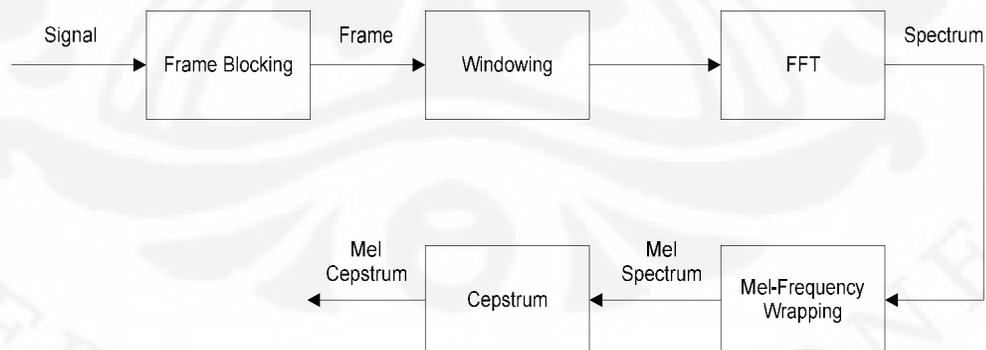
Reshapping merupakan proses perubahan ukuran matriks suatu citra menjadi ukuran matriks tertentu namun dengan jumlah *array* yang sama. Misalkan citra dengan matriks $M \times N$ diubah menjadi citra dengan ukuran $N \times M$.

2.4.3 Ekstraksi

Ekstraksi atau *feature extraction* merupakan proses di mana *sample-sample* citra darah yang telah melalui proses pengolahan citra akan diubah menjadi vektor-vektor data. Terdapat beberapa metode yang dapat digunakan untuk proses ini, antara lain *Linear Prediction Coding* (LPC) dan *Mel Frequency Cepstrum Coefficient* (MFCC). Metode yang akan digunakan pada proses ekstraksi dalam pengenalan citra darah ini adalah dengan menggunakan MFCC.

2.4.3.1 Mel Frequency Cepstrum Coefficient (MFCC)

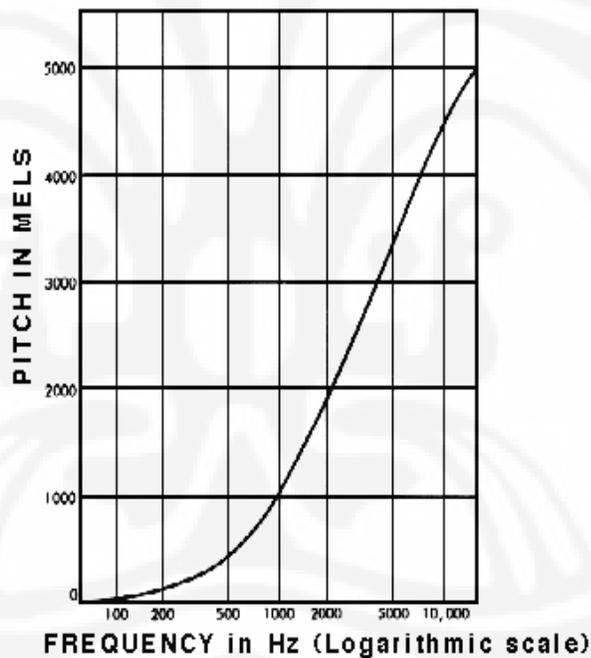
MFCC di sini bertujuan untuk menghasilkan *cepstrum* yang akan digunakan dalam membentuk *codeword*. MFCC biasanya digunakan pada sistem pengenalan suara dan didasarkan pada variasi dari batas *bandwidth* pendengaran manusia. Blok diagram dari MFCC ditunjukkan pada Gambar 2.10.



Gambar 2.10 Blok diagram MFCC

Proses MFCC diawali dengan membagi sinyal (citra) menjadi beberapa *frame* melalui proses *frame blocking*. Setelah itu dilakukan *windowing* pada setiap *frame*. *Windowing* bertujuan untuk meminimalisasi diskontinuitas sinyal dan distorsi spektral. Kemudian dari setiap *frame* dicari spektrum amplitudonya dengan terlebih dahulu mengubah masing-masing *frame* dari domain waktu ke domain frekuensi dengan menggunakan *Fast Fourier Transform* (FFT). Selanjutnya dilakukan proses *mel-frequency wrapping* untuk memperoleh sinyal spektrum dalam *mel-scale* dari hasil FFT. Langkah terakhir adalah mengubah hasil *log mel spectrum* ke dalam domain waktu. Hasil akhirnya adalah MFCC.

MFCC ditunjukkan dengan menggunakan skala *Mel-frequency* atau *Mel-scale*. Grafik dari *Mel-scale* dapat dilihat pada Gambar 2.11.



Gambar 2.11 Mel scale[12]

Mel scale adalah skala yang memiliki titik acuan yang jika dibandingkan dengan pengukuran frekuensi normal dapat didefinisikan dengan frekuensi 1000 Hz dan 40dB di atas batas pendengaran adalah sama dengan 1000 mels. Untuk mengubah frekuensi dalam Hertz menjadi mel digunakan persamaan berikut:

$$m = 1127,01048 \log_e (1 + f / 700) \dots\dots\dots (2.3)$$

Sedangkan inversnya adalah:

$$f = 700(e^{m/1127.01048} - 1) \dots\dots\dots (2.4)$$

2.4.3.2 Fast Fourier Transform (FFT) [11]

FFT merupakan salah satu teknik dalam transformasi *Fourier* untuk mengubah suatu nilai dari domain waktu menjadi domain frekuensi. Transformasi *Fourier*, $F(u)$, dengan satu variabel dari fungsi kontinyu, $f(x)$ didefinisikan dengan persamaan:

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx \dots\dots\dots (2.5)$$

Dan $f(x)$ dapat diperoleh dengan inversnya:

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} du \dots\dots\dots (2.6)$$

Kedua persamaan di atas disebut sebagai pasangan transformasi *Fourier*.

Transformasi *Fourier* dapat pula dituliskan dalam bentuk sebagai berikut

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \dots\dots\dots (2.7)$$

untuk setiap nilai f riil. Dan inversnya:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{-j2\pi ft} df \dots\dots\dots (2.8)$$

untuk setiap nilai t riil. Transformasi *Fourier* seringkali dituliskan dalam bentuk frekuensi angular di mana $\omega = 2\pi f$ dengan satuan dalam radian per sekon. Dan dengan mensubstitusi $\omega = 2\pi f$ maka dapat dihasilkan rumus berikut:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \dots\dots\dots (2.9)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{-j\omega t} d\omega \dots\dots\dots (2.10)$$

Discrete Fourier Transform (DFT) merupakan transformasi *Fourier* dari fungsi diskrit. DFT dengan satu variabel, $f(x)$, $x = 0, 1, 2, \dots, M - 1$, dapat didefinisikan dengan persamaan berikut:

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M} \dots\dots\dots (2.11)$$

untuk $u = 0, 1, 2, \dots, M - 1$. Dan inversnya adalah

$$f(x) = \sum_{x=0}^{M-1} F(u) e^{j2\pi ux/M} \dots\dots\dots (2.12)$$

untuk $x = 0, 1, 2, \dots, M - 1$. Perkalian dengan $1/M$ yang diletakkan di depan transformasi *Fourier* terkadang juga dapat diletakkan di depan inversnya.

Jumlah operasi untuk menghitung DFT berdasarkan persamaan (2.11) adalah sebanding dengan M^2 . Nilai ini diperoleh untuk setiap nilai M dari u , di mana diperlukan sebanyak M kompleks perkalian dari $f(x)$ dengan $e^{(-j2\pi ux/M)}$ dan penjumlahan $M-1$ dari hasilnya. Untuk hasil yang sama, dengan menggunakan *Fast Fourier Transform* (FFT) hanya diperlukan operasi sebanyak $M \log_2 M$.

Untuk lebih jelasnya, persamaan (2.11) dapat dituliskan sebagai berikut

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) W_M^{ux} \dots\dots\dots (2.13)$$

di mana

$$W_M = e^{-j2\pi / M} \dots\dots\dots (2.14)$$

dan M diasumsikan

$$M = 2^m \dots\dots\dots (2.15)$$

dengan m adalah bilangan integer positif. Kemudian M juga dapat diekspresikan

$$M = 2N \dots\dots\dots (2.16)$$

dengan N juga merupakan bilangan integer positif. Maka persamaan (2.13) dapat dituliskan sebagai berikut

$$F(u) = \frac{1}{2} \left[\frac{1}{N} \sum_{x=0}^{N-1} f(2x) W_N^{ux} + \frac{1}{N} \sum_{x=0}^{N-1} f(2x+1) W_N^{ux} W_{2N}^{ux} \right] \dots\dots\dots (2.17)$$

Dengan mendefinisikan

$$F_{\text{even}}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(2x) W_N^{ux} \dots\dots\dots (2.18)$$

untuk $u = 0, 1, 2, \dots, N-1$, dan

$$F_{\text{odd}}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(2x+1) W_N^{ux} \dots\dots\dots (2.19)$$

maka persamaan (2.17) dapat dituliskan sebagai berikut

$$F(u) = \frac{1}{2} [F_{\text{even}}(u) + F_{\text{odd}}(u) W_{2N}^{ux}] \dots\dots\dots (2.20)$$

Implementasi persamaan (2.18) sampai (2.20) akan menghasilkan algoritma FFT rangkap berturut-turut. Algoritma tersebut muncul dari metode perhitungan transformasi dua titik yang dapat dihitung dari transformasi satu titik, perhitungan transformasi empat titik dari transformasi dua titik, dan seterusnya untuk nilai M yang sebanding dengan hasil pangkat bilangan integer dari basis 2. Dengan demikian jumlah operasi perhitungan FFT, $a(n)$ dengan $n \geq 1$ adalah

$$a(n) = 2^m \log_2 2^m = M \log_2 M \dots\dots\dots (2.21)$$

FFT merupakan algoritma yang efisien dalam menghitung DFT dan inversnya. Perhitungan DFT memerlukan operasi sebanyak M^2 , sedangkan FFT dapat memenuhi hal yang sama dengan operasi sebanyak $M \log_2 M$. Dengan demikian FFT merupakan *fast algorithm* untuk mengimplementasikan DFT.

2.4.4 Kuantisasi Vektor

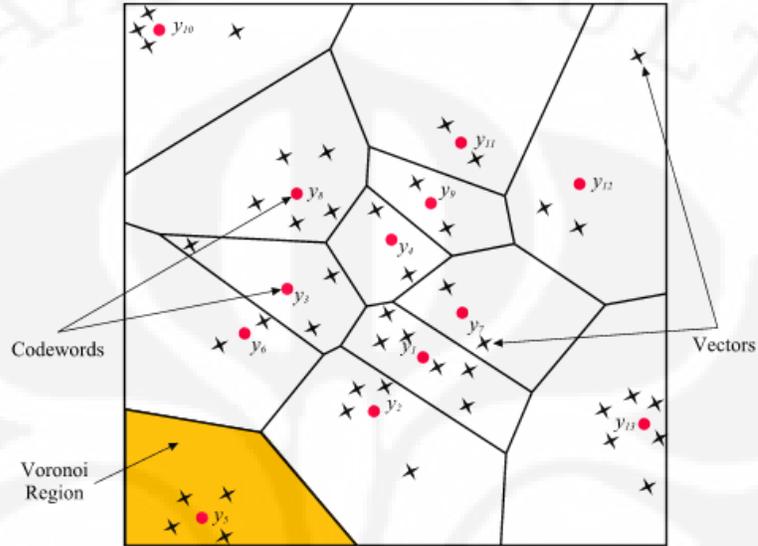
Kuantisasi vektor atau *Vector Quantization* (VQ) merupakan teknik kuantisasi klasik di mana dilakukan pemodelan dari fungsi kepadatan probabilitas dengan distribusi vektor. Kuantisasi vektor memetakan vektor dengan dimensi k pada ruang vektor R_k menjadi suatu bentuk vektor berhingga $Y = \{y_i : i = 1, 2, \dots, n\}$. Vektor y_i disebut sebagai vektor kode. Vektor-vektor ini merupakan vektor-vektor data yang diperoleh dari hasil ekstraksi dan disebut sebagai *codeword*. Kumpulan dari *codeword* disebut dengan *codebook*.

Gambar 2.12 menggambarkan vektor pada suatu ruang dengan garis horizontal menunjukkan nilai riil dan garis vertikal menunjukkan nilai imajiner dari vektor. Setiap *cluster* dari vektor menunjukkan *codeword*-nya. Setiap *codeword* berada pada daerah *Voroinoi*-nya masing-masing. Daerah ini dipisahkan dengan garis imajiner seperti yang terlihat pada Gambar 2.12. Vektor masukan ditandai dengan x sedangkan *codeword* ditandai dengan bulatan berwarna merah.

Representasi *codeword* ditentukan berdasarkan jarak *Euclidian* terdekat dari vektor masukan. Jarak *Euclidian* didefinisikan dengan persamaan berikut ini [13].

$$d(x, y_i) = \sqrt{\sum_{j=1}^k (x_j - y_{ij})^2} \dots\dots\dots (2.22)$$

Di mana x_j adalah komponen ke- j dari vektor masukan dan y_{ij} adalah komponen ke- j dari *codeword* y_i .



Gambar 2.12 Contoh *codeword* pada ruang dua dimensi [13]

Jarak dari suatu vektor ke *codeword* terdekat disebut dengan distorsi. Pada proses pengenalan, total distorsi yang paling kecil antara *codeword* dari *database* dan *codebook* VQ dari *input* merupakan hasil identifikasi.

Untuk memperbaiki VQ dalam proses pembuatan *codebook*, digunakan *General Lylod Algorithm* (GLA) atau biasa disebut *LBG Algorithm*. Prosedur implementasinya adalah sebagai berikut[14]:

- (1) Mendesain suatu vektor *codebook* yang merupakan *centroid* dari keseluruhan vektor *training*.
- (2) Mengubah ukuran *codebook* menjadi dua kalinya dengan membagi masing-masing *current codebook*, C_n , menurut aturan:

$$C_n^+ = C_n (1 + \varepsilon) \dots\dots\dots (2.23)$$

$$C_n^- = C_n (1 - \varepsilon) \dots\dots\dots (2.24)$$

di mana n bervariasi dari 1 sampai sebesar ukuran *codebook* dan ε adalah parameter *splitting* ($\varepsilon = 0,01$).

- (3) Melakukan *Nearest Neighbour Search* dengan mengelompokkan vektor-vektor *training* yang berkumpul pada blok-blok tertentu. Selanjutnya menentukan *codeword* dalam *current codebook* terdekat dan memberikan tanda vektor berupa sel yang diasosiasikan dengan *codeword* terdekat.
- (4) Melakukan *centroid update* dengan menentukan *centroid* baru pada masing-masing sel dengan menggunakan vektor *training* pada sel tersebut.
- (5) Melakukan iterasi pertama dengan mengulang langkah 3 dan 4 sampai jarak rata-rata di bawah *present treshold*.
- (6) Melakukan iterasi kedua dengan mengulang langkah 2, 3, dan 4 sampai *codebook* berukuran M .

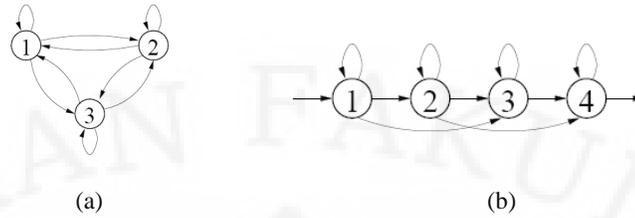
2.5 HIDDEN MARKOV MODEL

Hidden Markov Model (HMM) adalah model statistik di mana suatu sistem yang dimodelkan diasumsikan berupa proses Markov dengan parameter-parameter yang belum diketahui dan parameter-parameter yang tersembunyi tersebut harus ditentukan dari parameter yang *observable*. Parameter model yang diambil kemudian dapat digunakan untuk keperluan analisa selanjutnya, misalnya untuk aplikasi pengolahan citra.

Pada model Markov biasa, *state* (x) secara langsung dapat dilihat oleh *observer* dan oleh karena itu, probabilitas transisi menjadi satu-satunya parameter. Pada HMM, *state* tidak secara langsung dapat dilihat, tetapi bervariasi dipengaruhi oleh *state* yang terlihat. Setiap *state* memiliki distribusi probabilitas dari tanda-tanda *output* yang mungkin. Oleh karena itu, *sequence* dari tanda-tanda tersebut yang dihasilkan dari HMM akan memberikan informasi mengenai *sequence* dari *state*.

2.5.1 Model Markov[15]

Bentuk umum dari rantai *Markov* adalah bentuk *Ergodic* seperti yang dapat dilihat pada Gambar 2.13(a). Namun dapat juga dimodelkan dengan model *left-right Markov* seperti pada Gambar 2.13(b).



Gambar 2.13 Model Markov (a) Ergodic (b) left-right [13]

Matriks transisi dari model Ergodic dapat dinyatakan sebagai berikut:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \dots\dots\dots (2.25)$$

2.5.2 Elemen HMM[16]

Elemen-elemen Hidden Markov Model meliputi:

(1) N , jumlah state dalam model. Umumnya setiap state saling berhubungan sehingga suatu state dapat dicapai dari state yang lain. State individual dinotasikan sebagai $S = \{S_1, S_2, \dots, S_N\}$ dan state dalam waktu t sebagai q .

(2) M , jumlah simbol observasi yang berbeda tiap state. Simbol individual dinotasikan sebagai $V = \{V_1, V_2, \dots, V_M\}$.

(3) $A = \{a_{ij}\}$, distribusi probabilitas transisi state, di mana
 $A_{ij} = P[q_{t+1} = S_j \mid q_t = S_i] \dots\dots\dots (2.26)$
 dengan $1 \leq i, j \leq N$.

(4) $B = \{b_j(k)\}$, distribusi probabilitas simbol observasi pada state ke j , di mana
 $b_j(k) = P[v_k \text{ pada } t \mid q_t = S_j] \dots\dots\dots (2.27)$
 dengan $1 \leq j \leq N$ dan $1 \leq k \leq M$.

(5) $\pi = \{\pi_i\}$, distribusi initial-state, di mana
 $\pi = P[q_t = S_i] \dots\dots\dots (2.28)$
 dengan $1 \leq i \leq N$.

Elemen π , A , dan B merupakan parameter-parameter markov dalam HMM yang tidak diketahui atau tersembunyi (*hidden*). Dari ketiga parameter tersebut maka HMM dapat dituliskan dalam bentuk $\lambda = (A, B, \pi)$.

Jika diberikan suatu model $P(O|\lambda)$ dengan probabilitas urutan observasi $O = O_1 O_2 \dots O_T$, maka untuk mengetahui nilai probabilitas observasinya diperlukan suatu urutan *state* yang tetap, misalkan

$$Q = q_1 q_2 \dots q_T \dots \dots \dots (2.29)$$

di mana q_1 adalah *initial-state*. Maka probabilitas urutan observasi O untuk urutan *state* persamaan (2.29) adalah

$$P(O | Q, \lambda) = \prod_{t=1}^T P(O_t | q_t, \lambda) \dots \dots \dots (2.30)$$

Sehingga diperoleh

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \dots b_{q_T}(O_T) \dots \dots \dots (2.31)$$

Probabilitas dari urutan *state* Q maka dapat ditulis sebagai berikut

$$P(Q | \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \dots \dots \dots (2.32)$$

Probabilitas gabungan O dan Q yang merupakan probabilitas saat O dan Q muncul bersamaan adalah hasil perkalian dari keduanya atau dapat ditulis sebagai berikut

$$P(O, Q | \lambda) = P(O | Q, \lambda) P(Q, \lambda) \dots \dots \dots (2.33)$$

Probabilitas observasi O diperoleh dengan menjumlahkan probabilitas gabungan dari semua kemungkinan urutan *state* q , yaitu

$$P(O | \lambda) = \sum_{all Q} P(O | Q, \lambda) P(Q | \lambda) \dots \dots \dots (2.34)$$

Atau dapat juga dituliskan sebagai berikut

$$P(O | \lambda) = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T) \dots \dots (2.35)$$

Untuk menghitung persamaan (2.35) di atas diperlukan suatu prosedur yang efisien yang disebut dengan prosedur *forward-backward*. Prosedur ini menggunakan variabel *forward* $\alpha_t(i)$ yang didefinisikan sebagai

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) \dots \dots \dots (2.36)$$

dengan probabilitas urutan observasi sebagian adalah $O_1 O_2 \dots O_t$ (sampai waktu t) dan state S_i pada waktu t dengan model λ yang diberikan. Penyelesaian $\alpha_t(i)$ dapat dilakukan sebagai berikut:

- (1) Inisialisasi

$$\alpha_t(i) = \pi_i b_i(O_1) \dots\dots\dots (2.37)$$

dengan $1 \leq i \leq N$.

(2) Induksi

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \dots\dots\dots (2.38)$$

Dengan $1 \leq t \leq T-1, 1 \leq j \leq N$.

(3) Terminasi

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \dots\dots\dots (2.39)$$

Sedangkan variabel *backward* $\beta_t(i)$ didefinisikan sebagai

$$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T, q_t = S_i | \lambda) \dots\dots\dots (2.40)$$

dengan probabilitas urutan observasi parsial dari $t+1$ sampai akhir diperoleh dari *state* S_i pada waktu t dan model λ . Penyelesaian untuk $\beta_t(i)$ dapat dilakukan sebagai berikut:

(1) Inisialisasi

$$\beta_t(i) = 1 \dots\dots\dots (2.41)$$

dengan $1 \leq i \leq N$.

(2) Induksi

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \dots\dots\dots (2.42)$$

Dengan $t = T-1, T-2, \dots, 1, 1 \leq i \leq N$.

Dalam menyesuaikan parameter model $\lambda = (A, B, \pi)$ untuk memaksimalkan probabilitas urutan observasi $P(O | \lambda)$ berdasarkan model dapat digunakan metode *Baum-Welch*. Nilai $\xi_t(i, j)$ yang merupakan probabilitas dalam *state* S_i pada waktu t dan *state* S_j pada waktu $t+1$ dapat didefinisikan sebagai berikut:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \dots\dots\dots (2.43)$$

Dengan variabel *forward* dan *backward*, maka $\xi_t(i, j)$ dapat ditulis dalam bentuk berikut

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} \dots\dots\dots (2.44)$$

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \dots\dots\dots (2.45)$$

Jika probabilitas dalam *state* S_i pada waktu t dengan model serta urutan observasi yang diberikan disimbolkan dengan $\gamma_t(i)$ maka dapat dituliskan sebagai berikut

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \dots\dots\dots (2.46)$$

Kemudian jika $\gamma_t(i)$ dijumlahkan sepanjang waktu t maka diperoleh hasil yang dapat diinterpretasikan sebagai jumlah transisi dari *state* S_i yang diharapkan.

Begitu pula untuk penjumlahan $\xi_t(i, j)$ sepanjang waktu t (dari $t = 1$ sampai $t = T - 1$) dapat diinterpretasikan sebagai jumlah transisi dari *state* S_i sampai *state* S_j yang diharapkan. Maka dapat ditulis

$$\sum_{j=1}^N \gamma_t(i) = \text{jumlah transisi dari } S_i \text{ yang diharapkan} \dots\dots\dots (2.47)$$

$$\sum_{j=1}^N \xi_t(i, j) = \text{jumlah transisi dari } S_i \text{ sampai } S_j \text{ yang diharapkan} \dots\dots\dots (2.48)$$

Dengan menggunakan semua persamaan di atas maka dapat diperoleh re-estimasi dari parameter-parameter HMM. Persamaan re-estimasi untuk nilai π , A dan B adalah sebagai berikut:

$$\begin{aligned} \bar{\pi} &= \text{frekuensi(jumlah)pada state } S_i \text{ pada waktu } (t = 1) \\ &= \gamma_t(i) \dots\dots\dots (2.49) \end{aligned}$$

$\bar{a}_{ij} = \frac{\text{jumlah transisi dari state } Si \text{ sampai state } Sj \text{ yang diharapkan}}{\text{jumlah transisi dari state } Si \text{ yang diharapkan}}$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \dots\dots\dots (2.50)$$

$\bar{b}_j(k) = \frac{\text{jumlah waktu pada state } j \text{ dan simbol observasi } v_k \text{ yang diharapkan}}{\text{jumlah waktu pada state } j \text{ yang diharapkan}}$

$$= \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j) \text{ s.t. } O_t=v_k} \dots\dots\dots (2.51)$$

BAB III

RANCANG BANGUN PENGENALAN PENYAKIT DARAH DENGAN HMM

Proses pengenalan penyakit dengan citra darah ini secara garis besar dibagi menjadi dua tahap utama, yaitu proses pembentukan *database* dan proses pengenalan. Citra darah akan dianalisa dengan menggunakan metode *Hidden Markov Model* (HMM) yang akan diproses dengan menggunakan perangkat lunak untuk komputasi numerik.

Sebelum dilakukan proses pengenalan darah, terlebih dahulu dibuat *database* untuk menyimpan data *codeword* dan parameter HMM dari citra darah beberapa jenis penyakit. Dalam proses pembentukan *database* tersebut dilakukan proses pembelajaran (*training*) terhadap 15 *file* yang merupakan *sample* dari citra darah untuk tiap jenis penyakit. Parameter-parameter di dalam *database* selanjutnya akan digunakan sebagai pembanding dalam proses pengenalan.

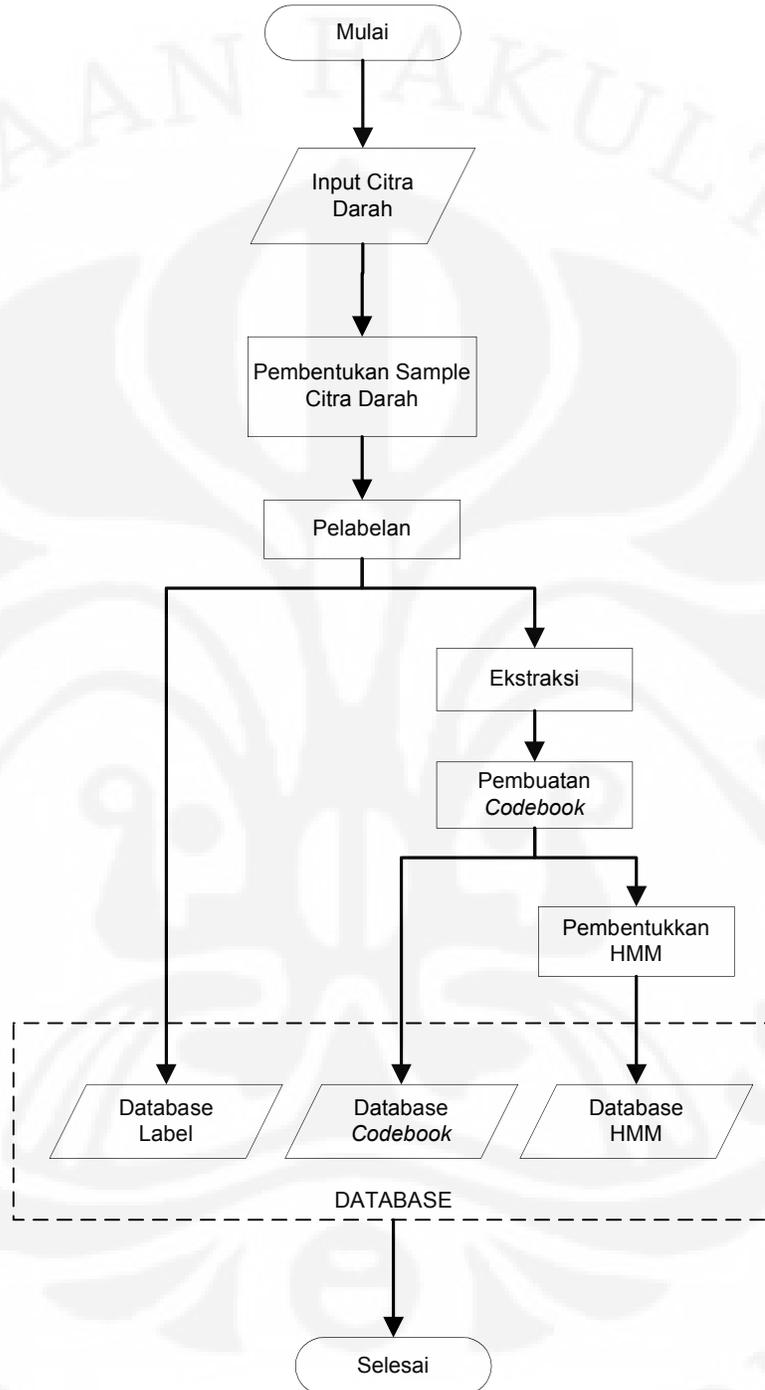
3.1 PEMBENTUKAN DATABASE

Sebelum dilakukan proses pengenalan, dilakukan proses pembentukan *database* untuk memperoleh data sebagai acuan. Terdapat 3 tahapan proses utama dalam pembentukan *database*, yaitu:

- (1) Proses pelabelan
- (2) Proses pembuatan *Codebook*
- (3) Proses pembentukan HMM

Diagram alir pembuatan *database* dapat dilihat pada Gambar 3.1.

Citra darah yang dimasukkan dibentuk menjadi *sample* citra darah dengan melibatkan proses pengolahan citra. Proses pengolahannya terdiri dari transformasi *grayscale*, *leveling*, *cropping*, dan *reshapping*. Hasil *sample* citra yang telah diolah tersebut yang selanjutnya akan dipakai pada proses-proses utama dalam pembentukan *database*.



Gambar 3.1 Diagram alir pembentukan *database*

Input citra RGB diubah menjadi bentuk *grayscale* sehingga hanya memiliki 256 derajat keabuan. Kemudian derajat keabuan tersebut diubah dengan *leveling*

dengan 5 skala (0, 1, 2, 3, dan 4). Selain itu, *input* citra yang utuh di-*cropping* menjadi beberapa *sample* citra. *Sample* yang dibentuk merupakan potongan pada bagian sel-sel darah putih yang menunjukkan karakteristik penyakit leukemia. Terakhir adalah proses *reshapping* yang akan mengubah ukuran array matriks *sample* citra menjadi satu baris.

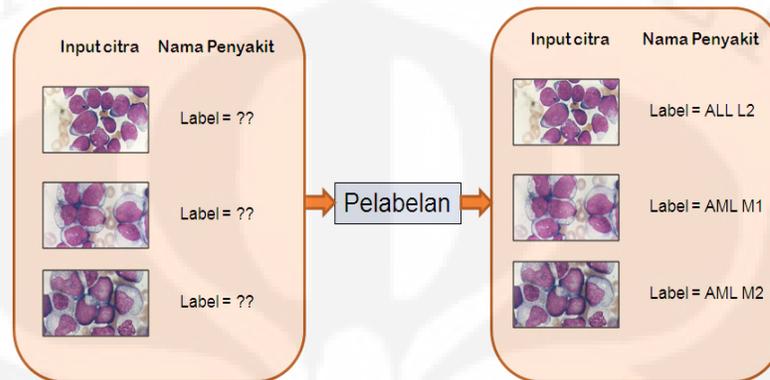
Algoritma proses pembuatan *sample* citra dari *input* citra darah adalah sebagai berikut:

```
definisikan parameter sampling;
untuk i = 1 sampai jumlah sampling
    citradarah [i] = nama penyakit.jpg;
    citradarah input = nama penyakit [i];
    make sample (citradarah input);
    sample [i] = citradarah input;
    imread nama penyakit.jpg;
    baca level tiap piksel;
        jika level antara 0 sampai 50
            gray level = 0;
        jika level antara 51 sampai 101
            gray level = 1;
        jika level antara 102 sampai 152
            gray level = 2;
        jika level antara 153 sampai 203
            gray level = 3;
        jika level antara 204 sampai 255
            gray level = 4;
    bentuk piksel matriks ke piksel baris;
kembali
```

3.1.1 Proses Pelabelan

Proses pelabelan merupakan proses pembuatan label yang bertujuan untuk membuat nama *file* dari setiap citra darah *input*. Citra darah *input* tersebut merupakan *sample* dari beberapa penyakit leukemia. Melalui proses pelabelan maka tiap *sample* yang belum teridentifikasi akan memiliki label yang secara tidak langsung menunjukkan karakteristik penyakit leukemia tertentu. Ilustrasi proses pelabelan ini dapat dilihat pada Gambar 3.2. Pada Gambar 3.2 contoh citra yang digambarkan masih berupa citra awal. Pada proses sebenarnya, citra tersebut

terlebih dulu melalui tahap pengolahan citra seperti yang telah dijelaskan di atas sebelum dilakukan proses pelabelan. Jadi proses pelabelan dilakukan pada citra *input* yang telah menjadi *sample* citra.



Gambar 3.2 Proses Pelabelan

Berikut ini merupakan algoritma dalam membuat label nama penyakit dari *sample* citra darah.

```

untuk i = 1 sampai 5
    input nama penyakit;
    tulis nama penyakit;
    nama label nama penyakit[i] = nama penyakit
kembali
    
```

Dalam perangkat lunak untuk komputasi numerik, *syntax* perintah yang dimasukkan untuk membuat label adalah:

```
>> make_labels(rep)
```

Banyaknya repetisi yang ingin dilakukan dimasukkan ke dalam “rep”. Setelah perintah tersebut dimasukkan maka akan diminta nama label yang menunjukkan identitas penyakit. Selanjutnya akan diminta masukan nama file yang termasuk dalam label tersebut. Jumlah nama file yang diminta adalah sebanyak repetisi yang diinginkan. Contoh tampilan pada *command window* yang menunjukkan proses pelabelan dengan 10 repetisi untuk label pertama dapat dilihat pada Gambar 3.3.

```
Command Window
>> make_labels(10)
Saving label 1
Write the name of the label
ALL L2
write the name of the file
L2_1
write the name of the file
L2_2
write the name of the file
L2_3
write the name of the file
L2_4
write the name of the file
L2_5
write the name of the file
L2_6
write the name of the file
L2_7
write the name of the file
L2_8
write the name of the file
L2_9
write the name of the file
L2_10
```

Gambar 3.3 Tampilan proses pelabelan

3.1.2 Proses Pembuatan *Codebook*

Proses pembuatan *codebook* ditentukan oleh hasil *sampling* dan ekstraksi yang dilakukan sebelumnya pada *sample* citra. Pembuatan *codebook* melibatkan proses yang disebut *vector quantization* (VQ). VQ merupakan proses pemetaan vektor dari ruang vektor yang besar menjadi daerah yang terbatas. Setiap daerah direpresentasikan dengan *centroid* yang disebut *codeword*. Kumpulan semua *codeword* disebut dengan *codebook*

Titik-titik *sample* yang berdekatan dikuantisasikan ke satu titik vektor sehingga diperoleh beberapa titik vektor atau *centroid*. Kemudian nilai-nilai *centroid* dari setiap *sample* untuk tiap jenis penyakit yang diperoleh dari proses pembelajaran tersebut akan disimpan menjadi sebuah *codebook*. *Codebook* yang dibentuk merupakan bagian dari *database* di mana terdapat kumpulan data-data dengan nama-nama *file* seperti M1_1, M1_2, M2_1 dan seterusnya.

Algoritma untuk proses FFT dan proses kuantisasi vektornya adalah sebagai berikut:

```
definisikan besar vektor
untuk i = 1 sampai jumlah sample
```

```

        hitung FFT untuk setiap sample [i];
        sample point [i] = nilai FFT;
    kembali
    definisikan cluster;
    untuk j = 1 sampai jumlah cluster
        hitung centroid;
        simpan centroid [j] berdasarkan urutan labelnya;
    kembali

```

Dalam perangkat lunak untuk komputasi numeriknya, *syntax* perintah yang dimasukkan untuk membuat *codebook* adalah:

```
>> make_codebook('VQ_filename',M,iteration)
```

Input “*VQ_filename*” pada fungsi di atas diisi dengan nama *file* di mana data *codebook* akan disimpan, input “*M*” menunjukkan ukuran dari *codebook* yang dibuat, dan input “*iteration*” merupakan jumlah iterasi untuk GLA yang umumnya bernilai 10. Contoh tampilan pada *command window* yang menunjukkan proses pembuatan *codebook* dengan nama *file* “*codebook*” dan ukuran *codebook* 32 dapat dilihat pada Gambar 3.4.

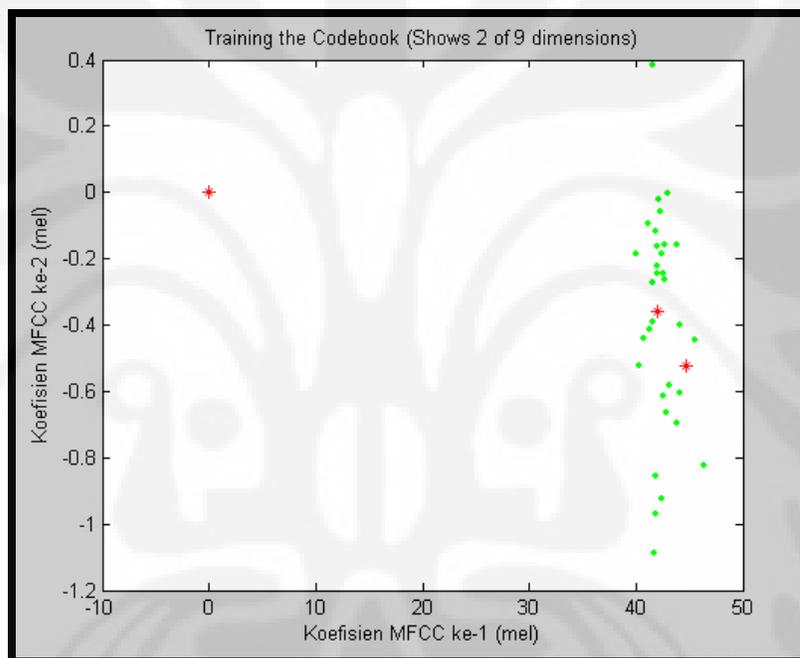
```

Command Window
>> make_codebook('codebook',32,10)
Extracting Features
Done!
number of points to train VQ:32
Training codebook of 1 bits
Distortion:1.131779
Distortion:1.070373
Distortion:1.021145
Distortion:0.993241
Distortion:0.973848
Distortion:0.955407
Distortion:0.955407
Distortion:0.955407
Distortion:0.955407
Distortion:0.955407
Training codebook of 2 bits
Distortion:0.480392
Distortion:0.477846
Distortion:0.477846

```

Gambar 3.4 Tampilan proses pemuatan *codebook*

Selanjutnya program akan melakukan proses pembelajaran terhadap *codebook* dan ditunjukkan dengan adanya nilai distorsi yang muncul dari hasil kuantisasi vektor. Selain itu pada tiap tahap pembelajaran akan ditampilkan pula plot nilai-nilai *codeword* yang diperoleh seperti yang dapat dilihat pada Gambar 3.5. Distorsi yang ada ditunjukkan oleh adanya perbedaan warna pada titik-titik vektornya. Proses ini kemudian diakhiri dengan terbentuknya daerah-daerah vektor.



Gambar 3.5 Plot titik-titik vektor pada proses kuantisasi vektor

3.1.3 Proses Pembentukan HMM

Proses pembentukan HMM dilakukan untuk mengetahui semua parameter HMM. Parameter-parameter HMM untuk setiap label adalah berdasarkan *centroid* yang diperoleh dari hasil pembentukan *database*. Jumlah nilai *centroid* dari setiap label akan menentukan jumlah *state* dari HMM. Dengan demikian dapat dituliskan algoritma proses pembentukan HMM sebagai berikut:

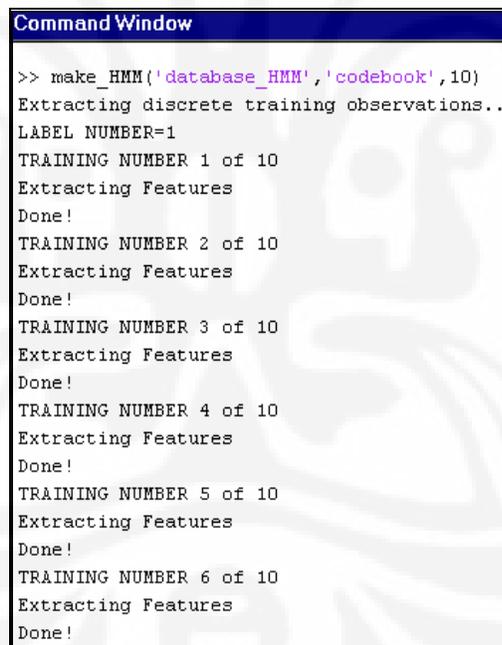
```
hitung jumlah centroid;
state = jumlah centroid;
```

Setelah diketahui jumlah *state* serta urutan penggalan *sampling* yang diwakili oleh tiap *centroid* dari semua *file* citra untuk semua jenis penyakit maka dilakukan proses pembentukan HMM sehingga diperoleh nilai-nilai parameter HMM. Seluruh parameter tersebut kemudian akan disimpan dalam *database*.

Dalam perangkat lunak, *syntax* perintah yang dimasukkan untuk proses pembentukan HMM adalah:

```
>> make_HMM('model_file','codebook',iteration)
```

Input “*model_file*” di atas diisi dengan nama *file* di mana data HMM akan disimpan, input “*codebook*” diisi dengan nama *file* di mana data *codebook* disimpan, dan input “*iteration*” merupakan jumlah iterasi untuk GLA yang umumnya bernilai 10. Contoh tampilan pada *command window* yang menunjukkan proses pembentukan HMM dengan nama *model_file* adalah “*database_HMM*” dan ukuran *codebook* 32 dapat dilihat pada Gambar 3.6.



```
Command Window
>> make_HMM('database_HMM','codebook',10)
Extracting discrete training observations..
LABEL NUMBER=1
TRAINING NUMBER 1 of 10
Extracting Features
Done!
TRAINING NUMBER 2 of 10
Extracting Features
Done!
TRAINING NUMBER 3 of 10
Extracting Features
Done!
TRAINING NUMBER 4 of 10
Extracting Features
Done!
TRAINING NUMBER 5 of 10
Extracting Features
Done!
TRAINING NUMBER 6 of 10
Extracting Features
Done!
```

Gambar 3.6 Tampilan proses pembentukan HMM

Program selanjutnya akan melakukan proses pembelajaran HMM yaitu perhitungan *log of probability* (LoP) untuk 10 iterasi pada tiap-tiap label. Tampilan yang muncul pada saat proses ini sedang berlangsung dapat dilihat pada gambar 3.7.

```
Command Window
TRAINING NUMBER 10 of 10
Extracting Features
Done!
TRAINING MODELS
Training label 1
Iteration:1
Iteration:2
Iteration:3
Iteration:4
Iteration:5
Iteration:6
Iteration:7
Iteration:8
Iteration:9
Iteration:10

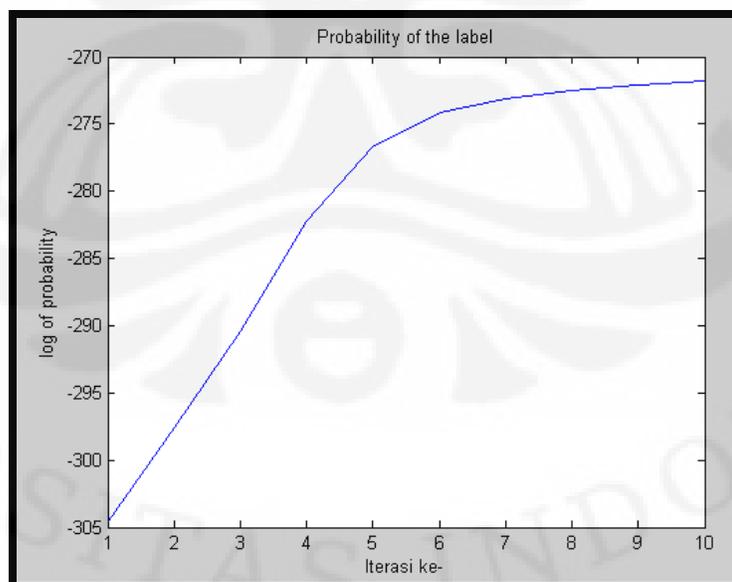
time =

    7.8430

Training label 2
Iteration:1
Iteration:2
```

Gambar 3.7 Tampilan saat perhitungan LoP tiap label

Setelah pembelajaran selesai kemudian akan ditampilkan grafik nilai *log of probability* hasil pembelajaran dari setiap label. Contoh grafiknya dapat dilihat pada Gambar 3.8.



Gambar 3.8 Grafik nilai *log of probability*

3.1.3.1 State dari Markov Chain

Setiap penggalan citra darah diubah ke dalam domain frekuensi dengan menggunakan metode FFT. Kemudian besaran vektor spektrum dari tiap citra darah dikuantisasi untuk memperoleh nilai *centroid*-nya. *Sample point* yang diperoleh dari beberapa penggalan citra darah yang mirip akan dikelompokkan ke dalam satu *cluster*.

Misalkan diperoleh nilai *centroid* sebagai berikut:

Penggalan citra darah C1 → *centroid* 1

Penggalan citra darah C2 → *centroid* 2

...

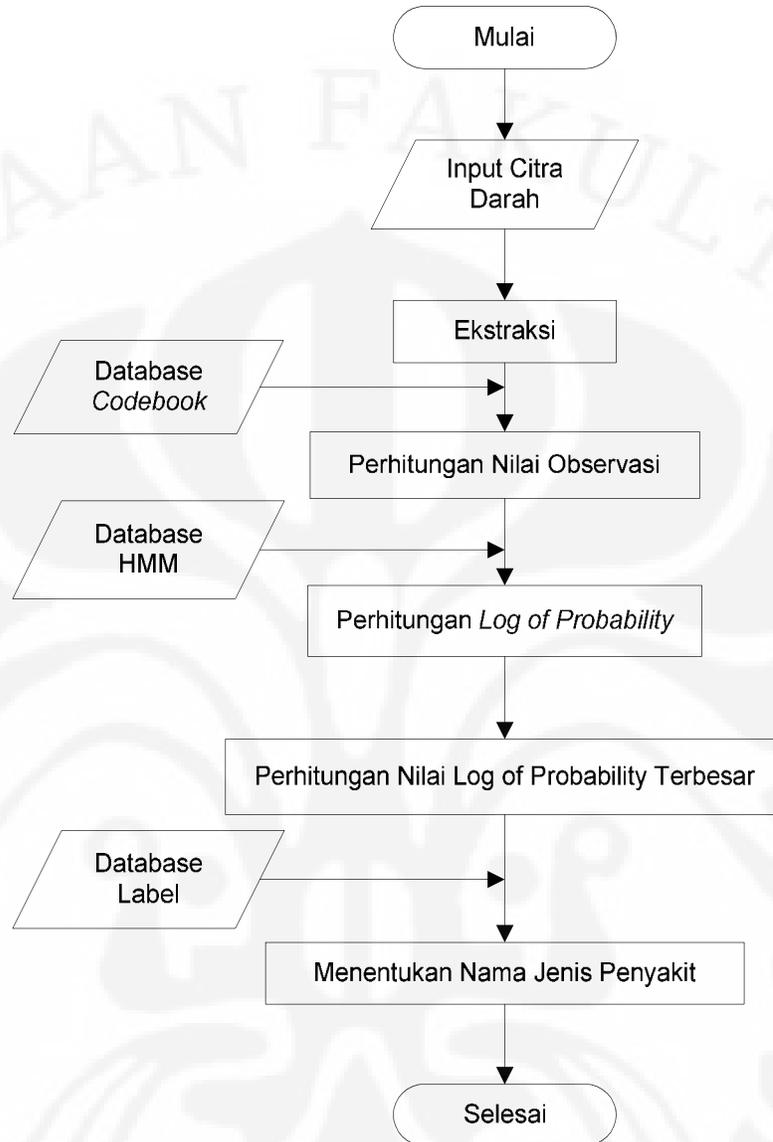
Penggalan citra darah C5 → *centroid* 5

Secara langsung penggalan-penggalan citra di atas menyatakan *state* dari *Markov Chain*. Namun dalam perhitungan, nilai *centroid*-nya yang akan mewakili penggalan citra darah tersebut dalam menentukan *state* dari *Markov Chain*.

3.2 PROSES PENGENALAN

Awal proses pengenalan tidak jauh beda dengan awal proses pembentukan *database*, namun pada proses pengenalan tidak lagi dilakukan proses pembelajaran. Diagram alir proses pengenalan dapat dilihat pada Gambar 3.9.

Pada proses pengenalan citra darah yang akan diidentifikasi terlebih dahulu melewati proses ekstraksi sehingga diperoleh *sample-sample* yang kemudian diubah ke dalam domain frekuensi dengan FFT. Hasilnya akan terbentuk titik-titik *sample* yang akan dicari nilai *centroid*-nya berdasarkan nilai codeword yang ada pada *database*. Dari nilai tersebut selanjutnya akan dihitung besar *Log of Probability* (LoP) untuk semua nilai parameter HMM yang didapatkan. Setelah itu nilai tersebut dibandingkan dengan LoP dari parameter HMM yang ada pada *database*. Nilai LoP yang paling besar adalah nilai LoP yang dicari. Dengan demikian maka didapatkan hasil identifikasi berupa nama label dari *file* yang dikenali yang sesuai dengan nilai LoP tersebut.



Gambar 3.9 Diagram alir proses pengenalan

Algoritma dari proses pengenalan penyakit dengan citra darah menggunakan metode HMM adalah sebagai berikut.

```

mulai
imread nama citrauji.jpg;
citradarah = nama citrauji.jpg;
penggal citra;
untuk g = 1 sampai jumlah penggalan citra
    citradarahcitrauji [g] = penggal citra;
kembali
  
```

```

definisikan nilai sampling dari citradarah_citrauji;
sampling citradarah_citrauji;
untuk i = 1 sampai jumlah penggal H
    hitung FFT untuk jumlah penggalan;
    matriks codeword_citrauji [i,m,n];
    cari centroid pada database untuk codeword_citrauji;
    definisikan observasi dari urutan dari centroid;
    definisikan parameter HMM;
    untuk h = 1 sampai jumlah_label
        baca data parameter HMM untuk jenis citra
        lainnya dari database;
        hitung log of probability (LoP) untuk semua
        label dan citrauji;
        LoP [jumlah_label] = tertinggi
    kembali
kembali
selesai

```

Dalam perangkat lunak untuk komputasi numerik, *syntax* perintah yang dimasukkan untuk proses pengenalan adalah:

```
>> recognition('model_file', 'codebook')
```

Input “*model_file*” di atas diisi dengan nama *file* di mana data HMM akan disimpan dan input “*codebook*” diisi dengan nama *file* di mana data *codebook* disimpan. Proses mulai berjalan saat dimasukkan nama *file* yang akan diidentifikasi. Setelah itu nilai hasil perhitungan *log of probability* untuk masing-masing label akan ditampilkan dan label yang akan diambil sebagai hasil identifikasi adalah label yang memiliki nilai *log of probability* terbesar. Contoh tampilan pada *command window* yang menunjukkan proses pengenalan dapat dilihat pada Gambar 3.10.

```
Command Window
>> recognition('database_HMM','codebook')
image labels:
names =
    '1:ALL L2'
    '2:AML M1'
    '3:AML M2'
    '4:AML M4'
    '5:AML M5'

Write the name of the file
L2_15
Extracting Features
Done!
Label   Log of Probability
-----
4       -68.845908
1       -69.643589
3       -78.584355
5       -79.163845
2       -83.115855

The image is 4:AML M4

ans =
    4
```

Gambar 3.10 Tampilan proses pengenalan

BAB IV

UJI COBA PERANGKAT LUNAK PENGENALAN PENYAKIT DARAH DENGAN HMM

Uji coba dilakukan terhadap 5 buah citra mikroskopis darah yang merepresentasikan 5 tipe leukemia. Dari tiap citra kemudian akan dilakukan uji coba pada 15 *sample* citra. Nama-nama file dari *sample* yang digunakan untuk uji coba dari kelima jenis penyakit leukemia dapat dilihat pada Tabel IV.1.

Tabel IV.1 Nama file dan label tiap jenis leukemia

No.	Jenis Leukemia	Nama Label	Nama File
1.	Acute Lymphoblastic Leukemia L2	ALL_L2	L2_1 – L2_15
2.	Acute Myeloblastic Leukemia M1	AML_M1	M1_1 – M1_15
3.	Acute Myeloblastic Leukemia M2	AML_M2	M2_1 – M2_15
4.	Acute Myeloblastic Leukemia M4	AML_M4	M4_1 – M4_15
5.	Acute Myeloblastic Leukemia M5	AML_M5	M5_1 – M5_15

Proses uji coba dilakukan dengan menggunakan variasi *codebook* dan repetisi. Prosesnya diawali dengan melakukan proses pembuatan *database* untuk tiap variasi *codebook* dan repetisi. Setelah itu baru dapat dilakukan proses pengenalan (*recognition*). Pada proses pengenalan, perangkat lunak ini akan mengolah *file input* data uji dengan melakukan perhitungan sampai diperoleh nilai *log of probability* terbesar dan *file* data uji dapat dikenali sesuai label yang memenuhi nilai *log of probability* tersebut.

Secara lengkap, proses uji coba meliputi:

- (3) Uji coba untuk ukuran *codebook* 32 dengan 5 repetisi
- (4) Uji coba untuk ukuran *codebook* 32 dengan 10 repetisi
- (5) Uji coba untuk ukuran *codebook* adalah 64 dengan 5 repetisi
- (6) Uji coba untuk ukuran *codebook* adalah 64 dengan 10 repetisi
- (7) Uji coba untuk ukuran *codebook* adalah 128 dengan 5 repetisi
- (8) Uji coba untuk ukuran *codebook* adalah 128 dengan 10 repetisi

4.1 HASIL UJI COBA PENGENALAN

Hasil uji coba pengenalan dengan variasi ukuran *codebook* sebesar 32, 64, dan 128 serta variasi jumlah repetisi sebanyak 5 dan 10 untuk masing-masing label penyakit dapat dilihat pada Tabel IV.2 sampai Tabel IV.6. Hasil uji coba pengenalan yang tidak teridentifikasi pada tiap-tiap label ditandai dengan warna merah.

Tabel IV.2 Hasil uji coba untuk penyakit ALL L2

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
1.	L2_1	ALL L2					
2.	L2_2	ALL L2					
3.	L2_3	ALL L2					
4.	L2_4	ALL L2					
5.	L2_5	ALL L2					
6.	L2_6	AML M1	ALL L2	AML M5	ALL L2	AML M1	ALL L2
7.	L2_7	AML M2	ALL L2	AML M2	ALL L2	AML M2	ALL L2
8.	L2_8	AML M5	ALL L2	AML M5	ALL L2	ALL L2	ALL L2
9.	L2_9	ALL L2					
10.	L2_10	AML M4	ALL L2				
11.	L2_11	AML M5	ALL L2	ALL L2	AML M5	ALL L2	ALL L2
12.	L2_12	AML M1	ALL L2	AML M1	ALL L2	AML M5	AML M5
13.	L2_13	AML M1	AML M5	AML M1	AML M1	ALL L2	ALL L2
14.	L2_14	ALL L2	ALL L2	ALL L2	AML M1	ALL L2	AML M1
15.	L2_15	AML M4	AML M4	AML M4	ALL L2	AML M4	AML M4

Tabel IV.3 Hasil uji coba untuk penyakit AML M1

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
1.	M1_1	AML M1					
2.	M1_2	AML M1					
3.	M1_3	AML M1					
4.	M1_4	AML M1					
5.	M1_5	AML M1					
6.	M1_6	AML M4	AML M1	AML M4	AML M1	AML M1	AML M1
7.	M1_7	AML M4	AML M1	AML M2	AML M1	AML M4	AML M1
8.	M1_8	AML M1					
9.	M1_9	AML M4	AML M1	AML M1	AML M1	ALL L2	AML M1
10.	M1_10	AML M1					

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
11.	M1_11	AML M5	AML M5	ALL L2	AML M5	ALL L2	AML M5
12.	M1_12	AML M2	AML M4	AML M2	AML M1	AML M1	AML M1
13.	M1_13	AML M4	AML M1	ALL L2	AML M1	AML M4	AML M1
14.	M1_14	ALL L2	ALL L2	ALL L2	AML M1	ALL L2	ALL L2
15.	M1_15	AML M2					

Tabel IV.4 Hasil uji coba untuk penyakit AML M2

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
1.	M2_1	AML M2					
2.	M2_2	AML M2					
3.	M2_3	AML M2					
4.	M2_4	AML M2					
5.	M2_5	AML M2					
6.	M2_6	AML M2					
7.	M2_7	AML M4	AML M2	AML M1	AML M2	AML M2	AML M2
8.	M2_8	AML M2	AML M2	AML M1	AML M2	AML M1	AML M2
9.	M2_9	AML M2	AML M2	AML M4	AML M2	AML M2	AML M2
10.	M2_10	AML M2					
11.	M2_11	AML M2	AML M2	AML M2	AML M2	ALL L2	AML M2
12.	M2_12	AML M1	AML M1	AML M1	ALL L2	AML M1	AML M1
13.	M2_13	AML M1	AML M1	AML M1	AML M1	AML M2	AML M2
14.	M2_14	ALL L2	AML M4	AML M2	AML M2	AML M4	AML M2
15.	M2_15	AML M4	AML M4	ALL L2	AML M2	ALL L2	AML M4

Tabel IV.5 Hasil uji coba untuk penyakit AML M4

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
1.	M4_1	AML M4					
2.	M4_2	AML M4					
3.	M4_3	AML M4					
4.	M4_4	AML M4					
5.	M4_5	AML M4					
6.	M4_6	ALL L2	AML M4	AML M1	AML M4	AML M2	AML M4
7.	M4_7	AML M2	AML M4				
8.	M4_8	AML M4	AML M4	AML M4	AML M4	ALL L2	AML M4
9.	M4_9	AML M4	AML M4	AML M4	AML M4	AML M2	AML M4
10.	M4_10	ALL L2	AML M4	AML M2	AML M4	AML M2	AML M4
11.	M4_11	ALL L2	AML M5	AML M1	ALL L2	AML M2	ALL L2

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
12.	M4_12	AML M4	AML M2	AML M1	AML M2	ALL L2	AML M2
13.	M4_13	AML M4	AML M2	AML M4	AML M2	AML M4	AML M4
14.	M4_14	AML M2	AML M2	AML M1	AML M2	AML M2	AML M2
15.	M4_15	AML M2	ALL L2	AML M1	AML M4	ALL L2	AML M4

Tabel IV.6 Hasil uji coba untuk penyakit AML M5

No.	Codebook	32		64		128	
	Repetisi Nama File	5	10	5	10	5	10
1.	M5_1	AML M5					
2.	M5_2	AML M5					
3.	M5_3	AML M5					
4.	M5_4	AML M5					
5.	M5_5	AML M5					
6.	M5_6	ALL L2	AML M5	AML M4	AML M5	ALL L2	AML M5
7.	M5_7	AML M5					
8.	M5_8	AML M5					
9.	M5_9	AML M2	AML M5	AML M1	AML M5	AML M2	AML M5
10.	M5_10	AML M5					
11.	M5_11	AML M1	AML M5				
12.	M5_12	AML M5	AML M5	AML M5	ALL L2	ALL L2	ALL L2
13.	M5_13	AML M5	AML M5	AML M5	AML M5	AML M4	AML M5
14.	M5_14	AML M5					
15.	M5_15	ALL L2	ALL L2	AML M2	ALL L2	ALL L2	AML M1

4.2 ANALISIS HASIL UJI COBA PENGENALAN

Sistem pengenalan penyakit pada skripsi ini menggunakan 5 jenis citra darah leukemia. Setiap jenis citra memiliki karakteristiknya masing-masing seperti yang dapat dilihat dari bentuk sel darah putih (*leukosit*). Karakteristik sel darah putih tersebut menjadi acuan dalam menentukan jenis penyakit leukemia dalam uji coba pengenalan ini.

Sistem pengenalan ini akan membandingkan *input* citra yang akan dikenali dengan data citra yang ada pada *database*. Dalam uji coba ini ada 2(dua) kemungkinan keadaan yang terjadi setelah proses pengenalan dilakukan, yaitu:

(1) Sistem mampu mengenali citra dengan benar

Keadaan ini terjadi jika hasil pengenalan sesuai dengan *input* yang diberikan.

Label yang muncul dari nilai *log of probability* terbesar pada proses

pengenalan adalah sama dengan label jenis penyakit leukemia dari *input* citra. Keadaan ini menandakan bahwa proses pengenalan yang dilakukan berhasil atau akurat.

(2) Sistem salah mengenali citra

Pada keadaan ini, hasil pengenalan ternyata berbeda dengan *input* yang diberikan. Label yang muncul berdasarkan nilai *log of probability* terbesar tidak sama dengan label atau jenis penyakit leukemia dari *input* citra. Keadaan ini menandakan bahwa proses pengenalan mengalami *error*.

Berdasarkan dua kemungkinan keadaan di atas maka dari hasil uji coba dapat diketahui tingkat keberhasilan atau keakuratan dari proses pengenalan yang dilakukan. Besar persentase akurasi dengan variasi besar *codebook* dan banyaknya repetisi yang diuji pada masing-masing label dapat dilihat pada Tabel IV.7. Sedangkan persentase akurasi keseluruhan dirangkum pada Tabel IV.8.

Tabel IV.7 Tabel persentase akurasi uji coba tiap-tiap label

Parameter Penyakit	Codebook 32		Codebook 64		Codebook 128	
	5 Repetisi	10 Repetisi	5 Repetisi	10 Repetisi	5 Repetisi	10 Repetisi
ALL L2	46.67%	86.67%	60.00%	80.00%	73.33%	80.00%
AML M1	46.67%	73.33%	53.33%	86.67%	60.00%	80.00%
AML M2	73.33%	73.33%	60.00%	86.67%	66.67%	86.67%
AML M4	60.00%	66.67%	60.00%	73.33%	46.67%	80.00%
AML M5	73.33%	93.33%	80.00%	86.67%	66.67%	86.67%

Tabel IV.8 Tabel persentase akurasi uji coba keseluruhan

Codebook \ Repetisi	32	64	128
5	60.00%	62.67%	62.67%
10	78.67%	82.67%	82.67%

4.2.1 Analisis Citra Darah Sebagai Data *input*

Seperti yang telah disebutkan sebelumnya bahwa data *input* yang digunakan dalam sistem pengenalan ini adalah citra darah mikroskopis untuk penyakit leukemia. Citra darah tersebut merupakan hasil akuisisi *sample* darah yang diambil dari sumsum tulang yang merupakan tempat diproduksi sel darah

putih (*leukosit*). Sel darah putih menjadi bagian penting dalam pengenalan ini karena informasi dari sel darah putih tersebut yang akan menentukan jenis penyakit leukemia yang dikandung dalam citra darah. Jadi proses ekstraksi citra yang dilakukan fokus pada sel-sel darah putih.

Perlu diketahui bahwa dalam skripsi ini digunakan 5 buah citra mikroskopis darah leukemia. Kelima buah citra tersebut terdiri atas 2 jenis leukemia yaitu leukemia limfositik akut (ALL) dan leukemia mielositik akut (AML). Citra darah pertama (L2) termasuk leukemia jenis ALL sedangkan 4 citra lainnya termasuk leukemia jenis AML. Keempat citra jenis AML yang digunakan memiliki perbedaan satu dengan lainnya karena merupakan 4 tipe AML yang berbeda-beda karakteristiknya, yaitu tipe M1, M2, M4, dan M5.

Dari hasil uji coba seperti yang dapat dilihat dari Tabel IV.2 sampai Tabel IV.6, diketahui bahwa beberapa citra uji tidak dikenali dengan benar atau terjadi kesalahan pengenalan dalam menentukan label. Hal ini mungkin karena proses *sampling* dilakukan secara acak dengan mengambil potongan sel darah putih (*leukosit*) dari setiap citra *input*. Sedangkan sel-sel darah putih hasil potongan tersebut mungkin memiliki karakteristik yang tidak jauh berbeda untuk beberapa label karena kesemua label yang diambil merupakan jenis penyakit leukemia seperti yang telah dipaparkan di atas.

Selain karena faktor internal dari citra darah, kesalahan pengenalan citra mungkin juga disebabkan akibat faktor eksternal yang ada dalam proses pengolahan citra. Adanya faktor eksternal yang menyebabkan kesalahan pengenalan ini merupakan kelemahan dalam sistem pengenalan penyakit pada skripsi ini. Faktor-faktor eksternal tersebut meliputi:

(1) Tipe citra

Tipe citra yang digunakan adalah citra dalam *grayscale* dengan 256 derajat keabuan. Citra *input* yang diberikan awalnya merupakan citra RGB yang kemudian dalam pengolahannya diubah ke dalam *grayscale*. Citra dalam *grayscale* ini memiliki keterbatasan level warna jika dibandingkan dengan RGB. Oleh karena itu, informasi citra darah dalam bentuk warna tidak dapat diperoleh dengan baik.

(2) Resolusi citra

Ukuran citra *input* yang digunakan adalah sebesar 234x400 piksel. Dengan resolusi citra sebesar itu bentuk atau nilai informasi yang didapat mungkin kurang teliti dibandingkan dengan menggunakan citra dengan resolusi yang lebih besar.

(3) Tingkat *leveling*

Citra *grayscale* dilevelkan ke dalam 5 level, yaitu menjadi bilangan integer 0 untuk level 0-50, 1 untuk level 51-101, 2 untuk level 102-152, 3 untuk level 153-204, dan 4 untuk level 205-255. Jika jumlah level semakin banyak maka perolehan karakteristik citra dapat lebih teliti karena matriks citra hasil *leveling* akan memiliki variasi level yang lebih beragam sehingga hasil pengenalan dapat menjadi lebih baik.

4.2.2 Analisis Pengaruh Perubahan *Codebook*

Perubahan *codebook* ternyata berpengaruh pada hasil uji coba pengenalan seperti yang dapat dilihat pada Tabel IV.7 dan Tabel IV.8. Secara keseluruhan, berdasarkan Tabel IV.8 terlihat bahwa semakin besar ukuran *codebook* maka akan menyebabkan tingkat akurasi yang semakin besar pula atau minimal sama.

Adanya perubahan ukuran *codebook* akan menghasilkan jumlah *codeword* (*centroid*) yang lebih banyak. Semakin banyaknya *centroid* akan menghasilkan nilai vektor data hasil vektor kuantisasi yang semakin teliti. Dengan kata lain, jarak antara vektor data dengan *centroid* semakin kecil. Hal ini ditunjukkan dengan semakin kecilnya nilai distorsi di akhir iterasi. Jadi semakin besar ukuran *codebook* dapat meningkatkan tingkat akurasi hasil pengujian.

Walaupun secara keseluruhan tingkat akurasi semakin besar akibat penambahan ukuran *codebook*, untuk beberapa uji coba pada label tertentu terjadi penurunan tingkat akurasi seperti yang dapat dilihat pada Tabel IV.7. Jadi, jika dilihat dari hasil uji coba tiap-tiap label, peningkatan ukuran *codebook* belum tentu meningkatkan tingkat akurasi.

Menurunnya tingkat akurasi yang terjadi pada beberapa hasil uji coba adalah karena terdapat beberapa *input* citra uji yang tidak teridentifikasi oleh sistem. Kesalahan pengenalan ini dapat terjadi karena adanya karakteristik dari *input* citra

uji untuk jenis label tertentu memiliki kemiripan dengan karakteristik citra untuk label lain pada *database*.

Adanya kemiripan karakteristik dari *input* citra uji dari satu label dengan *sample* citra pada *database* dari label lain menyebabkan *codeword* hasil citra uji dengan ukuran *codebook* yang semakin besar ternyata semakin mendekati *codeword* dari *sample* citra untuk label lain. Keadaan tersebut ditunjukkan oleh sistem saat membandingkan nilai observasi dan nilai *log of probability* (LoP). Sebagai contoh dapat dilihat hasil perhitungan nilai LoP untuk label AML M4 dengan *file input* citra uji adalah “M4_9” berikut ini. Untuk ukuran *codebook* 64 adalah sebagai berikut:

```
Write the name of the file
M4_9
Extracting Features
Done!
Label    Log of Probability
-----
4        -51.572374
3        -80.290456
2        -105.956585
1        -110.858274
5        -154.868126

The image is 4:AML M4
```

Sedangkan untuk ukuran *codebook* 128 adalah sebagai berikut:

```
Write the name of the file
M4_9
Extracting Features
Done!
Label    Log of Probability
-----
3        -75.916226
4        -84.157181
1        -86.323138
2        -129.607250
5        -156.708769

The image is 3:AML M2
```

Dari kedua hasil di atas dapat dilihat bahwa pada file *input* citra uji yang sama yaitu “M4_9”, nilai LoP untuk label 4 (AML M4) berubah dari -51.572374 untuk *codebook* 64 menjadi -84.157181 untuk *codebook* 128. Nilai LoP untuk label “AML M_4” pada ukuran *codebook* 128 lebih kecil dari LoP pada ukuran *codebook* 64. Selain itu terjadi pula perubahan nilai LoP untuk label 3 (AML M2)

yang menjadi semakin besar dan pada ukuran *codebook* 128 nilainya melebihi nilai LoP label 4 (AML M4). Dengan demikian sistem melakukan kesalahan pengenalan dan hasil akhir yang dikenali bukan AML M4 melainkan AML M2.

4.2.3 Analisis Pengaruh Perubahan Repetisi

Selain perubahan ukuran *codebook*, adanya perubahan jumlah repetisi juga berpengaruh pada tingkat akurasi hasil uji coba. Berdasarkan Tabel IV.8 diketahui bahwa meningkatnya jumlah repetisi akan meningkatkan tingkat akurasi pada keseluruhan sistem pengenalan.

Adanya penambahan jumlah repetisi secara langsung akan menambah jumlah data yang akan dimasukkan ke dalam *database*. Banyaknya data pada *database* ini ditentukan saat melakukan proses pelabelan, di mana banyaknya data *file* yang diminta adalah sebanyak repetisi yang diinginkan. Jadi jika awalnya repetisi adalah 5 sehingga data yang disimpan pada *database* juga sebanyak 5 maka saat diubah repetisinya menjadi 10, datanya juga bertambah menjadi 10.

Dengan lebih banyak repetisi maka secara tidak langsung akan menambah karakteristik label yang disimpan pada *database*. Karena data yang dimasukkan lebih banyak, maka karakteristik yang tadinya belum diambil saat proses pembentukan *database* dilakukan dengan repetisi yang lebih kecil, akan terpenuhi dengan repetisi yang lebih besar. Dari data yang lebih banyak tersebut akan mempengaruhi proses pembentukan *database* serta proses pengenalan sehingga nilai *log of probability* mengalami perubahan. Sebagai contoh dapat dilihat hasil perhitungan nilai LoP untuk label ALL L2 dengan *file input* citra uji adalah “L2_8” berikut ini. Untuk jumlah repetisi 5 hasilnya adalah sebagai berikut:

```
Write the name of the file
L2_8
Extracting Features
Done!
Label    Log of Probability
-----
5        -60.505884
1        -63.686977
4        -66.208925
2        -69.220469
3        -81.083453

The image is 5:AML M5
```

Sedangkan untuk jumlah repetisi 10 adalah sebagai berikut:

```
Write the name of the file
L2_8
Extracting Features
Done!
Label   Log of Probability
-----
1       -29.844061
5       -43.669107
4       -43.850861
2       -61.449613
3       -72.667078
```

The image is 1:ALL L2

Dari kedua hasil di atas dapat dilihat bahwa pada file *input* citra uji yang sama yaitu “L2_8”, nilai LoP untuk label 1 (ALL L2) berubah dari -63.686977 untuk 5 repetisi menjadi -29.844061 untuk 10 repetisi. Nilai LoP untuk label “ALL L2” pada jumlah repetisi 5 lebih kecil daripada nilai LoP pada jumlah repetisi 10. Dengan demikian sistem melakukan pengenalan dengan benar pada repetisi yang lebih besar (repetisi 10) karena hasil akhir yang dikenali adalah ALL L2, tidak lagi AML M5.

Dari uraian di atas maka dapat dikatakan bahwa peningkatan repetisi menyebabkan tingkat akurasi pengenalan akan semakin besar. Hal ini juga berlaku pada hasil uji coba untuk setiap label dan *codebook* yang repetisinya diubah seperti yang dapat dilihat pada Tabel IV.7. Berdasarkan Tabel IV.7 hanya terdapat satu label pada *codebook* 32 yang tidak menunjukkan perubahan tingkat akurasi melainkan sama. Namun hal tersebut tidak terlalu berpengaruh karena sebagian besar menunjukkan tingkat akurasi yang meningkat.

BAB V

KESIMPULAN

Berdasarkan analisis dari hasil pengenalan penyakit yang dirancang agar mampu mengenali penyakit dari citra darah dengan menggunakan metode HMM, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Perubahan ukuran *codebook* dengan variasi sebesar 32, 64 dan 128 serta perubahan jumlah repetisi dengan variasi bernilai 5 dan 10, akan mempengaruhi tingkat akurasi hasil pengenalan dengan tingkat akurasi secara keseluruhan berkisar antara 60% sampai 82,76%.
2. Tingkat akurasi pengenalan secara umum akan meningkat jika ukuran *codebook* bertambah atau jika jumlah repetisi bertambah.
3. Memperbanyak jumlah repetisi akan menghasilkan peningkatan akurasi yang lebih besar jika dibandingkan dengan peningkatan akurasi akibat penambahan ukuran *codebook*.

DAFTAR ACUAN

- [1] “_____”, “*Tak Perlu Darah, Ludah Pun Bisa Deteksi Penyakit*”, diakses 31 Desember 2007, dari Kompas Cyber Media.
<http://www.kompas.com/kesehatan/news/0404/01/100129.htm>
- [2] “_____”, “*Biologi Darah*”. Diakses 27 April 2008 dari Medicastore.
<http://www.medicastore.com/med/>
- [3] Zalwa Setiadi, “*Mimisan Yang Tidak Selalu Ringan*”. Diakses 27 April 2008, dari Balita-Anda. <http://www.mail-archive.com/balita-anda@balita-anda.com/msg102098.html>
- [4] Uci, “*Keguguran Berkali-kali? Waspadai Kelainan Darah*”. Diakses 27 April 2008, dari Balita-Anda. <http://www.mail-archive.com/balita-anda@balita-anda.com/msg168097.html>
- [5] Steven Dowshen, MD, “*Blood*”. Diakses 23 November 2007, dari Nemours Foundation Kids Health,
http://www.kidshealth.org/teen/your_body/bloodteory.html
- [6] Alison M. Stuebe, M.D., “*Anemia*”. Diakses 2 November 2007, dari VeriMed Healthcare Network.
http://pennhealth.com/health_info/pregnancy/000221.htm
- [7] “_____”, “*Leukemia*”. Diakses 27 April 2008, dari Medicastore.
http://www.medicastore.com/med/detail_pyk.php?idktg=12&judul=Leukemia&iddtl=45&UID=2007123019215061.5.16.31
- [8] “_____”, “*Stage of Acute Leukemia*”, Diakses 22 Mei 2008, dari University of Virginia Health System.
<http://www.healthsystem.virginia.edu/internet/hematology/HessIDB/home.cfm>
- [9] “_____”, “*Pengolahan citra*”. Diakses Mei 2008.
<http://www.google.co.id/gw+ln?u=http%3A%2Fstaffsite.gunadarma.ac.id%2Fsrini%2Findex.php>
- [10] Suksmono, “*Pendahuluan Pengenalan Pola*”. Diakses 28 November 2007.

<http://radar.ee.itb.ac.id/~suksmono/Lectures/ec4041/ppt/PP1.%20Pendahuluan%20Pengenalan%20Pola.pdf>

- [11] R. C. Gonzalez, R.E. Woods, *Digital Image Processing Second Edition* (New Jersey: Prentice Hall, 2002), hal. 154.
- [12] Appleton and Perera. *The Development and Practice of Electronic Music* (New Jersey: Prentice-Hall, 1975)
- [13] Mohamed Qasem, “*Vector Quantization*”. Diakses Desember 2007, dari Geocities
<http://www.geocities.com/mohamedqasem/vectorquantization/vq.html>
- [14] Patanne, Giuseppe, “*Enhanced LBG algorithm*”, Faculty of Engineering University of Catania, Italy, 1999.
- [15] Dr Philip Jackson. “*HMM Tutorial 1*”. Diakses 21 November 2007, dari University of Surrey Centre for Vision Speech & Signal Processing,
<http://www.ee.surrey.ac.uk/Teaching/Courses/eem.ssr/>
- [16] Lawrence R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition*. Proceeding of IEEE Vol.77 No. 2, February 1989.

DAFTAR PUSTAKA

Larry Waterbury, *Buku Saku Hematologi Edisi 3* (Jakarta: Penerbit Buku Kedokteran, 1995)

R. C. Gonzalez, R. E. Woods, *Digital Image Processing Second Edition* (New Jersey: Prentice Hall, 2002)

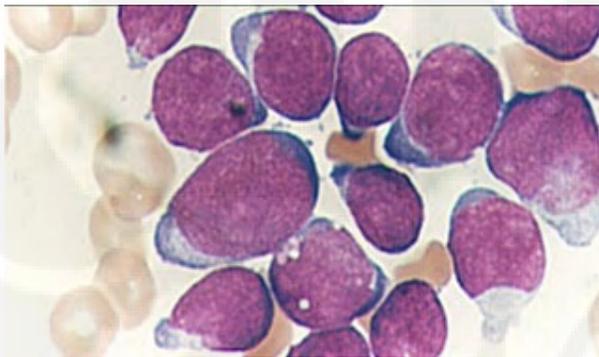
Robbins, et. Al, *Basic Pathology 7th Edition* (Pennsylvania: Saunders, 2003)

LAMPIRAN

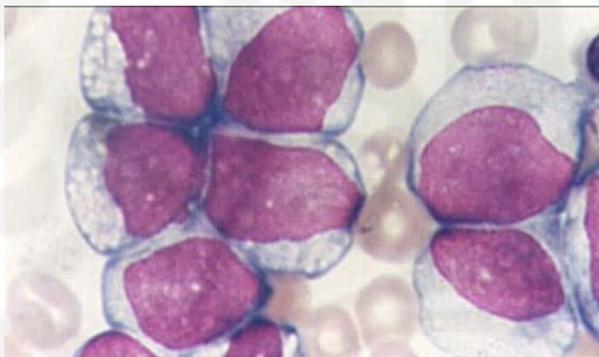
Lampiran 1 Citra Darah Leukemia Yang Digunakan

(Sumber: Atlas Hematology)

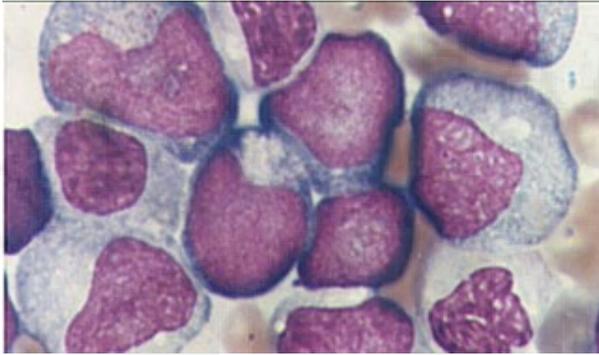
Citra Darah ALL L2



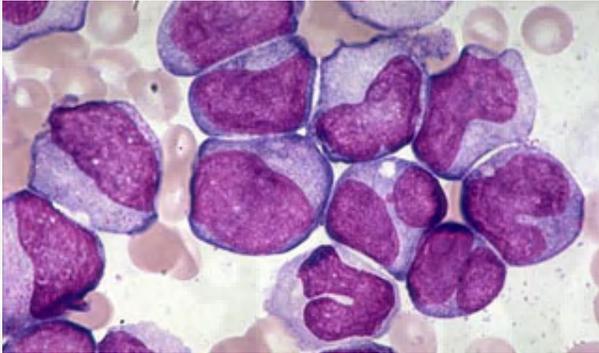
Citra Darah AML M1



Citra Darah AML M2



Citra Darah AML M4



Citra Darah AML M5

