

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

Judul Kegiatan : Pengenalan Aksara Batak Toba dengan Chain Code dan jaringan saraf tiruan Propagasi Balik.

Peneliti / Pelaksana

Nama Lengkap : DAURAT SINAGA
NIDN : 0616106902
Jabatan Fungsional :
Program Studi : Sistem Informasi
Nomor HP : 081228566601
Surel (e-mail) : parendeborngin@gmail.com

Anggota Peneliti (1)

Nama Lengkap : RICARDUS ANGGI PRAMUNENDAR
NIDN : 0613068601
Perguruan Tinggi : UNIVERSITAS DIAN NUSWANTORO

Institusi Mitra (jika ada)

Nama Institusi Mitra :
Alamat :
Penanggung Jawab :
Tahun Pelaksanaan : Tahun ke 1 dari rencana 1 tahun
Biaya Tahun Berjalan : Rp. 11.500.000,00
Biaya Keseluruhan : Rp. 15.000.000,00

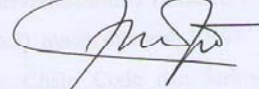
Mengetahui
Dekan

(Dr. Abdul Syukur, MM.)
NIP/NIK 0686.11.1992.017



Semarang, 11 - 12 - 2013,

Ketua Peneliti,



(DAURAT SINAGA)

NIP/NIK0686121998170

Menyetujui,
Ketua Lembaga Penelitian

(Tyas Catur Pramudi, S.si, M.Kom)
NIP/NIK 0686.11.1994.046



RINGKASAN

Mencintai budaya sendiri sebelum Negara/orang lain yang mencintai dan mengakui sebagai budayanya mereka, sangatlah perlu disampaikan kepada semua insan bangsa kita ini, terutama para pemuda dan generasi penerus Bangsa ini. Aksara Batak Toba atau orang Batak Toba sendiri menyebutnya Huruf Batak yang merupakan salah satu peninggalan budaya nenek moyang Bangsa Indonesia dan Batak Toba Khususnya, yang saat ini hampir punah, oleh karena semakin pudarnya rasacinta terhadap budaya sendiri atau mungkin karena kurangnya sarana yang dapat digunakan untuk membantu menterjemahkan Aksara Batak Toba ke Huruf Latin.

Penelitian pengenalan Aksara Batak Toba yang sudah ada saat ini adalah *Character Recognition of the Batak Toba Alphabet Using Signatures and Simplified Chain Code* , ini adalah penelitian yang pertama tentang pengenalan Aksara Batak Toba, penelitian ini menggunakan metode Signatures dan Simplified Chain Code. Dua bentuk dataset aksara Batak Toba yaitu aksara modern dan tradisional. Hasil akhir dari penelitian ini adalah waktu pengenalan hanya 0.06 detik dan akurasi yang bervariasi font-1 (modern) untuk ukuran 50 dan 60 lebih dari 90% sedangkan font-2 (tradisional), masih dibawah 80%.

Pengenalan Aksara Batak Toba dengan Chain Code dan Jaringan Saraf tiruan progasibalik adalah penelitian kedua yang mengambil objek Aksara Batak Toba. Bentuk Aksara Batak Toba yang menjadi objek kami adalah font-2 bentuk yang tradisional, dengan ukuran 30,40,50pt dan 60pt, dengan proses akuisisi citra digital lewat scanner dengan kualitas 150 dpi. Penggunaan Chain Code, untuk ekstraksi feature sebelum melakukan pelatihan dengan jaringan saraf tiruan sangat meningkatkan kecepatan pelatihan, tanpa Chain Code pelatihan dengan jaringan Saraf tiruan sungguhlah sangat lambat sekali, karena jumlah vektor yang sangat besar.

PRAKATA

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena dengan Rahmat dan karunianya kami dapat diselesaikan penelitian ini. Penelitian Pengenalan Karakter Batak Toba dengan Chain Code dan Jaringan Saraf Tiruan Propagasi Balik, adalah penelitian yang berobjek pada Budaya Bangsa Indonesia dan Budaya Batak Toba khususnya.

Karena kemajuan peradaban dan teknologi banyak budaya bangsa kita yang terlupakan, yang berdampak pada klaim sepihak orang (negara) lain sebagai budaya mereka. Harapan dan doa kami semoga dengan penelitian ini anak-anak negeri kita merasa terdorong untuk lebih mencintai budayanya sendiri.

Ucapan terimakasih yang sebesar-besarnya kepada Kepala Direktorat Jenderal Pendidikan Tinggi (Dikti) yang telah membiayai penelitian ini, kepada Rektor Universitas Dian Nuswantoro, kepada Kepala Lembaga Penelitian dan Pengabdian Masyarakat (LPPM), Universitas Dian Nuswantoro Semarang, yang telah memberikan dukungan dan Kepercayaan Kepada kami dalam melakukan penelitian ini dalam program Penelitian Dosen Pemula tahun 2013, serta kepada semua pihak yang secara langsung maupun tidak langsung memberikan dukungannya pada penelitian ini, doa dan hapan kami penelitian ini akan bermanfaat bagi kemajuan Budaya Bangsa Indonesia.

Semarang, 9 Desember 2013

Ketua Peneliti

DAFTAR ISI

HALAMAN PENGESAHAN	i
RINGKASAN	ii
PRAKATA.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	vii
DAFTAR LAMPIRAN.....	viii
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Analisis Masalah.....	2
1.3 Rumusan Masalah.....	4
1.3.1 Umum	4
1.3.2 Khusus.....	4
1.4 Tujuan Penelitian	4
1.4.1 Tujuan Umum	4
1.4.2 Tujuan Khusus	5
1.5 Manfaat Penelitian	5
1.5.1 Manfaat Umum	5
1.5.2 Manfaat Khusus	5
BAB II.....	6
2.1 Penelitian terkait	6
2.2 Pengenalan Karakter Tulisan tangan	7
2.3 Konsep dasar Pengenalan Pola	10
2.4 Pengolahan Citra.....	11
2.4.1 Citra Grayscale.....	13
2.4.2 Citra Threshold	13
2.4.3 Citra Biner.....	14
2.6 Algoritma Chain Code.....	16
2.7 Jaringan Saraf Tiruan.....	19
2.7.1 Faktor Bobot	20
2.7.2 Fungsi Aktivasi	20
2.8 Model Jaringan Syaraf Tiruan Propagasibalik.....	21

2.8.1	Algoritma Jaringan Syaraf Propagasibalik	22
2.8.2	Pembelajaran Terawasi (<i>Supervised Learning</i>)	25
2.8.3	Pembelajaran Tidak Terawasi (<i>UnSupervised Learning</i>)	25
2.8.4	Chain Code dan Jaringan Saraf Tiruan Propagasibalik	26
BAB III		27
3.1	Proses pengenalan Karakter	27
3.1.1	Akuisisi Citra Digital	27
3.1.2	Praproses	27
3.1.3	Segmentasi	27
3.1.4	Ekstraksi Fitur dengan Chain Code	27
3.1.5	Normalisasi	28
3.2	Proses Pelatihan dengan Jaringan Saraf Tiruan Propagasibalik	28
3.3	Proses Uji/Testing dengan Jaringan Saraf Tiruan Propagasibalik	28
BAB IV		29
4.1	Pengambilan Gambar	29
4.2	Praproses	29
4.3	Mengubah ukuran gambar	31
4.4	Mendeteksi tepi gambar (<i>Edge detection</i>) dan Chain Code	31
4.5	Proses Chain Code	32
4.6	Proses Normalisasi	37
4.7	Training dengan Jaringan Saraf Tiruan data tulisan tangan	38
4.8	Training dengan Jaringan Saraf Tiruan data Aksara cetak	40
4.9	Pelatihan Jaringan Saraf Tiruan	41
4.10	Tanpa Chain Code	41
4.11	Setelah Chain Code	43
4.12	Pengenalan Karakter	46
BAB V		47
5.1	Kesimpulan	47
5.2	Pekerjaan kedepan	48
LAMPIRAN		51

DAFTAR GAMBAR

Gambar 1 1 :Aksara Batak Toba Modern (Font-1).....	2
Gambar 1 2 : Aksara Batak Toba tradisional (Font-2).....	2
Gambar 1 3 : Akurasi Pengenalan Aksara Batak Toba, font-1 dengan SCC.....	3
Gambar 1 4 : Akurasi Pengenalan Aksara Batak Toba, font-2 dengan SCC.....	4
Gambar 2 1 : Contoh bentuk font karakter China.....	9
Gambar 2 2 : Sebagian karakter vocal Persia.....	9
Gambar 2 3 : Contoh sebagian Aksara Bali.....	9
Gambar 2 4 :Alur Konsep dasar pengenalan Pola.....	10
Gambar 2 5 : Warna Citra RGB.....	13
Gambar 2 6 : Ketetangaan Chain Code.....	16
Gambar 2 7 : Arah Algoritma Chain Code.....	16
Gambar 2 8 : 8 tetangga Chain Code.....	17
Gambar 2 9 : Arah Chain Code.....	17
Gambar 2 10 : Lokasi awal, P0 dan arah untuk mendapatkan kode rantai.....	18
Gambar 2 11 : Ilustrasi fungsi sigmoid biner dengan range (0,1).....	21
Gambar 2 12 : Ilustrasi fungsi sigmoid bipolar dengan range (-1,1).....	21
Gambar 2 13 : Arsitektur Jaringan Saraf Tiruan Propagasibalik.....	23
Gambar 4 1 : Karakter "A" dengan ukuran yang berbeda.....	29
Gambar 4 2 : Pixel Citra Berwarna.....	30
Gambar 4 3 : Citra keabuan (grayscale).....	30
Gambar 4 4 : Citra Binari.....	30
Gambar 4 5 : Proses perubahan ukuran citra.....	31
Gambar 4 6 : Hasil Deteksi tepi dengan Canny.....	32
Gambar 4 7 : Hasil proses Chain Code.....	36
Gambar 4 8 : Hasil Eliminasi angka frekuensi 1 (satu).....	37
Gambar 4 9 : Hasil akumulasi angka arah.....	37
Gambar 4 10 : Hasil Proses Normalisasi.....	37
Gambar 4 11 : Hasil frekuensi sesuai dengan jumlah N yg sudah ditentukan.....	38
Gambar 4 12 : Pelatihan dengan dataset tulisan tangan.....	39
Gambar 4 13 : Grafik MSE hasil pelatihan dataset tulisan tangan.....	40
Gambar 4 14 : Training data tanpa Chain Code.....	42
Gambar 4 15 : Training dengan menggunakan Chain Code.....	45
Gambar 4 16 : Mean Squared Error (MSE).....	45
Gambar 4 17 : Pengenalan Aksara "ma".....	46

DAFTAR TABEL

Table 1 : Contoh data dan target 25
Table 2 : Tabel pengenalan Aksara Batak Toba 48

DAFTAR LAMPIRAN

Lampiran 1 : Data training dengan Chain Code	51
Lampiran 2 : Data training tanpa Chain Code	60
Lampiran 3 : Dataset dengan variasi ukuran 6,8,10,12,18,20.....	69
Lampiran 4 : Dataset variasi ukuran 24,28,30,40	70
Lampiran 5 : Dataset variasi ukuran 50	71
Lampiran 6 : Dataset Vriasi Ukuran 60 pt.....	72
Lampiran 7 : Dataset variasi Tulisan tangan.....	73
Lampiran 8 : Dataset Variasi tulisan tangan sebagai testing.....	74

BAB I

PENDAHULUAN

1.1 Latar Belakang

Negara yang besar adalah Negara yang menghargai Budaya-nya sendiri, Indonesia adalah Negara yang kaya akan Budaya. Aksara Batak Toba adalah salah satu peninggalan Budaya yang hampir punah, kepunahan ini mungkin bisa disebabkan oleh semakin pudarnya rasacinta terhadap budaya sendiri atau tidak ada alat yang dapat digunakan untuk memudahkan memahami/menterjemahkan Aksara Batak Toba ke Aksara latin.

Aksara Batak Toba yang telah digunakan sebagai bahasa komunikasi dan korespondensi ratusan tahun lalu sampai sekarang masih ada peninggalan berupa, ramuan obat-obatan hingga ilmu perdukunan. Dari dalam Pustaka Batak (dokumen tulisan Batak Toba kuno), banyak yang dapat dipelajari dan digunakan untuk kebutuhan manusia jaman modern sekarang[2].

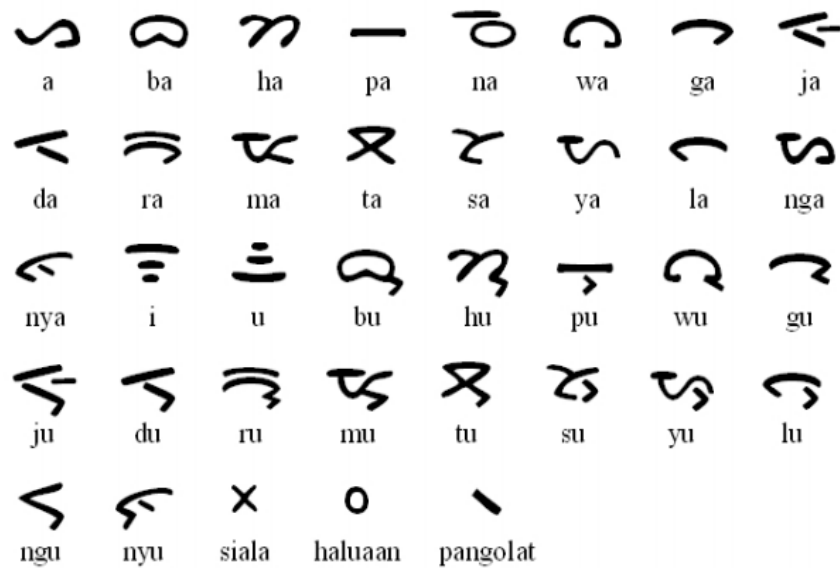
Penelitian pengenalan Aksara Batak Toba yang sudah ada saat ini adalah *Character Recognition of the Batak Toba Alphabet Using Signatures and Simplified Chain Code* [2] , dalam penelitian ini penulis menggunakan dua bentuk huruf (Font) Aksara Batak Toba, seperti pada (gambar 1.1 dan 1.2) yang menjadi dataset. Akurasi pengenalan Font-1, ukuran 60 pt dan 150 dpi, mencapai lebih dari 90 %, dengan Font-2 ukuran 60 pt dan 150 dpi, akurasi masih dibawah 80%, (seperti gambar 1.3 dan 1.4) . melihat keadaan ini penulis berharap dengan metode Chain Code (CC) dan Jaringan Saraf Tiruan (JST), maka akan ada peningkatan akurasi pengenalan Aksara Batak Toba baik Font-2, dengan ukuran Font serta akuisisi citra yang sama.

Dari tahun 1990, pengolahan citra dan tehnik pengenalan pola secara efisien dikombinasikan dengan metode-metode kecerdasan buatan (*Artificial Intillegence*). *An Overview of Character Recognition Focused on Off-Line Hanwriting* Metode yang umum digunakan dalam pengenalan Karakter dengan akurasi diatas 85 % adalah Jaringan Saraf Tiruan (*Neural Network*) algoritma Propagasibalik dan

Hidden Markov Model (HMM). Dalam penelitian ini penulis menggunakan Chain Code dan Jaringan Saraf Tiruan (JST) Propagasibalik. Gambar 1.1 dan 1.2 adalah bentuk Modern (font-1) dan tradisional (font-2).



Gambar 1 1 :Aksara Batak Toba Modern (Font-1)

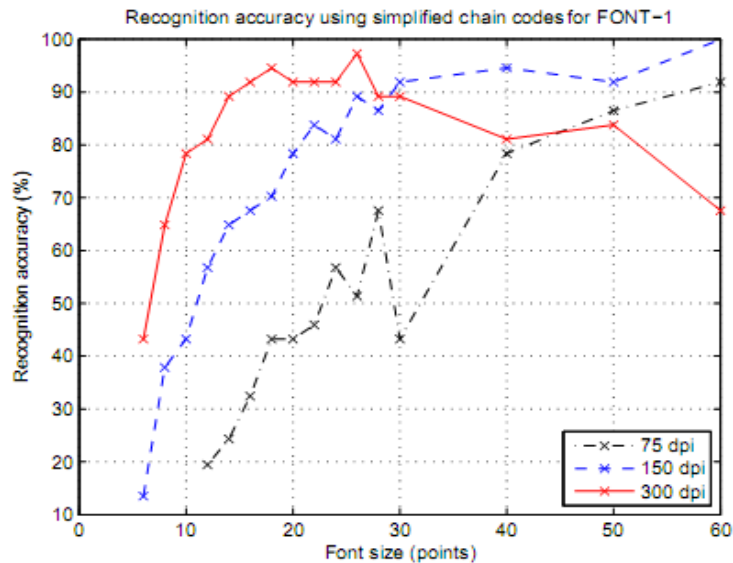


Gambar 1 2 : Aksara Batak Toba tradisional (Font-2)

1.2 Analisis Masalah

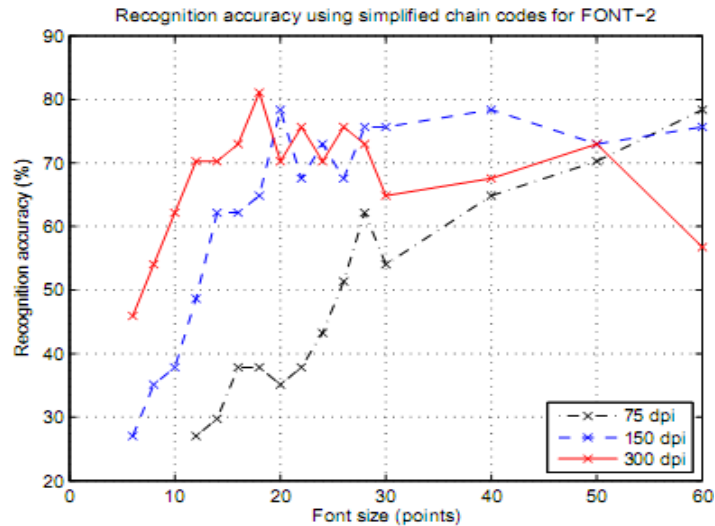
Beberapa analisis yang kami simpulkan dalam penelitian tentang pengenalan Aksara Batak Toba[2], adalahsebagai berikut :

1. Pengenalan Aksara Batak Toba dengan Signatures dan Simplified Chain Code adalah penelitian pertama pengenalan Aksara Batak Toba.
2. Dalam penelitian tersebut terdapat perbandingan dua metode yaitu Signature dengan Simplified Chain Code (SCC).
3. Dataset yang digunakan adalah Citra digital dengan tulisan hitam dan latar belakang putih. Karakter tersebut di pindahkan dengan menggunakan scanner 75,150,300 dpi (*dot per inci*), dengan ukuran 10,20,30,40,50, 60 pt.
4. Akurasi pengenalan Aksara Batak Toba untuk font-1 dengan SCC ukuran 30,40,50 dan 60pt sudah $\geq 90\%$ dengan kualitas gambar 150 dpi, sedangkan 300 dpi untuk 40, 50 dan 60pt masih dibawah 90 %. Seperti gambar dibawah ini :



Gambar 1 3 : Akurasi Pengenalan Aksara Batak Toba, font-1 dengan SCC

5. Akurasi pengenalan Aksara Batak Toba untuk FONT-2 dengan SCC untuk ukuran yang sama 30,40,50 dan 60pt masih kurang dari 80 % dengan kualitas gambar 150 dpi, sedangkan 300 dpi dibawah 80 % dan yang menarik adalah pada ukuran 18pt lebih dari 80 % sedangkan 24 dan 28pt dengan kualitas gambar 300 dpi terlihat lebih baik dari kualitas gambar 150 dpi. seperti yang ditampilkan dalam gambar dibawah ini :



Gambar 1 4 : Akurasi Pengenalan Aksara Batak Toba, font-2 dengan SCC

6. Dari analisis kami diatas, maka dalam penulisan ini kami menggunakan gabungan dua Metode yaitu Chain Code dan Jaringan Saraf Tiruan Propagasibalik, dengan data set yang sama untuk meningkatkan akurasi pengenalan dengan ukuran dan kualitas gambar yang sama.

1.3 Rumusan Masalah

1.3.1 Umum

Aksara Batak Toba adalah salah satu peninggalan warisan Budaya Batak yang hampir punah,

1.3.2 Khusus

Akurasi pengenalan Aksara Batak Toba untuk FONT-2 dengan SCC untuk ukuran yang sama 30,40,50 dan 60pt masih kurang dari 80 % dengan kualitas gambar 150 dpi, sedangkan 300 dpi dibawah 80 % dan yang menarik adalah pada ukuran 18pt lebih dari 80 % sedangkan 24 dan 28pt dengan kualitas gambar 300 dpi terlihat lebih baik dari kualitas gambar 150 dpi.

1.4 Tujuan Penelitian

1.4.1 Tujuan Umum

Dengan adanya penelitian ini semoga menjadi masukan bagi bangsa kita khususnya orang Batak Toba untuk menjaga dan melestarikan Budayanya.

1.4.2 Tujuan Khusus

Meningkatkan akurasi pengenalan Aksara Batak Toba font-2 ukuran 30,40,50 dan 60pt dengan kualitas gambar 150 dpi dan 300 dpi dengan Chain Code(CC) dan Jaringan Saraf Tiruan (JST) Propagasibalik.

1.5 Manfaat Penelitian

1.5.1 Manfaat Umum

Manfaat umum pada penelitian ini adalah penerapan Chain Code (CC) untuk mempercepat waktu proses pelatihan dan Jaringan Saraf Tiruan (JST) .

1.5.2 Manfaat Khusus

Penelitian ini diharapkan dapat membantu menterjemahkan Citra Aksara Batak Toba Font-2, ukuran 30,40,50 dan 60pt dengan kualitas gambar 150 dpi .

BAB II TINJAUAN PUSTAKA

2.1 Penelitian terkait

Penelitian tentang pengenalan Aksara Batak Toba yang sudah pernah dilakukan adalah *Character Recognition of the Batak Toba Alphabet Using Signatures and Simplified Chain Code* [1], Penelitian ini merupakan penelitian pertama tentang Pengenalan Aksara Batak Toba. Dalam penelitian ini penulis menggunakan dua Metode yaitu Signatures dan Simplified Chain Code.

Tujuan dari penelitian ini adalah untuk membangun tehnik pengenalan didasarkan pada ide ini untuk menggambarkan dan mengenali Aksara Batak Toba dengan skala bervariasi. Selain Metode *Signatures*, penulis juga mengusulkan dan menggunakan *Simplified Chain Code (SCC)* untuk pengenalan Aksara Batak Toba. Meskipun *Signatures* dan *Chain Code* telah umum digunakan untuk pengenalan karakter, dalam penelitian ini diperkenalkan konsep-konsep yang agak baru dan praktek dalam mengaplikasikan kedua metode.

Percobaan yang dilakukan dalam penelitian ini memprioritaskan kegiatan yang paling sering terjadi dalam kehidupan nyata.

- 1) Ukuran dari karakter input font-2 adalah 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 40, 50, dan 60 poin , tanpa berani, gaya italic atau lainnya. Karena karakter bergaya dan tulisan tangan yang dikecualikan, bentuk karakter dipertahankan terlepas dari ukuran.
- 2) Karakter warna hitam dengan latar belakang putih dan dicetak di atas kertas putih oleh off-the-rak printer elektronik. Print out dipindakan lewat scanner dengan 75, 150, dan 300 dot-per-inch (dpi) menghasilkan gambar resolusi warna terkompresi sebagai masukan untuk kedua tehnik.
- 3) Teks diketik dari kiri ke kanan dengan rata kiri.

Implementasi dan percobaan pada Aksara Batak Toba telah menunjukkan bahwa SCC dapat mencapai akurasi yang sebanding dengan Signatures. SCC bahkan bisa mendapatkan akurasi pengenalan 90% untuk karakter ukuran 12 poin dengan

parameter yang tepat. pada perbandingan waktu run-time untuk mengenali karakter, SCC lebih unggul daripada Signatures lebih dari sepuluh kali lipat.

SCC Berbeda dengan Signatures, waktu yang dibutuhkan untuk pengenalan Karakter 'a' sampai 1.2 detik, sementara hanya 0,06 detik yang diperlukan oleh SCC untuk mengenalinya, memberikan peningkatan 20 kali lipat yang luar biasa. Selain itu, bentuk yang sama yang diproduksi oleh nearestneighbor, metode interpolasi bilinear dan bicubic menunjukkan ukuran yang dapat membantu pengenalan karakter ukuran font yang kecil jika pemindaian tidak tersedia. Pada 600 dpi set oleh scanning atau mengubah ukuran dari Citra 300-dpi, keduanya tampil sama menggunakan SCC dengan akurasi 80% atau lebih tinggi untuk karakter ukuran <12 point dengan penambahan hanya kecil untuk menjalankan-waktu.

Dalam penelitian PENGENALAN AKSARA BATAK TOBA DENGAN CHAIN CODE DAN JARINGAN SARAF TIRUAN PROPAGASI BALIK, ini kami mencoba menggabungkan dua metode sebagai alat komputasi yang setiap metode ditugaskan berbeda guna mendapatkan akurasi yang lebih baik dari penelitian sebelumnya khususnya font-2. Metode yang kami gunakan adalah Chain Code dan Jaringan Saraf Tiruan Propagasibalik, kedua metode ini merupakan metode yang akrab bagi para peneliti khususnya penelitian dibidang pengolahan citra digital. Kombinasi dari kedua metode diatas akan di perlihatkan dalam bab berikutnya dengan data yang sama dengan penelitian sebelumnya, jumlah dataset dan sumberdata diambil dari hasil scanning lewat media scanner dengan ketentuan-ketentuan yang sama dengan penelitian sebelumnya, agar dapat dilihat jelas kekurangan, kelemahan serta kekuatan dari metode yang kami gunakan.

Tujuan penelitian ini sebagaimana yang sudah disampaikan di bab sebelumnya, menunjukkan bahwa dengan mengkombinasikan antara Chain Code dan Jaringan Saraf Tiruan Propagasibalik, pelatihan akan lebih cepat dan lebih akurat dalam pengenalan Aksara Batak Toba. Masa depan yang kami harapkan untuk peneliti-peneliti yang berbasis pengolahan Citra Digital, tehnik dan kombinasi kedua metode yang kami gunakan dapat dijadikan sebagai pengetahuan dan perbandingan metode.

2.2 Pengenalan Karakter Tulisan tangan

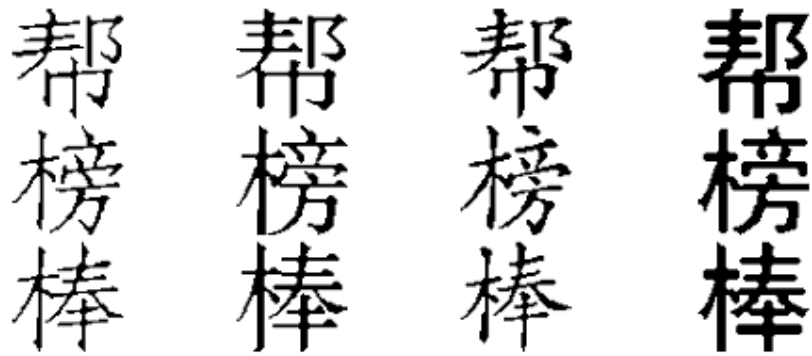
Penelitian yang membahas tentang pengenalan karakter tulisan tangan telah banyak dilkakukan, sejak tahun 1900-1980, sejarah pengenalan karakter dikenal,

ketika Ilmuan Rusia mencoba berusaha membangun sebuah alat bantu untuk rintangan penglihatan[11]. Pengenalan Karakter pertama pada pertengahan tahun 1940'an dilakukan seiring dengan perkembangan teknologi komputer digital.

Indonesia terdiri dari berbagai suku, setiap suku memiliki aksara tradisional yang pada jaman dahulu digunakan sebagai sarana komunikasi dan korespondensi, sekarang aksara peninggalan nenek moyang kita itu hampir punah karena sudah jarang digunakan. Beberapa aksara tradisional daerah , seperti Aksara Sunda, Jawa, Bali dan Batak (Toba, Mandailing, Karo, Simalungun, Pakpak) . Penelitian ini mengambil objek Aksara Batak Toba, sebagaimana yang telah dibahas pada bab pertama. Dalam penelitian ini penulis menggunakan dua metode yang masing-masing memiliki tugas sendiri-sendiri, adapun metode yang digunakan adalah Chain Code sebagai Metode Ekstraksi Ciri dan Jaringan Saraf Tiruan Propagasibalik sebagai Metode pengenalan karakter. Kedua metode ini akan penulis uraikan sesuai dengan fungsi masing-masing dalam penelitian ini.

Penelitian sebelumnya tentang Pengenalan Aksara Batak Toba adalah *Character Recognition of the Batak Toba alphabet using Sinatures and Simplifid Chain Code (SCC)*, dalam penelitian ini pengenalan Aksara Batak Toba menggunakan dua dataset dan ukuran font yang berbeda-beda seperti yang telah di singgung pada bab pertama. Kombinasi antara Chain Code dan Jaringan Saraf Tiruan Propagasibalik untuk Mempercepat pelatihan dan Pengenalan serta akurasi lebih baik dari penelitian sebelumnya adalah tujuan utama dari penelitian ini.

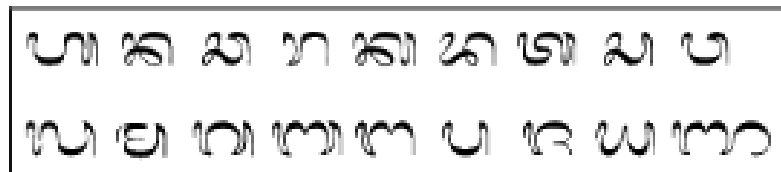
Proses Pengenalan Aksara Batak Toba dengan Chain Code dan Jaringan Saraf Tiruan Propagasibalik mempercepat pelatihan dan pengenalan serta akurasi pengenalanya. Dengan dua bentuk font sebagai data training dan sekaligus sebagai data testing. Berikut ini adalah penelitian tentang pengenalan Karakter tulisan tangan dari beberapa daerah dan negara, Penelitian yang sudah adalah sbb:



Gambar 2 1 : Contoh bentuk font karakter China

ت	پ	ب	آ و ا
'te'	'pe'	'be'	'alef'
ح	چ	ج	ث
'he'	'che'	'jim'	'ce'
ر	ذ	د	خ
're'	'zal'	'dal'	'khe'
ش	س	ژ	ز
'sheen'	'seen'	'jhe'	'ze'

Gambar 2 2 : Sebagian karakter vocal Persia



Gambar 2 3 : Contoh sebagian Aksara Bali

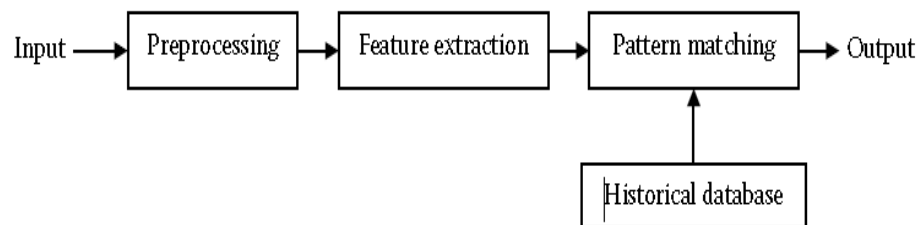
Dan masih banyak lagi penelitian tentang pengenalan Karakter tulisan tangan yang belum penulis sampaikan dalam penelitian ini.

2.3 Konsep dasar Pengenalan Pola

Metode Jaringan Saraf merupakan salah satu metode yang paling umum untuk Pengenalan pola. Pengenalan pola dapat defenisikan sebagai disiplin scientific yang tujuannya adalah klassifikasi benda (atau pola) menjadi beberapa kategori atau kelas. Tergantung pada masalah, obyek mungkin gambar atau sinyal gelombang atau jenis pengukuran yang perlu klasifikasikan [8].

Pencocokan pola umumnya menggunakan *Artificial Intelligence (AI)* dan *soft-computing* tehnik untuk tujuan pencocokan dua set pola. Tehnik ini memastikan respon yang tepat waktu oleh sistem, tidak seperti tehnik lain yang sering komputasi mahal dan karenanya lebih lambat. Dalam sebagian besar sistem kehidupan nyata kita bahas, soft-computing tehnik yang diterapkan untuk mengekstrak aturan dari data yang tersedia, maka aturan-aturan yang digunakan untuk tujuan pengakuan. Ini biasanya lebih baik daripada pencocokan input dengan setiap bagian dan setiap data yang tersedia dalam database.

Pencocokan pola tehnik tradisional melibatkan metode statistik di mana alat statistik yang digunakan untuk menentukan sejauh mana dua pola cocok. Pendekatan-pendekatan statistik cenderung sangat komputasi mahal dan karenanya tidak umum digunakan jika ada template banyak tersedia untuk setiap jenis pola dalam database.



Gambar 2 4 :Alur Konsep dasar pengenalan Pola

Struktur umum dari sistem pencocokan pola diberikan pada Gambar 2.3.1, yang menunjukkan semua langkah biasanya diikuti dengan sistem pengenalan. Pertama, preproces sistem masukan untuk menghilangkan suara apapun. Fiatur tersebut kemudian diekstraksi. Fitur-fitur diekstraksi kemudian dicocokkan dengan database dengan tehnik pencocokan pola. Hasil pencocokan menunjukkan output dari sistem[8].

2.4 Pengolahan Citra

Istilah citra atau image yang pada umumnya digunakan dalam bidang pengolahan citra diartikan sebagai suatu fungsi kontinu dari intensitas cahaya $f(x,y)$ dalam bidang dua dimensi dengan (x,y) menyatukan suatu koordinat dengan nilai, f pada setiap titik menyatukan intensitas atau tingkatan kecerahan atau derajat keabuan (brightness/gray level). Suatu citra digital adalah suatu citra kontinu yang diubah kedalam bentuk diskrit, baik koordinat maupun intensitas cahayanya. Kita dapat menganggap suatu citra digital sebagai suatu matriks, dimana indeks baris dan kolomnya menyatakan koordinat sebuah titik pada citra tersebut dan nilai masing-masing elemennya menyatakan intensitas cahaya pada titik tersebut.

Suatu titik pada sebuah citra digital sering disebut sebagai elemen citra (image-elemen), elemen gambar (picture-elemen), piksel (pixel). Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer menjadi citra yang kualitasnya lebih baik.

Operasi-operasi yang dilakukan didalam pengolahan citra banyak macamnya, namun secara umum operasi pengolahan citra dapat diklasifikasikan dalam beberapa yaitu:

1. Peningkatan kualitas citra (image enhancement)

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

- a. Perbaikan kontras gelap/terang
- b. Perbaikan tepian objek (edge enhancement)
- c. Penajaman (sharpening)
- d. Penapisan derau (noise filtering)

2. Perbaikan citra (image restoration)

Operasi ini bertujuan menghilangkan / meminimumkan cacat pada citra.

Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra, bedanya pada pemugaran citra penyebab degredasi gambar diketahui:

- a. Penghilang kesamaran (debluring)
- b. Penghilang derau (noise)
- c. Pelembutan citra (smoothing)

3. Pemampatan citra (image compression)

Jenis operasi ini agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Ukuran citra 258 kb (kilo byte) dapat direduksi menjadi 49 kb.

4. Segmentasi Citra (image analysis)

Jenis operasi ini bertujuan untuk memecah suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu. Jenis ini berkaitan erat dengan pengenalan pola

5. Analisis citra (image analysis)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Tehnik analisis citra mengakses ciri-ciri tertentu yang membantu dalam mengidentifikasi objek. Proses segmentasi kadang kala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh-contoh operasi analisis citra yaitu:

- a. Pendeteksian objek (edge detection)
- b. Ekstraksi batas (boundary)
- c. Representasi area wilayah (region)

6. Rekonstruksi citra (image reconstruction) menggunakan komputer menjadi citra yang kualitasnya lebih baik.

Citra dalam komputer tidak lebih dari sekumpulan sejumlah piksel dimana setiap triplet terdiri atas variasi tingkat keterangan (brightness) dari elemen red, green dan blue. Representasinya dalam citra, triplet akan terdiri dari 3 angka yang mengatur intensitas dari Red (R), Green (G) dan Blue (B) dari suatu triplet. Setiap triplet akan merepresentasikan 1 piksel (picture element). Suatu triplet dengan nilai 67, 228 dan 180 berarti akan mengeset nilai R ke nilai 67, G ke nilai 228 dan B ke nilai 180. Angka-angka RGB ini yang seringkali disebut dengan color values. Pada format .bmp citra setiap piksel pada citra direpresentasikan dengan dengan 24 bit, 8 bit untuk R, 8 bit untuk G dan 8 bit untuk B, dengan pengaturan seperti pada gambar dibawah ini[22].



Gambar 2 5 : Warna Citra RGB

2.4.1 Citra Grayscale

Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra grayscale berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra grayscale seringkali merupakan perhitungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik single band.

Citra grayscale disimpan dalam format 8 bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra grayscale dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B sehingga dapat dituliskan menjadi:

$$X = (R + G + B) / 3$$

$$\text{Warna} = \text{RGB}(X, X, X)$$

2.4.2 Citra Threshold

Citra Threshold Dilakukan dengan mempertegas citra dengan cara mengubah citra hasil yang memiliki derajat keabuan 256 (8bit), menjadi hanya dua buah yaitu hitam dan putih. Hal yang perlu diperhatikan pada Proses Thresholding adalah memilih sebuah nilai threshold dimana piksel yang bernilai dibawah nilai threshold akan diset menjadi hitam dan piksel yang bernilai diatas nilai threshold akan diset menjadi putih. Misalkan ditetapkan suatu nilai batas / ambang, sebesar 128 dimana elemen-elemen (piksel) pada citra batas nilainya lebih kecil dari pada nilai batas tersebut 'menyala', dan elemen-elemen lainnya dianggap 'dimatikan', dan keduanya diubah nilainya sesuai statusnya.

2.4.3 Citra Biner

Citra biner adalah citra dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas, biasanya bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (background) dan menyatakan warna tinta/objek (foreground) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra Threshold sebelumnya yakni:

Jika Citra RGB(0, 0, 0) maka X=1, Jika tidak maka X=0

2.5 Operasi Morfologi

Berdasarkan etimologi bahasa, Morphologi berasal dari kosakata bahasa Jerman yaitu *morphologie* yang terdiri dari *morph+logie/logy*. *Morph* adalah suatu kerja yang berupa singkatan dari *metamorphose* yang artinya perubahan bentuk atau karakter akibat perubahan pada struktur atau komposisi (transformasi). Sesuai dengan arti dasarnya, konsep Morphological Operation pada pengolahan citra adalah operasi-operasi perubahan bentuk pada binary image. Berbagai jenis Morphological Operation antara lain: Erosion, dilation, opening, closing, thinning, dan thickness. Saat ini yang dibahas adalah thinning[22]

1. Dilasi

Dilasi adalah operasi morfologi yang akan menambahkan pixel pada batas antar objek dalam suatu citra digital. Atau secara rinci Dilasi merupakan suatu proses menambahkan piksel pada batasan dari objek dalam suatu image sehingga nantinya apabila dilakukan operasi ini maka image hasilnya lebih besar ukurannya dibandingkan dengan image aslinya.

2. Erosi

Erosi merupakan kebalikkan dari Dilasi. Proses ini akan membuat ukuran sebuah citra menjadi lebih kecil. Berbeda dengan dilatasi, apabila erosi dilakukan maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari structuring element yang digunakan untuk memproses image tersebut.

3. Opening

Opening merupakan kombinasi proses dimana suatu citra digital dikenai operasi erosi dilanjutkan dengan dilasi. Operasi opening pada citra mempunyai efek

memperhalus batas-batas objek, memisahkan objek-objek yang sebelumnya bergandengan, dan menghilangkan objek-objek yang lebih kecil daripada ukuran structuring

4. Closing

Closing merupakan kombinasi dimana suatu citra dikenai operasi dilatasi dilanjutkan dengan erosi. Operasi closing juga cenderung akan memperhalus objek pada citra, namun dengan cara menyambung pecahan-pecahan (fuses narrow breaks and thin gulf) dan menghilangkan lubang-lubang kecil pada objek.

5. Thining

Thinning merupakan suatu proses penting sebelum melakukan proses-proses atau operasi-operasi pengolahan citra, seperti dalam proses pengenalan karakter optic, pengenalan sidik jari, pemrosesan teks, dan lain sebagainya. Tujuannya adalah mengurangi bagian yang tidak perlu (redundant) sehingga dihasilkan informasi yang esensial saja.

6. Shrinking

Shrinking merupakan erosi yang dimodifikasi sehingga piksel single tidak boleh dihapus. Hal ini berguna jika jumlah objek tidak boleh berubah.

7. Pruning

Pruning adalah transformasi yang menghilangkan endpoint dari citra yang telah mengalami proses skeletoning dan memproses sampai stabilitas tercapai. Proses Pruning merupakan proses pemangkasan cabang (branches) yang tidak diperlukan. Cabang yang tidak diperlukan biasanya muncul sebagai hasil dari Morphological Skeleton.

8. Thickening

Shrinking merupakan erosi yang dimodifikasi sehingga piksel single tidak boleh dihapus. Hal ini berguna jika jumlah objek tidak boleh berubah.

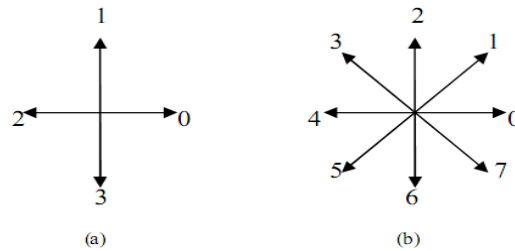
9. Skeletonizing

Skeleton adalah kerangka (atau sumbu medial) yang merepresentasikan sebuah bentuk atau citra biner, dihitung dengan menggunakan operator morfologi. Proses skeletoning dapat didefinisikan sebagai gabungan dari erosi dan opening.

2.6 Algoritma Chain Code

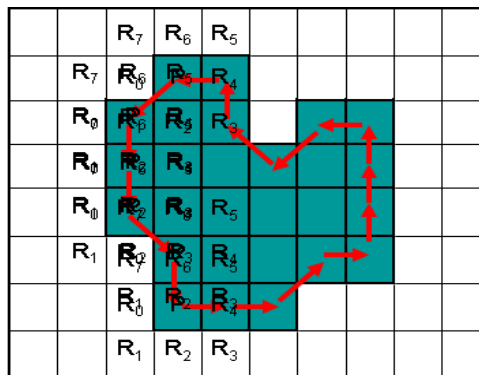
Pendekatan pertama untuk mewakili suatu gambar dengan menggunakan kode rantai diperkenalkan oleh Freeman pada tahun 1961 yang dikenal sebagai Freeman Chain Code (FCC). Kode ini mengikuti kontur dengan cara searah jarum jam counter dan melacak arah seperti yang kita pergi dari satu pixel kontur ke yang berikutnya. Kode melibatkan 4-terhubung dan 8 - jalan terhubung. Gambar 1 (a) menunjukkan 4-terhubung dan Gambar 1 (b) menunjukkan 8-terhubung dari FCC.

In this way a chain code $[d_i]$ is created, where d_i is the coding number of the direction of the line segment that connects boundary pixel (x_i, y_i) with the next one (x_{i+1}, y_{i+1}) , sweeping the boundary in, say, the clockwise sense. A disadvantage



Gambar 2 6 : Ketetapan Chain Code

dalam kelipatan 45° bahwa kita harus bergerak untuk pergi dari satu pixel kontur ke yang berikutnya. Gambar 2.7 menunjukkan contoh Kode Rantai Freeman menggunakan 8-terhubung jalan.



Gambar 2 7 : Arah Algoritma Chain Code

```

Algoritma
i=0
While (Ri==0) { i++ }
Move P to Ri
Set i=6 for next search

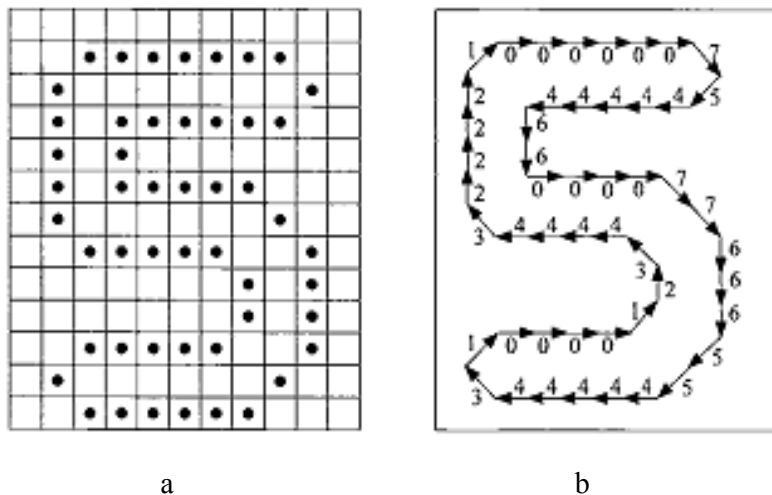
```

Mengidentifikasi P pixel yang dimiliki oleh "objek" kelas dan piksel tetangga (4 konektivitas tetangga) R₀ yang dimiliki oleh "background" class. Asumsikan bahwa pixel memiliki nilai "1" jika itu milik "objek" kelas dan "0" jika itu milik "latar belakang" class. Menetapkan 8-konektivitas tetangga P ke R₀, R₁, ..., R₇ sebagai berikut:

R7	R6	R5
R0	P	R4
R1	R2	R3

Gambar 2 8 : Delapan tetangga Chain Code

Algoritma delapan arah menganggab kode 1 yang berdekatan dengan yang lain. Setiap 1 kode dibandingkan dengan delapan arah lainnya. Gambar 2.6.4 menunjukkan delapan konektivitas arah yang digunakan dalam penelitian ini.



Gambar 2 9 : Arah Chain Code delapan arah

Gambar 2.9: Karakter "5" (a) versi resampled pada grid kasar, dan (b) kode rantai yang dihasilkan.

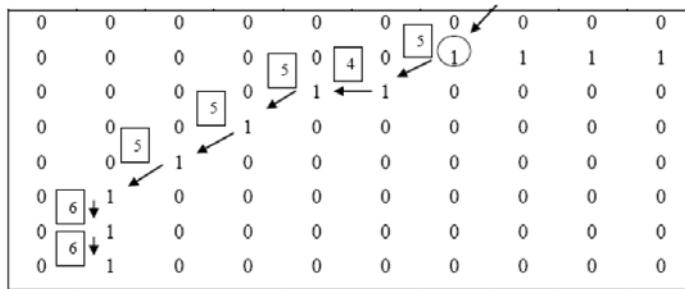
Uraian ini adalah bahwa kode rantai yang dihasilkan biasanya panjang dan pada saat yang sama sangat sensitif jika ada terdapat noise. Hal ini menyebabkan kode rantai dengan variasi karena kebisingan dan tidak selalu untuk kurva batas. Sebuah cara keluar adalah untuk sample kurva batas dengan memilih kotak dimensi yang lebih besar. Untuk setiap kotak grid semua titik di dalam kotak ditugaskan nilai dari masing-masing boxcenter.

Algoritma Chain Code yang digunakan dalam penelitian ini adalah :

- 1 Temukan pixel dalam objek yang memiliki nilai paling kiri dibaris paling atas, menyebutnya **P0** pixel.
- 2 Tentukan dir variabel (untuk arah), dan set ke setara dengan 7 (karena P0 adalah pixel kiri atas dalam obyek, arah ke piksel berikutnya harus 7).
- 3 Melintasi lingkungan 3x3 dari pixel saat ini dalam arah berlawanan arah jarum jam, dimulai pencarian di pixel arah $dir + 7 \pmod{8}$ jika dir atau bahkan $dir + 6 \pmod{8}$ jika dir aneh. Kehendak ini menetapkan arah arus ke arah pertama berlawanan arah jarum jam dari dir:

<i>dir</i>	0	1	2	3	4	5	6	7
$dir + 7 \pmod{8}$	7	0	1	2	3	4	5	6
$dir + 6 \pmod{8}$	6	7	0	1	2	3	4	5

- 4 Pixel latar depan yang pertama akan menjadi elemen batas baru. Memperbarui dir.
- 5 Berhenti ketika batas saat elemen Pn adalah sama dengan P1 elemen kedua dan pixel batas sebelumnya Pn-1 sama dengan elemen batas pertama P0.
- 6 Gambar. 2.6.5 di bawah ini menunjukkan lokasi awal (P0) dan arah diperoleh dengan menggunakan algoritma di atas .



Gambar 2 10 : Lokasi awal, P0 dan arah untuk mendapatkan kode rantai

2.7 Jaringan Saraf Tiruan

Implementasi jaringan saraf Propagasibalik dalam pengenalan pola sudah sering digunakan, berdasarkan hasil review yang dilakukan dari tahun 1990, pengolahan citra dan tehnik pengenalan pola secara efisien dikombinasikan dengan metode-metode kecerdasan buatan (*Artificial Intillegence*). *An Overview of Character Recognition Focused on Off-Line Handwriting* Metode yang umum digunakan dalam pengenalan Karakter dengan akurasi diatas 85 % adalah Neural Network (Jaringan Saraf) algoritma Propagasibalik dan Hidden Marcop Model (HMM). Dalam peneletian ini penulis menggunakan Chain Code dan Jaringan Saraf Tiruan (JST) Propagasibalik.

Jaringan syaraf tiruan terdiri dari beberapa neuron dan ada hubungan antar neuron- neuron seperti pada otak manusia. Neuron/selsaraf adalah sebuah unit pemroses informasi yang merupakan dasar operasi jaringan syaraf tiruan.

Jaringan syaraf tiruan terdiri atas beberapa elemen penghitung tak linier yang masing-masing dihubungkan melalui suatu pembobot dan tersusun secara paralel. Pembobot inilah yang nantinya akan berubah (beradaptasi) selama proses pelatihan.

Pelatihan perlu dilakukan pada suatu jaringan syaraf tiruan sebelum digunakan untuk menyelesaikan masalah. Hasil pelatihan jaringan syaraf tiruan dapat diperoleh tanggapa yang benar (yang diinginkan) terhadap masukan yang diberikan. Jaringan syaraf tiruan dapat memberikan tanggapan yang benar walaupun masukan yang diberikan terkena derau atau berubah oleh suatu keadaan. (Hermawan, 2006).

Penyelesaian masalah dengan jaringan syaraf tiruan tidak memerlukan pemrograman. Jaringan syaraf tiruan menyelesaikan masalah melalui proses belajar dari contoh-contoh pelatihan yang diberikan. Biasanya pada jaringan syaraf tiruan diberikan sebuah himpunan pola pelatihan yang terdiri dari sekumpulan contoh pola. Proses belajar jaringan syaraf tiruan berasal dari serangkaian contoh-contoh pola yang diberikan.metode pelatihan yang sering dipakai adalah metode belajar terbimbing. Selama proses belajar itu pola masukan disajikan bersama-sama dengan pola keluaran yang diinginkan. Jaringan akan menyesuaikan nilai bobotnya sebagai tanggapan atas pola masukan dan sasaran yang disajikan tersebut. (Hermawan, 2006)

2.7.1 Faktor Bobot

Bobot merupakan suatu nilai yang mendefinisikan tingkat atau kepentingan hubungan antara suatu node dengan node yang lain. Semakin besar bobot suatu hubungan menandakan semakin pentingnya hubungan kedua node tersebut.

Bobot merupakan suatu hubungan berupa bilangan real maupun integer, tergantung dari jenis permasalahan dan model yang digunakan. Bobot-bobot tersebut bisa ditentukan untuk berada didalam interval tertentu. selama proses pelatihan, bobot tersebut dapat menyesuaikan dengan pola-pola input.

Jaringan dengan sendirinya akan memperbaiki diri terus-menerus karena adanya kemampuan untuk belajar. Setiap ada suatu masalah baru, jaringan dapat belajar dari masalah baru tersebut, yaitu dengan mengatur kembali nilai bobot untuk menyesuaikan karakter nilai[16].

2.7.2 Fungsi Aktivasi

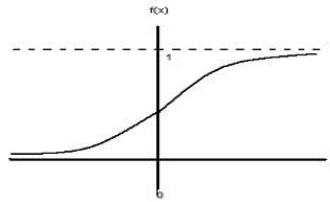
Setiap neuron mempunyai keadaan internal yang disebut level aktivasi atau level aktivitas yang merupakan fungsi input yang diterima. Secara tipikal suatu neuron mengirimkan aktivitasnya kebeberapa neuron lain sebagai sinyal. Yang perlu diperhatikan adalah bahwa neuron hanya dapat mengirimkan satu sinyal sesaat, walaupun sinyal tersebut dapat dipancarkan ke beberapa neuron yang lain.

Ada beberapa pilihan fungsi aktivasi yang digunakan dalam metode *Propagasibalik*, seperti fungsi sigmoid biner, dan sigmoid bipolar. Karakteristik yang harus dimiliki fungsi fungsi aktivasi tersebut adalah kontinue, diferensiabel, dan tidak menurun secara monoton. Fungsi aktivasi diharapkan dapat mendekati nilai-nilai maksimum dan minimum secara baik. Berikut ini adalah fungsi aktivasi yang sering digunakan yaitu :

1. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *Propagasibalik*. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Definisi fungsi sigmoid biner adalah segai berikut:

$f_1(x) = 1/(1+ e^{-x})$ dengan turunan $f_1'(x) = f_1(x) (1- f_1(x))$
berikut ini ilustrasi fungsi sigmoid biner:



Gambar 2 11 : Ilustrasi fungsi sigmoid biner dengan range (0,1)

2. Fungsi Sigmoid Bipolar

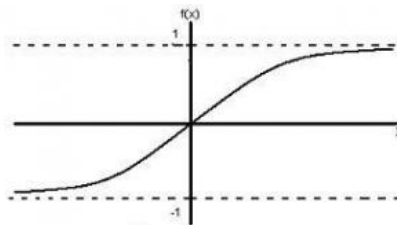
Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1. Definisi fungsi sigmoid bipolar adalah sebagai berikut:

$$f_2(x) = 2 f_1(x) - 1$$

dengan turunan

$$f_2'(x) = \frac{1}{2} (1+ f_2(x)) (1- f_2(x))$$

Berikut ini adalah ilustrasi fungsi sigmoid bipolar:



Gambar 2 12 : Ilustrasi fungsi sigmoid bipolar dengan range (-1,1)

2.8 Model Jaringan Syaraf Tiruan Propagasibalik

Pelatihan pada jaringan syaraf Propagasibalik, *feedforward* (umpan maju) dilakukan dalam rangka perhitungan bobot sehingga pada akhir pelatihan akan diperoleh bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimumkan *error* (kesalahan) yang terjadi. *Error* (kesalahan) dihitung berdasarkan rata-rata kuadrat kesalahan (MSE). Rata-rata kuadrat kesalahan juga dijadikan dasar perhitungan unjuk kerja fungsi aktivasi. Sebagian besar pelatihan untuk jaringan *feedforward* (umpan maju) menggunakan gradien dari fungsi aktivasi

untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimumkan kinerja. Gradien ini ditentukan dengan menggunakan suatu teknik yang disebut Propagasibalik.

Algoritma pelatihan standar Propagasi balik akan menggerakkan bobot dengan arah gradien negatif. Prinsip dasar dari algoritma *Propagasi balik* adalah *memperbaiki* bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat.

Pelatihan *Propagasibalik* meliputi 3 tahapan sebagai berikut.

1. Propagasi maju.

Pola masukan dihitung maju mulai dari *input layer* hingga *output layer* menggunakan fungsi aktivasi yang ditentukan.

2. Propagasi mundur.

Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi itu diPropagasi mundur. Dimulai dari garis yang berhubungan langsung dengan unit-unit di *output layer*.

3. Perubahan bobot.

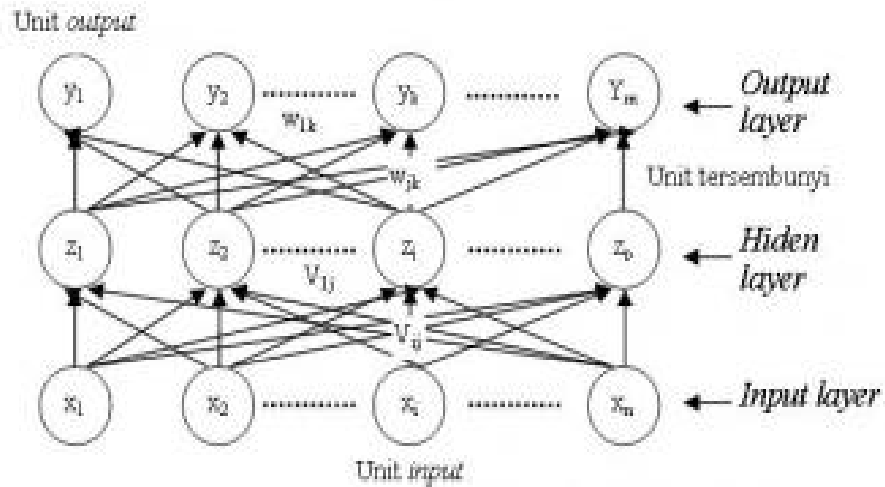
Modifikasi bobot untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. (Puspaningrum, 2006)

2.8.1 Algoritma Jaringan Syaraf Propagasibalik

Propagasi balik merupakan algoritma pembelajaran yang terawarisi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *Propagasi balik* menggunakan *error output* untuk mengubah nilai-nilai bobotnya dalam arah mundur (*backward*). Tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu untuk mendapatkan nilai error tersebut. Saat perambatan maju neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid biner yaitu:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

Arsitektur jaringan syaraf *Propagasibalik* seperti terlihat pada gambar dibawah ini:



Gambar 2 13 : Arsitektur Jaringan Saraf Tiruan Propagasibalik

Keterangan :

x_1 s/d x_n : *input layer*

z_1 s/d z_p : *hidden layer*

y_1 s/d y_m : *output layer*

Algoritma *Propagasibalik* [3]:

1. Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
2. Kerjakan langkah-langkah berikut selama kondisi berhenti belum terpenuhi.

Tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Feedforward

1. Tiap unit input (x_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut pada lapisan yang ada di atasnya (lapisan tersembunyi).
2. Tiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) menjumlahkan sinyal-sinyal input terbobot:

$$z_in_j = v_0j + \sum_{i=1}^n x_i v_{ij} \tag{2}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya: $z_j = f(z_in_j)$ dan kirimkan sinyal tersebut ke semua unit dilapisan atasnya (unit-unit output).

3. Tiap unit output (y_k , $k=1,2,3,\dots,m$) menjumlahkan sinyal-sinyal input terbobot

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (3)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya.

$$y_k = f(y_in_k)$$

dan kirimkan sinyal tersebut ke semua unit dilapisan atasnya (unit-unit output).

Propagasibalik

4. Tiap-tiap unit output (y_k , $k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola input pembelajaran. Hitung informasi errornya.

$$\sigma_k = (t_k - y_k) f'(y_in_k) \quad (3)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki w_{jk}) : $\Delta w_{jk} = \alpha \sigma_k z_{ij}$

$$(4)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}) : $\Delta w_{0k} = \alpha \sigma_k$

$$(5)$$

Kirimkan σ_k ini ke unit-unit yang ada dilapisan bawahnya.

5. Tiap-tiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya) :

$$\sigma_in_j = \sum_{k=1}^m \sigma_k w_{jk} \quad (6)$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*: $\sigma_j = \sigma_in_j f'(z_in_j)$

$$(7)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}): $\Delta v_{jk} = \alpha \sigma_j x_i$

$$(8)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}): $\Delta v_{0j} = \alpha \sigma_j$

$$(9)$$

Tiap unit output (y_k , $k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,3,\dots,p$).

$$w_{jk} \text{ (baru)} = w_{jk} \text{ (lama)} + \Delta w_{jk} \quad (10)$$

Tiap-tiap unit tersembunyi ($z_j, j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,3,\dots,n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \tag{11}$$

6. Tes kondisi berhenti.

2.8.2 Pembelajaran Terawasi (*Supervised Learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika output yang diharapkan telah diketahui sebelumnya.

Contoh: andaikan kita memiliki jaringan syaraf yang akan digunakan untuk mengenali pasangan pola, misalkan pada operasi AND:

Table 1 : Contoh data dan target

INPUT		TARGET
0	0	0
0	1	1
1	0	1
1	1	1

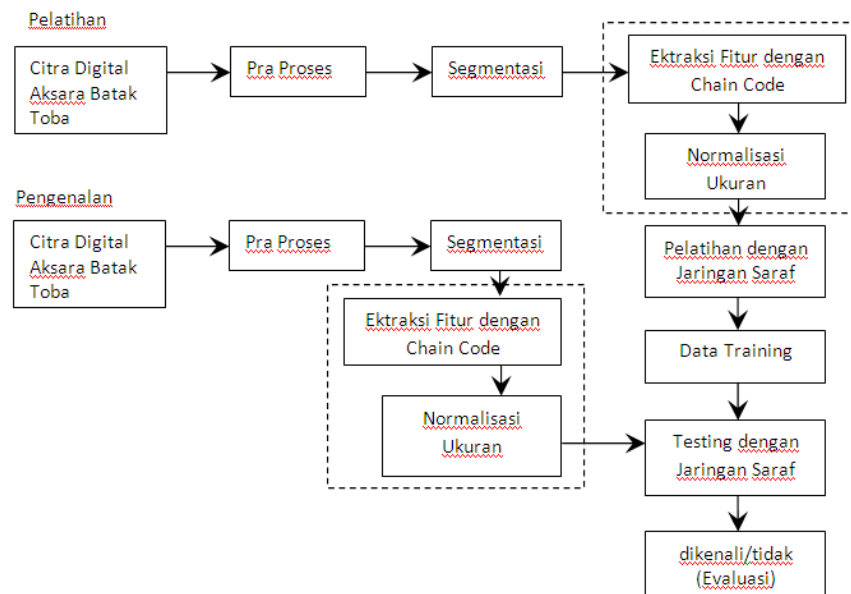
Pada proses pembelajaran, satu pola input akan diberikan ke satu neuron pada lapisan input. Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke neuron pada lapisan output. Lapisan output ini akan membangkitkan pola output yang nantinya akan dicocokkan dengan pola output targetnya. Apabila terjadi perbedaan antara pola output hasil pembelajaran dengan pola target, maka disini akan muncul error. Apabila nilai error ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

2.8.3 Pembelajaran Tidak Terawasi (*UnSupervised Learning*)

Pada metode pembelajaran yang tak terawasi ini tidak memerlukan target output. Pada metode ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola.

2.8.4 Chain Code dan Jaringan Saraf Tiruan Propagasibalik

Penelitian ini menunjukkan kerja sama antara dua Metode yaitu Chain Code dengan Jaringan Saraf Tiruan Propagasi balik dalam proses pengenalan Aksara Batak Toba, kedua metode tersebut mempunyai tugas yang berbeda satu sama lain. Chain Code digunakan untuk Mengekstraksi Fitur/bentuk final Karakter yang akan di Latih dan diuji sedangkan Jaringan Saraf digunakan untuk pelatihan dan pengenalan Karakter itu sendiri. Hasil dari proses Chain Code akan dijadikan sebagai dataset yang siap untuk dipelajari dan dikenali. Bagan pengenalan Aksara Batak Toba dengan Chain Code dan Jaringan Saraf Tiruan Propagasibalik, seperti pada gambar dibawah ini :



Gambar 2. 9 : Proses pengenalan Aksara Batak Toba

Berdasarkan analisis pada bab sebelumnya, maka dengan Metode Chain Code dan Jaringan Saraf Tiruan, diharapkan akan menambah akurasi Pengenalan Aksara Batak Toba font-2 dengan ukuran yang sama dan dataset yang sama. Jadi jelas bahwa perbedaan antara Pengenalan Aksara Batak Toba dengan dengan metode Simplifield Chain Code (CC) dengan Jaringan Saraf Tiruan (JST) adalah Akurasi Pengenalan Aksara pada font-2(font tradisional).

BAB III METODE PENELITIAN

3.1 Proses pengenalan Karakter

Dalam penelitian ini kami melakukan beberapa langkah :

3.1.1 Akuisisi Citra Digital

Adalah proses mendapatkan Citra digital. pada langkah persiapan Citra Digital dengan menggunakan segala perangkat standar seperti Scanner dan kamera adalah cara umum untuk memperoleh gambar, dalam penelitian ini gambar diperoleh dari Scanner Aksara Batak Toba Tulisan Tangan.

3.1.2 Praproses

Tahapan ini terlibat dengan pengolahan karakter input dan mengubahnya menjadi bentuk yang dapat digunakan sebagai masukan untuk sistem pengenalan, langkah ini melakukan pengolahan kecil seperti penghapusan gangguan. Dalam algoritma, kita berasumsi bahwa langkah ini telah dilakukan. Keluaran dari tahap ini adalah *image* tersegmentasi yang akan digunakan untuk proses selanjutnya dalam penelitian. *Image* tersegmentasi adalah *image* yang sudah dipisahkan dari latar belakang *Image*. Dalam penelitian ini, algoritma yang digunakan praproses adalah algoritma **otsu** metode ini digunakan untuk binarisasi citra yaitu **citra warna** menjadi **citra grayscale(keabuan)** dan **citra grayscale** menjadi **hitam putih**. Dalam praproses ini untuk menghilangkan noise atau gangguan pada citra hitam putih akan dibantu dengan menggunakan proses Morphologi[10].

3.1.3 Segmentasi

Image tersegmentasi adalah *image* yang sudah dipisahkan dari latar belakang *image*, kami menggunakan metode.

3.1.4 Ekstraksi Fitur dengan Chain Code

Di tahap ini, *image* akan diekstrak untuk mendapatkan nilai-nilai yang merepresentasikan ciri spesifik dari *image* tersebut. *Image* dari tahap ke-2 akan diperkecil ukuran *pixel*nya karena jumlah datanya yang terlalu besar untuk dijadikan *input*, sehingga *image* diperkecil menjadi 50x55 *pixel*. *Image* ini dipilih karena masih

dapat mewakili ciri citra asli, sesuai dengan rumus level maksimum dekomposisi yang dibatasi oleh persamaan yang berkorelasi (Kanata, 2008:7). *Image* ini akan diubah menjadi citra biner yang disimpan dalam matriks dengan nilai 0 (hitam) dan 1 (putih). Dalam penelitian ini penulis menggunakan metode Chain Code untuk Ekstraksi Ciri.

3.1.5 Normalisasi

Di tahap ini, vektor ciri *digital* yang telah di ekstraksi dengan Chain Code akan dilakukan proses normalisasi agar semua ukuran data menjadi sama.

$$F_i^N = \frac{f_i}{\sum f_i} \times N \quad (12)$$

Dimana :

F_i adalah frekuensi code ke- i

$\sum F_i$ adalah total frekuensi semua code

N adalah nilai frekuensi yang diinginkan

3.2 Proses Pelatihan dengan Jaringan Saraf Tiruan Propagasibalik

Proses pelatihan merupakan langkah penting dalam penelitian ini, karena tanpa pelatihan uji pengenalan karakter tidak dapat dilakukan. Dalam pelatihan dengan Jaringan Saraf Tiruan propabagasi balik data didapatkan hasil normalisasi dengan menggunakan rumus seperti pada persamaan 12 diatas.

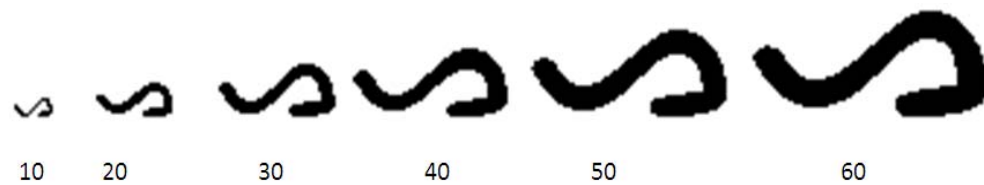
3.3 Proses Uji/Testing dengan Jaringan Saraf Tiruan Propagasibalik

Setelah melakukan propses pembelajaran terhadap jaringan saraf proses berikutnya adalah Pengenalan Aksara dalam proses ini citra yang sudah di latih sebelumnya akan dijadikan sebagai perbandingan untuk mencari dan mencocokkan data yang di testing dengan jaringan saraf Propagasibalik.

BAB IV PELAKSANAAN PENELITIAN

4.1 Pengambilan Gambar

Dalam penelitian ini kami mencoba pengenalan Aksara Batak Toba dengan menggunakan font tradisional, dataset menyesuaikan dengan penelitian sebelumnya yaitu pengambilan gambar dengan perangkat scanner. Gambar yang diambil dengan menggunakan format 300 dpi dan 150 dpi dengan ukuran font 10, 20, 30, 40, 50, 60 untuk kedua font. Gambar dengan format "JPG".



Gambar 4 1 : Aksara "A" dengan ukuran yang berbeda

4.2 Praproses

Sebelum langkah ekstraksi fitur dibutuhkan langkah praprosesing yang berfungsi untuk melakukan perubahan ukuran gambar, perubahan warna dan pembentukan ulang gambar. Berikut ini program matlab untuk menampilkan tampilan citra, asli(warna), keabuan dan binary(0 dan 1).

```
citra=['gambar.jpg'];  
I = imread(citra);  
I1 = rgb2gray(I);  
bw=im2bw(I1);  
figure,imshow(I); %menampilkan citra asli  
figure,imshow(I1); %menampilkan citra grayscale  
figure,imshow(bw); %menampilkan citra binary (0 dan 1)
```

R:255	R:251	R:255	R:253	R: 12	R: 1	R: 4	R: 0
G:255	G:251	G:255	G:253	G: 12	G: 1	G: 4	G: 0
B:255	B:251	B:255	B:253	B: 12	B: 1	B: 4	B: 0
R:233	R:255	R:255	R: 0	R: 5	R: 0	R: 15	R: 0
G:233	G:255	G:255	G: 0	G: 5	G: 0	G: 15	G: 0
B:233	B:255	B:255	B: 0	B: 5	B: 0	B: 15	B: 0
R:249	R: 5	R: 0	R: 9	R: 2	R: 6	R: 2	R: 4
G:249	G: 5	G: 0	G: 9	G: 2	G: 6	G: 2	G: 4
B:249	B: 5	B: 0	B: 9	B: 2	B: 6	B: 2	B: 4
R: 6	R: 19	R: 0	R: 9	R: 13	R: 2	R: 15	R: 0
G: 6	G: 19	G: 0	G: 9	G: 13	G: 2	G: 15	G: 0
B: 6	B: 19	B: 0	B: 9	B: 13	B: 2	B: 15	B: 0

Gambar 4 2 : Pixel Citra Berwarna

255	255	255	255	255	251	234	255	251	243	255	235
255	255	255	255	240	255	232	255	247	252	237	12
255	255	251	252	255	241	255	247	250	252	5	10
254	239	255	250	245	252	255	255	254	17	0	0
251	255	255	250	255	236	250	0	10	11	5	14
247	241	245	246	242	255	0	0	4	1	0	0
251	255	243	255	240	9	5	14	4	0	0	10
255	251	255	253	12	1	4	0	0	12	22	0
233	255	255	0	5	0	15	0	18	0	2	14
249	5	0	9	2	6	2	4	15	0	0	1
6	19	0	9	13	2	15	0	1	10	5	4
0	0	20	0	0	0	11	19	5	0	0	0
6	22	0	15	12	2	8	0	1	2	5	255

Gambar 4 3 : Citra keabuan (grayscale)

0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1

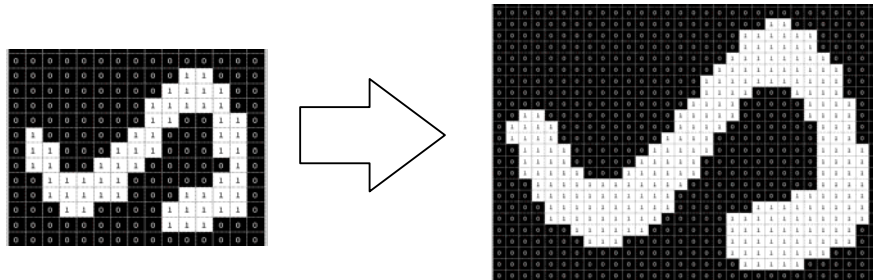
Gambar 4 4 : Citra Binari

4.3 Mengubah ukuran gambar

Gambar yang ekstraksi dengan Chain Code dan yang dilatih Jaringan Saraf Tiruan adalah gambar yang sudah dilakukan perubahan bentuk baik warna maupun ukuran sesuai dengan ukuran dataset yang digunakan pada penelitian sebelumnya. Dalam perubahan ukuran ini citra asli setelah proses *grayscale* dilakukan proses perubahan ukuran gambar (*resize*) dalam proses ini kami menggunakan metode Otsu. Dibawah ini kami tampilkan source code program Matlab untuk melakukan proses citra digital dari warna ke keabuan dan dari keabuan ke binary setelah itu dilakukan perubahan ukuran besar citra digital

Perubahan ukuran gambar dengan menggunakan Matlab sebagai berikut :

```
citra=['Imc1.jpg'];  
I = imread(citra);  
gr = rgb2gray(I);  
bw=im2bw(gr);  
ir=imresize(bw,[60 ,60]);
```



Gambar 4 5 : Proses perubahan ukuran citra

4.4 Mendeteksi tepi gambar (*Edge detection*) dan Chain Code

Selanjutnya citra digital dilakukan proses *Edge Detection* (proses pencarian tepi objek) dari hasil pencarian tepi objek ini kemudian dimasukkan ke dalam Chain Code untuk dilakukan penelusuran arah sesuai dengan algoritma 8 arah ketetanggan dalam Chain Code. Metode yang kami gunakan dalam proses deteksi tepi adalah Canny, kami memilih metode karena telah melakukan perbandingan dengan metode yang lain.

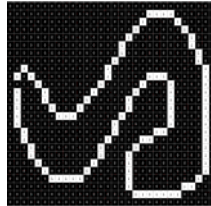
```
t = imread('gambar.jpg');  
gr=rgb2gray(t);  
se = strel('disk',2);  
erodedBW = imerode(gr,se);
```



```

bw=im2bw(erodedBW);
rc=imresize(bw,[60 ,60]);
I2 = edge(~rc,'canny');
figure,imshow(~rc)
imtool(I2);

```



Gambar 4 6 : Hasil Deteksi tepi dengan Canny

4.5 Proses Chain Code

Sebelum melakukan pelatihan dengan Jaringan Saraf Tiruan Propagasibalik terlebih dahulu dilakukan proses Chain Code, proses ini bertujuan agar jumlah data yang menjadi masukan pada saat input lebih kecil dengan demikian proses pembelajaran jaringan saraf akan lebih cepat. Dibawah ini adalah scrip program chaincode function delapan arah mata angin.

```

function out = chaincode8(image)
% Detailed explanation goes here
n=[0, 1; -1, 1; -1,0; -1,-1; 0,-1; 1,-1; 1,0; 1,1];

flag=1;
cc=[];
[x y]=find(image==1);
x=min(x);
imx=image(x,:);
y=min(find(imx==1));
first=[x y];
dir=7;
while flag==1
    tt=zeros(1,8);
    newdir=mod(dir+7-mod(dir,2),8);
    for i=0:7
        j=mod(newdir+i,8)+1;
        tt(i+1)=image(x+n(j,1),y+n(j,2));
    end
    d=min(find(tt==1));
    dir=mod(newdir+d-1,8);
    cc=[cc,dir];

```

```

x=x+n(dir+1,1);
y=y+n(dir+1,2);
if x==first(1) & y==first(2)
    flag=0;
end;
end
out=cc;
end

```

ini adalah program yang melakukan perhitungan secara otomatis frekwensi sebuah karakter.

```

clc;
clear all;
close all;
jk=1;
nml=zeros(62,100);
i=imread('Thin1.jpg');
test=im2bw(i,graythresh(i));
c2=chaincode8(test);
normalize(c2)

kounter=1;
[x,y]=size(ans);
c3=ans;
frek=zeros(2,y);
%% masukkan hasil Chain Code ke dalam array frek
for i=1:y
    frek(1,i)=c2(1,i);
end

%% hitung jumlah frekwensi angka Chain Code
for i=1:y
    if i==y
        frek(2,i)=kounter;
        kounter=1;
        break
    else
        if (c2(1,i)==c2(1,i+1))
            kounter=kounter+1;
        else
            frek(2,i)=kounter;
            kounter=1;
        end
    end
end

end
%% masukkan Chain Code dengan frekwensinya

```

```

frek2=zeros(2,y);
nn=1;
for i=1:y
    if (frek(2,i)>1)
        frek2(1,nn)=frek(1,i);
        frek2(2,nn)=frek(2,i);
        nn=nn+1;
    else
        i=1;
    end
end
end
%% hitung total frekwensinya
frek3=zeros(2,y);
fr2=0;
no=1;
for i=1:y
    if i==y
        break
    else
    if (frek2(1,i)==frek2(1,i+1))
        fr1=frek2(1,i);
        fr2=fr2+frek2(2,i);
        frek3(1,no)=fr1;
        frek3(2,no)=fr2;
    else
        frek3(1,no)=frek2(1,i);
        frek3(2,no)=fr2+frek2(2,i);
        no=no+1;
        fr1=0;
        fr2=0;
    end
end
end
end
%% lakukan normalisasi dengan rumus
frek4=zeros(2,y);
totfrek=0;
for i=1:y
    frek4(1,i)=frek3(1,i)
    totfrek=totfrek+frek3(2,i);
end
end
for i=1:y
    frek4(2,i)=round((frek3(2,i)/totfrek)*100);
end
end
%% normalisasi siap digunakan
a=1;
b=1;
for i=1:100
    ul=frek4(2,i);

```

```

if (ul>1)
  for a=1:(ul)
    ss=frek4(1,i);
    nml(jk,b)=ss;
    b=b+1;
  end
else
  nml(jk,b)=frek4(1,i);
  b=b+1;
end
end

```

hasil dari proses perhitungan Chain Code untuk Karakter “A” diatas adalah :

5	6	5	5	5	6	5	6	5	6
5	6	5	6	5	6	5	6	5	5
6	5	6	5	6	5	5	6	5	6
5	5	5	5	4	4	4	4	3	3
2	3	2	3	2	2	2	3	2	2
3	2	3	3	3	3	4	5	6	6
7	6	6	6	6	6	6	6	6	6
6	6	6	6	6	6	6	6	6	6
6	6	6	7	6	7	6	7	7	7
0	7	0	7	0	0	0	0	0	0
1	1	1	1	1	1	1	1	2	1
1	2	1	2	1	2	1	2	1	2
1	2	1	1	2	1	2	1	2	1
1	1	2	1	0	0	0	0	0	0
7	7	6	6	7	6	6	6	7	6
6	6	6	6	6	6	6	6	6	6
5	5	4	5	4	4	5	4	4	5
4	5	4	5	5	6	6	6	6	6
6	6	6	6	7	7	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	3	2	3
2	2	3	3	2	3	3	3	4	4
4	4	4	4	4	4	4	4	4	4
4									

Dari hasil diatas dilakukan normalisasi yang bertujuan untuk melakukan penyederhanaan angka-angka agar tidak terlalu banyak. Dengan cara melakukan pengumpulan angka yang sama lalu dijumlahkan.

Proses yang dilakukan untuk mendapatkan angka yang sudah ternormalisasi adalah :

1. Data dari proses deteksi tepi, diproses dengan ChainCode 8 arah.
2. Hitung jumlah data sesuai arah, mulai dari 0-7, data yang tampil hanya satu kali ditandai dan selanjutnya dihilangkan. Seperti pada gambar 4.7 menjelaskan bahwa baris pertama adalah nomor urut dan baris kedua adalah angka hasil proses Chain Code dan baris ketiga adalah angka frekwensi jumlah angka hasil proses Chain Code.
3. Setiap angka yang berfrekwensi satu akan dihilangkan.
4. Setelah proses eliminasi angka dengan frekwensi satu, jumlahkan semua angka yang sama seperti gambar 4.8.
5. Hasil penjumlahan angka yang sama seperti gambar 4.9, frekwensi angka dijumlahkan, dari gambar 4.9 berarti yang dijumlahkan adalah angka 11,4,2,5 ...13, hasil penjumlahan dari frekwensi angka tersebut adalah 195.
6. Angka-angka tersebut masukkan ke dalam persamaan 12.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
5	6	5	5	5	6	5	6	5	6	5	6	5	6	5	6	5	6	5	5	6	5	6	5	6
1	1	0	0	3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	2	1	1	1	1	1
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
5	5	6	5	6	5	5	5	5	4	4	4	4	3	3	2	3	2	3	2	2	2	3	2	2
0	2	1	1	1	0	0	0	4	0	0	0	4	0	2	1	1	1	1	0	0	3	1	0	2
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
3	2	3	3	3	3	4	5	6	6	7	6	6	6	6	6	6	6	6	6	6	6	6	6	6
1	1	0	0	0	4	1	1	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
6	6	6	6	6	6	6	6	7	6	7	6	7	7	7	0	7	0	7	0	0	0	0	0	0
0	0	0	0	0	0	0	22	1	1	1	1	0	0	3	1	1	1	1	0	0	0	0	0	6
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
1	1	1	1	1	1	1	1	2	1	1	2	1	2	1	2	1	2	1	2	1	2	1	1	2
0	0	0	0	0	0	0	8	1	0	2	1	1	1	1	1	1	1	1	1	1	1	1	0	2
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
1	2	1	2	1	1	1	2	1	0	0	0	0	0	0	7	7	6	6	7	6	6	6	7	6
1	1	1	1	0	0	3	1	1	0	0	0	0	0	6	0	2	0	2	1	0	0	3	1	0
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
6	6	6	6	6	6	6	6	6	6	5	5	4	5	4	4	5	4	4	5	4	5	4	5	5
0	0	0	0	0	0	0	0	0	11	0	2	1	1	0	2	1	0	2	1	1	1	1	0	2
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
6	6	6	6	6	6	6	6	6	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	9	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	14
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
1	0	1	1	0	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	0	2	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	1	1	1
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271				
2	2	3	3	2	3	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4				
0	2	0	2	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	13				

Gambar 4 7 : Hasil proses Chain Code

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	no.urut		
5	5	5	5	4	3	2	2	3	6	6	7	0	1	1	1	1	0	7	6	angka arah		
3	2	2	4	4	2	3	2	4	2	22	3	6	8	2	2	3	6	2	2	frekwensi		
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36					no.urut		
6	6	5	4	4	5	6	7	0	1	1	2	2	3	3	4					angka arah		
3	11	2	2	2	2	9	2	14	2	2	40	2	2	3	13					frekwensi		

Gambar 4 8 : Hasil Eliminasi angka frewensi 1 (satu)

5	4	3	2	3	6	7	0	1	0	7	6	5	4	5	6	7	0	1	2	3	4	arah
11	4	2	5	4	24	3	6	15	6	2	16	2	4	2	9	2	14	4	42	5	13	frekwensi

Gambar 4 9 : Hasil akumulasi angka arah

4.6 Proses Normalisasi

Setelah didapatkan frekwensi dari sebuah karakter selanjutnya adalah melakukan normalisasi angka yang bertujuan untuk menyamakan ukuran panjang dari setiap angka hasil Chain Code. Urutan jumlah angka yang sudah sama dijadikan sebagai dataset untuk pelatihan dan pengenalan karakter, persamaan yang digunakan untuk normalisasi ini adalah persamaan (12), dengan demikian apabila hasil frekwensi diatas dimasukkan kedalam persamaan (12) akan menjadi sebagai berikut :

Frekwensi hasil chain code diatas (gambar 4.3.8) dimasukkan kedalam persamaan (12), hasilnya seperti terlihat seperti gambar dibawah ini :

$$F_i^* = \frac{11}{195} * 20 = 1,12 \dots \quad F_i^* = \frac{13}{195} * 20 = 1,33$$

Penjelasan persamaan diatas F adalah frekwensi, i=nilai awal frekwensi,n=nilai akhir dan 20 adalah panjang data yang diinginkan, dari gambar 4.3.6 diatas angka awal adalah 11 dan angka akhir 13, jumlah dari semua angka-angka frekwensinya 195, merupakan hasil penjumlahan dari angka-angka frekwensi 11+4+2+5+4 .. +5+13 =195. Hasil proses persamaan setelah dimasukkan frekwensi 11 s/d 13 diatas didapatkan hasil seperti terlihat pada gambar dibawah :

11	4	2	5	4	24	3	6	15	6													
1,128	0,41026	0,21	0,51	0,4103	2,46	0,31	0,62	1,538	0,62													
2	16	2	4	2	9	2	14	4	42	5	13											
0,21	1,64	0,21	0,41	0,21	0,92	0,21	1,4359	0,41	4,31	0,51	1,33											

Gambar 4 10 : Hasil Proses Normalisasi

Melakukan pembulatan keatas untuk mendapatkan angka yang bulat misalnya 1,12 menjadi 1 dan 0,5 menjadi 1 daseterusnya, dengan demikian diperoleh angka-angka yang menjadi frekwensi akhir dari sebuah perhitungan Chain Code.

Seperti dibawah ini :

5	4	3	2	3	6	7	0	1	0	7	
11	4	2	5	4	24	3	6	15	6	2	
1,128	0,41026	0,21	0,51	0,4103	2,46	0,31	0,62	1,538	0,62	0,21	
1	0	0	1	0	3	0	1	2	1	0	
6	5	4	5	6	7	0	1	2	3	4	arah
16	2	4	2	9	2	14	4	42	5	13	frekwens
1,64	0,21	0,41	0,21	0,92	0,21	1,4359	0,41	4,31	0,51	1,33	20
2	0	0	0	1	0	2	0	4	1	1	20

Gambar 4 11 : Hasil frekwensi sesuai dengan jumlah N yg sudah ditentukan

5=1, 2=1, 6=3, 0=1, 1=2, 0=1, 6=2, 6=1, 0=2, 2=4, 3=1, 4=1

Hasil akhirnya adalah : 5 2 6 6 6 0 1 1 0 6 6 6 0 0 2 2 2 2 3 4

Data yang sudah di normalisasi ini akan dijadikan dataset untuk inputan pelatihan dengan jaringan saraf, hal yang sama dilakukan terhadap semua karakter yang akan dilatih. Data yang sudah di normalisasi ini akan dijadikan dataset untuk inputan pelatihan dengan jaringan saraf.

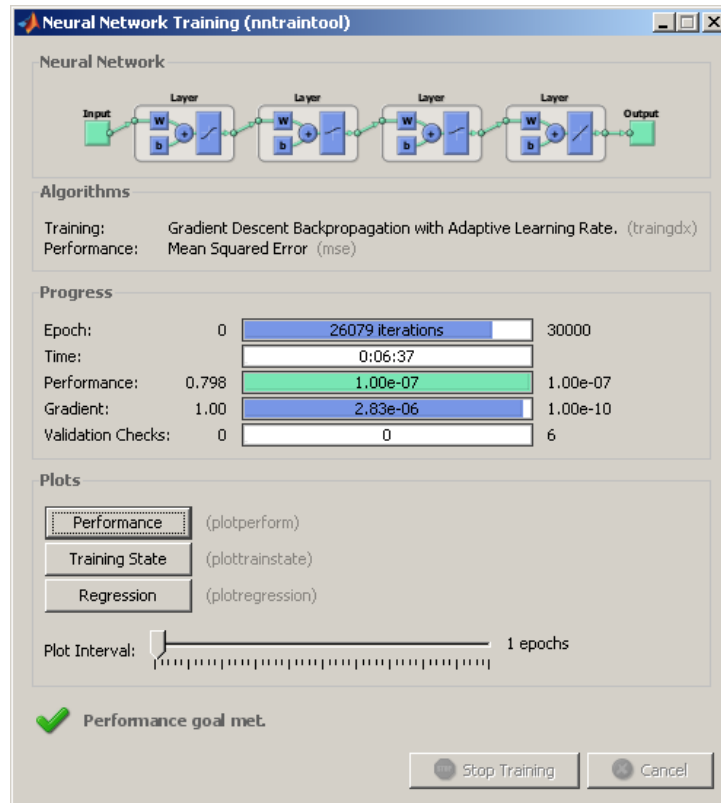
4.7 Training dengan Jaringan Saraf Tiruan data tulisan tangan

Untuk melihat sebuah perbandingan pelatihan dan pengenalan aksara Batak Toba selain dari dataset yang sama dengan penelitian sebelumnya, kami juga mencoba dataset tulisan tangan (Lampiran 7). Pelatihan tulisan tangan dengan memasukkan 10 variasi huruf yang terdiri dari 10 induk huruf dan 6 induk huruf cetak dengan ukuran 60 point, induk huruf yang kami latih adalah a,ha,ma,na,ra,ta,ba,wa,la,nga dan 6 induk huruf cetak yaitu a,ha,ma,ba,ra,ta. Kesimpulan yang dapat kami ambil adalah :

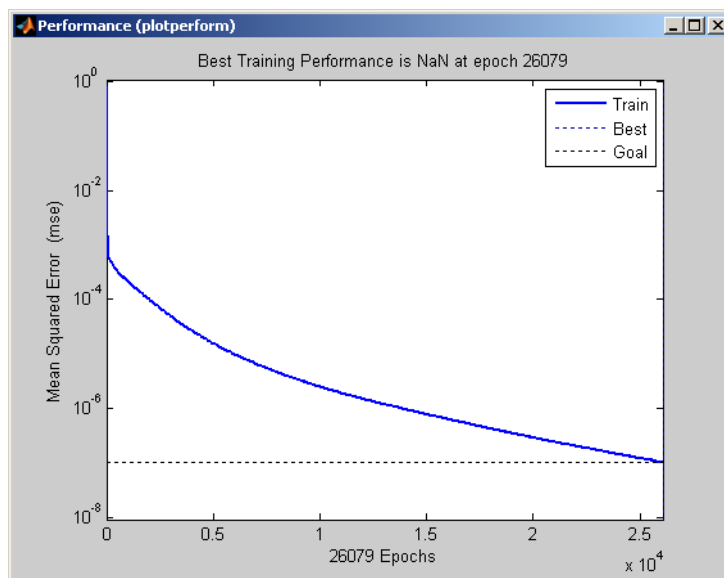
1. Settingan kebutuhan jaringan harus dibedakan agar pelatihan tidak terlalu lama.
2. Iterasi pelatihan, Target Error dan jumlah neuron berbeda dengan pelatihan menggunakan dataset aksara cetak.

Dibawah ini adalah gambar hasil proses pelatihan jaringan saraf dengan menggunakan dataset tulisan tangan dengan setingan sebagai berikut :

```
net = newff(minmax(pn),[ 100 100 5 1],{'tansig' 'logsig'  
'logsig' 'purelin'},'traingdx');  
net.trainParam.epochs      = 30000;  
net.trainParam.goal        = 1e-7;  
net.trainParam.lr          = 0.5;  
net.trainParam.mc          = 0.7;  
net.trainParam.show        = 250;
```



Gambar 4 12 : Pelatihan dengan dataset tulisan tangan



Gambar 4 13 : Grafik MSE hasil pelatihan dataset tulisan tangan

Hasil pengujian dengan menggunakan salah satu variasi tulisan tangan yang dilatih yaitu aksara “ a”, hasil dari pengujian ini dikenali pada posisi target yaitu 1-10 dan 101, seperti dibawah ini :

Nilai Output Data Uji y1 :1.0183

data termasuk Pola ke:1 2 3 4 5 6 7 8 9 10 101

Hasilnya adalah :aaaaaaaaal

Sedangkan hasil testing dengan menggunakan tulisan tangan yang tidak sama dengan data yang di training dengan settingan jaringan saraf yang sama diatas jumlah data 106 yang diujikan 8 karakter dan yang dikenali 4, maka akurasi = $4/8 \times 100 = 50\%$.

Kemungkinan peningkatan akurasi pengenalan karakter dengan menggunakan tulisan tangan masih bisa ditingkatkan dengan melakukan perubahan settingan jaringan saraf tiruan.

4.8 Training dengan Jaringan Saraf Tiruan data Aksara cetak

Dari hasil proses Chain Code dan Normalisasi data didapatkan sebuah matriks dengan format baris dan kolom.

P dalam program dibawah adalah hasil proses normalisasi Chain Code yang sudah di mulai dari karakter “a s/d nya”, matrik tersebut dijadikan sebagai data training Jaringan Saraf Tiruan Propagasibalik, dengan ketentuan sebagai berikut :

Jumlah lapisan 5 dengan rincian masing masing lapisan sebagai berikut lapisan pertama berjumlah 150 neuron, lapisan kedua,ketiga 150 neuron dan lapisan ketiga 7

neuron serta lapisan yang ke empat 1 neuron. Nilai ambang yang digunakan adalah 0.5, sedangkan iterasi(epoch) sebesar 15000, adapun aktivasi yang digunakan adalah sebagai berikut:

```
net = newff(minmax(pn),[150 150 150 7 1],{'tansig' 'logsig' 'logsig' 'logsig'  
'purelin'},'traingdm');  
% Melihat bobot-bobot awal input, lapisan,dan bias (boleh dihilangkan)sis  
BobotAwal_Input          = net.IW{1,1} ;  
BobotAwal_Bias_Input     = net.b{1,1} ;  
BobotAwal_Lapisan1      = net.LW{2,1} ;  
BobotAwal_Bias_Lapisan1 = net.b{2,1} ;  
BobotAwal_Lapisan2      = net.LW{3,1};  
BobotAwal_Bias_Lapisan2 = net.b{3,1};  
BobotAwal_Lapisan3      = net.LW{4,1};  
BobotAwal_Bias_Lapisan3 = net.b{4,1};  
  
% Set max epoh, target error, learning rate,momentum dan epoh show  
net.trainParam.epochs = 15000;  
net.trainParam.goal    = 1e-11;  
net.trainParam.lr      = 0.5;  
net.trainParam.mc      = 0.8;  
net.trainParam.show    = 250;
```

4.9 Pelatihan Jaringan Saraf Tiruan

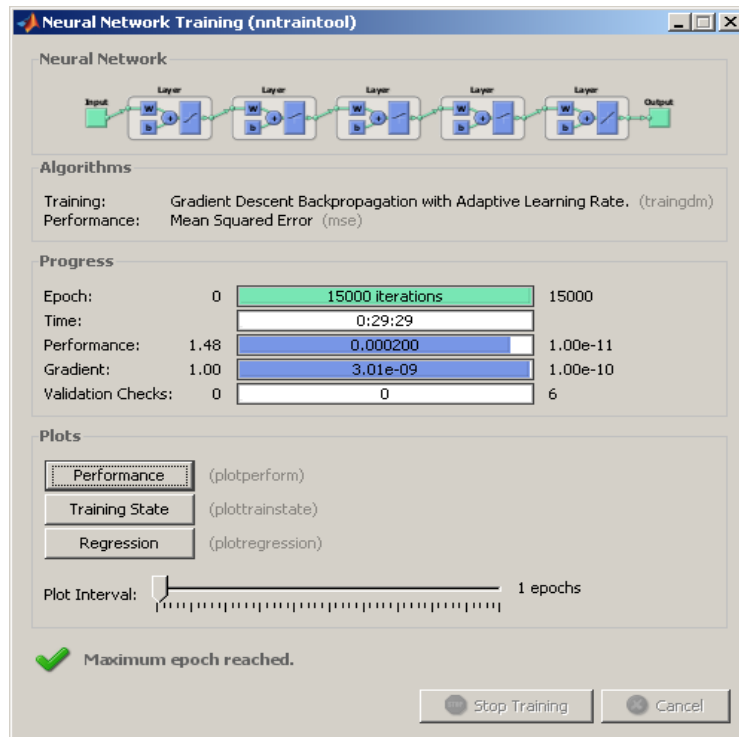
4.10 Tanpa Chain Code.

Untuk mengetahui perbedaan penggunaan Chain Code dan Jaringan Saraf Tiruan Propagasibalik dalam pelatihan dan tanpa menggunakan Chain Code dalam pengenalan Aksara Batak Toba, penulis telah melakukan percobaan pembelajaran data sesuai dengan penelitian sebelumnya. Proses yang dilakukan adalah mempersiapkan data yaitu citra digital melakukan binarisasi, segmentasi, feature ekstraksi lalu training dengan jaringan saraf tiruan.

Proses pengenalan Aksara Batak Toba dengan Jaringan Saraf Tiruan saja, diawali dengan pengambilan gambar dengan menggunakan scanner, praproses,segmentasi, recognisi dan evaluas, berikut ini adalah cara yang digunakan penulis untuk

mengetahui perbandingan antara pengenalan Aksara Batak Toba tanpa menggunakan Chain Code. proses yang dilakukan adalah sebagai berikut :

Citra hasil proses segmentasi dan feature ekstraksi dengan ukuran 60 X 60, dijadikan sebagai data inputan untuk Jaringan Saraf Tiruan Propagasibalik, gambar dibawah ini adalah hasil proses training jaringan saraf tiruan propagasibalik, data inputan berjumlah 5 karakter (lampiran 2) :



Gambar 4 14 : Training data tanpa Chain Code

Matrik karakter “a” diatas, dijadikan sebagai data training Jaringan Saraf Tiruan Propagasibalik, dengan ketentuan sebagai berikut :

```
net = newff(minmax(pn),[ 150 150 150 7 1],{'tansig' 'logsig' 'logsig' 'logsig'
'purelin'},'traingdm');
```

% Set max epoch, target error, learning rate, momentum dan epoch show

```
net.trainParam.epochs = 15000;
```

```
net.trainParam.goal = 1e-11;
```

```
net.trainParam.lr = 0.5;
```

```
net.trainParam.mc = 0.8;
```

```
net.trainParam.show = 250;
```

`% Melihat bobot-bobot awal input, lapisan,dan bias (boleh dihilangkan)sis`

```
BobotAwal_Input = net.IW{1,1} ;
```

```
BobotAwal_Bias_Input = net.b{1,1} ;
```

```
BobotAwal_Lapisan1 = net.LW{2,1} ;
```

```
BobotAwal_Bias_Lapisan1 = net.b{2,1} ;
```

```
BobotAwal_Lapisan2 = net.LW{3,1};
```

```
BobotAwal_Bias_Lapisan2 = net.b{3,1};
```

```
BobotAwal_Lapisan2 = net.LW{4,1};
```

```
BobotAwal_Bias_Lapisan2 = net.b{4,1};
```

Jumlah lapisan 5 dengan rincian masing masing lapisan sebagai berikut lapisan pertama berjumlah 150 neuron, lapisan kedua dan tiga 150 neuron dan lapisan ketiga 7 neuron serta lapisan yang ke empat 1 neuron. Nilai ambang yang digunakan adalah 0.5, sedangkan iterasi(epoch) sebesar 15000, adapun aktivasi yang digunakan adalah sebagai berikut:

```
net = newff(minmax(pn),[150 150 170 7 1],{'tansig' 'logsig' 'logsig' 'logsig' 'purelin'},  
'traingdm').
```

Data yang dilatih berjumlah 5 karakter dengan panjang 3600 tiap karakter, setelah dilatih hasilnya seperti gambar dibawah ini, waktu yang digunakan sangat lama sekali yaitu 29 menit untuk 5 baris/karakter.

Inilah percobaan yang kami lakukan melakukan pelatihan dengan jaringan saraf Propagasibalik dengan 5 jumlah data training dan jumlah neuron tiap layer 150 dua layer terakhir 7 neuron dan 1 neuron untuk output layer. Kelemahan dari jaringan saraf tanpa Chain Code terletak pada waktu training yang digunakan terlalu lama, dari percobaan diatas waktu untuk training 5 karakter 28 menit.

4.11 Setelah Chain Code

Dataset yang digunakan untuk training adalah hasil proses Chain Code (lampiran 1), data ini terdiri dari 93 Karakter masing-masing 31 font aksara Batak tradisional ukuran 40,50 dan 60 pt. dengan panjang masing-masing 102.

Setiap karakter telah dijadikan menjadi matrik dengan ukuran 1x102 matrik ini akan menjadi data inputan untuk di latih dengan jaringan saraf tiruan propagasibalik. Dan setelah melakukan setting puluhan kali kami mendapatkan susunan layer dan neuron

serta variable yang lain dan adapun settingan kebutuhan jaringan saraf tiruan saat pelatihan, kami telah menentukan adalah sebagai berikut :

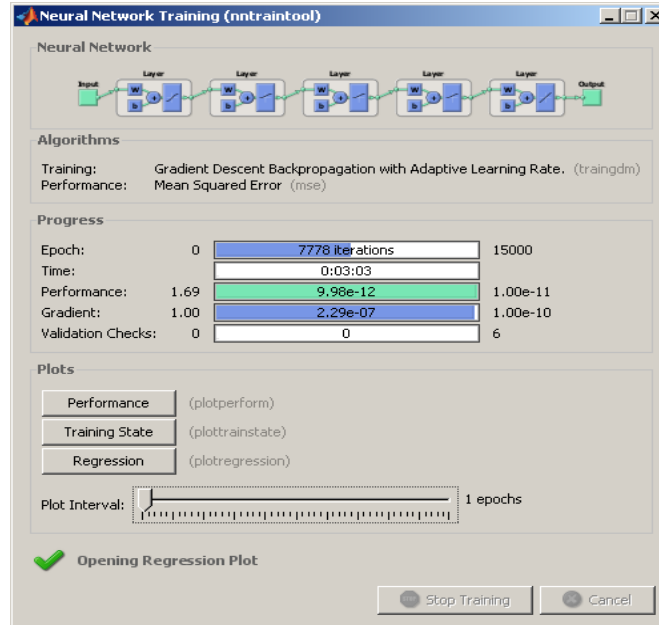
```
net.trainParam.epochs    = 15000;
net.trainParam.goal      = 1e-11;
net.trainParam.lr        = 0.5;
net.trainParam.mc        = 0.8;
net.trainParam.show      = 250;

BobotAwal_Input          = net.IW{1,1} ;
BobotAwal_Bias_Input     = net.b{1,1} ;
BobotAwal_Lapisan1      = net.LW{2,1} ;
BobotAwal_Bias_Lapisan1 = net.b{2,1} ;
BobotAwal_Lapisan2      = net.LW{3,1};
BobotAwal_Bias_Lapisan2 = net.b{3,1};
BobotAwal_Lapisan3      = net.LW{4,1};
BobotAwal_Bias_Lapisan3 = net.b{4,1};
```

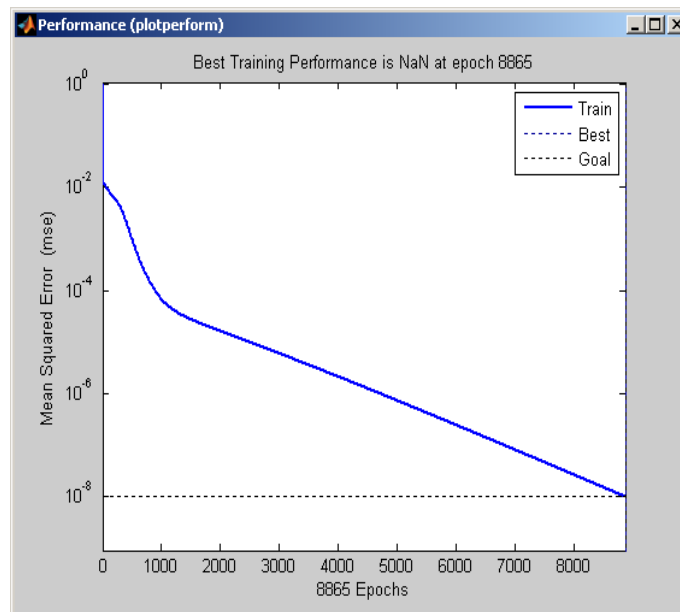
Jumlah lapisan 5 dengan rincian masing masing lapisan sebagai berikut lapisan pertama berjumlah 150 neuron, lapisan kedua 150 neuron, ketiga 150 neuron dan lapisan empat 7 neuron serta lapisan yang ke lima 1 neuron. Nilai ambang yang digunakan adalah 0.5, sedangkan iterasi(epoch) sebesar 15000, adapun aktivasi yang digunakan adalah sebagai berikut:

```
net = newff(minmax(pn),[150 150 150 7 1],{'tansig' 'logsig' 'logsig' 'logsig' 'purelin'},
'traingdm').
```

Dari hasil uji coba diatas, Penulis menyimpulkan bahwa : pengenalan Aksara Batak Toba dengan menggabungkan Chain Code dan Jaringan saraf tiruan lebih lebih cepat dibandingkan tanpa menggunakan Chain Code. Berikut ini adalah tampilan uji coba pelatihan dengan Jaringan Saraf Tiruan Propagasibalik yang datanya telah lebih dahulu diproses dengan Chain Code dan dinormalisasi.



Gambar 4 15 : Training dengan menggunakan Chain Code



Gambar 4 16 : Mean Squared Error (MSE)

4.12 Pengenalan Karakter

Proses pengenalan Karakter adalah proses yang utama dalam penelitian ini, kami menggunakan Aksara Batak Toba FONT-2 dengan ukuran 40,50pt dan 60pt sebagai data latih/test sesuai dengan penelitian sebelumnya. Proses pengenalan yang kami lakukan sebagaimana yang sudah kami sampaikan pada bab sebelumnya yaitu dengan mengambil sebuah karakter "a" dan hasil proses Chain Code akan dimasukkan kedalam Jaringan Saraf Tiruan untuk dilakukan pengujian. Hasil pengenalan Aksara Batak Toba Seperti gambar dibawah ini :

```
Program dapat mengenali data pelatihan 100%
SAATNYA UNTUK PENGENALAN POLA
=====
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
a  ra hu tu ma su pu bu yu wu ta ha gu lu sa
=====
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
bu da ja gu a  yu i  da ju  u ba lu ra bu d
=====
Nilai Output Data Uji y1 :5
data termasuk Pola ke:5 48
Hasilnya adalah          :mm
```

Gambar 4 17 : Pengenalan Aksara "ma"

y1 adalah nilai output data uji berdasarkan target yang sudah ditentukan, aa adalah karakter set sebagai informasi Karakter yang dikenali, sedangkan 1 dan 34 adalah posisi karakter berada sesuai dengan target yang sudah ditentukan untuk mewakili setiap karakter yang di latih.

Hasil dari proses pengenalan ini yang telah dicoba dengan 93 data dan semuanya dikenali 100%.

BAB V PENUTUP

5.1 Kesimpulan

Dari proses yang telah kami lakukan, dapat kami simpulkan bahwa :

1. Chain Code adalah metode yang tepat untuk digunakan sebelum pelatihan Pola dengan Jaringan Saraf Tiruan propagasibalik, karena dengan Chain Code waktu pelatihan akan lebih cepat.
2. Jaringan Saraf Tiruan Propagasibalik adalah metode yang tepat digunakan dalam pengenalan Aksara Batak Toba.
3. Dengan menggunakan Jaringan Saraf Tiruan Propagasibalik 5 layer, dimana tiga layer pertama berjumlah 150 neuron dan layer ke 4 7 neuron serta layer output 1 neuron, jumlah learning rate 0,5 dengan error 0,1 dan 102 inputan serta 93 dataset, pengenalan Aksara Batak Toba Font-2 dengan ukuran 40,50 pt dan 60 pt dikenali dengan akurasi 100%. testing/uji pengenalan 93 karakter semuanya dikenali 100%. Persamaan yang digunakan untuk menentukan akurasi pengenalan Aksara Batak Toba adalah :

$$\text{Akurasi} = \frac{\text{jumlah data yg dikenali}}{\text{jumlah data yg di uji}} \times 100$$

Persamaan diatas diambil dari sebuah penelitian pengenalan karakter tulisan tangan yaitu “*Handwrite Character Recognition System using Chain Code and Correlation Coefficient*” [24].

Dibawah ini kami tampilkan table pengenalan Aksara Batak Toba denga data tulisan tangan dan aksara cetak.

Table 2 : Tabel pengenalan Aksara Batak Toba

NO	Variasi aksara	Lapisan NN-BP	Jumlah Neuron	Jumlah data / panjang/ waktu training	Parameter kebutuhan jaringan	Akurasi
1	Huruf Cetak dgn ukuran Font 30,40 50 dan 60 pt	5 layer	1=150 2=150 3=150 4=7 5=1	93/ 102 / 4 menit	epochs = 15000; goal = 1e-11; lr = 0.5; mc = 0.8;	100 %
2	Huruf tulisan tangan dgn ukuran 60x60	4 layer	1=100 2=100 3=5 4=1	106/ 100 / 26 menit	epochs = 30000; goal = 1e-7; lr = 0.5; mc = 0.7;	50 %

5.2 Pekerjaan kedepan

- Pengenalan Aksara Batak Toba dengan menggunakan tulisan tangan langsung dikertas lalu discan dengan ukuran 150 dpi dan dijadikan sebagai dataset.
- Menambah variasi bentuk karakter dengan ukuran font yang bervariasi pula .

DAFTAR REFERENSI

- [1] Character Recognition of the Batak Toba Alphabet Using Signatures and Simplified ChainCode (Panggabean dan Rønningen, 2009)
- [2] WM. Hutagalung.(2009). Pustaha Batak Tarombo dohot Turiturian ni Bangso Batak,Tulus Jaya
- [3] Character Recognition of the Batak Toba Alphabet Using Signatures and Simplified ChainCode (Panggabean dan Rønningen, 2009)
- [4] Kusumadewi Sri and Purnomo Hari, *Aplikasi Logika Fuzzy untuk Pendukung*.: Penerbit Graha Ilmu, 2010
- [5] Suyanto, *Artificial Intelligenc: Searching, Reasoning, Planning, and Learning*. Bandung: Penerbit Informatika, 2007.
- [6] Sri Kusumadewi. (2003). Artificial Intelegence, Yogyakarta: Graha Ilmu.
- [7] Siang, J.J.(2009). Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan Matlab.
- [8] Heaton, J. (2008). *Introduction to Neural Network with Java*. United States: Heaton Research, Inc.
- [9] Anupan Shukla, R. T. (2010). *Real Life Aplications of Soft Computing*. Boca Raton: CRC Press.
- [10] Gonzales.(2003). Digital Image Processing Using MATLAB.London.Pearson Education LTD.
- [11] Putra Darma. (2009), *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.
- [12] An Overview of Character Recognition Focused on Off-Line Hanwriting (Nafiz,Fatos 2001)
- [13] Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika
- [14] Acharya Tinku, Ray Ajoy K. (2005). *Image Processing, Principle and Application*. New Jersey: John Wiley & Sons,Inc.
- [15] Hermawan, A. (2006). *Jaringan Syaraf Tiruan Teori dan Aplikasi*. Yogyakarta: Penerbit Andi
- [16] Kusumadewi, S. (2003). *Artificial Intellegence (Tehnik dan Aplikasinya)*. Yogyakarta: Graha Ilmu

- [17] Puspaningrum, D. (2006). *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta : Andi Offset
- [18] (Neural Network Classifiers for Optical Chinese Character Recognition, Richard Romero, Robert Berger, Robert Thibadeau and David Touretzky,2006)
- [19] (Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes, H. Izakian, S. A. Monadjemi, B. Tork Ladani, and K. Zamanifar,2008)
- [20] Ni Kadek Ayu W.2011,Pembentukan Pola Khusus untuk ekstraksi cirri pada Sistem Pengenalan Akasara Bali Cetak.
- [21] S, Riyanto dkk. 2005. Step by Step Pengolahan Citra Digital. Yogyakarta: Andi.
- [22] Basuki, A. 2005. Pengolahan Citra Digital menggunakan VB. Yogyakarta: Graha Ilmu
- [23] Sitorus, S dkk. 2006. Pengolahan Citra Digital. Medan:USU Press.
- [24] H. Izakian, S. A. Monadjemi, B. Tork Ladani, and K. Zamanifar,2008,Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes
- [25] *Handwrite Character Recognition System using Chain Code and Correlation Coefficient*(Ravi Sheth,N C Chauhan,Mahesh M.Goyani,Kinjal A.Mehta, 2011)

LAMPIRAN

Lampiran 1 : Data training dengan Chain Code

p=[5,5,6,5,6,5,5,5,5,5,4,3,2,2,2,3,3,4,5,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,0,0
,1,1,1,1,1,1,2,1,2,1,1,0,0,0,0,7,6,6,6,6,6,6,6,5,4,4,4,5,6,6,6,6,6,7,0,0,0
,0,0,1,0,1,1,1,2,3,3,3,4,4,4;

4,4,4,4,4,5,6,6,6,6,6,0,0,1,0,0,0,0,0,0,4,4,4,5,4,5,5,5,6,6,6,6,6,0,1,
1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,7,7,5,5,5,6,6,6,7,0,1,1,1,1,1,1,1,2,2,2,3,
3,3,0,1,2,2,2,3,3,3,4,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0;

5,5,6,6,6,6,7,0,1,0,0,0,0,7,5,5,5,5,5,6,6,6,7,0,0,1,1,1,1,1,1,1,1,1,1,
0,0,0,0,7,6,6,5,5,5,5,5,6,6,6,6,7,0,0,7,6,5,6,6,7,0,1,1,2,1,2,2,2,2,3,3,
2,2,1,2,2,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,4;

4,4,4,4,4,4,4,5,5,6,6,7,7,7,7,7,7,6,5,4,5,5,5,5,6,6,6,6,7,0,0,1,1,1,
0,0,0,7,7,7,7,7,7,6,5,5,6,6,7,0,0,1,1,1,1,1,1,1,2,3,3,3,3,4,3,3,2,1,1,1,
1,1,1,1,1,2,3,3,4,0;

4,5,6,6,6,6,7,0,1,7,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,0,1,1,0,0,0,7,7,0,0,
1,2,2,2,3,4,3,4,3,4,2,1,0,0,1,1,2,2,2,3,4,4,4,4,4,4,5,5,5,6,5,6,5,5,3,2,2,
2,2,2,2,2,1,7,0,0,0,1,0,1,1,2,2,2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4;

5,6,6,7,0,0,0,0,0,7,5,5,5,6,5,6,6,6,6,7,7,0,0,0,0,0,0,1,1,2,3,3,4,4,
4,4,3,2,2,1,1,1,1,0,7,6,6,6,6,7,7,7,6,5,5,5,6,6,6,7,0,1,1,1,1,1,2,2,2,3,
3,3,2,2,2,2,1,0,0,1,1,2,3,4,4,4,4,4,4,4,4,4,4,3,3,3,4,4,4,4,4;

6,6,6,6,6,6,6,6,6,7,0,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,6,7,7,6,6,5,5,6,6,
6,7,0,0,1,1,1,1,1,2,2,3,3,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,2,2,2,3,
4,4;

4,4,4,5,5,5,5,5,6,6,5,6,7,7,7,0,
0,1,1,1,1,1,1,0,0,0,0,0,7,7,7,7,7,7,0,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0;

6,6,6,6,7,6,6,6,6,6,6,7,6,7,7,0,0,1,1,1,1,2,1,2,1,1,0,0,7,6,3,3,4,4,5,6,
6,7,7,7,5,5,5,6,6,6,7,0,1,1,1,1,1,2,2,2,1,2,2,2,2,2,2,3,3,3,3,4,4,4,5,5,
6,5,5,5,4,4,3,2,2,2,2,0,0,1,1,1,2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,0,0;

4,4,4,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,0,0,1,1,2,2,3,4,4,2,2,2,
2,2,1,1,1,0,0,0,0,0,0,0,7,7,7,7,7,6,6,6,6,5,4,4,4,5,6,6,6,6,7,7,7,7,7,7,
0,1,2,3,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,0,0;

4,4,4,4,4,4,5,5,6,6,7,7,7,7,7,7,7,7,7,7,5,5,5,5,5,5,6,6,6,6,7,0,0,0,
1,1,1,1,1,1,0,0,0,7,7,7,7,7,7,7,7,0,0,1,1,2,2,3,3,3,3,3,3,3,2,2,1,1,1,1,
1,1,1,1,1,2,2,3,3,4,0,0,0;

5,5,6,6,6,6,6,6,7,0,1,1,1,0,0,0,0,7,5,5,6,5,6,6,6,6,6,6,7,1,1,1,1,1,1,
1,1,1,0,0,0,0,7,6,6,6,6,5,5,5,6,6,6,6,6,6,6,6,7,1,1,1,1,2,2,2,1,2,2,2,2,2,
2,2,2,2,2,2,2,2,3,3,3,4,4,4,4,4,4,4,4,4,5,5,4,3,3,4,4,4,4,4,4,4;

4,5,5,5,5,6,5,6,6,6,6,6,6,6,6,7,0,1,1,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,
0,7,0,7,5,5,5,5,6,6,6,6,6,6,7,7,7,7,7,7,7,7,1,1,2,2,2,2,3,3,3,2,1,1,2,2,
2,2,2,2,3,3,3,3,3,3,4,3,4,0;

4,4,4,4,4,5,5,5,6,6,6,6,6,6,7,7,7,7,7,7,0,1,1,2,2,3,3,3,3,3,2,1,0,0,0,0,

0,0,0,0,0,0,0,7,6,6,6,6,6,6,6,6,7,7,7,6,5,5,5,6,6,6,6,7,0,1,1,1,1,2,2,
2,3,3,2,2,7,7,0,1,1,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4;

6,6,6,6,6,0,0,0,0,0,0,7,6,5,5,5,5,5,6,6,6,6,6,7,7,7,0,0,0,0,0,0,0,1,
1,2,2,2,3,3,4,4,4,4,4,2,2,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,2,2,2,3,3,4,4,
4,4,4,4,4,4,4,4,4,5,4,4,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0;

5,6,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,4,4,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,3,4,0;

4,4,4,4,4,4,4,4,4,4,4,4,4,5,6,6,6,6,7,6,6,7,7,0,0,7,7,0,7,0,7,0,0,7,0,7,6,5,
6,6,7,7,0,1,1,2,2,3,3,3,3,3,4,3,4,3,4,3,4,3,4,4,1,0,0,0,0,0,0,0,6,6,7,0,
0,0,0,0,0,0,0,0,0,1,1,2,2,3,4,4,4,4,4,4,4,4,2,2,1,1,2,2,3,3,4,4;

5,6,6,6,6,6,6,7,7,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,0,1,1,1,2,2,2,1,0,0,0,0,
7,6,6,6,6,6,7,6,0,1,2,2,2,2,2,2,2,2,2,2,3,3,4,4,4,4,5,5,6,6,6,6,5,5,5,
4,4,3,2,2,2,2,2,2,0,0,0,1,2,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,4,4;

4,4,4,4,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,6,7,7,
7,7,7,7,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,1,1,2,1,1,2,2,1,2,2,2,2,
2,2,2,2,3,2,2,3,2,3,2,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,0;

4,4,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,0,1,2,2,2,2,3,4,4,4,
2,2,2,2,2,2,1,1,1,0,0,0,0,0,0,7,7,7,7,6,6,6,6,5,4,4,4,5,6,6,6,6,7,0,0,0,
0,0,1,2,2,2,2,2,2,2,2,2,2,2,2,3,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4;

6,6,6,6,6,7,0,7,7,7,7,7,6,6,6,5,5,5,5,5,5,6,5,6,6,6,6,7,0,0,0,0,1,1,1,1,
1,1,1,1,7,0,7,7,7,7,7,0,7,0,4,3,4,3,3,3,3,3,4,3,5,5,5,5,5,5,5,5,4,4,4,4,
3,2,2,2,2,1,2,1,1,1,1,1,1,2,2,2,3,3,3,3,3,4,3,2,2,2,2,2,0,0;

4,
0,0,7,5,6,6,6,6,7,7,0,7,7,0,7,7,6,5,5,5,6,5,6,6,7,7,0,0,0,1,1,1,1,1,2,2,
3,3,3,3,3,4,3,4,3,4,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,3,4,4,4,4;

4,4,5,5,6,6,6,6,7,0,0,0,0,0,0,0,0,0,0,0,6,6,6,4,4,5,6,6,6,7,0,0,0,7,6,5,4,
5,5,6,6,7,7,0,0,0,0,0,0,1,1,2,2,3,2,2,2,0,0,1,1,2,2,3,3,4,3,2,2,0,0,0,0,0,
0,0,0,1,1,2,3,3,4,0;

4,4,4,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,1,1,1,1,1,1,0,0,0,1,0,0,0,
0,0,0,0,0,0,0,0,7,0,0,7,7,5,5,5,6,5,6,6,6,6,6,6,6,6,7,0,1,1,1,1,1,2,1,2,2,
2,2,2,2,2,2,2,2,3,3,3,3,4,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0;

5,5,5,5,5,5,6,6,6,6,6,6,6,7,7,7,7,0,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
7,7,7,7,7,7,7,0,0,0,0,0,1,1,1,1,1,2,1,2,2,2,2,2,3,3,3,3,4,3,3,3,3,3,3,4,
3,3,3,3,3,3,3,3,3,3,3,4,3,3,3,3,3,4,3,3,3,4,4,4,0,0,0,0,0;

4,4,4,4,5,6,6,6,7,0,0,0,0,0,0,0,0,6,4,4,4,4,4,4,4,4,5,5,5,6,6,6,6,7,0,1,1,1,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,7,5,5,5,5,6,6,7,7,5,6,6,7,0,1,1,1,1,2,3,3,2,1,1,
1,2,2,3,3,3,0,0,1,2,3,3,4;

5,6,6,6,7,7,6,6,4,4,4,4,6,6,6,7,7,0,7,6,6,6,5,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,
6,7,7,0,1,1,2,2,3,4,4,4,4,4,4,
4,4,3,2,2,2,0,1,1,2,3,3,4,2,2,1,1,2,3,4,4,4,4,4,4,4,4,4,4,4,0,0,0;

4,4,4,4,4,5,4,4,5,5,5,6,6,6,6,6,6,6,7,0,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,0,1,
1,2,2,2,3,3,3,3,3,3,3,3,3,4,3,1,1,0,0,0,6,6,6,6,6,6,0,0,0,0,0,0,0,0,0,0,0,0,
0,1,1,2,2,2,3,3,4,4,4,4,4,4,4,4,4,2,2,2,1,1,2,2,2,2,3,4,4,4,4,4;

4,4,5,6,6,6,6,7,0,0,6,6,6,6,6,6,7,7,7,0,1,0,0,0,7,5,5,6,6,7,0,1,1,1,1,1,
1,2,2,3,4,4,3,1,1,1,0,0,0,0,1,1,2,2,3,4,4,4,4,4,4,5,5,5,5,5,6,5,5,4,3,2,2,
2,2,2,0,0,0,0,0,1,1,2,2,3,3,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0;

4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,0,1,1,1,0,0,0,0,7,7,
7,0,0,0,0,7,7,5,5,5,6,6,6,7,0,0,1,1,1,1,1,2,2,2,3,3,3,2,2,2,2,2,2,2,2,
2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0,0;

4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,5,5,6,6,6,6,6,6,7,0,0,0,0,0,0,0,
7,7,7,7,6,6,7,7,7,7,7,7,7,7,7,7,7,7,0,0,1,1,2,2,2,2,3,3,3,3,3,3,3,3,3,
3,3,3,2,1,0,0,1,0,0,0,0,0,0,0,0,1,1,1,2,2,2,2,3,3,3,4,4,4,4;

4,4,4,4,4,4,5,4,4,5,5,6,6,6,6,6,7,0,0,7,7,7,7,7,7,7,7,7,7,7,7,0,7,7,0,1,
2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,2,1,0,0,7,6,6,6,6,6,7,0,0,0,0,0,0,0,0,0,
0,1,2,2,2,2,3,4,4,4,4,4,4,4,4,4,2,2,2,1,1,2,2,2,2,3,4,4,4,0;

4,4,4,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,7,1,1,1,1,1,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,7,7,5,5,5,5,6,6,7,7,7,7,7,7,7,7,0,1,2,2,2,2,3,3,3,
2,2,1,1,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,0,0,0;

5,5,6,5,6,5,6,5,5,5,5,5,4,4,2,2,2,2,3,3,4,6,6,6,6,6,6,6,6,6,6,7,7,7,7,0,
0,0,1,1,1,1,1,2,1,2,1,1,0,0,0,7,6,6,6,6,6,6,6,5,4,4,6,6,6,6,0,0,0,0,0,0,
1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,4,4,4,4,4;

5,6,6,6,6,6,7,0,7,6,6,6,6,6,7,7,0,0,1,1,1,1,2,1,1,1,0,0,0,7,7,6,6,5,3,3,
4,5,6,7,7,5,5,5,6,6,7,1,1,1,1,1,2,2,2,1,2,2,2,2,2,2,2,2,3,3,3,3,4,4,4,5,5,
5,6,5,5,5,4,4,3,2,2,2,2,0,0,1,1,2,2,3,4,4,4,4,4,4,4,4,4,4,4,0;

4,5,5,6,6,6,6,6,7,0,0,0,0,0,0,0,0,0,0,6,6,6,4,4,5,6,6,6,0,0,0,6,5,4,5,6,6,
6,7,0,0,0,0,0,0,1,2,2,3,3,2,2,0,0,0,1,2,2,2,3,4,4,2,2,2,2,0,0,0,0,0,0,
0,0,1,1,2,2,3,3,4,0;

4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,4,5,5,6,6,6,6,6,6,6,6,6,7,0,0,0,0,0,0,7,6,6,6,
6,7,7,7,7,7,7,7,0,7,7,7,7,7,7,7,7,0,0,1,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,
3,3,3,1,0,0,1,0,0,0,0,0,0,1,1,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,0;

4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,6,6,6,6,7,7,7,7,7,7,7,7,0,7,0,7,7,5,5,6,6,6,7,
7,0,1,1,2,1,2,2,3,3,3,3,3,4,3,4,3,4,4,0,0,7,0,7,0,7,7,7,7,7,6,6,5,6,5,5,
4,3,3,2,2,2,1,1,3,3,4,3,4,3,3,3,3,3,3,3,2,2,2,2,1,1,0,0,0;

6,7,7,7,7,7,7,7,7,7,7,6,6,6,6,6,
6,6,6,6,6,6,4,4,4,4,4,4,4,4,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2,3,3,
3,3,3,3,3,3,3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,0;

4,4,5,5,5,5,5,6,7,7,0,0,0,1,1,1,
1,1,1,0,0,0,0,0,7,7,7,7,7,7,0,0,0,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0;

4,4,5,5,6,6,6,6,6,7,7,7,7,7,7,0,1,1,2,2,3,3,3,3,3,2,0,0,0,0,0,0,0,0,0,0,0,
7,6,6,6,6,6,6,7,6,7,7,6,5,5,5,5,6,6,7,0,1,1,1,1,1,2,2,2,2,3,2,3,3,2,2,1,
7,7,7,0,1,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0;

4,4,4,4,5,6,6,6,6,6,7,1,0,0,0,0,0,7,5,4,4,4,4,5,5,5,6,6,6,6,6,6,0,1,1,1,
0,0,0,0,0,0,0,0,0,0,0,7,5,5,5,5,6,6,7,0,1,1,1,1,1,1,2,2,2,2,3,3,3,4,4,
4,4,3,0,0,0,0,0,0,1,2,2,2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4;

4,4,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,0,1,1,1,1,0,0,0,7,7,7,7,0,

0,0,0,0,0,7,7,5,5,5,5,6,6,7,0,0,1,1,1,1,1,2,2,2,2,2,3,3,3,3,2,2,2,2,2,2,
2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0;

4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,4,5,6,6,6,6,7,0,0,0,0,0,7,7,7,7,7,0,7,7,5,5,6,
6,7,0,1,1,1,1,2,2,3,3,3,3,4,3,4,3,7,0,7,0,7,7,7,7,6,6,5,5,5,5,4,3,2,2,1,
1,3,3,4,3,3,3,3,3,4,4,4,4,4,4,3,2,2,2,2,1,0,1,0,0,0,0,0,0,0,0,0;

4,4,4,4,5,5,5,5,5,5,5,5,5,5,6,7,7,
7,7,7,7,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,
2,2,2,3,2,2,3,3,3,3,4,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0;

4,5,6,6,6,6,6,6,7,0,1,1,1,0,0,0,0,7,5,5,5,6,6,6,6,6,6,6,0,1,1,1,1,1,1,
1,1,1,0,7,6,6,6,6,6,5,5,5,5,6,6,6,6,6,6,7,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,3,3,3,4,4,4,4,4,4,5,5,4,3,3,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0;

4,5,5,5,5,5,5,5,5,4,4,3,3,3,3,4,3,3,3,4,4,4,5,5,6,6,6,6,7,7,7,7,7,7,7,7,
6,5,5,5,5,5,5,5,5,6,6,6,6,6,7,7,0,0,1,1,1,1,1,1,1,1,1,1,0,7,7,0,7,7,7,7,7,
0,0,1,1,3,3,3,3,3,3,3,3,3,2,2,2,1,1,1,1,1,1,1,1,1,2,3,3,4,4,4;

4,4,4,4,4,4,5,6,6,6,6,6,7,0,0,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,0,1,1,0,0,7,
0,7,0,0,1,1,2,2,3,3,3,4,3,4,2,2,2,1,0,0,1,1,2,2,2,2,3,4,4,4,4,5,5,5,5,6,
6,5,4,2,2,2,2,2,2,2,0,0,0,0,0,0,0,1,1,2,2,3,4,4,4,4,4,4,4,0,0;

4,4,4,5,6,6,6,6,7,0,0,0,0,0,0,0,4,4,4,4,4,5,5,6,6,6,6,6,0,1,1,0,0,0,0,
0,0,0,0,0,0,0,0,0,5,5,5,5,6,7,7,0,6,5,5,6,7,0,1,1,1,1,1,2,3,3,2,1,1,2,3,
3,3,4,0,0,0,1,2,2,3,3,4,0;

5,4,5,6,6,6,6,7,0,0,1,1,0,0,0,0,7,5,5,5,5,5,6,6,6,6,7,0,1,1,1,1,1,1,
1,1,1,1,1,1,1,0,0,0,0,6,6,6,5,5,5,5,5,6,6,6,6,7,0,0,7,6,5,6,6,7,0,1,1,2,
2,2,2,3,3,2,2,2,2,2,2,3,3,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4;

5,4,6,6,6,6,6,6,6,6,
6,6,6,6,6,6,6,6,6,6,6,6,6,6,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,3,4,4,0,0;

5,5,6,5,5,6,5,5,5,5,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
7,7,7,7,7,7,7,7,7,7,7,7,7,7,0,0,1,1,1,1,1,1,1,2,2,2,3,3,3,3,3,3,3,3,3,3,
3,4,4,0,0,0;

5,6,6,6,7,0,0,6,6,6,6,6,6,6,7,7,0,0,0,1,0,0,0,7,5,5,6,6,6,7,0,1,1,1,1,1,
1,1,2,2,3,3,4,4,3,4,1,1,0,0,1,0,1,1,2,3,4,4,4,4,4,4,5,5,5,5,5,6,5,4,3,2,2,
2,2,0,0,0,0,0,0,1,2,2,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0,0;

4,5,5,5,5,5,6,6,6,6,6,7,7,7,7,7,7,7,
7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,3,
3,3,2,2,2,2,2,1,1,1,1,1,0;

4,6,6,6,6,6,6,6,7,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,7,7,6,5,5,6,6,6,7,0,1,1,1,1,1,2,2,2,3,
3,3,2,2,2,1,0;

4,4,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,0,0,0,0,1,2,2,2,2,3,4,4,4,2,2,
2,2,2,1,1,1,0,0,0,0,0,0,0,7,7,7,7,6,6,6,6,6,5,4,4,4,5,6,6,6,6,6,7,0,0,0,0,
0,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,4,3,4,4,4,4,4,4,4,4;

5,6,6,6,6,6,0,0,0,0,0,0,0,7,6,5,5,5,5,5,6,5,6,6,6,6,6,7,7,7,0,0,0,0,0,0,
0,0,0,1,1,2,2,2,3,3,3,4,4,4,4,4,4,3,2,2,2,1,1,1,1,1,1,1,0,1,0,0,0,0,0,1,1,
1,2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,0;

4,4,4,4,5,5,6,7,7,7,7,7,7,7,7,6,6,5,5,4,4,5,5,5,6,6,6,6,7,0,0,0,1,1,1,0,
0,0,7,7,7,7,7,7,7,5,5,5,6,6,7,7,0,1,1,1,1,1,1,3,3,3,3,3,3,2,1,1,1,1,1,1,
1,1,1,2,2,3,3,4;

4,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,7,0,0,0,1,1,2,2,2,3,4,4,4,2,2,2,2,2,
1,1,1,0,0,0,0,0,0,0,0,7,7,7,7,6,6,6,6,5,4,4,4,5,6,6,6,6,7,7,7,7,7,7,0,
1,2,2,3,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,0;

4,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,7,1,1,1,1,1,1,0,1,0,0,0,0,0,0,
0,0,0,0,7,0,0,7,7,7,5,5,6,5,6,6,6,6,6,6,7,1,1,1,1,1,1,2,2,1,2,2,2,2,2,2,
2,2,2,2,2,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0;

5,6,6,6,6,6,7,7,6,6,6,6,6,6,6,6,6,6,6,7,7,0,1,1,1,1,2,2,2,2,2,1,0,0,7,
7,6,6,6,6,6,6,6,7,1,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,4,4,4,5,6,6,6,6,5,5,4,
3,2,2,2,2,2,2,2,0,0,1,1,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,4,4,0,0;

6,6,7,0,0,0,0,0,7,0,7,5,5,5,5,6,6,6,7,7,0,0,0,0,0,0,0,1,2,2,3,4,4,4,4,
2,2,1,1,1,1,1,0,1,7,6,6,6,7,7,7,5,5,5,5,6,6,7,7,1,1,1,1,1,1,2,2,3,3,3,3,
3,2,2,2,0,0,1,1,2,3,4,4,4,4,4,4,4,4,4,4,3,3,4,4,4,4,4,4,4,4,4,4; % size60

5,6,5,5,6,5,5,5,5,5,4,3,2,2,2,3,3,4,4,6,6,6,6,6,6,6,6,6,7,7,0,0,1,
1,1,1,2,1,1,2,1,1,2,1,1,0,0,0,7,6,6,6,6,6,6,5,5,4,4,4,5,6,6,6,6,6,0,
0,0,0,1,0,0,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,4,4;

4,4,4,4,5,4,4,5,6,6,6,7,0,7,5,5,5,6,6,0,1,0,0,0,0,1,0,7,0,0,0,0,0,7,5,
5,6,7,7,5,6,7,0,1,1,1,1,2,3,3,7,7,6,5,5,5,5,4,3,2,1,3,3,2,1,1,3,4,4,4,4,
4,4,3,4,5,4,4,4,4,5,4,2,2,1,1,1,3,4,3,2,2,2,1,0,0,1,0,0,0,0;

4,6,6,6,6,7,7,0,0,0,7,0,0,0,7,0,6,6,5,5,5,5,5,5,6,6,6,6,6,6,7,7,7,0,0,7,
0,0,0,0,0,0,1,1,1,3,3,4,4,4,4,4,3,2,2,2,1,1,1,1,0,1,1,0,0,0,1,0,0,0,1,0,
0,1,1,1,3,3,4,4,5,4,4,4,5,4,5,4,4,4,3,3,4,3,4,4,4,4,4,4,4,4,0;

4,4,4,5,5,6,6,6,6,6,6,7,7,0,0,0,6,6,6,6,6,6,6,6,6,6,7,7,7,0,0,0,1,1,0,
0,0,0,0,7,5,5,6,6,6,7,7,7,3,3,3,2,2,2,1,1,3,4,4,4,4,4,5,5,4,4,4,4,3,3,3,
2,2,2,2,2,2,2,2,2,2,4,4,4,3,3,2,2,2,2,2,2,1,1,0,0,0,0,0,0,0;

6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
6,6,6,6,6,7,7,6,6,7,7,7,7,7,7,3,3,3,3,3,3,2,2,3,3,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2;

4,4,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,6,6,7,7,0,1,1,
1,1,1,0,0,1,0,0,7,7,7,7,7,0,0,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,3,3,3,3,3,3,3,4,0,0,0,0,0;

4,4,4,4,4,5,6,6,6,7,7,7,7,7,7,7,5,5,4,4,5,5,5,6,6,6,6,7,0,0,0,1,1,1,0,0,
0,7,7,7,7,7,6,5,5,6,6,6,7,7,1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,3,2,1,1,1,1,1,
1,1,1,2,3,3,4,0,0;

4,5,5,6,6,6,6,6,7,0,0,0,1,0,0,0,0,7,6,6,5,4,4,5,6,6,6,6,7,7,0,0,6,5,6,
6,6,7,7,0,0,0,0,0,1,1,1,2,3,3,2,0,0,1,1,2,3,3,4,2,2,0,0,0,7,0,0,0,0,0,1,
1,2,3,3,3,3,4,0;

5,5,5,5,5,6,
6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,3,3,3,3,3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,
2,2;

6,6,6,7,0,0,0,0,0,7,0,6,5,5,5,5,5,6,6,6,6,7,7,0,7,0,0,0,0,1,1,3,3,4,4,3,

4,4,2,2,1,1,0,1,0,1,6,6,6,6,7,7,7,6,5,5,5,6,6,6,7,7,0,1,1,1,1,1,1,1,2,2,2,
3,3,3,2,2,2,1,0,1,1,2,3,3,4,4,4,4,4,4,5,4,4,4,4,3,3,4,4,3,4;

5,6,6,6,6,7,6,4,4,4,4,5,6,6,6,6,7,7,0,7,6,6,4,4,4,4,4,4,4,4,4,4,6,6,6,
6,6,7,7,0,0,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,1,2,2,3,4,4,4,4,4,
4,4,4,2,2,0,1,1,1,2,3,3,4,4,2,2,1,1,2,2,3,4,4,4,4,4,4,4,0,0;

4,4,6,0,1,2,1,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,1,2,
2,2,2,2,2,4,0,0;

4,4,5,6,6,6,6,7,7,6,6,6,6,7,6,7,6,7,7,7,0,1,1,1,1,1,1,1,0,0,0,7,6,5,3,4,
5,5,6,6,7,7,6,5,5,6,6,6,7,7,1,1,1,1,1,2,2,2,2,1,1,2,2,2,2,2,3,3,3,4,3,4,
4,4,5,5,5,5,6,5,5,5,5,4,3,2,2,2,2,0,0,1,1,1,3,3,4,4,4,4,4,4;

4,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,0,0,0,1,1,1,0,0,0,0,0,7,
7,7,7,0,0,0,0,0,7,5,5,6,6,6,7,0,1,1,1,1,1,2,2,2,3,3,3,3,3,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,3,3,3,3,3,4,0,0,0;

4,4,4,4,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,0,0,0,1,1,2,2,2,3,4,4,4,2,
2,2,2,2,2,1,1,0,1,0,0,0,0,0,0,0,0,7,7,7,7,6,6,6,6,5,4,4,4,5,6,6,6,7,0,0,
0,0,0,1,1,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,0;

4,4,4,4,4,5,5,5,5,6,6,6,6,6,7,7,7,7,7,7,0,1,2,2,3,3,3,3,2,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,7,5,5,6,6,6,7,6,6,6,5,5,5,6,6,6,7,7,0,1,1,1,1,2,2,2,3,2,
2,0,0,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0;

4,5,6,6,6,6,7,0,1,0,0,0,0,5,5,5,5,5,6,6,6,6,6,6,0,1,1,1,1,1,1,1,1,1,1,1,0,
0,0,0,7,6,6,5,5,5,5,6,6,6,7,7,0,7,5,6,5,6,6,6,7,0,1,1,1,2,1,2,2,2,3,3,3,
2,1,2,1,2,2,2,2,2,3,3,4,4,4,4,4,4,4,5,4,4,4,4,4,4,4,4,4,4,4,0;

4,4,4,4,4,4,4,5,4,4,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,0,1,1,1,1,1,0,0,
0,1,0,0,0,0,0,0,0,7,0,0,0,7,0,6,6,5,5,6,6,6,6,6,6,6,6,7,1,1,1,1,1,1,1,
2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,4,3,4,4,3,4,4,4,4,4,4,4,4,4,0,0,0;

5,5,5,5,5,5,6,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,6,6,7,7,7,7,7,7,0,7,0,0,
7,0,0,0,0,0,0,0,0,1,0,0,1,0,0,1,1,1,1,2,1,2,1,2,2,2,2,2,2,2,2,2,2,3,2,2,
2,3,2,3,3,3,3,3,3,4,0,0,0;

5,5,6,6,6,6,6,6,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,6,6,7,7,6,5,5,6,6,6,
6,6,6,7,0,1,1,1,1,1,3,3,2,2,2,0,0,0,0,0,0,0,0,0,0,7,0,1,1,2,2,2,3,3,4,4,
4,0;

4,4,4,4,4,4,5,4,5,4,4,5,4,4,5,5,5,5,6,6,6,6,6,7,0,0,7,6,7,7,7,7,7,7,7,
0,7,7,7,7,0,1,1,2,2,3,3,3,3,3,4,4,3,2,1,0,0,0,1,0,1,7,6,6,6,6,6,7,0,
0,0,0,0,0,0,0,0,1,1,2,3,3,4,4,4,4,4,4,5,3,2,2,2,1,1,2,2,2,3,3;

5,5,6,6,6,7,7,7,7,7,7,7,7,5,6,5,5,5,5,5,6,5,6,6,7,7,7,0,0,0,1,1,1,1,1,1,
1,1,0,7,7,7,0,7,7,7,0,0,0,1,1,1,2,2,3,3,3,3,3,3,3,3,2,2,2,1,1,1,2,1,1,1,
2,2,3,3,4,4,4,4,5,5,5,5,4,5,4,4,4,3,3,3,3,4,3,3,4,4,4,0,0,0;

4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,1,1,2,2,3,4,4,3,2,2,2,2,1,
1,1,0,0,0,0,0,0,0,0,7,7,7,6,6,6,5,4,4,4,6,6,6,6,7,7,7,7,7,7,0,1,2,2,
2,3,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,4,3,4,4,4,4,4,4,4,4,4;

4,4,4,4,4,4,4,4,5,4,4,5,4,4,4,5,4,4,4,4,5,4,5,5,6,6,6,6,6,6,6,7,7,0,0,0,1,
0,0,0,0,0,7,7,7,7,7,7,7,7,7,7,7,0,1,1,2,2,2,3,3,3,3,3,3,3,3,3,3,3,4,
3,2,0,0,0,0,1,0,0,1,0,0,0,0,1,1,1,1,2,2,2,3,3,4,4,4,4,4,4,0;

6,7,7,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,2,3,3,4,4,4,4,4,4,
 5,4,4,3,2,2,2,2,1,0,1,1,1,2,2,3,3,4,2,1,1,2,3,3,4,4,4,4,4,4,0;

4,6,6,6,6,6,6,0,1,1,1,0,0,0,0,0,6,6,5,5,5,5,6,6,6,6,6,6,7,1,1,1,1,1,1,
 1,1,1,1,1,0,0,0,0,6,6,6,6,5,5,5,6,6,6,6,6,7,1,1,1,1,2,2,1,2,2,2,2,2,2,
 2,2,2,2,2,2,2,3,3,3,4,4,4,4,4,4,4,5,5,4,4,3,4,4,4,4,4,4,4,4;

4,4,4,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,0,0,1,1,1,1,
 0,0,0,0,7,7,7,0,0,0,0,6,6,5,5,6,6,6,7,7,0,1,1,1,1,1,1,2,2,2,3,3,3,3,3,3,
 2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4;

4,4,4,5,5,5,5,6,6,6,6,6,6,6,6,7,7,7,7,7,7,0,1,1,3,3,3,3,2,1,0,1,0,0,0,0,
 0,0,0,0,0,0,0,6,5,6,6,6,7,7,6,5,5,6,6,6,7,0,1,1,1,1,2,2,2,2,3,2,1,7,0,1,
 2,2,2,2,3,3,3,4,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0,0;

5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,6,0,0,0,0,0,0,0,0,0,0,0,
 0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
 2,2,4,0;

5,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,6,6,7,7,7,7,7,
 0,7,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,1,1,1,1,1,2,2,1,2,2,2,2,2,2,
 3,2,2,3,3,3,3,3,3,3,3,3,4,4,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,0,0;

5,6,6,6,6,7,0,0,1,1,0,0,0,0,6,5,5,5,5,5,5,6,6,6,7,0,1,1,1,1,1,1,1,1,
 1,0,0,0,0,7,6,6,6,5,5,5,5,5,6,6,6,6,7,0,7,6,5,6,6,7,7,0,1,1,2,2,2,3,
 3,3,2,1,2,2,2,2,2,2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4;

4,5,5,5,5,6,6,5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,0,0,1,1,2,2,3,4,5,3,2,2,
 2,2,2,2,1,1,1,0,1,0,0,0,0,0,0,0,7,7,7,7,7,6,6,6,6,5,3,4,4,5,6,6,6,7,0,0,
 0,0,1,1,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4;

5,5,5,4,5,4,3,4,3,3,4,3,4,4,5,6,6,7,7,0,7,7,5,6,5,5,5,6,6,7,0,1,0,1,1,1,
 0,7,0,7,7,0,0,1,2,3,2,3,3,3,2,2,1,1,2,1,2,1,2,3,4,0,7,6,5,6,5,6,5,6,6,
 7,7,7,6,7,6,5,4,4,3,3,4,3,4,5,5,5,4,5,4,3,2,2,1,1,1,2,1,3,3;

6,6,6,6,6,6,6,6,6,6,7,0,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,6,6,6,6,6,6,
 7,0,1,1,1,1,2,2,2,2,3,3,3,2,2,2,2,0,0,0,0,0,0,0,0,0,0,1,2,2,2,3,4,4,4,4,
 4,0,0,0;

5,4,4,5,5,6,0,1,1,1,1,0,1,0,0,0,0,
 0,0,0,0,0,0,0,6,6,5,5,5,6,6,6,6,6,6,6,6,1,1,1,1,2,2,2,2,2,2,2,2,2,2,
 3,3,2,3,3,3,3,3,4,4,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0;

5,5,6,6,5,5,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,0,7,7,7,7,
 7,7,7,0,7,7,7,0,0,0,0,0,0,1,1,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,3,3,4,
 3,3,3,3,3,4,3,3,3,3,3,3,4,3,3,3,4,3,3,3,3,3,3,3,4,4,4,4,4,0;

4,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,7,7,0,7,0,0,1,1,2,2,3,3,4,2,2,2,2,1,
 1,1,0,0,0,0,0,0,0,0,7,7,7,6,6,6,4,4,4,5,6,6,6,6,6,7,7,7,7,7,0,1,1,2,3,2,
 2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,0,0;

4,4,4,5,4,4,5,4,4,5,4,4,5,4,4,5,5,5,6,6,6,6,6,7,0,0,7,7,7,6,7,7,0,7,7,0,
 7,7,0,7,0,7,0,1,1,2,3,3,3,3,3,3,4,4,3,3,4,3,3,0,0,1,0,1,0,0,1,7,6,6,6,6,
 6,0,0,0,0,0,0,0,0,0,1,1,2,2,2,3,4,4,4,4,4,4,4,2,2,1,1,2,2;

4,4,5,5,5,5,6,6,6,6,6,6,6,7,0,1,1,0,1,0,0,1,0,0,0,0,0,0,0,0,7,0,0,0,7,
 7,6,5,5,5,6,6,6,6,6,6,7,7,7,7,7,3,3,3,3,3,3,2,2,2,2,2,1,1,1,2,3,3,4,4,4,
 3,4,4,4,4,4,4,4,4,4,4,5,4,4,4,5,5,4,3,2,2,2,2,2,2,1,1,1,1;

4,4,4,4,5,4,4,4,5,4,4,4,5,4,4,4,5,4,4,4,5,4,4,5,5,5,6,6,6,6,6,7,7,0,0,0,
1,0,0,0,7,7,7,0,7,7,7,7,7,7,7,7,7,7,7,7,7,0,7,7,0,1,2,2,2,2,2,3,3,3,3,3,3,
3,4,3,3,2,1,1,0,0,0,1,0,0,1,0,0,1,1,1,1,2,2,2,3,3,4,4,4,4,0;
4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,4,5,6,6,6,6,7,6,7,7,7,7,7,0,7,7,0,7,0,7,6,5,6,
6,6,7,0,0,1,1,2,1,2,2,3,3,3,3,3,3,4,3,4,3,4,3,0,0,0,0,0,0,0,6,6,6,7,0,0,
0,0,0,0,0,0,1,1,1,2,3,4,4,4,4,4,4,2,1,2,3,4,4,4,4,0,0,0,0,0;
4,4,4,4,4,5,5,5,6,6,6,6,6,7,7,0,0,7,5,5,5,5,6,6,6,6,6,6,7,1,1,1,1,1,0,0,
0,1,0,0,0,0,0,0,0,7,0,0,0,0,7,7,6,5,5,5,5,6,6,7,7,1,1,1,1,1,1,1,1,2,2,2,
2,2,3,3,3,2,1,0,1,1,2,2,3,3,3,4,4,3,4,4,4,4,4,4,4,4,4,4,4;
4,4,5,4,4,4,4,4,4,4,4,4,5,4,4,4,4,5,5,5,6,6,6,6,6,7,7,0,0,0,0,1,0,0,7,7,6,
7,7,7,0,7,7,7,6,5,5,6,6,7,7,0,0,1,1,1,1,2,1,2,2,2,2,3,3,3,3,3,3,4,4,3,
4,2,0,0,0,0,0,1,0,0,0,0,1,1,1,2,2,3,3,4,4,4,4,4,4,4,4,4,4,0,0;
4,4,4,4,4,4,4,4,4,4,6,6,6,6,6,0,0,0,6,6,6,6,6,6,6,6,6,6,6,6,6,7,6,7,7,7,
7,7,1,1,1,1,1,0,0,7,0,7,0,7,0,4,3,4,3,4,3,4,4,5,5,5,5,5,3,3,3,3,2,3,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,4,4,4,2,2,2,2,2,0,0,0,0,0,0,0,0,0;
4,4,4,5,5,6,6,6,7,0,0,0,4,5,5,5,5,6,6,6,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,7,0,4,5,5,5,6,7,7,6,5,6,7,3,2,1,2,3,3,2,1,1,1,0,4,3,4,4,4,4,4,4,4,4,
4,4,4,4,4,5,5,4,2,2,2,1,1,1,1,0,4,4,4,3,2,2,2,1,1,0,0,0,0,0;
4,4,5,5,6,6,6,7,7,7,7,7,7,7,7,6,5,5,5,4,4,5,5,6,6,6,6,7,0,0,1,1,1,1,1,0,
0,0,0,0,7,7,7,7,7,7,7,7,0,1,1,1,2,2,2,3,3,3,3,3,3,3,1,1,1,1,1,1,1,2,2,
2,3,3,4,0,0;
5,6,6,7,0,0,6,6,6,6,6,6,7,7,7,7,0,0,1,0,0,0,0,7,5,5,6,6,6,7,7,1,1,1,1,
1,1,3,3,4,4,3,4,2,2,1,0,1,0,0,0,1,1,2,2,2,3,4,4,4,5,4,4,5,5,5,5,6,5,5,
5,4,3,2,2,2,2,0,7,0,0,0,1,0,1,1,2,3,4,4,4,4,4,4,4,4,4,4,4,4,4;
5,6,6,6,7,7,0,0,0,0,7,0,0,7,6,5,5,5,5,5,5,6,6,5,6,6,6,6,7,7,7,0,7,0,0,0,
7,0,0,1,1,1,3,3,4,4,4,3,3,2,2,2,1,1,1,1,1,0,1,1,0,0,1,0,0,0,1,0,0,1,1,2,
2,3,3,4,4,4,4,5,4,4,4,5,4,4,5,4,4,4,3,3,3,4,4,3,4,4,4,4,4,4,4]; % size 30

Lampiran 4 : Dataset variasi ukuran 24,28,30,40

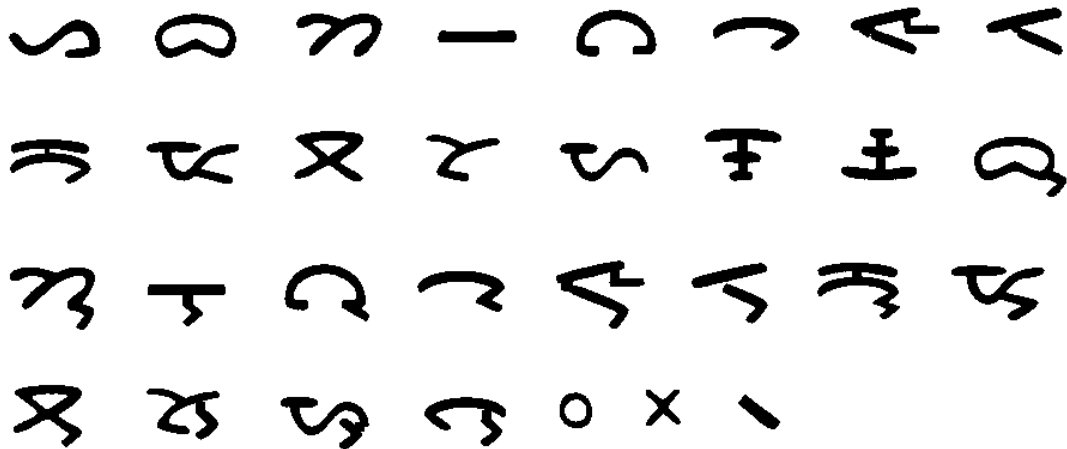
5 0 7 - 0 7 4 1 3 4 2 2 5 =
= 0 3 7 7 0 2 4 1 3 4 2 2 5 0
o x \

5 0 7 - 0 7 4 1 3 4 2 2
5 = = 0 3 7 7 0 2 4 1 3 4 2
4 5 0 0 x \

5 0 7 - 0 7 4 1 3 4 2 2
5 = = 0 3 7 7 0 2 4 1 3 4
4 5 0 0 x \

5 0 7 - 0 7 4 1 3 4 2 2
2 5 = = 0 3 7 7 0 2 4 1 3
= 4 5 0 0 x \

Lampiran 5 : Dataset variasi ukuran 50



Lampiran 6 : Dataset Vriasi Ukuran 60 pt

S O 2 1 C C
A V H R X 2 S
H H O 2 F C
C A V H R X R
S C O X 1

Lampiran 7 : Dataset variasi Tulisan tangan

Handwritten Variations										
u	u	u	u	u	u	u	u	u	u	a
h	h	h	h	h	h	h	h	h	h	ha
m	m	m	m	m	m	m	m	m	m	ma
n	n	n	n	n	n	n	n	n	n	na
r	r	r	r	r	r	r	r	r	r	ra
t	t	t	t	t	t	t	t	t	t	ta
b	b	b	b	b	b	b	b	b	b	ba
c	c	c	c	c	c	c	c	c	c	wa
i	i	i	i	i	i	i	i	i	i	i
v	v	v	v	v	v	v	v	v	v	nga
l	l	l	l	l	l	l	l	l	l	la
p	p	p	p	p	p	p	p	p	p	pa
s	s	s	s	s	s	s	s	s	s	sa
d	d	d	d	d	d	d	d	d	d	da
g	g	g	g	g	g	g	g	g	g	ga
u	u	u	u	u	u	u	u	u	u	u
y	y	y	y	y	y	y	y	y	y	ya
n	n	n	n	n	n	n	n	n	n	nya
j	j	j	j	j	j	j	j	j	j	ja

Lampiran 8 : Dataset Variasi tulisan tangan sebagai testing

