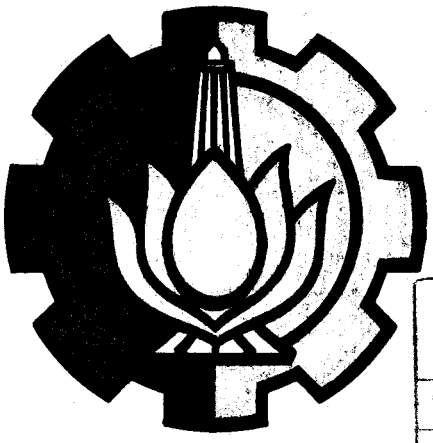
TUGAS AKHIR

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENDETEKSIAN DUPLIKASI PADA BASIS DATA CITRA DOKUMEN DENGAN TEKNIK SHAPE CODING



DISUSUN OLEH :

MOCHAMAD NDARU PURNOMO SIDI
NRP. 2693.100.049

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2000

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENDETEKSIAN DUPLIKASI PADA BASIS DATA CITRA DOKUMEN DENGAN TEKNIK SHAPE CODING

TUGAS AKHIR
Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Inormatika
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Esther Hanaya, M.Sc.	Rully Soelaiman, S.Kom
Nip. 130.816.212	Nip. 132.085.802

SURABAYA
2000

kupersembahkan tugas akhir ini untuk

papa dan mama yang telah

mendidikku dan membesarkanku dengan

penuh kasih sayang

juga untuk mas novi dan dik angga yang tercinta

ABSTRAKSI

Teknologi pengolahan citra dokumen telah berkembang sedemikian rupa sehingga saat ini sudah bukan hal yang istimewa lagi bagi suatu organisasi untuk mengubah sejumlah besar dokumen kertas yang dimilikinya ke dalam bentuk citra dokumen melalui proses scanning, dan menyimpannya ke dalam sebuah basis data.

Ke dalam basis data ini kemudian ditambahkan informasi index untuk digunakan dalam proses pencarian dan pencegahan duplikasi. Tingkat kerumitan proses penambahan informasi index bergantung dari tujuan yang hendak dicapai. Mulai dari tujuan sederhana seperti sekedar menyimpan dokumen ke dalam bentuk citra (*archival purpose*), sampai ke tujuan yang lebih kompleks seperti pembuatan sistem yang mampu mengambil informasi dalam citra dokumen secara otomatis berdasarkan query.

Pemeliharaan integritas basis data semacam ini sangatlah sulit, terutama dalam lingkungan terdistribusi, dimana salinan dari dokumen yang sama bisa mengalami proses scanning berulang kali pada situs yang berlainan.

Dalam Tugas Akhir ini, digunakan suatu metode untuk mendeteksi dan mencegah terjadinya duplikasi citra dokumen dalam suatu basis data yang menampung ribuan citra dokumen. Metode ini didasarkan pada teknik shape coding yang mengambil sebuah signature dari tiap citra dokumen dan menggunakannya sebagai informasi index ke dalam basis data. Perangkat lunak yang dibuat akan mampu mendeteksi duplikasi citra yang identik dan berbeda dalam hal resolusi dan kualitas (*image variant*). Metode ini mempunyai sejumlah keuntungan dibandingkan dengan OCR (*optical character recognition*) dan metode lain yang didasarkan pada pengenalan (*recognition-based*) dalam hal kecepatan dan toleransi terhadap degradasi citra [1].

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT yang berkat rahmat dan karunia-Nya, akhirnya Tugas Akhir yang berjudul "**Perancangan Dan Pembuatan Perangkat Lunak Pendeteksian Duplikasi Pada Basis Data Citra Dokumen Dengan Teknik Shape Coding**" ini dapat diselesaikan. Penulisan Tugas Akhir ini disusun guna memenuhi salah satu persyaratan untuk meraih gelar sarjana pada Jurusan Teknik Informatika Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya.

Ucapan terima kasih penulis sampaikan untuk:

1. Bapak Ir. Arif Djunaidy, M.Sc, Ph.D, selaku Ketua Jurusan Teknik Informatika-ITS dan dosen wali selama perkuliahan di Jurusan Teknik Informatika-ITS.
2. Bapak Rully Soelaiman S.Kom, selaku dosen pembimbing yang telah memberikan bimbingan dan motivasi.
3. Ibu Ir. Ester Hanaya, selaku dosen pembimbing yang telah memberikan bimbingan dan motivasi.
4. Bapak dan Ibu Dosen di Jurusan Teknik Informatika-ITS.
5. Seluruh staf tata usaha Jurusan Teknik Informatika-ITS.
6. Keluarga di Sidoarjo, atas semua doa dan dorongannya demi selesainya Tugas Akhir ini.
7. Sucahyo Aji Condro, atas segala saran, bantuan, kritikan dan kecaman yang diberikan kepada penulis selama pengerjaan Tugas Akhir ini, dalam wujudnya yang negatif, menjengkelkan, sekaligus sugestif dan menimbulkan motivasi yang tinggi.
8. Edi Setiawan, atas semua dorongan, motivasi dan humor-humornya.
9. Semua rekan kehidupan malam yang bersama-sama mengerjakan Tugas Akhir, atas kerelaannya meminjamkan komputernya masing-masing kepada penulis untuk hal-hal di luar pengerjaan Tugas Akhir.
10. Edi Susanto, atas pinjaman printer HP Deskjet-nya.
11. Ibu SM atas karyanya yang telah banyak memberikan inspirasi dan motivasi kepada penulis.

12. Seluruh warga TC angkatan 93 yang telah membantu dan memotivasi penulis, serta teman-teman di lab Komisi, yang telah memberikan semangat dan fasilitas komputer dan internet untuk mendownload program-program yang diperlukan.
13. Semua pihak yang telah ikut membantu penulis selama menjalani studi dan tinggal di Surabaya.

Salah satu ciri karya manusia adalah tidak akan pernah sempurna karena manusia memiliki kelebihan dan kekurangan masing-masing. Demikian pula halnya dengan Tugas Akhir ini masih banyak kekurangannya. Untuk itu kritik dan saran yang bermanfaat sangat diharapkan demi penyempurnaannya. Harapan penulis juga, semoga tulisan ini bermanfaat bagi pembaca.

Surabaya, Januari 2000

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

Bab ini membahas mengenai beberapa hal dasar yang meliputi latar belakang, permasalahan, tujuan, batasan permasalahan, metodologi serta sistematika pembahasan buku Tugas Akhir. Dari pembahasan tersebut diharapkan, gambaran umum permasalahan dan pemecahan yang diambil akan dapat dipahami dengan baik.

1.1 Latar Belakang

Perkembangan pesat dalam teknologi komputasi dan kapasitas media penyimpanan, telah semakin memungkinkan penyimpanan dan administrasi citra dokumen dalam jumlah besar ke dalam suatu basis data. Kini keberadaan basis data yang menampung ribuan citra dokumen sudah bukan merupakan hal yang istimewa lagi. Dan dalam lingkungan basis data terdistribusi, hal-hal seperti manajemen, penyimpanan dan pengambilan citra dokumen menjadi isu yang sangat penting.

Dalam metode tradisional, proses index entry untuk basis data citra dokumen umumnya dilakukan secara manual, dengan informasi index seperti data administratif, nomor ID dokumen, beberapa kata kunci dan lain-lain. Namun ketika harus berhadapan dengan basis data yang menampung ribuan citra dokumen, metode index entry seperti ini sangat menghabiskan biaya.

Dalam suatu sistem berbasis citra, kita tidak bisa menerapkan cara-cara tradisional dalam hal organisasi, pencarian dan pengambilan data. Ini disebabkan karena besarnya ukuran yang dimiliki suatu citra. Suatu dokumen dalam bentuk citra

yang terdiri dari kumpulan pixel, tidak memiliki informasi isi yang biasanya ada pada dokumen dalam bentuk teks. Jika tidak bisa menyediakan informasi index yang akurat dan unik untuk setiap citra, maka proses index entry, administrasi dan perawatan basis data akan menjadi sulit.

1.2 Perumusan Masalah

Dalam suatu lingkungan basis data terdistribusi, terdapat kemungkinan bagi beberapa dokumen untuk memiliki salinan yang sama di beberapa situs. Ketika dilakukan proses pengubahan dokumen menjadi citra dokumen dan ditambahkan ke dalam basis data, besar kemungkinan salinan-salinan yang sama tersebut dimasukkan ke dalam basis data. Hal ini sedapat mungkin dicegah karena alasan-alasan yang menyangkut biaya penyimpanan, biaya indexing citra-citra yang pada dasarnya memiliki isi yang sama, biaya operasi basis data serta integritas basis data.

Definisi kesamaan (*similarity*) antar citra dokumen bergantung dari interpretasi yang digunakan. Citra dokumen memiliki beberapa tingkatan kesamaan [1], yaitu:

1. Tingkatan pertama meliputi citra dokumen yang identik atau sama persis sampai ke representasi citra yang paling sederhana yaitu pixel. Duplikat ini bisa muncul karena sebuah dokumen di-scan sekali dan hasilnya didistribusikan ke situs yang lain. Dalam kasus ini bisa diasumsikan bahwa citra tersebut adalah identik kecuali mungkin dalam variasi format file citra. Duplikat semacam ini cukup mudah untuk diidentifikasi dengan menggunakan pembandingan byte-per-byte atau pixel-per-pixel.
2. Tingkatan kedua meliputi citra dokumen *image-variant* yang berasal dari beberapa salinan dokumen yang sama. Hal ini bisa terjadi, sebagai contoh, untuk dokumen

yang dipublikasikan ke berbagai situs (contoh: laporan teknis, salinan memo, dll). Citra yang dihasilkan dari proses scanning pada dokumen-dokumen tersebut memiliki kesamaan dalam isi dan struktur, namun secara umum berbeda pada tingkat representasi citra. Perbedaan pada tingkat representasi citra bisa disebabkan karena:

a. dokumen asli telah dilubangi, disobek, dilapisi dengan selotip, kehilangan beberapa halaman, atau ditambahi dengan sampul.
b. dokumen asli telah difotokopi beberapa kali dan hasil fotokopinya kemudian difotokopi lagi.
c. dokumen asli telah di-scan pada perangkat keras yang berbeda, dengan resolusi, iluminasi dan kontras yang berbeda serta mengalami bermacam degradasi.

3. Tingkatan ketiga meliputi dokumen yang berbeda dalam isi dan struktur. Hal ini dapat terjadi jika suatu dokumen di-scan dan citranya dimasukkan ke dalam basis data. Kemudian dokumen yang sama diubah dan diformat ulang, di-scan dan hasilnya dimasukkan juga ke dalam basis data.

1.3 Tujuan Tugas Akhir

Keseluruhan perangkat lunak yang dibahas dalam Tugas Akhir ini bertujuan untuk:

1. Mengambil signature dari suatu citra dokumen yang akan ditambahkan ke dalam basis data.
2. Dengan menggunakan signature tersebut, menentukan apakah citra dokumen tersebut merupakan duplikat dari citra dokumen lain yang ada dalam basis data.

3. Menentukan dan menampilkan citra-citra dalam basis data yang mungkin mengandung duplikat, sesuai dengan urutan (*ranking*) kesamaan.

1.4 Batasan Masalah

Untuk mencegah meluasnya permasalahan yang dibahas, maka dalam Tugas Akhir ini diadakan batasan-batasan:

1. Algoritma deteksi duplikat dibatasi pada tingkat kesamaan pertama dan kedua, yang meliputi duplikat identik dan *image-variant*.
2. Algoritma analisa citra dokumen dibatasi pada citra dokumen cetak (*printed document*) dengan abjad latin.
3. Citra dokumen input dibatasi pada citra binari yang tidak memiliki kemiringan dan bingkai.
4. Walaupun suatu citra dokumen juga memiliki informasi kuantitatif dasar seperti jumlah halaman, dalam Tugas Akhir ini pengambilan informasi dibatasi dari analisa citra itu sendiri.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam menyusun Tugas Akhir ini adalah sebagai berikut:

1. Studi literatur dari berbagai buku dan jurnal.

   Pada studi literatur ini dicari dan dipelajari berbagai buku, jurnal dan dokumen mengenai analisa citra dokumen dan analisa n-gram.

2. Perancangan algoritma program dan struktur data

   Pada tahap ini ditentukan algoritma program yang akan digunakan lalu dirancang struktur datanya.

3. Pembuatan perangkat lunak.

   Pada tahap ini dibuat perangkat lunak yang dimulai dengan dibuat modul-modul yang ada kemudian digabungkan secara keseluruhan.

4. Pengujian perangkat lunak.

   Pada tiap-tiap modul yang telah dibuat diuji apakah sudah bekerja dengan benar dan kemudian modul-modul tersebut digabungkan untuk diuji secara keseluruhan.

5. Perbaikan untuk meningkatkan kinerja.

   Pada tahap ini rutin-rutin perangkat lunak yang tidak perlu dapat dihilangkan dan jika perlu dilakukan perubahan atau penambahan rutin.

6. Pembuatan laporan.

   Pada tahap ini ditulis laporan mengenai dasar teori, perancangan, pembuatan dan analisa perangkat lunak.

1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini dibagi menjadi 6 bab, dengan pembagian sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, permasalahan, tujuan, batasan permasalahan dan metodologi yang digunakan untuk menyelesaikan Tugas Akhir.

BAB II BASIS DATA CITRA DOKUMEN

Bab ini menjelaskan tentang citra dokumen, basis data citra dokumen dan penggunaan analisa n-gram dalam proses index entry untuk basis data citra dokumen.

BAB III SEGMENTASI CITRA DOKUMEN

Bab ini menjelaskan tentang segmentasi citra dokumen yang digolongkan sebagai analisa layout citra dokumen, algoritma segmentasi citra dokumen serta algoritma-algoritma pendukungnya.

BAB IV PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Bab ini menjelaskan mengenai perancangan dan pembuatan perangkat lunak untuk mendeteksi duplikasi dalam basis data citra dokumen dengan menggunakan teknik shape coding.

BAB V UJI COBA DAN EVALUASI PERANGKAT LUNAK

Bab ini menjelaskan cara-cara pelaksanaan uji coba, hasil uji coba dan evaluasi perangkat lunak yang membahas mengenai faktor-faktor yang mempengaruhi unjuk kerja perangkat lunak.

BAB VI KESIMPULAN DAN SARAN

Bab ini merupakan akhir dari penulisan buku ini berisi kesimpulan dan saran pengembangan berikutnya dari hasil Tugas Akhir ini.

BAB II

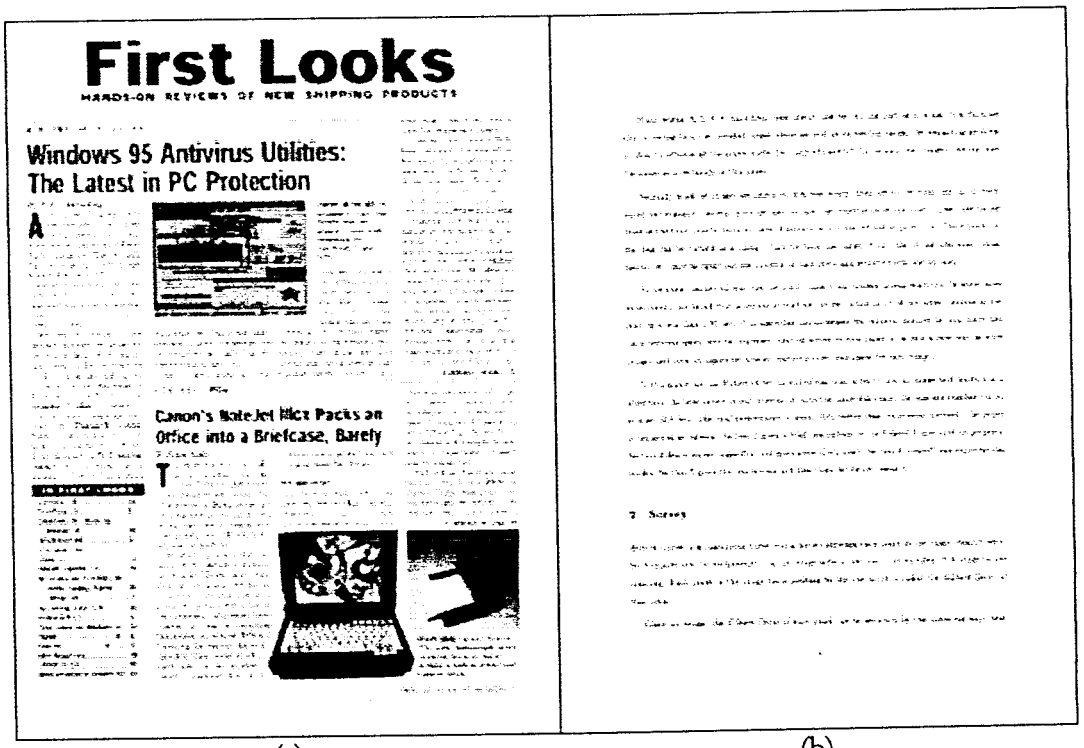
BASIS DATA CITRA DOKUMEN

Bab ini membahas mengenai citra dokumen, basis data citra dokumen dan analisa n-gram untuk index entry. Pembahasan tentang citra dokumen meliputi jenis-jenis citra dokumen yang dibedakan dari cara pembuatan dan materi teks yang terkandung di dalamnya. Pembahasan tentang basis data citra dokumen meliputi cara pembuatan, informasi index yang digunakan, keuntungannya dibandingkan dengan sistem dokumen konvensional serta hambatan-hambatan dalam konversi sistem dokumen konvensional ke sistem basis data citra dokumen. Dan terakhir dibahas mengenai analisa n-gram yang digunakan dalam proses index entry dan pencarian tingkat kesamaan (*similarity*) antar citra dokumen.

2.1 Citra Dokumen

Citra dokumen didefinisikan sebagai dokumen yang direpresentasikan melalui citra digital. Dokumen di sini dapat berupa semua material yang mengandung teks, mulai dari dokumen bisnis, memo, halaman majalah, halaman surat kabar, gambar teknik, halaman jurnal, sampul majalah, halaman buku, lembar kwitansi, surat perjanjian, dokumen kantor dan lain-lain. Sebuah dokumen biasanya memiliki struktur layout yang padat dengan tingkatan redundansi simbol yang tinggi, karena berisikan blok-blok teks seperti karakter, kata, kalimat dan paragraf yang digabung dengan blok non-teks seperti gambar, form, tabel dan lain-lain. Secara informal bisa

dikatakan bahwa sebuah dokumen berisikan komponen-komponen yang mewakili simbol-simbol sebuah bahasa [4]. Contoh dokumen diperlihatkan pada gambar 2.1.



(a) (b)

Gambar 2.1 Contoh citra dokumen dari (a) majalah dan (b) jurnal.

2.1.1 Scanner dan Scanning

Untuk mengubah sebuah dokumen menjadi citra dokumen diperlukan sebuah scanner dan prosesnya sendiri disebut sebagai scanning. Scanner merupakan perangkat keras yang mengambil sebuah dokumen sebagai input dan menghasilkan aliran informasi elektronis yang mewakili citra dari dokumen tersebut. Ada dua hal yang perlu diperhatikan dalam proses scanning, yaitu:

- besarnya resolusi yang digunakan, yang dinyatakan dengan dpi (dot per inch). Semakin besar ukuran dpi, citra yang dihasilkan semakin besar tapi juga nampak

semakin halus. Resolusi yang biasa digunakan untuk membuat citra dokumen yang digunakan dalam pustaka digital adalah 300 dpi.

- jumlah warna yang digunakan, yang dinyatakan dalam bpp (bit per pixel). Ukuran bpp yang paling kecil adalah 1 bpp atau 2 warna yang biasanya diwakili oleh warna hitam dan putih. Semakin besar jumlah bpp yang digunakan semakin besar pula ruang media penyimpanan yang digunakan. Citra dokumen umumnya merupakan citra binari atau citra yang hanya memiliki dua warna. Citra binari pada resolusi 300 dpi mampu merepresentasikan suatu dokumen dengan baik. Penggunaan citra binari juga dapat mengurangi jumlah kebutuhan ruang penyimpanan dan mempermudah analisa citra dokumen.

2.1.2 Binarisasi Citra

Seperti telah disebutkan di atas, umumnya citra dokumen direpresentasikan dalam citra binari. Karena itu perlu dilakukan suatu proses yang mengubah citra hasil scanning menjadi citra binari. Proses ini disebut sebagai binarisasi citra.

Suatu citra dari hasil scanning dapat diubah menjadi citra binari dengan menerapkan operasi threshold global. Nilai pixel yang berada di bawah threshold akan diubah menjadi pixel hitam dan yang berada di atas threshold akan diubah menjadi pixel putih. Proses ini bisa dirumuskan sebagai berikut:

$$x_{ij} = \begin{cases} 0; & x_{ij} < threshold \\ 1; & x_{ij} \geq threshold \end{cases} \qquad (2.1)$$

dengan keterangan:

x_{ij} = nilai pixel pada baris ke-i kolom ke-j

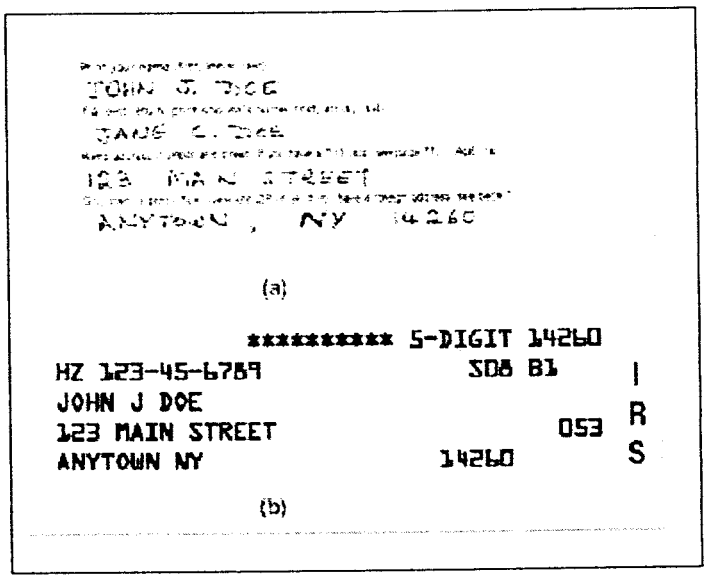
Nilai threshold itu sendiri bisa ditentukan sebelumnya atau dihitung dari histogram citra, yaitu berdasarkan lembah pada histogram citra tersebut. Sebuah threshold global, yang bisa digunakan untuk keseluruhan citra, kadang-kadang tidak berhasil didapatkan karena warna latar belakang yang bermacam-macam. Salah satu pemecahannya adalah dengan membandingkan nilai abu-abu dari suatu pixel dengan rata-rata nilai abu-abu pada beberapa pixel di sekelilingnya. Jika pixel tersebut lebih gelap dari pixel tetangganya, maka pixel tersebut dianggap hitam, jika tidak maka akan dianggap putih.

2.1.3 Dokumen Cetak dan Tulisan Tangan

Pada analisa citra dokumen, pokok permasalahan dalam menganalisa suatu blok teks terletak pada tingkat keteraturan bentuk karakter dan jarak antar karakter. Berdasarkan dua hal tersebut, blok-blok teks dalam suatu citra dokumen dapat digolongkan menjadi dua, yaitu:

- Blok teks cetak, jika karakter di dalamnya memiliki keteraturan dalam bentuk dan jarak antar karakter. Blok teks seperti ini dapat dihasilkan dari mesin ketik, printer, plotter dan mesin cetak. Sebuah dokumen yang didominasi oleh blok teks cetak disebut sebagai dokumen cetak (*printed document*).
- Blok teks tulisan tangan, jika karakter di dalamnya berasal dari tulisan tangan yang umumnya tidak memiliki keteraturan dalam bentuk dan kadang-kadang tidak memiliki jeda antar karakter, seperti dalam tulisan bersambung. Sebuah dokumen yang didominasi oleh blok teks tulisan tangan disebut sebagai dokumen tulisan tangan (*handwritten document*).

Contoh dokumen yang berisikan blok teks cetak dan tulisan tangan diperlihatkan pada gambar 2.2.



Gambar 2.2 Citra dokumen berisikan blok teks tulisan tangan dan cetak

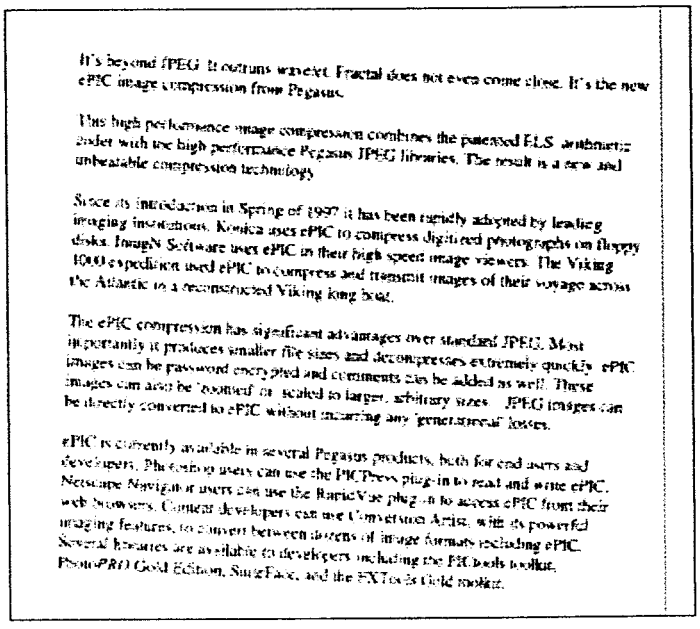
2.1.4 Masalah Pada Citra Dokumen

Citra dokumen yang dihasilkan dari proses scanning dapat mengalami beberapa masalah yang bisa mempersulit analisa citra, yaitu:

1. Citra yang dihasilkan memiliki kemiringan (*skew*) yang disebabkan karena kurang tepatnya posisi dokumen pada saat proses scanning. Perangkat lunak yang disertakan dalam paket scanner biasanya tidak memiliki fasilitas untuk menghilangkan kemiringan (*deskew*). Ini disebabkan karena definisi miring itu sendiri baru menjadi jelas jika yang menjadi input untuk scanner adalah sebuah dokumen, tapi tidak untuk jenis input lain seperti gambar atau foto. Menghilangkan kemiringan dari sebuah citra dokumen telah menjadi sebuah

obyek penelitian tersendiri dalam analisa citra dokumen. Contoh citra dokumen yang memiliki kemiringan diperlihatkan dalam gambar 2.3.

2. Citra memiliki noise, misalnya disebabkan oleh proses binarisasi citra yang dihasilkan dari scanning dokumen dengan noda pada latar belakang.
3. Citra mengalami degradasi, misalnya yang disebabkan oleh:
   - proses scanning pada dokumen yang merupakan generasi hasil fotokopi yang kesekian.
   - karakter yang memisah karena binarisasi citra abu-abu.
   - karakter yang menggabung (*merge*) karena blur atau latar belakang antar karakter yang gelap



It's beyond JPEG. It outruns wavelet. Fractal does not even come close. It's the new ePIC image compression from Pegasus.

This high performance image compression combines the patented ELS arithmetic coder with the high performance Pegasus JPEG libraries. The result is a new and unbeatable compression technology.

Since its introduction in Spring of 1997 it has been rapidly adopted by leading imaging institutions. Konica uses ePIC to compress digitized photographs on floppy disks. ImageN Software uses ePIC in their high speed image viewers. The Viking 1000 expedition used ePIC to compress and transmit images of their voyage across the Atlantic in a reconstructed Viking long boat.

The ePIC compression has significant advantages over standard JPEG. Most importantly it produces smaller file sizes and decompresses extremely quickly. ePIC images can be password encrypted and comments can be added as well. These images can also be 'zoomed in' or scaled to larger, arbitrary sizes. JPEG images can be directly converted to ePIC without incurring any 'generational' losses.

ePIC is currently available in several Pegasus products, both for end users and developers. Plug-n-play users can use the PICPress plug-in to read and write ePIC. Netscape Navigator users can use the PanicView plug-in to access ePIC from their web browsers. Content developers can use Conversion Artist, with its powerful imaging features, to convert between dozens of image formats including ePIC. Several libraries are available to developers including the PIC tools toolkit, PhotoPRO Gold Edition, SnapFace, and the FXTools Gold toolkit.

Gambar 2.3 Contoh citra dokumen yang memiliki kemiringan.

Dalam pengembangan algoritma untuk analisa citra dokumen, perlu dibuat sampel citra dokumen yang ideal, tanpa kemiringan, noise dan degradasi. Untuk membuat citra dokumen ideal bisa digunakan software-software tertentu tanpa

melalui proses scanning menggunakan scanner. Citra dokumen yang tidak dihasilkan dari proses scanning disebut juga sebagai citra dokumen sintetis. Dengan menggunakan citra dokumen sintetis, kita bisa berkonsentrasi pada analisa citra dokumen itu sendiri, tanpa perlu berhadapan dengan masalah-masalah yang telah disebutkan di atas.

Jika diinginkan, secara manual bisa ditambahkan efek-efek tertentu seperti noise, degradasi serta kemiringan ke dalam sebuah citra dokumen sintetis. Dengan demikian, citra dokumen sintetis dapat pula digunakan untuk mensimulasikan berbagai macam kondisi yang mungkin dihadapi oleh suatu algoritma analisa citra dokumen.

2.2 Basis Data Citra Dokumen

Hampir semua organisasi memiliki koleksi dokumen kertas. Jumlah koleksi yang dimiliki adalah beragam, dari yang hanya memerlukan sebuah lemari sebagai tempat penyimpanan, sampai ke yang berjumlah ratusan atau bahkan ribuan dokumen. Contoh paling umum dari sistem yang memiliki koleksi dokumen kertas dalam jumlah besar adalah perpustakaan yang dapat memiliki ratusan bahkan ribuan koleksi dokumen cetak berupa buku, majalah, jurnal serta ensiklopedia. Dalam kaitannya dengan basis data citra dokumen, sistem yang menggunakan dokumen kertas disebut sebagai sistem konvensional. Dalam sistem konvensional yang memiliki koleksi dalam jumlah besar, masalah administrasi dan perawatan menjadi isu utama, di samping masalah kemudahan dalam pencarian informasi.

2.2.1 Permasalahan Dalam Sistem Konvensional

Dalam sistem konvensional, umumnya secara manual diciptakan sistem administrasi yang mencatat semua informasi utama dari semua koleksi dan membagi-baginya ke dalam beberapa subyek atau kategori. Hal ini ditujukan untuk:

- memudahkan administrasi dan perawatan koleksi
- memudahkan pencarian informasi berdasarkan subyek atau kategori tertentu

Umumnya informasi yang dicatat adalah judul dokumen, nama pembuat dokumen atau orang yang bertanggung jawab atas dokumen tersebut, beserta informasi penting lain yang berhubungan dengan sistem administrasi itu sendiri.

Sistem administrasi sistem dokumen konvensional, memiliki kekurangan dalam hal pencarian informasi. Sistem ini tidak dapat sepenuhnya menjamin ketepatan hasil pencarian informasi. Ini disebabkan karena penentuan macam informasi yang dicatat dilakukan secara manual. Dan dalam beberapa kasus, informasi yang dicatat tersebut tidak terlalu mencerminkan isi sesungguhnya dari dokumen yang bersangkutan.

Kadang-kadang terjadi, ketika seseorang mencari suatu informasi dengan berbekalkan sebuah sebuah kata kunci yang terdapat dalam daftar subyek atau kategori. Ternyata hasil dari proses pencarian menunjuk ke beberapa dokumen yang di dalamnya tidak ditemukan informasi yang sedang dicarinya. Ini disebabkan karena dokumen yang berisikan informasi yang sedang dicarinya berada dalam subyek atau kategori yang lain.

Dengan pertimbangan tersebut, maka diperlukan sebuah sistem yang mampu melakukan pencatatan informasi dari dalam citra dokumen itu sendiri secara

otomatis. Dengan demikian hasil dari proses pencarian informasi akan mengacu langsung pada isi dokumen dan bukan pada pembagian subyek atau kategori.

2.2.2 Signature Citra Dokumen

Pada saat sebuah citra dokumen akan ditambahkan ke dalam suatu basis data, maka sistem perlu mengetahui apakah citra tersebut telah memiliki duplikat dalam basis data. Untuk keperluan ini dapat digunakan informasi index yang ada dalam setiap citra dokumen. Informasi ini dapat dihasilkan dengan cara entry manual atau secara otomatis dengan menggunakan algoritma tertentu. Namun dalam kenyataannya, sulit untuk menentukan informasi index yang unik untuk setiap citra dokumen yang akan ditambahkan ke dalam basis data. Hal ini disebabkan oleh besarnya volume informasi yang terdapat dalam tiap citra dokumen.

Untuk memutuskan informasi index apa yang akan digunakan untuk mendeteksi duplikat, perlu dipertimbangkan:

- ukuran informasi index dan
- waktu yang diperlukan untuk membuat serta membandingkannya dengan citra dokumen lain yang telah ada dalam basis data.

Karena itu kadang-kadang digunakan dua buah informasi index, sebagai filter primer dan filter sekunder dalam proses pencarian duplikat. Filter primer berfungsi untuk menghasilkan daftar citra duplikat dan untuk itu digunakan informasi index yang berukuran kecil (dan karena itu relatif kurang unik) serta cepat dalam proses pengambilan dan pembandingan. Untuk mengurangi jumlah match yang dihasilkan oleh filter primer, maka digunakan filter sekunder. Karena itu filter sekunder

menggunakan informasi index yang berukuran lebih besar (dan karena itu relatif lebih unik) namun lebih lama dalam proses pengambilan dan pembandingan.

Beberapa contoh informasi index dalam suatu citra dokumen, dijelaskan di bawah ini:

1. Informasi index mendasar seperti nomer dokumen, data, judul, nama pembuat dokumen dan jumlah halaman dokumen. Informasi index ini dimasukkan secara manual pada saat menambahkan citra ke dalam basis data. Namun untuk basis data yang melibatkan ribuan citra dokumen, index entry secara manual seperti ini akan memakan banyak biaya.
2. Informasi index yang berasal dari pengambilan signature dari citra dokumen, yang ditentukan berdasarkan feature set tertentu. Pengambilan signature dilakukan secara otomatis oleh sebuah algoritma. Tingkat keunikan signature, besarnya ukuran, serta kecepatan dalam proses pengambilan dan pembandingannya ditentukan oleh banyaknya feature set yang digunakan. Feature set yang jumlahnya sedikit menghasilkan signature yang berukuran kecil, cepat dalam proses pengambilan serta pembandingannya, namun memiliki tingkat keunikan yang rendah. Sedangkan feature set yang jumlahnya lebih besar menghasilkan signature yang berukuran besar, lebih lama dalam proses pengambilan dan pembandingan, namun memiliki tingkat keunikan yang lebih tinggi.
3. Informasi index yang berasal dari teks yang dihasilkan oleh proses OCR (Optical Character Recognition) terhadap citra dokumen. Proses OCR sendiri dapat digunakan untuk pengambilan signature dengan feature set berupa semua karakter alfabet. Walaupun proses pembandingan teks hasil OCR bisa dilakukan

dengan relatif cepat, kinerja OCR sendiri tidak bisa diandalkan dalam hal kecepatan dan akurasi. Hal ini disebabkan oleh:

- pertama, karena analisa mendalam oleh OCR tidak bisa diandalkan dalam mengambil suatu feature (yang untuk OCR berarti sebuah karakter) dari citra dokumen yang mengalami degradasi.
- kedua, karena informasi index yang dihasilkan akan menjadi sangat besar.

Karena itulah informasi index dari hasil OCR dirasakan kurang tepat untuk filter primer namun bisa digunakan untuk filter sekunder, untuk mengurangi jumlah match yang dihasilkan filter primer.

2.2.3 Proses Pembuatan Sistem

Proses pembuatan sistem basis data citra dokumen dilakukan dalam beberapa tahapan, yaitu:

1. Scanning.

   Proses scanning dilakukan dengan melakukan optical scanning terhadap dokumen kertas menggunakan scanner pada resolusi 300 dpi. Hasil scanning ini disimpan dalam file citra binari menggunakan algoritma kompresi lossless. Format standar yang digunakan untuk file citra dokumen adalah TIFF dengan algoritma kompresi CCITT Fax 4 yang merupakan standar kompresi dalam proses pengiriman fax.

2. Indexing.

   Sistem mengambil sebuah signature dari citra dokumen yang akan ditambahkan ke dalam basis data, dan menggunakan signature tersebut sebagai informasi index. Dengan menggunakan signature tersebut, sistem akan melakukan

pembandingan dengan signature yang dimiliki oleh citra dokumen lain dalam basis data. Dari sini bisa diketahui apakah citra dokumen yang akan ditambahkan tersebut sudah ada dalam basis data atau tidak. Dari proses pembandingkan ini, sistem akan memunculkan daftar citra dokumen yang memiliki nilai kesamaan di atas batas tertentu. Citra dokumen yang sama atau identik diharapkan untuk memiliki nilai "kesamaan" 100%.

3. Penyimpanan (*storing*).

   Bila ternyata dari proses pembandingan signature, tidak ditemukan citra dokumen dalam basis data dengan nilai kesamaan di atas batas tertentu, itu berarti citra baru tersebut tidak dianggap sebagai duplikat dan dimasukkan ke dalam basis data dengan signaturenya sebagai informasi index.

2.2.4 Manfaat Sistem

Banyak keuntungan yang bisa dicapai dari penggunaan sistem basis data citra dokumen [8] jika dibandingkan dengan sistem konvensional:

- Sistem basis data citra dokumen tidak memerlukan banyak ruang untuk penyimpanan fisik. Seluruh informasi yang ada dalam basis data dapat disimpan dalam bentuk CD, harddisk atau pun media simpanan lainnya. Bagian yang paling mahal dalam pembuatan sistem basis data adalah pada saat investasi awal untuk pengadaan peralatan, pelatihan personel untuk pengoperasiannya dan waktu yang diperlukan untuk proses scanning semua dokumen yang dimiliki ke dalam format citra digital. Namun tentu saja, semua hal tersebut masih lebih murah jika dibandingkan dengan pengadaan ruang secara fisik untuk menampung jutaan dokumen dalam wujud buku, jurnal dan lain-lain.

- Kemampuan untuk mengakses semua informasi dari remote site. Jika sistem dapat diatur dengan baik, maka komputer dari seluruh penjuru dunia akan mampu untuk berkomunikasi dengan sistem basis data citra dokumen dan mengakses semua informasi yang ada di dalamnya. Satu-satunya faktor pembatas adalah bandwidth jaringan. Namun hal ini mulai dapat diatasi dengan digunakannya jaringan fiber-optic. Dan karena informasi yang ada dalam sistem bisa dari mana saja, tidak dibutuhkan bangunan fisik yang perlu dikunjungi oleh user. User tak perlu pergi untuk mendapatkan informasi, tapi informasilah yang akan pergi kepada user.
- Dapat melakukan pencarian suatu informasi dengan menggunakan keseluruhan teks dari suatu buku, dokumen atau artikel. Hal ini memungkinkan, dalam kasus yang paling sederhana, pencarian kata kunci menggunakan keseluruhan dokumen daripada hanya menggunakan judul atau nama subyek.
- Penggunaan tampilan informasi yang efisien dan user-friendly. Karena informasi disimpan dalam bentuk kumpulan byte dalam sebuah komputer, maka dapat diatur dalam bentuk tampilan apa saja yang diinginkan. Daftar dokumen dan buku tertentu dapat ditampilkan dalam bentuk yang intuitif dan mudah digunakan. Teknik grafik dan multimedia juga bisa digunakan secara efektif untuk menampilkan informasi dalam user interface.

2.2.5 Hambatan Implementasi Sistem

Secara keseluruhan konsep sistem basis data citra dokumen adalah bagus, namun banyak rincian yang perlu diselesaikan untuk membuat suatu sistem yang tahan error dan mudah digunakan. Banyak dari permasalahan ini yang dikerjakan

dalam bidang lain di luar ilmu informatika dan hasilnya bisa diterapkan dalam proyek sistem basis data citra dokumen. Salah satunya meliputi pembuatan interface yang mudah digunakan; penggunaan ruang penyimpanan yang efisien, pengambilan informasi yang efisien, penyimpanan storage secara terdistribusi, pengambilan informasi dan pencarian; serta bandwith jaringan.

Salah satu permasalahan utama dalam sistem ini [8] adalah kenyataan bahwa banyak buku dan artikel yang tidak hanya berisikan teks, namun juga ilustrasi, gambar, diagram dan grafik. Sulit bagi sistem untuk melakukan klasifikasi dan indexing terhadap kandungan non-teks tersebut.

Permasalahan utama lainnya adalah besarnya investasi awal berupa waktu, uang dan sumber daya manusia yang diperlukan untuk mengubah dokumen kertas yang ada ke dalam bentuk citra digital. Tergantung dari jumlah koleksi dokumen yang dimiliki, bisa diperlukan puluhan sampai ratusan jam kerja untuk melakukan proses scanning. Hal ini bisa menjadi hambatan terbesar, yang menjadi sumber keraguan utama bagi kebanyakan organisasi saat harus memutuskan apakah akan mengkonversi sistem dokumen konvensional yang mereka miliki ke dalam sistem basis data dokumen citra.

2.3 Analisa N-Gram Dalam Index

2.3.1 Definisi N-Gram

Bila terdapat suatu string s dengan dengan panjang m, maka n-gram dari string s merupakan kumpulan substring N yang didefinisikan di bawah ini:

$$N_i = s_i, s_{i+1}, s_{i+2}, \ldots, s_{i+n-1} \; ; i = 1 \; .. \; m\text{-}n\text{+}1 \qquad (2.2)$$

dengan keterangan:

N_i = n-gram ke-i

s_i = karakter string s pada posisi index ke-i

m = panjang string s

n = panjang substring atau n-gram

Dengan kata lain, n-gram dari string s merupakan kumpulan substring n karakter dari string s, yang dimulai dari karakter pertama sampai karakter ke m-n+1. Untuk jelasnya, diperlihatkan pada contoh di bawah ini:

Jika string s adalah:

```
kertas gunting dan batu
```

Maka n-gram dari string s, dengan n = 4, adalah:

```
kert  erta  rtas  tas_  as_g  s_gu  gun_  gunt
unti  ntin  ting  ing_  ng_b  g_ba  bat_  batu
```

Aplikasi n-gram yang paling utama adalah dalam pembandingan string. Membandingkan n-gram dari dua buah string, mempunyai kelebihan dibandingkan dengan membandingkan kedua string itu sendiri. Dengan membandingkan kedua string itu sendiri, maka satu perubahan karakter pada salah satu string akan berakibat kedua string tersebut dinyatakan berbeda. Berbeda dengan pembandingan n-gram dari kedua string tersebut. Untuk jelasnya, diperlihatkan pada contoh di bawah ini:

Jika string a adalah :

```
merah biru
```

Jika string b adalah :

```
meras biru
```

Maka n-gram dari *a* dan *b* dengan $n = 4$, adalah :

```
mera  erah  rah_  ah_b  h_bi  bir_  biru
mera  eras  ras_  as_b  s_bi  bir_  biru
```

Saat 4-gram dari *a* dibandingan dengan 4-gram dari *b*, terdapat 3 hit dan 4 miss. Maka dinyatakan bahwa string *b* memiliki kesamaan 3/7 atau 43% dengan string *a*.

2.3.2 Penggunaan N-Gram Dalam Index

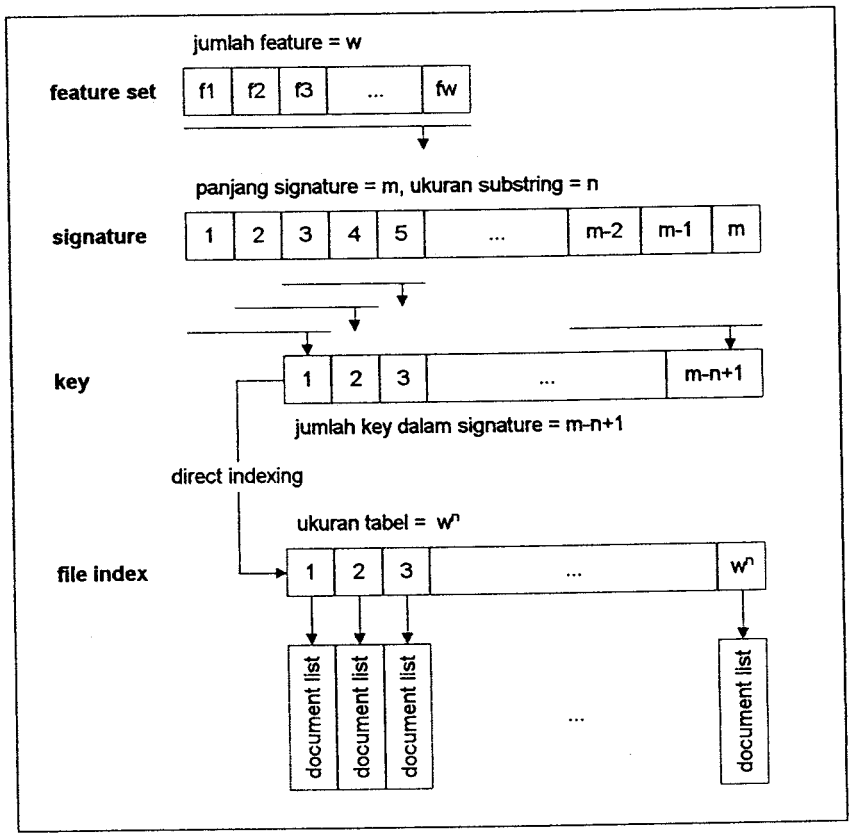
Implementasi n-gram dalam index basis data citra dokumen dilakukan dengan proses indexing yang didasarkan pada n-gram dari signature citra dokumen daripada keseluruhan signature itu sendiri. Cara ini mempunyai beberapa kelebihan, yang terutama adalah menambahkan tingkat ketelitian dalam proses query.

Seperti yang telah dijelaskan dalam sub bab 2.2.4, bahwa salah satu keuntungan penggunaan basis data citra dokumen jika dibandingkan dengan sistem dokumen konvensional adalah dapat dilakukan pencarian suatu informasi dengan menggunakan keseluruhan teks dari suatu buku, dokumen atau artikel. Dan dengan menggunakan analisa n-gram, sistem dapat secara otomatis melakukan klasifikasi, indeks dan pemberian label untuk semua buku, dokumen, artikel dan lain-lain tanpa memerlukan campur tangan manusia. Tentu saja ini bergantung pada jenis dan panjang signature yang digunakan.

Analisa n-gram juga bisa digunakan untuk melakukan pencarian berdasar subyek yang sangat efisien. Jika seorang user berhasil menemukan sebuah dokumen tentang suatu subyek, mereka dapat menggunakan analisa n-gram untuk menemukan semua dokumen dengan subyek yang sama. Kesamaan sebuah citra dokumen dengan citra dokumen lainnya didefinisikan sebagai jumlah hit atau n-gram yang sama antar signature kedua dokumen dibagi jumlah n-gram untuk sebuah signature.

2.3.3 Skema Indexing Menggunakan N-Gram

Diasumsikan bahwa sebuah citra dokumen akan ditambahkan ke dalam basis data. Dari dalam citra tersebut berhasil diambil sebuah signature *s* dengan panjang *m* dan didasarkan pada feature set *f* yang terdiri dari *w* feature. Kemudian dari signature *s* dihasilkan n-gram *N*. Maka proses entry n-gram *N* ke dalam index diperlihatkan dalam skema index pada gambar 2.4.



Gambar 2.4 Skema indexing n-gram dari signature

Dari gambar 2.4, nampak bahwa dari signature *s* dihasilkan m-n+1 n-gram yang kemudian berfungsi sebagai *key* untuk file index. File index sendiri berisikan semua substring dengan panjang *n* yang didapatkan dari *w* feature. Dengan *w* feature

dan substring dengan panjang n, maka jumlah total key yang dihasilkan adalah u^n, yang masing-masing menunjuk pada sebuah list dokumen.

Ketika sekumpulan key digunakan dalam sebuah query, setiap key akan menghasilkan sekian hit dari file index. Setiap hit akan dihitung untuk proses *vote* terhadap citra dokumen yang dihasilkan, dan dari sini bisa ditampilkan suatu daftar citra dokumen yang urut berdasarkan jumlah hit. Kesamaan antar dua citra dokumen didefinisikan sebagai jumlah key yang hit dibagi jumlah total key dalam signature.

BAB III

SEGMENTASI CITRA DOKUMEN

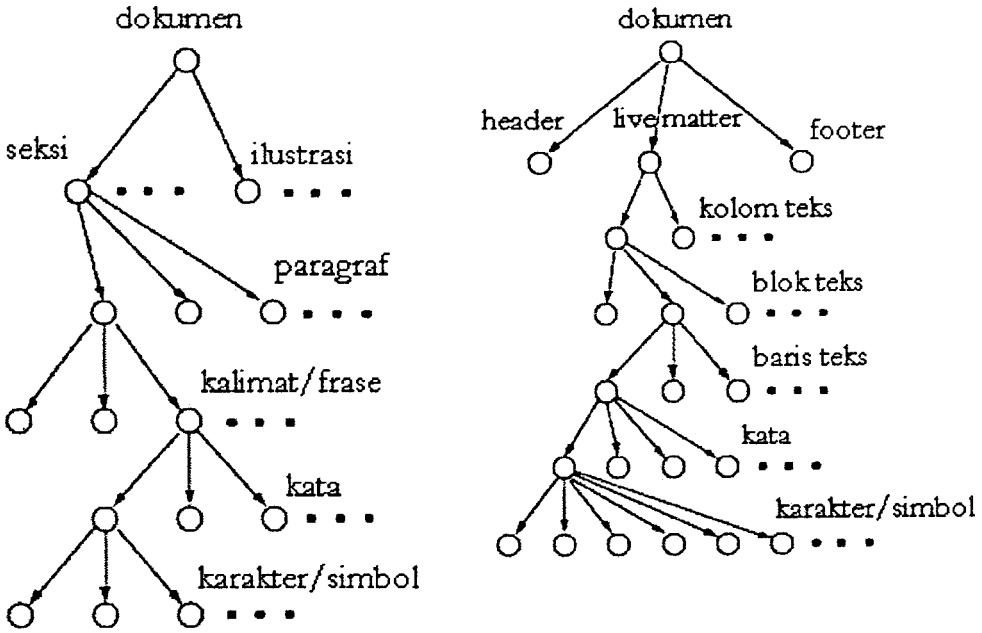
Bab ini membahas mengenai segmentasi citra dokumen beserta algoritma-algoritma yang mendukungnya. Segmentasi citra dokumen tergolong sebagai bagian dari analisa layout citra dokumen, yang mengidentifikasi bermacam-macam obyek dalam citra dokumen dan menjelaskan hubungan antar obyek-obyek tersebut. Dalam analisa layout citra dokumen, sebuah obyek didefinisikan sebagai suatu region persegi panjang yang menunjuk pada suatu karakter, kata, baris teks, paragraf, kolom teks, region non-tekstual dan lain-lain.

3.1 Analisa Citra Dokumen

Analisa citra dokumen menguraikan sebuah citra dokumen menjadi dua macam struktur [9], yaitu:

- Struktur logikal (*logical structure*), yang memberikan nama-nama pada bagian citra dokumen yang berisikan informasi dan menerangkan hubungan antara satu bagian dengan bagian lainnya secara logikal, seperti dalam urutan pembacaan.
- Struktur layout (*layout structure*) atau struktur fisik, yang menerangkan bagaimana informasi dari isi citra dokumen diletakkan secara fisik, seperti dalam **hal** tampilan dan lokasi.

Masing-masing struktur tersebut dapat direpresentasikan dalam wujud sebuah tree, seperti yang diperlihatkan dalam gambar 3.1.



(a) (b)

Gambar 3.1 Struktur (a) logikal dan (b) layout suatu citra dokumen yang diwujudkan dalam bentuk struktur tree.

Dalam tree untuk struktur logikal, setiap node daun menunjuk pada sebuah simbol. Di atasnya terdapat node-node yang mewakili kata, frase dan kalimat, paragraf dan ilustrasi, seksi serta dokumen itu sendiri. Untuk struktur layout, setiap node daun dalam tree-nya menunjuk pada kotak pembatas yang berisikan simbol. Di atasnya terdapat node-node yang menunjuk pada kotak pembatas yang berisikan kata, baris teks, blok teks dan semua kolom teks dalam dokumen. Root dari tree struktur layout menunjuk pada kotak pembatas terbesar yang berisikan citra dokumen itu sendiri. Dari sini bisa dilihat bahwa terdapat hubungan antara struktur logikal dan layout suatu dokumen. Hal ini disebabkan karena sifat alami dari proses pencetakan dokumen itu sendiri.

Umumnya dalam pembuatan dokumen kertas yang berasal dokumen teks elektronik, proses pencetakan harus menerjemahkan struktur logikal dokumen tersebut ke dalam struktur layout. Untuk itu proses pencetakan harus mengikuti beberapa aturan atau protokol yang merincikan kebutuhan layout dokumen. Kebutuhan tersebut meliputi: jenis font, ukuran font, style untuk simbol-simbol, jumlah dan lebar kolom, header, footer serta margin yang diperlukan dalam menghasilkan baris dan blok teks. Di samping itu juga ada aturan pembuatan spasi untuk simbol, kata, baris teks, blok teks dan kolom teks. Dalam hampir semua kasus, jarak spasi antar simbol lebih kecil dari jarak spasi antar kata. Demikian pula, jarak spasi antar baris teks lebih kecil dari jarak spasi antar blok teks atau kolom teks. Kecenderungan inilah yang digunakan sebagai pengetahuan awal dalam hampir semua algoritma OCR dan segmentasi dokumen.

Segmentasi citra dokumen digolongkan dalam analisa layout citra dokumen. Algoritma untuk segmentasi citra dokumen dapat dibedakan ke dalam dua kelompok. Kelompok pertama menggunakan metode atas bawah (*top down*) yang didasarkan pada model (*model-driven*), sedangkan kelompok kedua menggunakan metode dasar ke atas (*bottom up*) yang didasarkan pada data (*data-driven*).

Algoritma segmentasi yang digunakan dalam Tugas Akhir ini didasarkan pada algoritma RXYC (*Recursive XY Cuts*) [4] yang digolongkan sebagai metode top down. Tapi sebelum membahas mengenai algoritma RXYC, akan dibahas lebih dulu mengenai analisa projection profile yang menjadi dasar algoritma RXYC.

3.2 Analisa Projection Profile

Untuk sebuah citra binari B dengan tinggi m pixel dan lebar n pixel, maka:

- horizontal projection profile histogram merupakan sebuah vektor v sedemikian rupa sehingga

$$v_i = \sum_{j=1}^{n} B_{ij} \; ; B_{ij} = w \; ; i = 1 \; .. \; m \tag{3.1}$$

v_i = jumlah semua pixel dengan warna w pada baris i.

Vektor v dikatakan juga sebagai hasil proyeksi semua pixel dengan warna w terhadap sumbu y.

- vertical projection profile histogram merupakan sebuah vektor v sedemikian rupa sehingga

$$v_j = \sum_{i=1}^{m} B_{ij} \; ; B_{ij} = w \; ; j = 1 \; .. \; n \tag{3.2}$$

v_j = jumlah semua pixel dengan warna w pada kolom j.

Vektor v dikatakan juga sebagai hasil proyeksi semua pixel dengan warna w terhadap sumbu x.

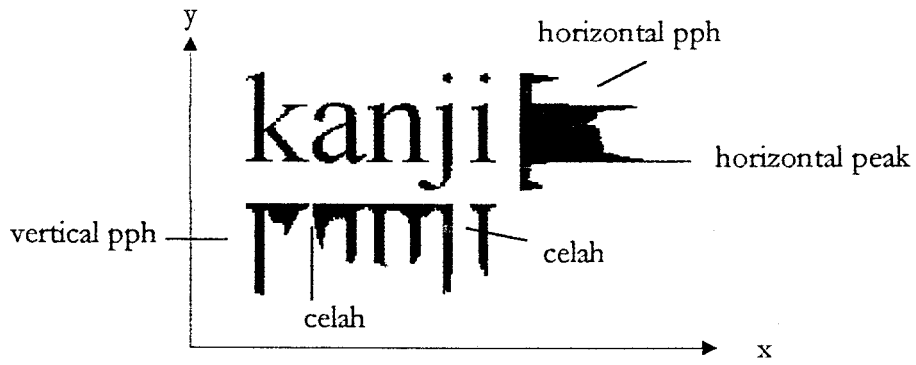
- Untuk analisa citra dokumen, warna w yang diproyeksikan adalah warna latar depan.

Beberapa properti yang menyertai suatu projection profile histogram adalah:

- profile, yaitu nilai yang berada dalam salah satu dimensi vektor v.
- celah (*gap*), yaitu rangkaian profile yang bernilai 0.
- puncak (*peak*), yaitu profile tertinggi dalam histogram.

Contoh projection profile histogram dan properti-propertinya dapat dilihat pada gambar 3.1.

Penggunaan projection profile histogram untuk segmentasi citra dokumen berawal dari kenyataan bahwa umumnya citra dokumen mengandung struktur layout horizontal dan vertikal, yang diletakkan dalam blok-blok persegi panjang yang paralel dengan sumbu x dan y. Contohnya seperti blok-blok paragraf yang berisikan baris-baris karakter.



Gambar 3.1 Projection profile histogram (pph) horizontal dan vertikal dari citra binari sebuah teks "kanji".

3.2.1 Binarisasi Projection Profile Histogram

Binarisasi dari projection profile histogram bertujuan untuk 'menyeleksi' profile dalam histogram. Nilai profile yang lebih besar daripada atau sama dengan threshold diubah menjadi 1 dan yang lebih kecil diubah menjadi 0. Proses ini bisa dirumuskan sebagai berikut:

$$b_i = \begin{cases} 0 ; v_i < threshold \\ 1 ; v_i \geq threshold \end{cases} \quad (3.3)$$

dengan keterangan:

b_i = nilai dimensi ke-i pada vektor histogram hasil binarisasi

v_i = nilai dimensi ke-i pada vektor histogram asli

Contoh proses binarisasi histogram diperlihatkan di bawah ini:

Histogram asli:

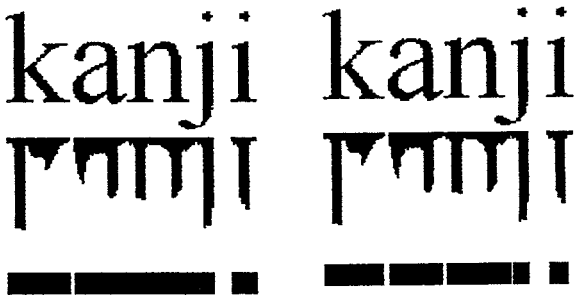
```
00011398749874498221100001128476872231100000
```

Threshold = 3

Histogram hasil binarisasi:

```
00000011111111111000000000001111110000000000
```

Contoh hasil penerapan binarisasi pada horizontal projection profile histogram sebuah baris teks 'kanji' diperlihatkan pada gambar 3.2.



(a) (b)

Gambar 3.2 Contoh hasil binarisasi profile histogram pada citra teks 'kanji' dengan threshold (a) 0,015 dan (b) 0,1 kali rata-rata profile.

Dengan menggunakan threshold yang tepat, proses binarisasi histogram dapat digunakan untuk menghilangkan pengaruh noise pada projection profile histogram. Threshold yang terlalu kecil, beresiko membiarkan hasil proyeksi yang muncul akibat noise. Sedangkan threshold yang terlalu besar beresiko menghilangkan hasil proyeksi yang muncul sedikit dalam suatu baris teks, seperti dalam kasus baris teks panjang yang hanya memiliki satu bagian ascender atau descender. Dalam contoh pada gambar 3.2 (b) dapat dilihat bahwa dengan nilai

threshold yang tepat, kita dapat menemukan kembali letak gap antar karakter yang tidak ditemukan dalam histogram asli.

3.2.2 Algoritma Run Length Smearing (RLS)

Proses RLS bertujuan menyatukan dua kumpulan non-zero profile yang dipisahkan oleh kumpulan zero profile (celah) yang panjangnya di bawah threshold. Contoh penggunaan RLS pada histogram hasil binarisasi diperlihatkan di bawah ini:

Histogram hasil binarisasi:

000000111111001110000000000100111000000000

Histogram hasil RLS dengan threshold = 5

000000111111111110000000000111111000000000

Histogram hasil RLS dengan threshold = 12

000000111111111111111111111111111000000000

Setelah proses RLS, kumpulan non-zero profile pada histogram digunakan untuk menentukan letak cut (pemilahan). Dengan menggabungkan koordinator cut dari hasil RLS pada sumbu x dan y, akan dihasilkan koordinat-koordinat kotak pembatas pada citra bersangkutan.

Dengan threshold yang berbeda-beda, kita bisa menggunakan RLS untuk proses segmentasi blok, baris, kata atau karakter. Pada tabel 3.1 diperlihatkan bahwa kita bisa melakukan segmentasi blok atau baris atau kata dengan menggunakan threshold yang berbeda-beda dalam proses RLS.

Tabel 3.1 Bounding box untuk baris teks "Citra dokumen" berdasarkan threshold untuk RLSA.

	threshold	**hasil**	**hasil segmentasi pada citra**
(a)	20	segmentasi baris	
(b)	5	segmentasi kata	
(c)	1	segmentasi karakter	

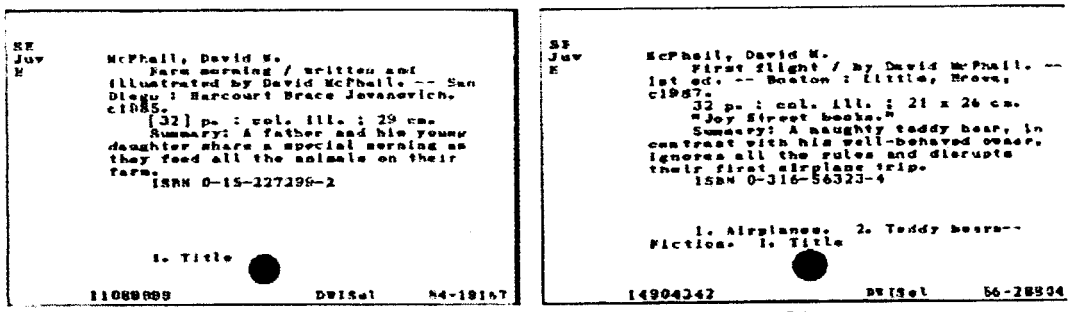
Pada contoh (c), blok "ku" pada kata "dokumen" tidak dapat dipisahkan. Seperti telah dijelaskan dalam binarisasi, letak cut antara karakter "k" dan "u" dapat ditemukan dengan threshold binarisasi yang lebih besar.

3.2.3 Keterbatasan Analisa Projection Profile

Dari uraian di atas, dapat disimpulkan bahwa penggunaan analisa projection profile untuk analisa layout citra dokumen memiliki beberapa keterbatasan. Beberapa di antaranya yaitu:

- tidak dapat menangani citra dokumen yang memiliki kemiringan. Sulit untuk menentukan posisi cuts dari histogram yang dihasilkan, karena semakin miring dokumen semakin besar kemungkinan histogram yang dihasilkan tidak memiliki celah.

- tidak dapat menangani citra dokumen yang memiliki bingkai (*frame*), karena berarti tidak akan ada celah pada histogram yang dihasilkan. Contoh dari citra dokumen yang memiliki bingkai diperlihatkan pada gambar 3.3.
- karakter yang terpisah akan tersegmentasi menjadi dua karakter
- karakter-karakter yang menggabung (*merge*) akan tersegmentasi menjadi satu blok.



(a) (b)

Gambar 3.3. Contoh citra dokumen yang memiliki bingkai

3.3 Algoritma RXYC (Recursive XY Cuts)

Algoritma yang umum digunakan dalam segmentasi citra dokumen adalah algoritma top down. Algoritma ini memulai dari keseluruhan citra dokumen dan seterusnya melakukan dekomposisi terhadap citra menjadi region-region yang lebih kecil. Setiap region dapat berisikan karakter, kata, baris teks, paragraf, kolom teks atau region non-teks. Jenis segmentasi ini memerlukan pengetahuan awal tentang karakteristik citra, yang diperlukan untuk mengambil properti yang akan digunakan untuk operasi pemisahan region.

Algoritma RXYC merupakan algoritma top down yang paling populer. Algoritma ini secara rekursif membagi citra dokumen menjadi blok-blok. Pada tiap tahapan rekursif, algoritma menghitung projection profile histogram untuk arah

vertikal dan horizontal. Kemudian algoritma meletakkan cut pada celah yang lebih lebar dari threshold tertentu pada kedua histogram dan proses dilanjutkan secara rekursif sampai tidak ada lagi celah yang cukup lebar dalam kedua histogram. Dari cara pembagian yang terstruktur ini, citra dokumen dapat direpresentasikan dalam wujud sebuah tree yang node-nodenya mewakili blok-blok persegi panjang dalam citra. Node-node daun mewakili blok-blok yang tidak dapat didekomposisi lagi (yaitu karakter) dan root mewakili blok persegi panjang terbesar, yaitu citra dokumen itu sendiri. Tree yang dihasilkan dari prosedur ini dapat disamakan dengan tree struktur layout pada gambar 3.1.b.

Keputusan tentang jenis cut yang dilakukan tergantung dari susunan pixel dan kedalaman proses rekursi. *Peak* (puncak) lokal pada projection profile horizontal dan vertikal dapat mewakili garis pembatas antar dua region yang berbeda (contohnya seperti pemisahan antar dua paragraf). Lokalitas nilai peak tersebut tergantung dari kedalaman proses rekursi, yaitu pada awalnya algoritma harus mencari peak tertinggi untuk bisa mengisolasi suatu region dalam citra. Pada level rekursi yang lebih dalam, algoritma harus mencari nilai peak yang lebih rendah untuk bisa mencari baris-baris teks.

Permasalahan utama yang ada pada algoritma RXYC adalah:

- Pada tiap tingkatan dekomposisi, sistem harus memilih dekomposisi yang tepat, dikarenakan perbedaan model dekomposisi untuk kolom teks, paragraf, baris teks, kata dan karakter. Di beberapa kasus, terdapat model dekomposisi yang tidak mencerminkan hubungan langsung antara jenis obyek dan tingkatan dekomposisi.

- Teknik pemilahan X-Y secara rekursif di atas, tidak bisa diterapkan untuk beberapa jenis topologi layout dokumen. Khususnya jika terdapat noise dalam citra dokumen.
- Penggunaan projection profile dalam menentukan posisi cut, mengakibatkan metode RXYC memiliki bermacam keterbatasan yang juga dimiliki oleh analisa projection profile, seperti yang telah dijelaskan dalam sub bab 3.1.3

3.4 Analisa Connected Component

Dalam algoritma RXYC, hal yang paling sulit adalah menentukan nilai threshold yang paling tepat untuk digunakan dalam meletakkan cut. Untuk segmentasi halaman, baris dan kata, penggunaan nilai konstan untuk threshold sudah cukup untuk mencapai hasil yang diinginkan. Namun, untuk segmentasi karakter tidak berlaku demikian. Jika tresholdnya terlalu longgar, maka sebuah karakter dapat tersegmentasi menjadi 3 atau 4 bagian. Sedangkan jika thresholdnya terlalu ketat, maka beberapa karakter dapat tersegmentasi menjadi satu karakter.

Dari sini disimpulkan bahwa analisa projection profile tidak cukup untuk segmentasi karakter. Sebagai alternatif, digunakan analisa connected component yang dapat menghasilkan segmentasi karakter sampai ke tingkatan yang lebih bisa diterima. Dengan demikian kita bisa menggabungkan kecepatan dan kesederhanaan yang dimiliki analisa projection profile dengan ketepatan yang dimiliki analisa connected component.

3.4.1 4-Neighbor dan 8-Neighbor

Untuk sebuah pixel yang terletak pada koordinat (i,j), maka:

- 4-neighbor dari pixel tersebut merupakan titik-titik (i',j') sedemikian rupa sehingga:

  $|i-i'| + |j-j'| = 1$

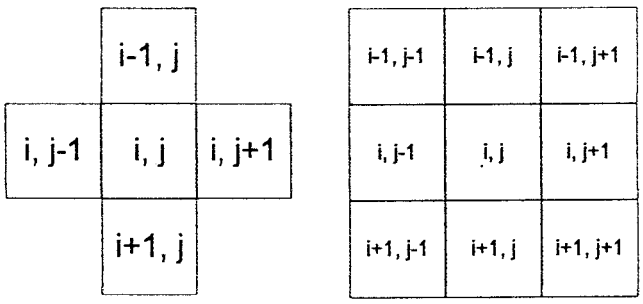
  dan koordinat 4-neighbor dari pixel tersebut adalah $(i\pm i, j)$ dan $(i,j\pm 1)$

- 8-neighbor dari pixel tersebut merupakan titik-titik (i',j') sedemikian rupa sehingga:

  $\max(|i-i'|,|j-j'|) = 1$

  dan koordinat 8-neighbor dari pixel tersebut adalah $(i, j\pm 1)$, $(i\pm 1, j)$ dan $(i\pm 1, j\pm 1)$

Kondisi 4-neighbor dan 8-neighbor diperlihatkan pada gambar 3.4.



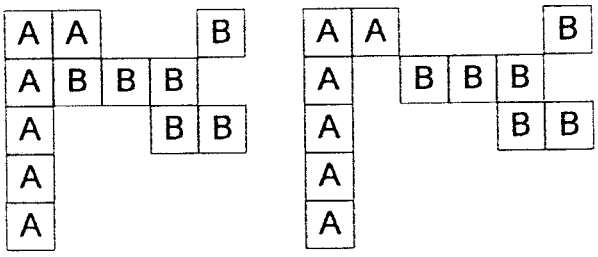
Gambar 3.4 Pixel (i, j) dengan (a) 4-neighbor dan (b) 8-neighbor yang dimilikinya.

3.4.2 4-Adjacent dan 8-Adjacent

Untuk dua himpunan pixel A dan B yang saling terpisah, maka:

- A adalah 4-adjacent terhadap B jika terdapat sebuah pixel dalam A yang merupakan 4-neighbor dari sebuah pixel dalam B
- A adalah 8-adjacent terhadap B jika terdapat sebuah pixel dalam A yang merupakan 8-neighbor dari sebuah pixel dalam B.

Contoh 4-adjacent dan 8-adjacent diperlihatkan pada gambar 3.5.



(a) (b)

Gambar 3.5 Himpunan pixel A yang (a) 4-adjacent dan (b) 8-adjacent dengan himpunan pixel B

3.4.3 Connected Component

Dalam sebuah citra binari *B*, dikatakan terdapat:

- 4-path dari pixel (i_0, j_0) ke pixel (i_n, j_n) jika terdapat urutan pixel:

  $(i_0, j_0), (i_1, j_1), \cdots (i_{n-1}, j_{n-1}), (i_n, j_n)$

  sedemikian rupa sehingga pixel (i_k, j_k) merupakan 4-neighbor dari pixel (i_{k+1}, j_{k+1})

  untuk

  $k = 0, 1, \ldots n-1$

- 8-path dari pixel (i_0, j_0) ke pixel (i_n, j_n) jika terdapat urutan pixel:

  $(i_0, j_0), (i_1, j_1), \cdots (i_{n-1}, j_{n-1}), (i_n, j_n)$

  sedemikian rupa sehingga pixel (i_k, j_k) merupakan 8-neighbor dari pixel (i_{k+1}, j_{k+1})

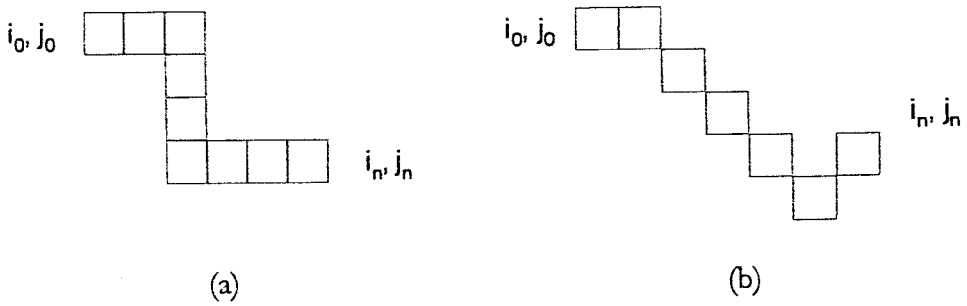
  untuk

  $k = 0, 1, \ldots n-1$

Contoh 4-path dan 8-path ditunjukkan pada gambar 3.6.

Jika dalam suatu citra binari *B*, semua pixel yang bernilai 1 disebut sebagai latar depan dan ditandai sebagai *S*, maka untuk suatu pixel *p* dan *q* dalam *S*:

- *p* adalah 4-connected terhadap *q*, jika terdapat 4-path dari *p* ke *q* yang hanya berisikan pixel dari S.
- *p* adalah 8-connected terhadap *q*, jika terdapat 8-path dari *p* ke *q* yang hanya berisikan pixel dari S.



Gambar 3.6 (a) 4-path dan (b) 8-path dari pixel (i_0, j_0) ke pixel (i_n, j_n)

Dalam hal ini *p* dikatakan memiliki relasi 'is-connected-to' dengan *q*. Relasi 'is-connected-to' merupakan relasi yang ekivalen, karena:

- refleksif, *p* connected dengan dirinya sendiri melalui sebuah path yang panjangnya nol.
- simetris, jika *p* connected ke *q*, maka *q* connected ke *p* melalui path sebaliknya
- transitif, jika *p* connected ke *q* dan *q* connected ke *r*, maka p connected ke r melalui sambungan path dari *p* ke *q* dan *q* ke *r*.

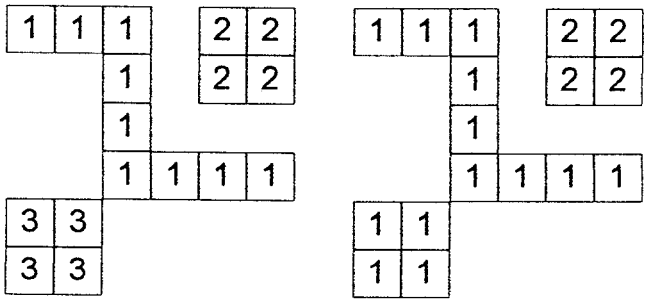
Karena relasi 'is-connected-to' merupakan relasi ekivalen, relasi ini membagi *S* menjadi himpunan kelas atau komponen yang ekivalen yang disebut sebagai connected component.

Jika himpunan *S'* merupakan komplemen dari *S*, yaitu himpunan semua pixel dalam *B* yang bernilai 0, maka:

- S' juga bisa dibagi menjadi himpunan connected component
- Citra dikelilingi oleh suatu bingkai berisikan pixel bernilai 0
- Komponen-komponen dari S' yang adjacent dengan bingkai tersebut disebut sebagai latar belakang dari B
- Komponen lainnya dalam S' disebut sebagai lubang

3.4.4 Component Labeling

Untuk sebuah citra binari B, component labeling memberikan label yang unik untuk semua pixel dalam tiap connected component. Contoh hasil component labeling diperlihatkan pada gambar 3.7.



Gambar 3.7 Hasil component labeling pada latar depan suatu citra binari menggunakan (a) 4-connected dan (b) 8-connected.

Pemberian label dapat dilakukan dengan menggunakan metode depth first labeling secara rekursif, yang diperlihatkan dalam tiga langkah berikut:

- scan keseluruhan citra binari dari atas ke bawah, kiri ke kanan sampai menemukan pixel dengan nilai 1 (0).
- jika pixel tersebut belum ditandai, maka tandai pixel tersebut dengan label baru yang belum digunakan.

- secara rekursif, periksa semua (8, 4) neighbor dari pixel tersebut yang bernilai 1 (0) dan tandai dengan label baru tersebut.

Namun demikian, algoritma rekursif mempunyai beberapa kerugian:

- kecepatan, memerlukan jumlah iterasi sebanding dengan diameter terbesar yang dimiliki suatu connected component dalam citra
- topologi, tidak bisa menentukan komponen-komponen pixel bernilai 0 mana yang merupakan lubang dalam komponen-komponen pixel bernilai 1.

3.4.5 Segmentasi Karakter Dengan Analisa Connected Component

Seperti disebutkan pada sub bab 3.4, analisa connected component mempunyai keunggulan terhadap analisa projection profile dalam hal segmentasi karakter. Hal ini karena analisa projection profile tidak bisa meletakkan cut di antara dua karakter jika tidak terdapat gap yang cukup dalam vertical projection profile histogram. Sebagai alternatif, setelah dilakukan segmentasi baris menggunakan analisa projection profile, selanjutnya digunakan analisa connected component untuk melakukan component labeling terhadap semua komponen dalam baris tersebut. Hasil dari component labeling ini bisa disamakan dengan segmentasi karakter, karena setiap komponen mewakili satu karakter. Yang menjadi perkecualian adalah karakter yang terdiri dari multi-komponen (seperti: i, j, titik dua, titik koma dll).

Perbandingan hasil segmentasi karakter menggunakan analisa projection profile dan analisa connected component diperlihatkan pada gambar 3.8. Kegagalan analisa projection profile pada gambar 3.8 (a) untuk meletakkan cut di antara teks 'anj' disebabkan karena tidak adanya celah pada vertical projection profile histogram di antara ketiga karakter tersebut.

(a) (b)

Gambar 3.8 Hasil segmentasi karakter pada citra teks 'kanji' menggunakan (a) analisa projection profile dan (b) analisa connected component 8-arah.

Namun demikian, analisa connected component masih tetap belum bisa digunakan apa adanya untuk memisahkan dua karakter yang bersentuhan (*merge*) atau menyatukan karakter yang pecah (*break*).

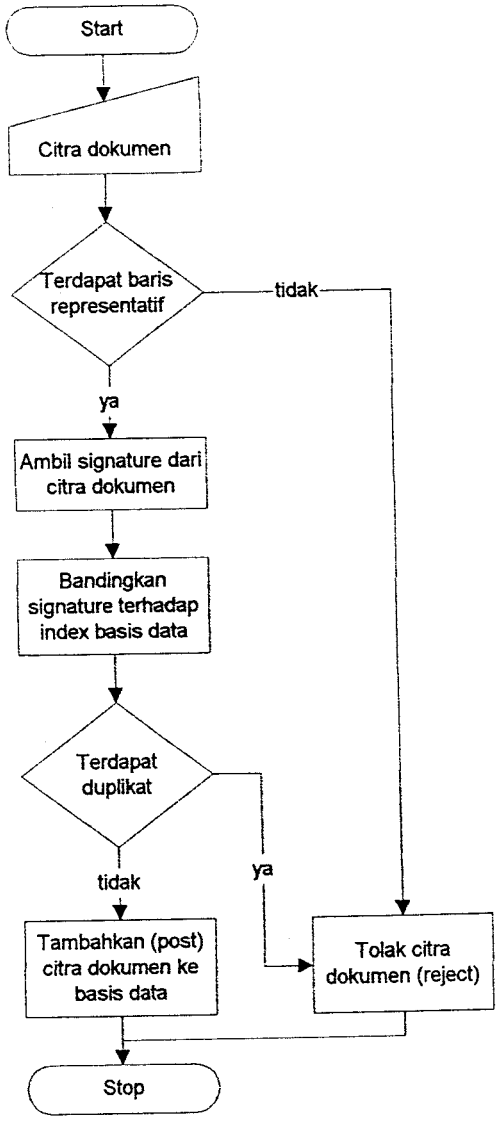
BAB IV

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Bab ini membahas mengenai perancangan dan pembuatan perangkat lunak untuk pendeteksian duplikasi pada basis data citra dokumen dengan menggunakan shape coding. Pembahasan tentang sistem duplikasi ini meliputi teknik shape coding, penentuan baris representatif, pengambilan string shape code sebagai signature, pembandingan signature dan index entry.

4.1 Keseluruhan Proses Dalam Perangkat Lunak

Keseluruhan proses dalam perangkat lunak yang dibuat dalam Tugas Akhir ini diperlihatkan pada gambar 4.1. Proses utama dalam gambar 4.1 adalah menentukan letak baris teks yang dianggap representatif dalam sebuah citra dokumen. Jika tidak berhasil ditemukan, maka citra dokumen akan ditolak (*rejected*). Dan bila berhasil, maka diambil signature dari baris teks tersebut dengan menggunakan teknik shape coding. Teknik shape coding ini memberi label pada setiap karakter dalam baris teks berdasarkan properti sederhana yang dimiliki suatu baris teks dan karakter didalamnya. Dan hasilnya adalah sebuah signature dengan feature set yang termasuk sederhana (10 feature) jika dibandingkan dengan feature set yang digunakan dalam OCR, namun sifatnya cukup unik dan mampu menangani kasus-kasus dimana citra dokumen input memiliki noise atau terdegradasi.



Gambar 4.1 Bagan alur keseluruhan proses dalam perangkat lunak

Signature yang berhasil diambil tersebut akan dibandingkan dengan signature dari citra dokumen lain yang diletakkan pada file index. Dengan menggunakan analisa n-gram akan dihasilkan daftar citra dokumen yang memiliki kesamaan atau yang signaturenya menghasilkan sekian hit terhadap citra dokumen input, yang diurutkan berdasarkan ranking kesamaan. Bila tidak ada citra dokumen yang

memiliki hit lebih dari batas tertentu, maka citra input akan ditambahkan ke dalam basis data. Bila ternyata ada, maka akan diserahkan pada user apakah akan ditambahkan ke dalam basis data atau ditolak.

4.2 Perancangan Perangkat Lunak

Tahap perancangan perangkat lunak dibagi dalam beberapa tahap, yaitu:

- memilih jenis dan jumlah feature set yang digunakan dalam signature.
- menentukan properti baris teks dan karakter yang digunakan dalam teknik shape coding.
- menentukan metode yang digunakan untuk pengambilan signature
- menentukan metode yang digunakan untuk proses signature posting dan signature matching.

4.2.1 Pemilihan Jenis Signature

Metode pengambilan signature dalam Tugas Akhir ini perlu memenuhi beberapa kondisi tertentu. Ada bermacam-macam algoritma pengambilan signature, yang bergantung pada jumlah feature set yang digunakan. Yang perlu diingat adalah jumlah citra dokumen yang akan dimasukkan ke dalam basis data dapat mencapai ribuan, dan banyak di antaranya mengalami degradasi. Karena itu signature digunakan dalam Tugas Akhir ini harus memenuhi beberapa syarat, yaitu:

- *Robust.*. Signature harus tetap bisa diambil sekalipun dokumen mengalami degradasi.

- Unik. Sebenarnya sulit untuk mengharapkan suatu signature adalah unik, kecuali jika digunakan feature set yang berukuran sangat besar. Namun demikian, jenis signature ini diharapkan untuk maksimal berhubungan dengan sekian puluh citra dokumen.
- *Compact.* Ukuran signature yang dihasilkan harus sekecil mungkin, karena diperlukan ruang untuk menyimpan signature dari ribuan citra dokumen.

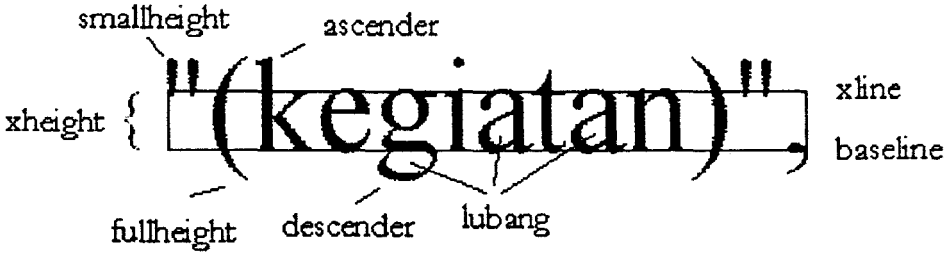
Algoritma pengambilan signature itu sendiri juga harus memenuhi beberapa syarat, yaitu:

- Cepat. Kita tidak bisa menggunakan algoritma yang menghabiskan beberapa menit untuk mengambil signature dan beberapa menit lagi untuk membandingkannya dengan tiap citra dalam basis data. Aplikasi dalam Tugas Akhir ini memerlukan pengambilan signature yang cepat serta waktu pembandingan yang hampir konstan.
- *Scalable.* Algoritma dapat diterapkan untuk basis data yang berisikan puluhan, ratusan atau ribuan dokumen.
- Akurat. User masih dapat menerima jika algoritma gagal dalam mendeteksi sejumlah duplikat, yang mengakibatkan citra dari dokumen yang sama dimasukkan dua kali. Namun mengidentifikasi citra dari dokumen yang berbeda sebagai duplikat (*false alarms*), tidaklah dapat diterima.

4.2.2 Teknik Shape Coding

Dengan memperhatikan syarat-syarat untuk sebuah signature dan algoritma pengambilannya, maka diputuskan untuk menggunakan feature set yang sederhana

namun mampu menangani degradasi dalam citra dokumen. Feature set digunakan tersebut didasarkan pada properti yang dimiliki oleh sebuah baris teks [3], seperti diperlihatkan pada gambar 4.2. Masing-masing properti tersebut dijelaskan dalam tabel 4.1.



Gambar 4.2 Contoh baris teks dan properti yang dimilikinya.

Tabel 4.1 Beberapa macam properti dalam sebuah baris teks

Properti	**Definisi**
xline	Garis tempat meletakkan bagian atas base huruf besar. Juga merupakan garis tempat meletakkan bagian atas huruf 'x'.
baseline	Garis tempat meletakkan bagian bawah base huruf besar
x-height	Tinggi suatu huruf tanpa ascender dan descender, seperti : 'x', yang juga merupakan tinggi base baris teks atau jarak antara xline dan baseline.
ascender	Bagian huruf kecil yang berada di atas x-line. Contohnya, separuh atas bagian vertikal huruf b atau h.
descender	Bagian huruf kecil yang berada di bawah x-height. Contohnya, separuh bawah vertikal huruf y atau g.

Properti-properti sederhana dalam baris teks pada gambar 4.2 masih perlu digabungkan dengan properti yang dimiliki oleh simbol-simbol yang ada didalamnya (dalam hal ini setiap dari simbol tersebut mewakili sebuah karakter), sebelum digunakan dalam signature. Hal ini disebabkan karena signature suatu citra dokumen didasarkan pada kode-kode yang dimiliki suatu karakter dalam citra dan bukan pada properti baris teks semata. Algoritma pemberian label untuk setiap karakter dalam baris teks yang didasarkan pada penggabungan properti baris teks dan properti setiap karakter di dalamnya disebut sebagai teknik shape coding.

Properti-properti karakter dalam suatu baris teks [1] meliputi:

- Tinggi karakter. Properti ini membedakan karakter-karakter atas smallheight, xheight dan fullheight. Karakter smallheight memiliki tinggi kurang dari xheight, sedangkan karakter fullheight lebih tinggi dari xheight dan menempati kesemua area untuk ascender, xheight dan descender.
- Area yang ditempati karakter. Properti ini adalah membedakan karakter-karakter dalam baris teks atas apakah mereka memiliki bagian ascender, descender, xheight atau perpaduan dari ketiganya.
- Keberadaan lubang dalam karakter.
- Keberadaan multi komponen. Contoh karakter yang memiliki lebih dari satu komponen adalah: i, j, titik dua, titik koma, dll.

Dari penggabungan properti tiap karakter dengan properti baris teks, dihasilkan 9 shape code plus satu shape code untuk spasi. Kesepuluh shape code tersebut [1] diperlihatkan pada tabel 4.2. Karakter-karakter yang diwakili oleh masing-masing shape code merupakan karakter yang sering digunakan dalam suatu dokumen.

Tabel 4.2 Macam-macam shape code yang digunakan

Shape code	Nama	Karakter
0	spasi	<spasi>
1	karakter ascender	1 2 3 5 7 C E F G H I J K L M N S T U V W X Y Z f h k l t
2	karakter descender	y
3	karakter x-height	< > * + c m n r s u v w x z
4	karakter ascender yang memiliki lubang	# $ & 0 4 6 8 9 A B D O P Q R b d
5	karakter descender yang memiliki lubang	g p q
6	karakter x-height yang memiliki lubang	a e o
7	karakter fullheight	[ ] ( ) { } |
8	karakter smallheight	, . _ - " '
9	karakter multi komponen	! % ? : ; = i j

4.2.3 Pengambilan Signature

Pengambilan signature merupakan proses utama yang harus dilakukan sebelum proses pembandingan signature. Dalam proses ini program mencari baris teks yang representatif dari citra dokumen input dan mengambil signature berupa string shape code dari baris teks tersebut dengan jenis dan jumlah feature set yang

telah dijelaskan dalam sub bab 4.2.2. Keseluruhan proses pengambilan signature diperlihatkan pada pseudo-code Pascal-like di bawah ini:

```
function getSignature : string;
  segmentasi baris;
  cari tinggi baris dengan frekwensi tertinggi (tft);
  for tiap baris dalam citra dokumen do
    if tinggi baris = tft ± konstanta then
      if baris memiliki ascender dan descender then
        segmentasi karakter;
        ambil string shape code;
        if panjang string shape code >= 50 then
          baris merupakan baris representatif;
        end if;
        ambil baris representatif ketiga sebagai baris signature;
        ambil signature;
      end if;
    end if;
  end for;
  return string shape code string dari baris signature;
end function;
```

Dalam mencari baris teks yang dianggap representatif (*representative line*), digunakan ketentuan-ketentuan sebagai berikut:

- memiliki tinggi baris yang sama dengan tinggi baris dengan frekwensi kemunculan tertinggi. Hal ini diperlukan agar baris representatif tidak berasal dari baris teks yang terlalu kecil, terlalu besar atau berasal dari blok non-teks seperti gambar atau tabel.
- memiliki variasi keberadaan xheight, ascender dan descender. Hal ini diperlukan agar string shape code yang dihasilkan bisa memiliki tingkat diskriminasi atau keunikan yang cukup dalam membedakan antar citra dokumen.
- memiliki string shape code yang panjangnya lebih dari 50 karakter. Panjang ini dirasakan cukup untuk menjamin keunikan signature. Penggunaan string shape code yang lebih panjang, akan menghasilkan signature yang lebih unik namun sekaligus menambah ukuran signature dan index.

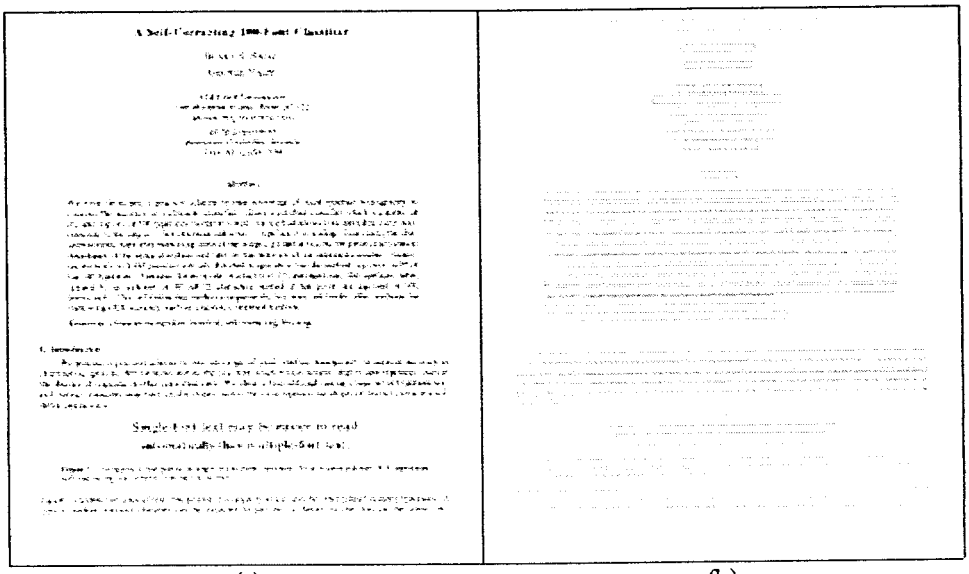
- merupakan baris ketiga yang memenuhi ketiga syarat di atas. Hal ini diperlukan untuk menghindari pengambilan baris representative yang berisikan informasi yang sering muncul (seperti header).

Untuk menjamin integritas sistem, maka keempat ketentuan di atas harus dilakukan secara konsisten dalam proses penambahan citra dokumen ke dalam basis data. Apabila tidak ditemukan baris yang memenuhi keempat syarat di atas, maka tidak ada signature yang diambil dan citra dokumen input akan ditolak oleh sistem (*rejected*). Apabila ditemukan maka signature tersebut akan dibandingkan dengan signature dari citra dokumen lain yang tersimpan dalam file index.

4.2.3.1 Segmentasi Baris

Segmentasi baris (*line segmentation*) bertujuan untuk mencari kotak pembatas (*bounding box*) untuk semua baris teks pada citra dokumen input, seperti yang diperlihatkan pada gambar 4.3. Algoritma segmentasi baris dalam program menggunakan dasar yang sama dengan metode RXYC (*Recursive XY Cuts*) yang telah dibahas dalam bab 3.3. Metode RXYC menghasilkan struktur layout citra dokumen yang diwujudkan dalam struktur tree. Namun berdasarkan pemikiran bahwa program hanya memerlukan kotak pembatas dari baris-baris dalam citra dokumen, maka program melewati proses segmentasi blok atau paragraf, dan langsung menuju proses segmentasi baris dengan hasil akhir berupa struktur link list yang tiap nodenya berisikan koordinat kotak pembatas untuk setiap baris dalam citra dokumen. Secara keseluruhan proses segmentasi baris dapat dilihat pada Pascal-like pseudo-code di bawah ini:

```
function segmentasi_baris(koordinat citra) : list_koordinat;
  x_profile := calculate_profile_on_x_axis(koordinat citra);
  binarize(x_profile,binarize_threshold);
  rlsa(x_profile,x_column_threshold);
  remove_thin_hill(x_profile,thin_threshold);
  x_cuts := find_cuts(x_profile);
  inisialisasi xy_cuts;
  for tiap koordinat pada x_cuts do
    y_profile := calculate_profile_on_y_axis(x_cuts coord);
    binarize(y_profile,binarize_threshold);
    rlsa(y_profile,1);
    remove_thin_hill(y_profile,smooth_threshold);
    y_cuts := find_cuts(y_profile);
    xy_cuts := xy_cuts + y_cuts;
  end for;
  if xy_cuts.coordinates_count > 1 then
    for tiap koordinat pada xy_cuts do
      line_segments := line_segmentation(xy_cuts coordinates);
  else
    line_segments := xy_cuts;
  end if;
end procedure;
```



(a) (b)

Gambar 4.3 Contoh sebuah (a) citra dokumen input dan (b) hasil segmentasi baris berupa kotak-kotak pembatas pada setiap baris (teks dan non-teks).

Pada pseudo-code di atas, nampak bahwa yang membedakan antara metode yang digunakan dalam program dengan metode RXYC adalah nilai threshold RLS (*run length smearing*) horizontal dan vertikal. Jika metode RXYC menggunakan

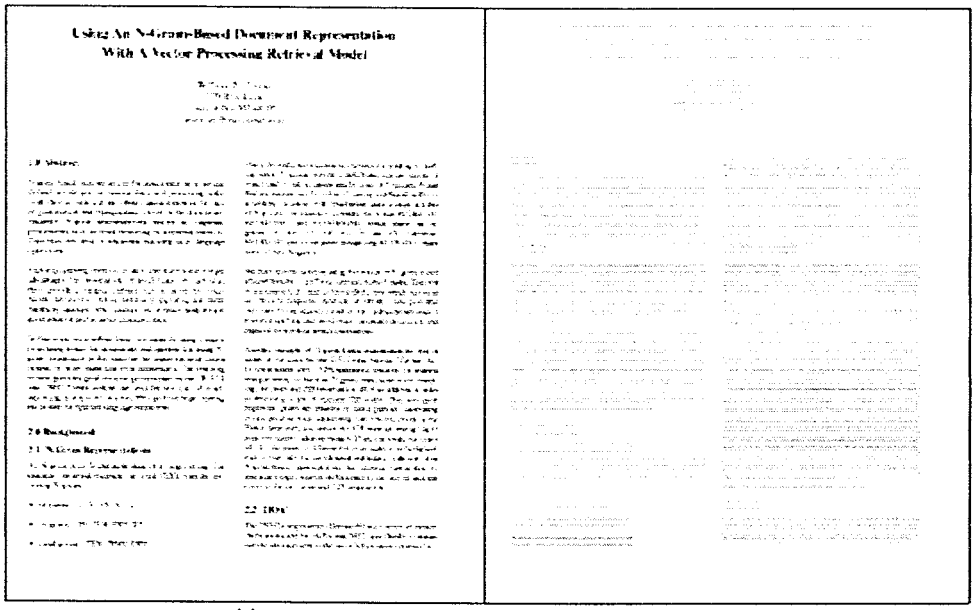
threshold RLS yang berubah-ubah untuk melakukan segmentasi blok, paragraf, baris, kata dan karakter secara berturut-turut, maka metode dalam program menggunakan nilai threshold RLS yang berubah-ubah untuk proses RLS horizontal dan konstan untuk proses RLS vertikal.

Penentuan nilai threshold RLS adalah sebagai berikut:

- Proses smearing horizontal membutuhkan threshold yang cukup untuk memisahkan kolom-kolom utama sambil tetap menyatukan baris, kata dan karakter. Penggunaan threshold yang konstan, walaupun tepat untuk citra dokumen tertentu, menjadi kurang tepat untuk citra dokumen yang memiliki dimensi dan resolusi lain. Jadi yang diperlukan adalah threshold yang otomatis menyesuaikan diri untuk setiap citra dokumen.
- Proses smearing vertikal bisa selalu dilakukan dengan threshold yang konstan, yaitu 1. Hal ini didasari oleh asumsi bahwa suatu baris teks dipisahkan dengan baris teks lainnya dengan jarak vertikal berselang minimal 1 pixel. Asumsi ini berlaku untuk segala macam dimensi dan resolusi citra dokumen. Tentu saja asumsi ini tidak berlaku jika terdapat baris-baris yang menyatu (*merge*) pada beberapa bagian, misalnya yang terjadi karena degradasi dalam citra dokumen.

Implementasi metode segmentasi baris di atas, berhasil dengan baik untuk citra dokumen yang memiliki satu kolom, seperti yang diperlihatkan pada gambar 4.3. Sedangkan untuk citra dokumen yang memiliki lebih dari satu kolom, metode segmentasi baris akan berhasil dengan baik hanya jika ditemukan threshold smearing horizontal yang pas untuk bisa memisahkan kolom-kolom. Jika kondisi tersebut dipenuhi, maka algoritma segmentasi di atas akan mampu menangani citra dokumen dengan struktur layout rumit sekalipun, contohnya seperti struktur layout citra

dokumen yang terdiri dari kolom-kolom yang di dalamnya terdiri lagi dari kolom-kolom. Contoh hasil segmentasi karakter pada citra dokumen yang memiliki lebih dari satu kolom diperlihatkan pada gambar 4.4.

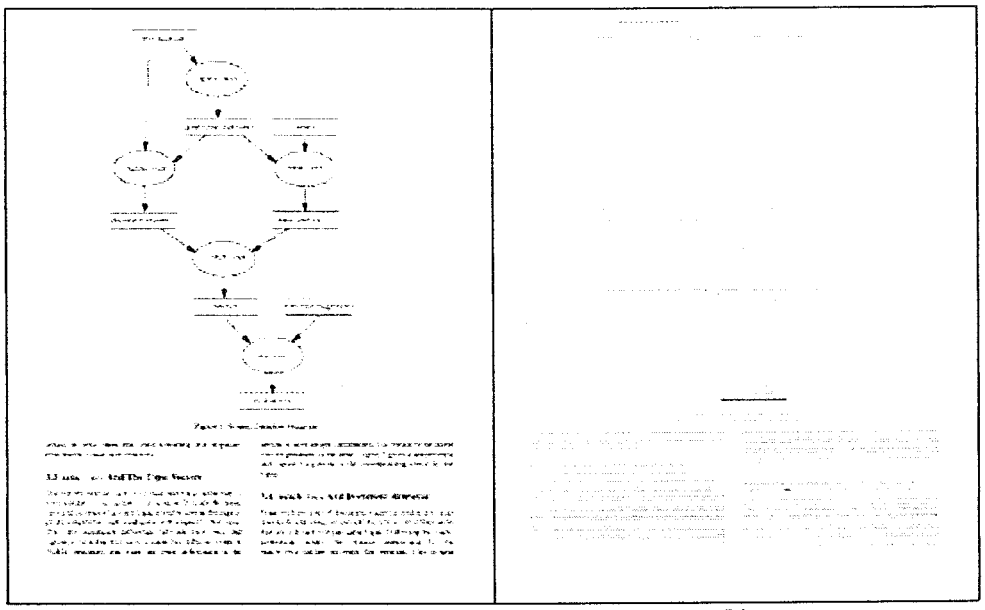


(a) (b)

Gambar 4.4 Contoh (a) citra dokumen input yang memiliki dua kolom dan (b) hasil segmentasi baris citra tersebut.

4.2.3.2 Pemilihan Baris Berdasarkan Tinggi

Tahap ini bertujuan untuk menyeleksi baris-baris yang dihasilkan dalam segmentasi baris berdasarkan tinggi tertentu. Hal ini perlu dilakukan karena dalam menghasilkan kotak-kotak pembatas untuk semua baris, metode segmentasi baris yang diimplementasikan dalam program masih belum bisa membedakan yang mana baris teks, dan yang mana baris non-teks. Hal ini disebabkan oleh penggunaan nilai konstan 1 untuk threshold RLS vertikal, yang mengakibatkan penempatan cut vertikal pada sembarang gap vertikal dengan lebar gap minimal 1 pixel. Contoh dari kondisi tersebut diperlihatkan dalam gambar 4.5.



(a) (b)

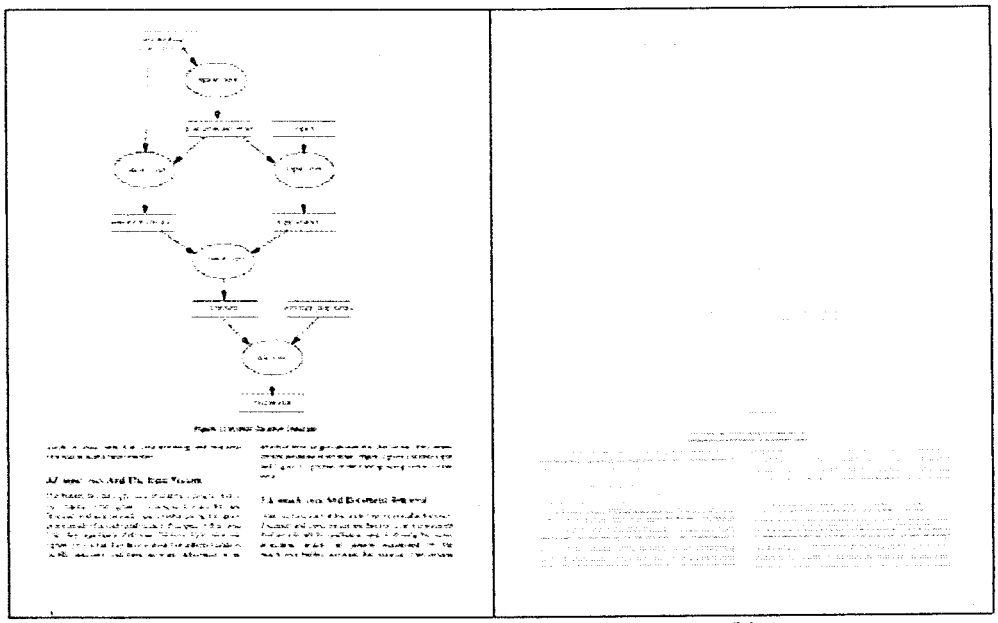
Gambar 4.5 Contoh (a) citra dokumen input yang memiliki blok non-teks dan (b) hasil segmentasi baris citra tersebut.

Berdasarkan keberadaan baris teks dan non-teks, citra dokumen dapat dibagi menjadi empat:

- citra dokumen yang berisikan baris teks saja
- citra dokumen yang berisikan baris non-teks saja
- citra dokumen yang lebih banyak berisikan baris teks
- citra dokumen yang lebih banyak berisikan baris non-teks

Dari pengamatan pada citra dokumen jenis 1 dan 3 dapat disimpulkan bahwa salah satu ciri baris teks adalah memiliki tinggi dengan frekwensi kemunculan terbanyak. Dengan menggunakan kesimpulan tersebut, baris non-teks dapat dipisahkan dari hasil segmentasi baris. Selain berguna untuk memisahkan baris non-teks dari hasil segmentasi baris, pemilihan baris berdasarkan tinggi juga berguna untuk memisahkan baris teks yang memiliki ukuran typeface yang terlalu besar atau

terlalu kecil. Dengan demikian baris-baris yang berisikan judul, header atau footer akan turut dipisahkan dari hasil segmentasi baris. Contoh hasil pemilihan baris berdasarkan tinggi diperlihatkan pada gambar 4.6.



(a) (b)

Gambar 4.6 Contoh (a) citra dokumen input yang memiliki blok non-teks dan (b) hasil pemilihan baris berdasarkan tinggi baris yang memiliki frekwensi kemunculan paling tinggi.

Metode ini kurang tepat jika diterapkan pada citra dokumen jenis 2 dan 4. Untuk citra dokumen jenis 2, tinggi baris yang paling sering digunakan adalah tinggi baris non-teks. Sedangkan untuk citra dokumen jenis 4 yang memiliki baris-baris non-teks dengan struktur layout seperti baris teks (misal: berjajar dari atas ke bawah), akan mengakibatkan tinggi baris non teks menjadi tinggi baris yang paling sering digunakan sehingga justru baris-baris teks yang akan dipisahkan dari hasil segmentasi baris.

4.2.3.3 Pemilihan Baris Berdasarkan Keberadaan Ascender dan Descender

Tahap ini bertujuan untuk sekali lagi menyeleksi baris-baris teks yang dihasilkan oleh proses sebelumnya. Kali ini didasarkan pada keberadaan kombinasi ascender dan descender pada setiap baris teks. Kondisi ini perlu dipenuhi agar nanti dapat dihasilkan string shape code yang paling unik (*maximum discrimination*).

Untuk mengetahui keberadaan ascender atau descender dalam suatu baris teks, bergantung pada keberhasilan untuk menentukan letak xline dan baseline. Letak kedua garis tersebut dapat diketahui dengan cara melakukan proyeksi vertikal terhadap baris teks. Dari projection profile histogram yang dihasilkan akan muncul dua *peak profile* pada separuh atas dan separuh bawah projection profile histogram. Hal ini diperlihatkan pada gambar 4.7.

Gambar 4.7 Contoh projection profile histogram vertikal dari suatu baris teks 'kagaha' serta peak profile atas dan bawah yang dimilikinya.

Gambar 4.8 Hasil pengukuran letak xline dan baseline dari baris teks yang diperlihatkan pada gambar 4.7

Jika projection profile histogram pada gambar 4.7 dibagi menjadi separuh bagian atas dan separuh bagian bawah, maka dapat dilihat bahwa peak profile pada separuh bagian atas menunjukkan letak xline, dan peak profile pada separuh bagian

bawah menunjukkan letak baseline. Hasil penentuan letak xline dan baseline dari baris teks pada gambar 4.7 diperlihatkan pada gambar 4.8.

Untuk mendukung penggunaan projection profile histogram vertikal dalam mengetahui letak xline dan baseline, sebuah baris perlu memenuhi beberapa kondisi, yaitu:

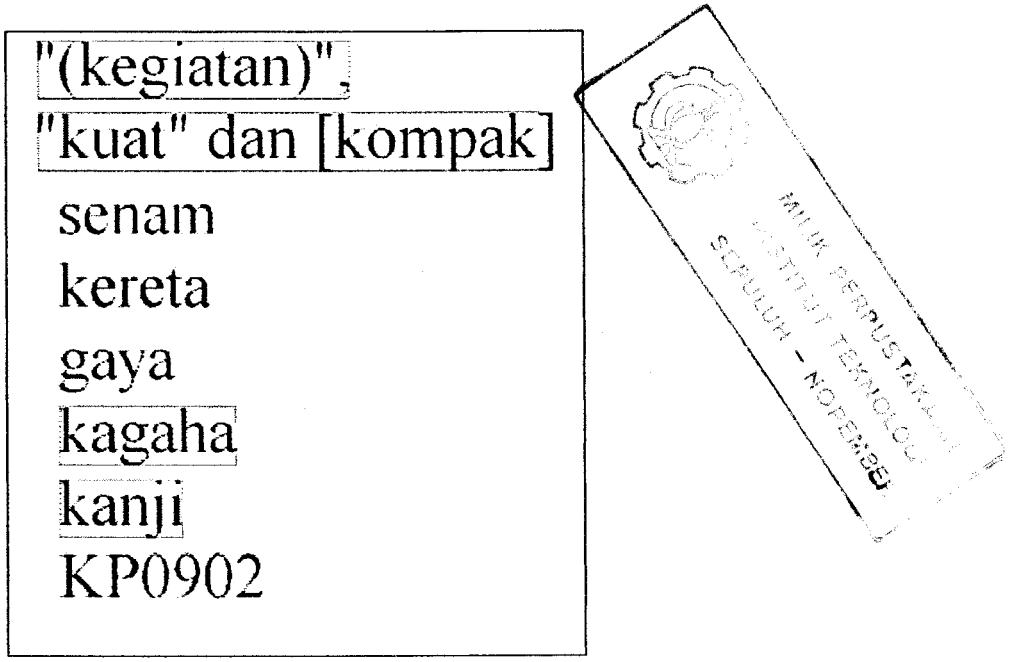
- berisikan banyak karakter. Semakin panjang baris teks tersebut, semakin tepat letak xline dan baseline yang dihasilkan.
- letak posisi base dari tiap karakternya adalah sama. Jenis typeface yang memiliki perbedaan posisi base untuk tiap karakternya, mengakibatkan posisi baseline yang dihasilkan dari analisa projection profile histogram menjadi kurang tepat.

Setelah mengetahui letak xline dan baseline, program dapat mengetahui keberadaan ascender dan descender dalam suatu baris teks. Berdasarkan keberadaan ascender dan descender, baris teks dapat dibagi menjadi 4 jenis seperti yang diperlihatkan pada tabel 4.3.

Tabel 4.3 Pembagian jenis baris teks berdasarkan keberadaan ascender dan descender.

Jenis	**Memiliki**		**Contoh**
	Ascender	**Descender**	
1			senam – sama
2	ya		Kereta – kekal – 0123456789
3		ya	gaya – uang
4	ya	ya	Kompak – pulang

Program mendefinisikan suatu baris teks sebagai jenis 4 jika tinggi ascender baris (jarak antara batas atas baris dengan xline) dan tinggi descender baris (jarak antara batas bawah baris dengan baseline) lebih besar dari 1/5 tinggi baris. Angka 1/5 merupakan konstanta yang didapatkan dari hasil percobaan, untuk mendapatkan hasil yang paling tepat untuk jenis-jenis typeface yang umum digunakan. Contoh hasil pemilihan baris yang memiliki ascender dan descender sekaligus diperlihatkan pada gambar 4.9.



Gambar 4.9 Hasil penyaringan baris. Baris yang diberi garis luar adalah baris yang memiliki variasi ascender dan descender (jenis 4).

4.2.3.4 Pengambilan String Shape Code

Tahap ini bertujuan untuk mengambil string shape code dari suatu baris teks yang dianggap representatif. Salah satu syarat untuk baris yang representatif adalah memiliki string shape code dengan panjang minimal 50 karakter. Jenis dan jumlah feature set yang digunakan telah dijelaskan dalam sub bab 4.2.2.

Dengan menggunakan analisa connected component 8 arah, dapat dihasilkan kotak-kotak pembatas untuk setiap komponen dalam baris. Setiap komponen mewakili satu karakter, kecuali pada karakter multi-komponen seperti 'i', 'j', '?', '!', titik koma, titik dua lain-lain. Untuk menghasilkan shape code untuk setiap karakter seperti yang diperlihatkan pada tabel 4.2, terdapat 4 hal yang perlu diketahui oleh program:

1. Keberadaan ascender, descender, xline, baseline serta xheight. Ukuran yang digunakan untuk menentukan keberadaan properti-properti tersebut diperlihatkan pada tabel 4.5.
2. Keberadaan spasi. Spasi dicari dengan mengetahui lebar rata-rata tiap komponen. Jika jarak antara sisi kanan suatu karakter dengan sisi kiri karakter berikutnya > lebar rata-rata, maka dianggap terdapat spasi di antara kedua karakter tersebut.
3. Keberadaan lubang pada suatu karakter, yang dapat diketahui dengan melakukan analisa connected component 4 arah terhadap latar belakang dari karakter. Bila terdapat komponen latar belakang yang memiliki batas atas, bawah, kiri dan kanan yang tidak sama dengan karakter tersebut, maka karakter tersebut dikatakan memiliki lubang.
4. Keberadaan multi-komponen. Untuk mengetahui apakah suatu komponen merupakan bagian dari suatu karakter multi-komponen, dapat dilakukan proyeksi horizontal terhadap region yang dibatasi oleh batas kiri dan kanan komponen serta batas atas dan bawah baris. Bila dari projection profile histogram ditemukan gap, maka komponen tersebut merupakan bagian dari karakter multi-komponen.

Tabel 4.5 Ukuran yang digunakan untuk menentukan jenis shape code berdasarkan keberadaan ascender, descender, xline, baseline serta xheight.

Jenis karakter	**Tinggi**	**Posisi (character span)**
Karakter dengan ascender	> xheight	dari batas atas baris sampai dengan baseline
Karakter dengan descender	> xheight	dari xline sampai dengan batas bawah garis
Karakter x-height (tanpa ascender dan descender)	xheight	memenuhi base (antara xline dan baseline)
Karakter fullheight	xheight + tinggi ascender + tinggi descender	memenuhi bagian ascender, base dan bagian descender
Karakter smallheight	< xheight	di atas xline, di antara xline dan baseline dan di bawah baseline

4.2.4 Signature Matching Dan Posting

Setelah berhasil mendapatkan signature, ada dua hal yang bisa dilakukan, yaitu:

1. *Signature posting*, yaitu menggunakan signature tersebut sebagai informasi index dari citra dokumen yang akan ditambahkan ke dalam basis data.
2. *Signature matching*, yaitu menggunakan signature tersebut untuk mencari citra duplikat dalam basis data dan

Untuk menambahkan tingkat *robustness*, maka proses signature matching dan posting didasarkan pada semua n-gram dari signature tersebut dan bukan pada keseluruhan signature [1]. Setiap n-gram dari signature berfungsi sebagai kunci index (*index key*) ke dalam basis data. Bila ada satu shape code yang hilang dari atau ditambahkan ke signature hanya akan mempengaruhi beberapa key namun tak akan mempengaruhi keseluruhan signature.

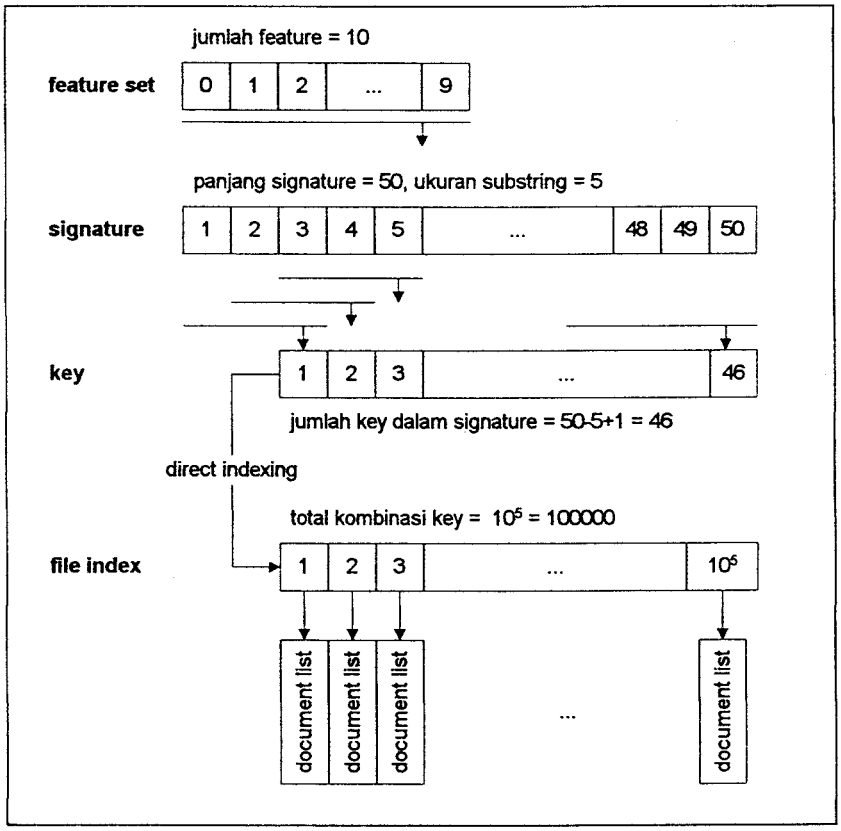
4.2.4.1 Signature Posting

Berdasarkan skema indexing yang diperlihatkan pada gambar 2.4, terdapat beberapa variable *algorithm-dependent* yang dilibatkan dalam analisa n-gram untuk signature, yaitu:

- panjang signature yaitu 50
- panjang masing-masing n-gram yaitu 5, sehingga n-gram yang dihasilkan disebut sebagai 5-gram
- jumlah alfabet atau shape code yang digunakan dalam feature set yaitu 10

Dengan demikian dari sebuah signature dihasilkan 50-5+1 atau 46 n-gram yang akan ditambahkan sebagai *key* ke dalam index. Jumlah total kombinasi key dari feature set yang beranggotakan 10 shape code dengan key yang panjangnya 5 adalah 10^5 atau 100.000 key. Sebagian dari 100.000 key tersebut secara teoritis tidak akan pernah muncul dalam suatu signature, contohnya seperti key yang memiliki deretan shape code 0 (00000, 01000, 00090 dan lain-lain). Masing-masing dari 100.000 key ini menunjuk ke sebuah document list, yang berisikan semua nomor id dokumen yang memiliki key bersangkutan dalam signaturenya. Document list inilah yang

merupakan representasi index basis data citra dokumen yang didasarkan pada analisa n-gram. Semua ini diperlihatkan dalam gambar 4.10.



Gambar 4.10 Skema index signature berdasarkan skema index pada gambar 2.4.

Dalam Tugas Akhir ini, document list diwujudkan dalam struktur tabel yang memiliki field-field sebagai berikut:

- *document_id* menunjuk pada untuk suatu citra dokumen
- *occurrence* menunjukkan jumlah kemunculan key yang berhubungan dengan document list dalam signature citra dokumen yang ditunjuk oleh *document_id.*

Dari yang diperlihatkan pada gambar 4.10, proses signature posting mengambil setiap key dari signature citra dokumen, mencari document list yang berhubungan dengan key tersebut lalu memasukkan id citra dokumen ke dalam

document list tersebut. Bila ternyata document list telah memiliki entry yang berisikan id citra dokumen tersebut, maka nilai *occurrence* untuk id dokumen tersebut dinaikkan satu. Nilai occurrence ini mewakili jumlah kemunculan key yang berhubungan dengan document list dalam dokumen yang diwakili oleh id dokumen. Keseluruhan proses signature posting diperlihatkan dalam Pascal-like pseudo code di bawah ini:

```
procedure post_signature;
  new_document_id := id untuk citra dokumen baru;
  for setiap key dalam signature do
    document_list := document list yang berhubungan dengan key;
    if terdapat new_document_id dalam document_list then
      curr_record := record yang berisikan new_document_id;
      curr_record.occurrence := curr_record.occurrence+1;
    else
      tambahkan record baru pada document_list;
      curr_record := record baru;
      curr_record.document_id := new_document_id;
      curr_record.occurrence := 1;
    end if;
  end for;
end procedure;
```

4.2.4.2 Signature Matching

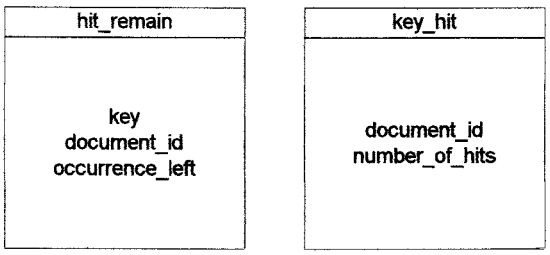
Dari gambar 4.10 diketahui bahwa semua signature citra dokumen dalam basis data diletakkan dalam document list dengan menggunakan analisa n-gram. Karena itu proses signature matching melakukan mengambil setiap key dari signature citra dokumen kandidat dan menggunakannya dalam proses query terhadap document list. Setiap key menghasilkan sejumlah dari hit dari document list dan setiap hit dihitung sebagai vote untuk citra dokumen kandidat. Nilai total vote dihitung, lalu program menampilkan daftar citra dengan sekian hit, diurutkan berdasarkan nilai total vote.

Proses signature matching memerlukan dua buah list temporary. List pertama menampung daftar id dokumen yang memiliki kesamaan key dengan

dokumen kandidat serta jumlah hit untuk masing-masing id dokumen. List kedua mencatat jumlah hit tersisa yang dimiliki suatu citra dokumen selama proses pembandingan. Dalam implementasinya, kedua list tersebut diwujudkan dalam bentuk tabel, yaitu tabel *key_hit* dan tabel *hit_remain* yang strukturnya diperlihatkan pada gambar 4.11.

Keseluruhan proses signature matching diperlihatkan Pascal-like pseudo-code di bawah ini:

```
procedure match_signature;
  inisialisasi tabel hit_remain;
  inisialisasi tabel key_hit;
  for tiap key dalam signature do
    query untuk document_list.key = key;
    for setiap record dalam query subset do
      cari record dalam hit_remain dengan key+document_id yang
      sama dengan query subset;
      kurangi jumlah hit yang ada dalam hit_remain;
      if jumlah hit dalam hit_remain > 0 then
        cari record dalam key_hit dengan document_id yang sama;
        naikkan jumlah hit;
      endif;
    end for;
  end for;
end procedure;
```



Gambar 4.11 Struktur tabel hit_remain dan key_hit.

4.3 Pembuatan Perangkat Lunak

Pembuatan perangkat lunak dibagi dalam beberapa tahap:

Komponen PixelGraphicLibrary digunakan karena mendukung manipulasi file citra format TIFF dengan RLE PackBits, serta manipulasi citra multipage. Sedangkan komponen Apollo digunakan karena mendukung penggunaan file tabel DBF beserta multi compound index (.CDX). Apollo mendukung semua versi format file DBF serta teknologi Rushmore, yang digunakan dalam Microsoft Foxpro 2.x ke atas.

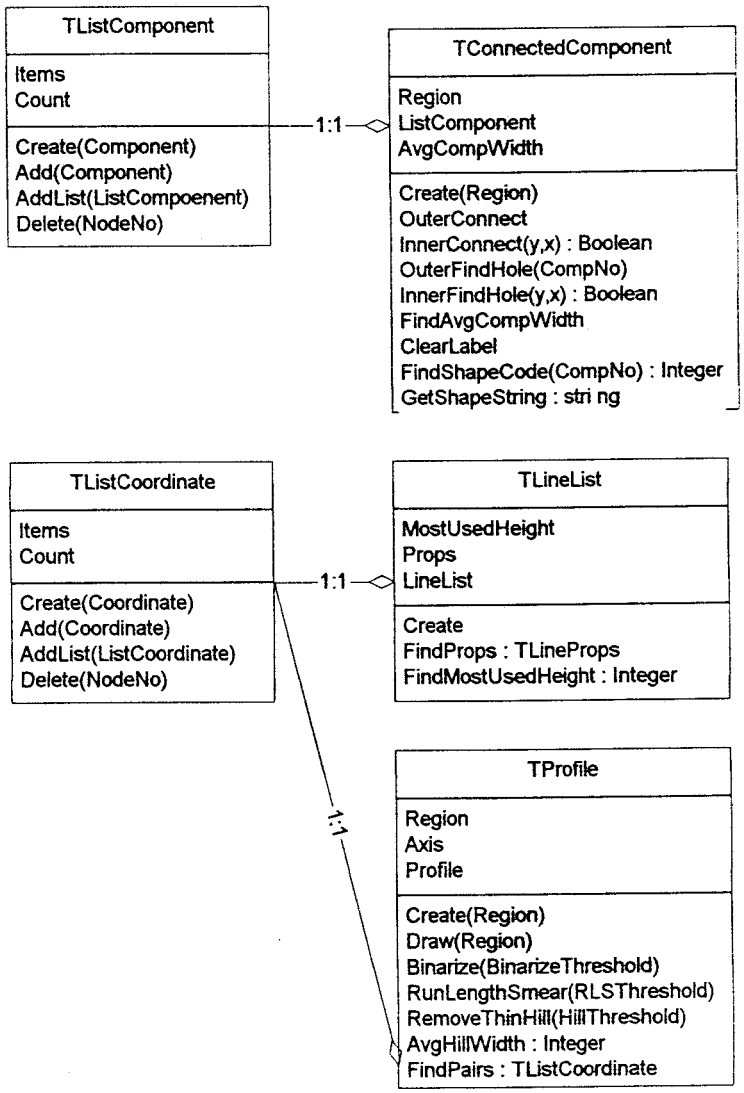
4.3.3 Class Yang Digunakan

Program menggunakan lima class untuk implementasi rancangan perangkat lunak. Daftar class dan deskripsi object-nya diperlihatkan dalam tabel 4.6.

Tabel 4.6 Daftar class dan deskripsi object-nya.

Class	Deskripsi object
TprojectionProfile	Mewakili sebuah projection profile histogram yang dihasilkan dari proyeksi sebuah region dalam citra input terhadap sebuah sumbu (y atau x)
TconnectedComponent	Mewakili sebuah list yang berisikan semua komponen yang dihasilkan dari proses component labeling terhadap sebuah region dalam citra input
TlineList	Mewakili sebuah list yang berisikan koordinat semua kotak pembatas yang dihasilkan dari proses segmentasi baris terhadap sebuah region dalam citra input
TlistCoordinate	Mewakili sebuah single link-list yang setiap node-nya berisikan koordinat sebuah region.
TlistConnected	Mewakili sebuah single link-list yang setiap node-nya berisikan koordinat sebuah komponen dari hasil proses component labeling terhadap suatu region dalam citra input.

Diagram seluruh class yang digunakan dalam program diperlihatkan dalam gambar 4.12.



Gambar 4.12 Diagram class yang digunakan dalam perangkat lunak

Dalam penerapan rancangan diagram class pada gambar 4.12 dengan bahasa pemrograman Borland Delphi 3.0, digunakan class *TObject* yang merupakan struktur data paling dasar bagi semua class. Class Tobject menyediakan interface yang memberikan kemudahan bagi programmer dalam pendefinisian suatu class serta

pembuatan object atau instance dari class tersebut. Hal ini dilakukan dengan menangani secara internal semua hal yang berkaitan dengan penyediaan, inisialisasi dan pelepasan memori yang digunakan untuk object suatu class.

4.3.3.1 Class TProjectionProfile

Properti utama dari class TprojectionProfile adalah sebuah array integer *Profile* yang menampung projection profile histogram dari hasil proyeksi sebuah region dalam citra input terhadap suatu sumbu (y atau x). Implementasi class TProjectionProfile dalam program diperlihatkan melalui struktur data berikut:

```
AxisType = (xAxis,yAxis);
   TprojectionProfile = class(TObject)
     private
       iProfile : PintArray;
       iAxis : AxisType;
       iRegion : Tcoordinate;
       iStart,iFinish : Integer;
     public
       constructor Create(SourcePG : TPixelGraphic;
                          Region:Tcoordinate; aAxis:AxisType);
       destructor Destroy; override;
       procedure Free;
       procedure Draw(var SourcePG : TPixelGraphic;
                      yTopLeft : Integer; xTopLeft : Integer);
       procedure Binarize(NormalizePercentage : real);
       procedure RunLengthSmear(RunLengthSmearThreshold:Integer);
       procedure RemoveThinHill(HillThreshold : Integer);
       function AvgHillWidth : Integer;
       function FindPair : TListCoordinate;
       property Region : Integer read iRegion;
       property Axis : AxisType read iAxis;
       property Profile : PIntArray read iProfile;
   end;
```

Daftar property class TprojectionProfile diperlihatkan pada tabel 4.7, sedangkan daftar method-nya diperlihatkan pada tabel 4.8.

Tabel 4.7 Daftar property dalam class TprojectionProfile

Properti	Deskripsi
Region	Berisikan koordinat batas (*border*) suatu region dalam citra input.
Axis	Berisikan sumbu arah proyeksi (sumbu y atau x).
Profile	Berisikan projection profile histogram yang dihasilkan dari proyeksi terhadap sumbu *Axis* pada daerah *Region* dalam citra input. Wujud fisiknya berupa sebuah array integer.

Tabel 4.8 Daftar method dalam class TprojectionProfile

Method	Deskripsi
Constructor Create	Mengambil koordinat sebuah region pada citra input dan menyimpannya dalam *Region*. Setelah itu menghitung projection profile histogram terhadap sumbu *Axis* untuk daerah *Region* dan menyimpannya dalam *Profile*.
Procedure Draw	Menggambar histogram *Profile* ke citra output.
Procedure Binarize	Melakukan binarisasi terhadap histogram *Profile*.
Procedure RunLengthSmear	Melakukan RLS (*run length smearing*) terhadap histogram *Profile*.
Procedure RemoveThinHill	Dijalankan setelah RunLengthSmear, guna menghapus deretan non-zero profile dalam histogram *Profile* yang lebarnya kurang dari threshold.
Function FindPair	Dijalankan setelah RunLengthSmear, untuk membuat sebuah list yang tiap node-nya berisikan deretan non-zero profile dalam histogram *Profile*.

4.3.3.2 Class TConnectedComponent

Properti utama dari class TprojectionProfile adalah sebuah list *ListComponent* yang mewakili semua 8-connected component yang dihasilkan oleh proses

component labeling rekursif terhadap sebuah region pada citra input. Implementasi class TprojectionProfile dalam program diperlihatkan dari struktur data berikut

```
TConnectedComponent = class(TObject)
private
  iNewLabel,iSumCompWidth,iAvgCompWidth : Integer;
  iImageValue,iImageLabel : TPixelGraphic;
  iRegion,iCompRegion : TCoordinate;
  iListComponent,iListHole : TListComponent;
public
  constructor Create(SourcePG:TPixelGraphic;
                     Region:Tcoordinate);
  destructor Destroy; override;
  procedure Free;
  procedure OuterConnect;
  function InnerConnect(y : Integer; x : Integer) : Boolean;
  function OuterFindHole(i : Integer) : Boolean;
  function InnerFindHole(y : Integer; x : Integer) : Boolean;
  procedure FindAvgCompWidth;
  function FindShapeCode(i:Integer;
                         theLine:LineProps):Integer;
  function GetShapeString : string;
  property Region : Tcoordinate read iRegion;
  property ListComponent:TListComponent read iListComponent;
  property AvgCompWidth : Integer read iAvgCompWidth;
  property SumCompWidth : Integer read iSumCompWidth;
end;
```

Tabel 4.9 Daftar properti dalam class TconnectedComponent

Properti	Deskripsi
Region	Berisikan koordinat daerah dalam citra input. List komponen dalam object instance dihasilkan dari proses component labeling terhadap daerah *Region.*
ListComponent	Berisikan sebuah list yang node-node-nya berisikan koordinat dari dari semua komponen yang berada dalam daerah *Region* dalam citra input.
AvgCompWidth	Berisikan lebar rata-rata dari semua komponen dalam *ListComponent.*
SumCompWidth	Berisikan jumlah lebar dari semua komponen dalam *ListComponent.*

Sebuah object dari class ini mewakili sebuah list 8-connected component. Daftar property class TconnectComponent diperlihatkan pada tabel 4.9, sedangkan daftar method-nya diperlihatkan pada tabel 4.10.

Tabel 4.10 Daftar method dalam class TconnectedComponent

Method	Deskripsi
Constructor Create	Mengambil sebuah daerah pada citra input dan menyimpan koordinatnya dalam *Region*. Setelah itu constructor menginisialisasi sebuah list *ListComponent* yang akan menampung semua 8-connected component dalam daerah *Region*. Terakhir constructor menginisialisasi sebuah array dua dimensi *iImageLabel* yang memiliki dimensi yang sama dengan citra yang berada dalam daerah *Region*. Array ini digunakan untuk menyimpan informasi label untuk tiap pixel citra dalam daerah *Region*.
Procedure OuterConnect	Melakukan component labeling secara rekursif dalam daerah *Region*, kemudian meletakkan hasilnya dalam *ListComponent*.
Function OuterFindHole	Mengambil sebuah komponen dalam *ListComponent*, lalu mengembalikan nilai *True* jika komponen tersebut memiliki lubang dan *False* jika tidak.
Procedure FindAvgCompWidth	Menghitung lebar rata-rata dari semua komponen dalam *ListComponent* dan meletakkan hasilnya dalam *AvgCompWidth*. Nilai lebar rata-rata ini digunakan untuk menentukan letak shape spasi dalam string shape code.
Function FindShapeCode	Mengambil sebuah komponen dari *ListComponent* dan mengembalikan nilai shape code untuk komponen tersebut.
Function GetShapeString	Mengembalikan string shape code untuk semua komponen dalam *ListComponent*.

4.3.3.3 Class TLineList

Properti utama dari class TlineList adalah sebuah list *LineList* yang setiap nodenya berisikan koordinat dari semua bounding box yang dihasilkan oleh proses segmentasi baris. Implementasi class dalam program diperlihatkan pada struktur data berikut:

```
TLineList = class(TObject)
  private
    iMostUsedHeight,iMostUsedWidth : Integer;
    iProps : LineProps;
    iLineList : TListCoordinate;
  public
    constructor Create;
    destructor Destroy; override;
    procedure Free;
    procedure FindMostUsedHeight;
    procedure FindMostUsedWidth;
    procedure FindProps(LineNumber:Integer);
    property MostUsedHeight : Integer read iMostUsedHeight;
    property MostUsedWidth : Integer read iMostUsedWidth;
    property Props : LineProps read iProps;
    property LineList:TListCoordinate read iLineList;
  end;
```

Tabel 4.11 Daftar properti dalam class TlineList

Properti	Deskripsi
LineList	Berisikan sebuah list yang nodenya berisikan semua koordinat bounding box dari hasil proses segmentasi baris.
MostUsedHeight	Berisikan tinggi baris yang paling sering muncul dalam *LineList*
MostUsedWidth	Berisikan yang paling sering muncul dalam *LineList*
Props	Berisikan semua properti yang dimiliki oleh sebuah baris teks seperti tinggi baris, posisi garis tengah, posisi garis tengah atas, posisi baris tengah bawah, posisi xline, posisi baseline, posisi garis tengah base, tinggi base, tinggi ascender, tinggi descender, posisi garis tengah ascender, posisi garis tengah descender serta keberadaaan ascender dan descender.

Daftar property class TlineList diperlihatkan pada tabel 4.11, sedangkan daftar method-nya diperlihatkan pada tabel 4.12.

Tabel 4.12 Daftar method dalam class TlineList.

Method	Deskripsi
Constructor Create	Melakukan inisialiasi pada *LineList*, *MostUsedHeight* serta *MostUsedWidth*.
Procedure FindMostUsedHeight	Menghitung tinggi baris yang paling sering muncul dalam *LineList* dan meletakkan hasilnya pada properti *MostUsedHeight*.
Procedure FindMostUsedWidth	Menghitung lebar baris yang paling sering muncul dalam *LineList* dan meletakkan hasilnya pada properti *MostUsedWidth*.
Procedure FindProps	Mengambil sebuah baris dalam *LineList* lalu menghitung semua properti dalam baris teks tersebut dan meletakkan hasilnya dalam properti *Props*.

4.3.4 Sistem Basis Data

Dalam proses signature posting dan signature matching banyak dilakukan operasi pencarian data dan query. Hal ini menunjuk pada penggunaan struktur tabel dalam pengimplementasian kedua proses tersebut. Dan cara termudah dan tercepat untuk menerapkan struktur data tabel adalah dengan memanfaatkan sistem basis data relational yang telah ada. Dalam menentukan sistem basis data relasional yang akan digunakan, perlu dipertimbangkan hal-hal berikut:

- Batasan ukuran file tabel dan index, jumlah record yang bisa diletakkan dalam satu tabel.

- Waktu yang diperlukan serta manajemen memori dalam membuka file yang berukuran besar.
- Waktu yang diperlukan untuk menambahkan record baru serta update index.
- Optimasi penggunaan index dalam operasi pencarian data dan query.

Point terakhir merupakan hal yang paling, karena begitu seringnya dilakukan proses pencarian dan query dalam proses signature matching dan signature posting.

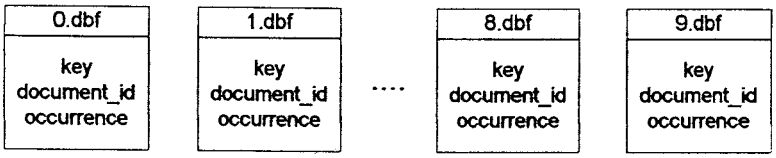
Dengan menggunakan pertimbangan-pertimbangan di atas, maka program menggunakan sistem basis data relasional Apollo 4.5 yang digunakan sebagai komponen dalam lingkungan Borland Delphi 3.0. Apollo 4.5 memiliki beberapa kelebihan dibandingkan dengan Borland Database Engine (BDE) dalam hal penggunaan index untuk optimasi proses pencarian dan query. Apollo juga mendukung sepenuhnya penggunaan tabel dengan format file DBF beserta multiple compound index yang digunakan dalam Microsoft Foxpro versi 2.x ke atas. Dukungan terhadap versi DBF ini juga diikuti dengan dukungan penggunaan teknologi Rushmore untuk optimasi dalam operasi pencarian dan query.

4.3.5 Implementasi Document List Dengan Tabel

Jumlah keseluruhan document list adalah 100.000, sesuai dengan jumlah keseluruhan kemungkinan kombinasi key dengan panjang 5 terhadap feature set yang beranggotakan 10 alfabet. Jika setiap document list diimplementasikan dengan satu file tabel maka sistem akan membutuhkan 100.000 file.

Dalam sistem ini, 100.000 document list tersebut dibagi ke dalam 10 file tabel. Pembagian ini didasarkan oleh shape code pertama yang ada dalam sebuah

key, yang sekaligus menjadi nama dari ke-10 file tersebut. Masing-masing tabel memiliki struktur yang sama, yang diperlihatkan pada gambar 4.15.



Gambar 4.15 Struktur tabel yang mewakili document list.

BAB V

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Bab ini membahas tentang cara-cara pelaksanaan uji coba, hasil uji coba dan evaluasi perangkat lunak, yang menjelaskan mengenai faktor-faktor yang mempengaruhi unjuk kerja perangkat lunak.

5.1 Spesifikasi Sistem

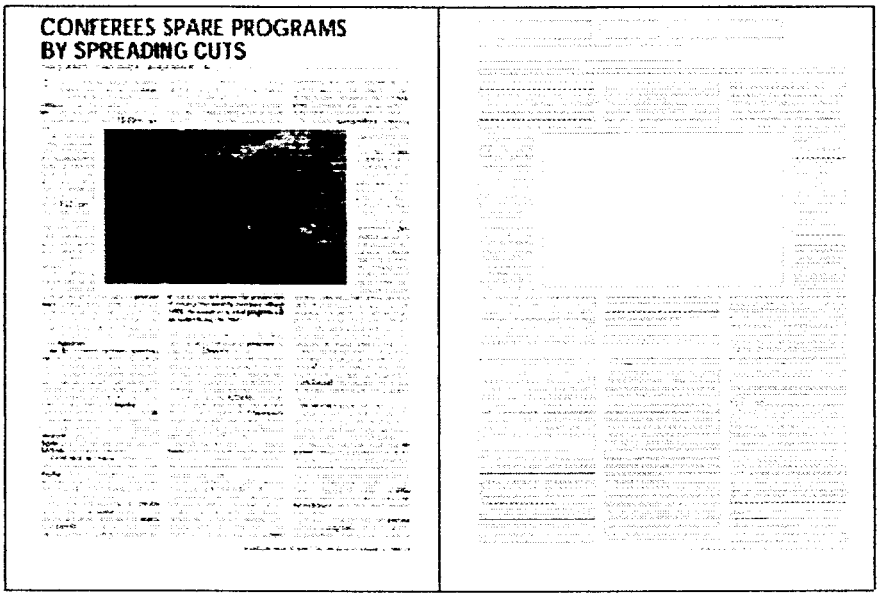
Untuk uji coba perangkat lunak digunakan perangkat keras dengan prosesor AMD DX4-133, RAM 16 MB, harddisk 1,2 GB, VGA card 1MB, serta monitor SVGA. Program dijalankan dalam lingkungan sistem operasi Windows 95 dengan tampilan 256 warna.

5.2 Uji Coba Pengambilan Signature

Pengambilan signature dari sebuah citra dokumen kandidat merupakan langkah awal dari proses pencarian duplikat dari citra tersebut. Umumnya untuk citra yang bersih dan tidak terdegradasi, pengambilan signature menunjukkan hasil seperti yang diharapkan. Penambahkan efek degradasi pada citra yang sama, mengakibatkan perubahan pada signature yang dihasilkan. Mulai dari hilangnya atau bertambahnya beberapa shape code sampai pada berubahnya posisi baris yang dianggap representatif. Berikut ini akan dibahas mengenai hasil uji coba untuk masing-masing proses dalam pengambilan signature.

5.1.1 Uji Coba Segmentasi Baris

Uji coba proses segmentasi baris menunjukkan hasil yang sesuai dengan yang diharapkan pada citra dokumen yang bersih; tanpa noise, kemiringan dan degradasi yang berarti. Program dapat menentukan koordinat kotak pembatas untuk semua baris (teks dan non teks) dalam citra dengan satu kolom atau lebih. Dan untuk struktur layout dokumen yang rumit sekalipun, bisa diatasi dengan baik berkat penggunaan metode rekursif. Salah satu contohnya diperlihatkan dalam gambar 5.1.

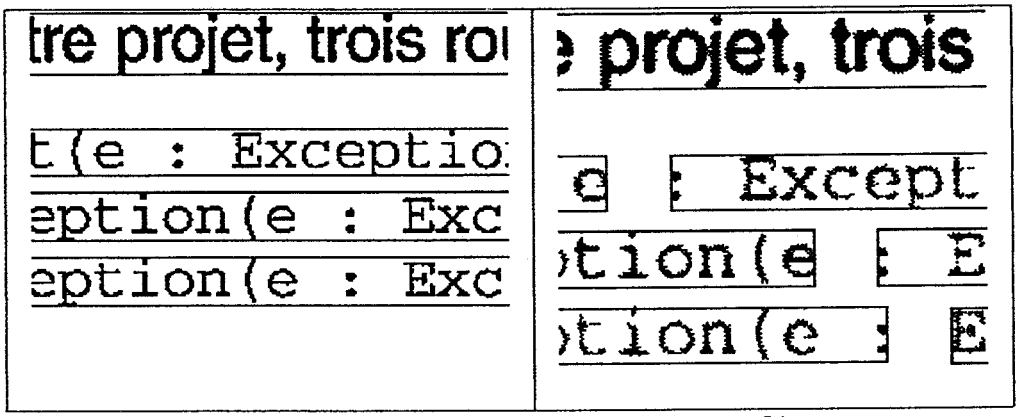


(a) (b)

Gambar 5.1. Salah satu contoh hasil segmentasi baris pada citra dokumen.

Hal yang mempengaruhi munculnya hasil segmentasi yang tidak sama dengan yang diharapkan, adalah nilai threshold proses RLS terhadap projection profile histogram vertikal. Penggunaan nilai konstan untuk threshold ini dapat mengakibatkan segmentasi baris pada suatu citra dokumen menjadi segmentasi kata pada citra lain dari dokumen sama dengan resolusi lebih tinggi. Keadaan ini diperlihatkan dalam gambar 5.2. Hal ini bisa mengakibatkan sebuah kotak pembatas

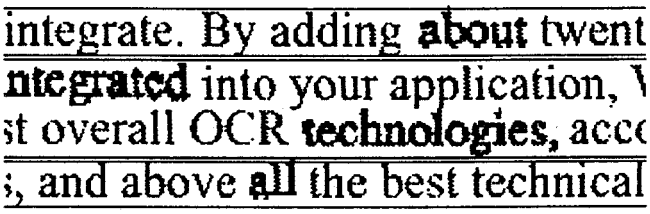
untuk sebuah baris dalam sebuah citra dokumen terbagi menjadi beberapa baris dalam citra dari dokumen yang sama dengan dimensi lebih besar. Terbaginya kotak pembatas baris ini dapat mengakibatkan jumlah shape code yang dimiliki baris tersebut tidak lagi memenuhi persyaratan untuk baris yang representatif. Dengan demikian baris representatif yang dihasilkan menjadi lain, yang akhirnya mengakibatkan signature yang dihasilkan menjadi berbeda dan kedua citra tersebut akan dinyatakan sebagai berbeda pula. Namun seperti yang telah disebutkan dalam sub bab 4.2.1, hal ini masih bisa diterima karena tidak termasuk sebagai *false alarm*.



Gambar 5.2 Perbedaan hasil segmentasi baris pada citra (a) dan citra (b) yang berasal dokumen yang sama dengan resolusi lebih tinggi.

Dalam uji coba lain, digunakan citra dokumen terdegradasi yang memiliki satu atau lebih baris teks yang bagian ascender atau descendernya menyatu dengan bagian ascender atau descender dari baris teks lainnya. Penggunaan analisa projection profile untuk segmentasi tidak bisa digunakan untuk memisahkan baris-baris tersebut, sekalipun baris-baris tersebut tidak benar-benar menyatu menurut penilaian mata manusia. Kondisi ini diperlihatkan pada gambar 5.3. Selama hal ini

tidak mempengaruhi hasil penentuan letak baris teks yang representatif, maka tidak menjadi masalah.



Gambar 5.3 Contoh baris (kedua dan ketiga) yang tidak berhasil dipisahkan oleh proses segmentasi baris yang didasarkan pada analisa projection profile.

5.1.2 Uji Coba Pemilihan Baris Berdasarkan Tinggi

Uji coba pemilihan baris berdasarkan tinggi memberikan hasil yang diinginkan untuk citra dokumen yang memiliki lebih banyak blok teks daripada blok non-teks dari hasil segmentasi baris. Dalam kasus-kasus tersebut, proses ini berhasil memisahkan blok-blok non teks serta baris-baris teks yang terlalu kecil atau terlalu besar dari hasil segmentasi baris sebelumnya.

Untuk input berupa citra dokumen yang memiliki lebih banyak blok non-teks daripada blok teks, program akan menetapkan tinggi baris yang dimiliki oleh blok non-teks sebagai tinggi baris yang paling sering digunakan. Dalam kasus ini, jika tinggi baris dalam blok teks terlalu kecil atau terlalu besar jika dibandingkan dengan tinggi baris yang paling sering digunakan, maka justru baris-baris teks tersebut yang akan dipisahkan dari hasil segmentasi baris.

5.1.3 Uji Coba Pemilihan Baris Berdasarkan Keberadaan Ascender dan Descender

Uji coba penentuan baris teks yang memiliki kombinasi ascender dan descender dinilai cukup berhasil untuk jenis typeface yang sering digunakan seperti Times New Roman atau Garamond. Ini juga termasuk jenis-jenis typeface yang memiliki kemiripan dalam perbandingan letak ascenderline, xline, baseline serta descenderline.

Penentuan keberadaan ascender dan descender dalam baris teks ditentukan oleh letak xline dan baseline. Penentuan letak xline dan baseline ini dilakukan dengan proyeksi horizontal baris teks. Maka semakin panjang baris teks tersebut, semakin tepat letak hasil pencarian posisi xline dan baseline.

Dalam kasus baris teks yang memiliki sedikit bagian ascender dan descender, proses segmentasi baris akan menganggap baris tersebut tidak memiliki ascender atau descender, sehingga baris ini tidak ikut terpilih. Namun hal ini justru membantu untuk hanya mengikutkan baris-baris teks dengan kombinasi ascender dan descender yang cukup dismriminatif ke proses berikutnya.

Dalam beberapa kasus, baris-baris teks pendek yang tidak memiliki keberadaan ascender dan descender juga turut terpilih. Namun hal ini tidak menjadi masalah, karena baris-baris teks ini tidak akan disertakan dalam penentuan baris representatif, berhubung jumlah shape code yang dihasilkannya kurang dari 50.

5.1.4 Uji Coba Pengambilan String Shape Code

Kesulitan yang dialami oleh proses pengambilan string shape code adalah saat harus berhadapan dengan satu karakter yang memisah menjadi beberapa komponen

atau dua karakter atau lebih yang menyambung menjadi satu komponen. Berkat penggunaan analisa n-gram pada signature, satu karakter yang hilang atau muncul hanya akan mempengaruhi beberapa key saja, namun tidak untuk keseluruhan signature. Sehingga jika dibandingkan dengan signature citra dari dokumen yang sama, akan menghasilkan hit yang cukup untuk bisa dikatakan sebagai citra duplikat. Tentu saja hal ini tidak berlaku jika terlalu banyak karakter yang memisah atau menyatu dalam satu baris teks. Atau jika degradasi pada citra tersebut mempengaruhi posisi baris representatif yang dihasilkan.

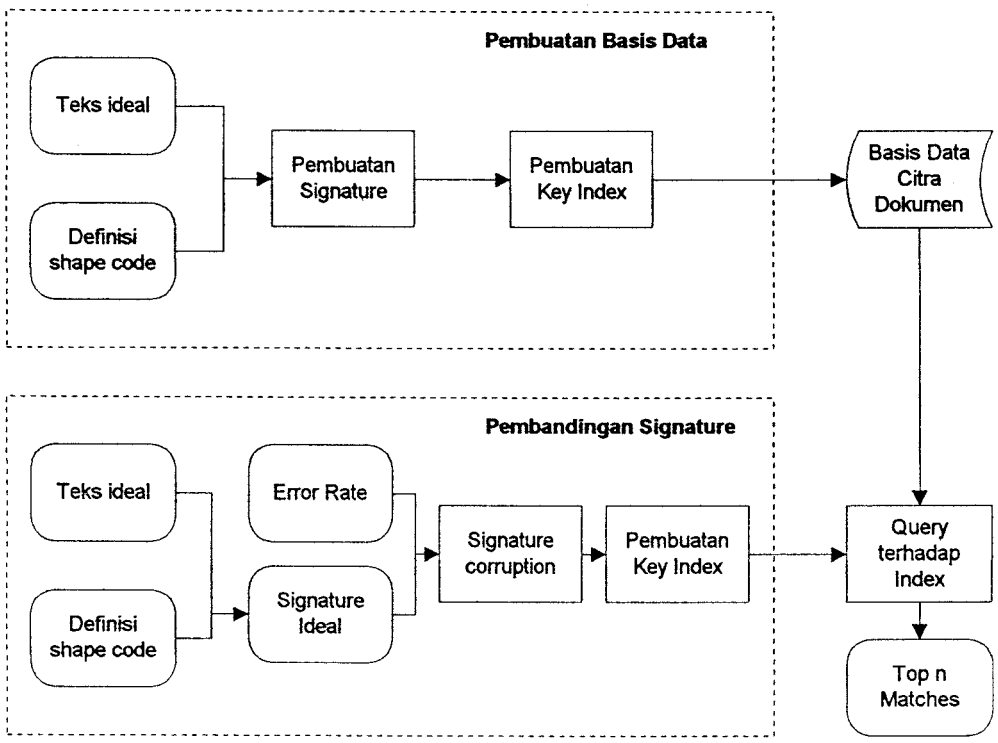
5.2 Uji Coba Pembandingan Signature

Untuk menunjukkan kelayakan teknik shape coding yang digunakan dalam Tugas Akhir ini, perlu dilakukan beberapa percobaan dengan menggunakan signature yang memiliki sifat ideal dan yang terkorupsi. Untuk itu diperlukan basis data citra dokumen yang memiliki banyak citra, dari ratusan hingga ribuan. Karena besarnya biaya (waktu dan ruang penyimpanan) yang diperlukan untuk membuat basis data semacam itu, maka digunakan simulasi.

Dengan simulasi, kita tidak hanya bisa mengurangi biaya uji coba tapi juga sekaligus dapat berkonsentrasi pada uji coba pembandingan signature itu sendiri tanpa harus memperhitungkan masalah pengambilan signature dari tiap citra dokumen. Simulator yang digunakan akan mencoba bermacam variasi dalam signature posting dan signature matching. Tujuan akhirnya adalah untuk membuktikan bahwa signature yang dihasilkan oleh teknik shape coding sudah cukup unik untuk digunakan dalam proses indexing dan index matching dalam basis data citra dokumen yang berukuran besar.

5.2.1 Model Input dan Error

Simulator yang dibuat mensimulasikan pembuatan basis data citra dokumen dengan cara menggantikan citra dokumen dengan file teks ASCII. Jika n adalah jumlah dokumen yang akan disimulasikan dan m adalah jumlah baris yang digunakan untuk mensimulasikan sebuah dokumen, maka jumlah baris teks yang diperlukan untuk simulasi basis data adalah mn. Dan jika r adalah jumlah baris representatif untuk signature yang mewakili sebuah dokumen, maka jumlah baris representatif yang dimasukkan ke dalam basis data adalah mr.



Gambar 5.4 Skema simulator untuk proses pembuatan basis data (signature posting) dan pembandingan signature (signature matching).

Dari masing-masing mr baris ini diambil signature berdasarkan shape code pada tabel 4.2, untuk mensimulasikan pengambilan signature yang sempurna.

Masing-masing signature dari *mr* baris tersebut kemudian disimpan ke dalam basis data untuk mensimulasikan proses signature posting. Hal ini diperlihatkan dalam gambar 5.4.

Untuk mensimulasikan proses signature matching, dilakukan proses pembandingan signature dan pendeteksian *duplikat* dan *non-duplikat* dengan mengambil *s* signature dari file teks yang sama atau file teks lain. Untuk mensimulasikan proses pembandingan signature duplikat, digunakan signature yang dihasilkan dari baris yang diambil dari file teks yang sama dengan yang digunakan untuk membuat basis data. Sedangkan untuk mensimulasikan proses pembandingan signature non-duplikat, digunakan signature yang dihasilkan dari baris yang diambil dari file teks lain.

Untuk menguji tingkat *robustness* dalam pengambilan signature, ke dalam simulator ditambahkan model noise dan degradasi. Model ini akan mengkorupsi sebuah signature kandidat dengan cara menambahkan secara random beberapa error tertentu ke dalam signature tersebut. Jenis error tersebut bisa berupa:

- penggantian (*replacement*), untuk mensimulasikan kesalahan pengenalan shape code,
- penyisipan (*insertion*), untuk mensimulasikan karakter yang memisah menjadi dua atau lebih komponen dan
- penghapusan (*deletion*), untuk mensimulasikan beberapa karakter yang menyambung menjadi satu komponen.

Untuk lebih menunjukkan kelayakan teknik shape coding, jumlah baris yang diambil dari file teks harus cukup besar, dari ratusan hingga ribuan dokumen. Semakin besar jumlah dokumen yang disimulasikan, maka hasil ujicobanya akan lebih

meyakinkan. Sama dengan yang digunakan oleh sistem sesungguhnya, simulasi ini juga menggunakan 10 shape code sebagai feature set dengan panjang key untuk analisa n-gram adalah 5.

5.2.2 Simulasi

Simulasi dibagi dalam dua bagian. Dalam bagian pertama disimulasikan sebuah basis data citra dokumen dengan properti sebagai berikut:

- memiliki 5000 citra.
- sebuah citra diwakili oleh 1 baris teks
- signature sebuah citra diwakili oleh 1 baris teks (yang mewakili citra itu sendir)
- variabel simulasi adalah jumlah error yang dimasukkan ke dalam signature

Sedang dalam bagian kedua, disimulasikan sebuah basis data citra dokumen dengan properti sebagai berikut:

- memiliki 1000 citra
- sebuah citra diwakili oleh 10 baris teks
- variabel simulasi adalah jumlah baris yang diperlukan untuk sebuah signature

Selain ketentuan minimal untuk jumlah karakter dalam suatu baris representatif, bisa juga ditambahkan ketentuan lain seperti keberadaan variasi ascender dan descender dalam sebuah baris, serta jumlah minimal ascender dan descender yang harus dimiliki sebuah baris. Dalam simulasi ini, pengambilan baris representatif hanya memperhitungkan jumlah karakter dalam baris saja.

5.2.2.1 Simulasi Bagian Pertama

Simulasi bagian pertama dilakukan dalam dua tahap. Pada tahap pertama, dilakukan query terhadap basis data dengan menggunakan signature duplikat, duplikat dengan error, non-duplikat dan non-duplikat dengan error. Dari proses hasil query masing-masing signature akan diperiksa distribusi nilai vote (jumlah hit) yang dihasilkan. Sedangkan pada tahap kedua dilakukan query terhadap sekian signature duplikat untuk memeriksa distribusi posisi 20 besar dalam peringkat nilai vote yang dihasilkan.

A. Tahap Pertama

Simulasi tahap pertama memiliki dua uji coba. Dalam uji coba pertama dilakukan query terhadap basis data menggunakan signature yang diketahui sebagai duplikat. Proses query ini kemudian diulangi beberapa kali, masing-masing dengan menambahkan beberapa error ke dalam signature.

Contoh hasil voting terhadap sebuah signature duplikat (baris ke-12) ditunjukkan dalam tabel 5.1 dengan penambahan 0, 5, 10, 15 dan 20 error dalam signature. Tabel ini menunjukkan bahwa sekalipun jumlah hit atau persentase kesamaan sangat berkurang karena penambahan 5 error saja, tetap saja dokumen duplikat (baris ke-12) memiliki peringkat lebih tinggi dibanding dokumen lainnya.

Uji coba kedua dilakukan untuk menguji tingkat *robustness* dari feature set yang digunakan. Untuk itu semua proses query dalam uji coba pertama diulangi sekali lagi, namun kali ini menggunakan signature yang diketahui sebagai non-duplikat. Hasil voting terhadap signature tersebut ditunjukkan pada tabel 5.2. Dalam contoh ini digunakan baris ke-101 dari file teks lain. Dari tabel 5.2 terlihat dari kombinasi

jumlah hit yang rendah serta kesamaan dokumen id antar kelompok peringkat atas, yang menandakan bahwa signature kandidat tersebut bukan merupakan suatu duplikat.

Tabel 5.1 Peringkat query hit berdasarkan signature duplikat dari baris 12

Error	Document id / (jumlah hit)								
	1	2	3	4	5	6	7	8	9
0	12 (46)	1954 (8)	8795 (8)	3137 (7)	11487 (7)	8195 (7)	1965 (6)	10388 (6)	91 (6)
5	12 (32)	1954 (8)	11487 (7)	1965 (6)	10388 (6)	1690 (6)	7854 (6)	1635 (5)	5204 (5)
10	12 (28)	1965 (6)	11487 (6)	10388 (6)	5204 (5)	1954 (5)	1919 (5)	771 (5)	7829 (5)
15	12 (17)	7708 (4)	6998 (4)	966 (3)	1023 (3)	653 (3)	6856 (3)	5222 (3)	12050 (3)
20	12 (11)	5031 (6)	11551 (6)	2802 (5)	7897 (5)	5204 (5)	3868 (5)	2994 (5)	771 (5)

B. Tahap Kedua

Dalam tahap kedua, dilakukan query menggunakan *n* signature duplikat yang dipilih secara random. Setiap proses query untuk masing-masing dari *n* signature tersebut diulangi beberapa kali dengan menambahkan 5, 10, 15 dan 20 error ke dalam signature. Tabel 5.3 memperlihatkan hasil uji coba yang menampakkan distribusi ranking pertama, kedua, kelima, kesepuluh dan kedua puluh pada signature yang benar (tanpa penambahan error), dibandingkan dengan signature yang dipengaruhi oleh error. Dalam contoh ini digunakan 100 signature duplikat yang diambil secara random. Dari tabel 5.3 bisa dilihat bahwa hasil yang benar (duplikat yang

sesungguhnya) tetap dalam ranking teratas untuk penambahan 10 error atau kurang. Untuk penambahan diatas 10 sampai 20 error, maka hasil yang benar akan tetap berada dalam 20 besar. Namun akan sulit untuk menentukan duplikat yang sesungguhnya, karena kurang signifikannya jumlah hit dari kelompok peringkat atas.

Tabel 5.2 Peringkat query hit berdasarkan signature non-duplikat dari baris 101.

Error	Document id / (jumlah hit)								
	1	2	3	4	5	6	7	8	9
0	6590 (14)	7911 (13)	1256 (12)	2756 (11)	8422 (10)	6990 (10)	6874 (10)	5910 (10)	7381 (10)
5	6590 (10)	7911 (9)	6874 (9)	2756 (9)	839 (9)	4455 (8)	526 (8)	11247 (8)	8930 (8)
10	6590 (11)	11009 (10)	8030 (9)	7738 (9)	11256 (9)	11847 (8)	7736 (8)	2899 (8)	2434 (8)
15	10115 (8)	11542 (7)	1241 (7)	8946 (7)	8698 (7)	2637 (8)	11386 (6)	10921 (6)	10830 (6)
20	11674 (6)	7675 (6)	6590 (6)	12096 (4)	8254 (4)	8228 (4)	11675 (4)	11009 (4)	4455 (4)

Tabel 5.3 Distribusi posisi peringkat untuk duplikat sesungguhnya dari satu sampai 20 besar.

Error	Ranking 1	2 Besar	5 Besar	10 Besar	20 Besar
0	100	100	100	100	100
5	100	100	100	100	100
10	100	100	100	100	100
15	51	58	69	77	100
20	17	21	24	30	100

5.2.2.2 Simulasi Bagian Kedua

Simulasi bagian kedua melakukan hal yang sama dengan simulasi bagian pertama tahap pertama, yaitu melakukan penghitungan query hit untuk signature dari citra duplikat dan non-duplikat. Perbedaannya terletak pada tujuannya yang menguji pengaruh panjang signature terhadap peringkat persentase kesamaan yang dihasilkan dari proses signature matching. Dengan demikian yang menjadi variabel simulasi kali ini adalah jumlah baris teks yang mewakili sebuah signature citra dokumen.

Tabel 5.4 Peringkat query hit terhadap signature duplikat dari dokumen id 315 dengan variasi jumlah baris per signature.

Baris	Document ID / (persentase kesamaan)								
	1	2	3	4	5	6	7	8	9
1	315 (100)	26 (24)	358 (24)	553 (22)	143 (20)	185 (20)	994 (20)	408 (17)	648 (17)
2	315 (100)	55 (24)	753 (18)	897 (16)	486 (18)	595 (17)	143 (17)	79 (17)	304 (17)
3	315 (100)	55 (25)	624 (23)	993 (22)	1154 (21)	250 (19)	85 (19)	994 (19)	753 (18)
4	315 (100)	993 (21)	551 (19)	1480 (19)	897 (19)	677 (19)	1154 (18)	874 (18)	410 (18)
5	315 (100)	1154 (22)	2115 (22)	1287 (22)	993 (22)	874 (21)	2145 (21)	2106 (21)	281 (20)

Contoh hasil voting terhadap sebuah signature dari citra duplikat (citra 315) dengan variasi 1, 2, 3, 4 dan 5 baris per signature ditunjukkan dalam tabel 5.4. Dari tabel tersebut dapat dilihat bahwa presentase kesamaan untuk semua variabel simulasi adalah hampir sama. Dari sini bisa disimpulkan bahwa panjang signature tidak terlalu

berpengaruh pada peringkat persentase kesamaan, walaupun tetap berpengaruh pada daftar peringkat dokumen dalam basis data yang memiliki kesamaan dengan dokumen kandidat.

Untuk hasil voting terhadap sebuah signature dari citra non-duplikat (citra 15) dengan variasi 1, 2, 3, 4 dan 5 baris per signature ditunjukkan dalam tabel 5.5. Dari tabel tersebut dapat dilihat bahwa presentase kesamaan untuk semua variabel simulasi adalah hampir sama. Dari sini bisa disimpulkan bahwa panjang signature tidak terlalu berpengaruh pada peringkat persentase kesamaan, walaupun ada sedikit peningkatan persentase kesamaan pada saat signature bertambah panjang. Selain itu perbedaan besar nampak pada daftar peringkat dokumen dalam basis data yang memiliki kesamaan dengan dokumen kandidat.

Tabel 5.5 Peringkat query hit terhadap signature non-duplikat dari dokumen id 15 dengan variasi jumlah baris per signature.

Baris	Document ID / (persentase kesamaan)								
	1	2	3	4	5	6	7	8	9
1	749 (17)	261 (17)	653 (15)	648 (15)	486 (15)	314 (15)	101 (13)	644 (13)	638 (13)
2	245 (17)	476 (15)	229 (15)	314 (15)	521 (14)	387 (14)	297 (14)	698 (14)	1083 (14)
3	530 (18)	314 (18)	234 (16)	1102 (16)	464 (16)	257 (16)	1077 (16)	1037 (16)	989 (16)
4	1499 (21)	989 (19)	31 (19)	314 (18)	1322 (18)	446 (18)	243 (18)	23 (18)	464 (18)
5	1077 (23)	1449 (23)	2314 (22)	2382 (21)	1906 (21)	2170 (20)	2445 (20)	2053 (20)	1740 (20)

5.2.2.3 Kesimpulan Hasil Simulasi

Dalam kenyataan, dua citra yang berasal dari dokumen yang sama dapat memiliki beberapa perbedaan, biasanya disebabkan oleh beberapa faktor seperti penambahan catatan, fotokopi, proses penuaan kertas serta perbedaan karakteristik dalam proses scanning meliputi resolusi dan kemiringan. Adalah wajar jika terdapat perbedaan pada signature-signature yang diambil dari citra-citra yang berasal dari dokumen yang sama. Karena itu penggunaan error sampai mencapai 20 buah dalam uji coba di atas dianggap telah mencukupi.

BAB VI

KESIMPULAN DAN SARAN

Bab ini merupakan akhir dari buku ini yang berisikan kesimpulan dan saran pengembangan berikut untuk Tugas Akhir ini. Di antaranya meliputi kesimpulan untuk penggunaan analisa projection profile dan analisa connected component dalam segmentasi citra dokumen, serta kesimpulan untuk penggunaan teknik shape coding dalam pengambilan signature.

6.1 Kesimpulan

Secara keseluruhan, penggunaan teknik shape coding dapat menghasilkan signature yang cukup unik untuk tiap citra dokumen dalam sebuah basis data berukuran besar. Dalam kasus signature yang ideal maupun yang memiliki error, proses deteksi duplikasi dengan teknik shape coding berhasil menentukan apakah citra kandidat merupakan duplikat atau tidak.

6.1.1 Segmentasi Citra Dokumen

Penggunaan analisa projection profile dalam segmentasi citra dokumen mempunyai keuntungan yang jelas, yaitu cepat dan sederhana. Hal ini sangat sesuai dengan keseluruhan proses deteksi duplikasi, yang membutuhkan algoritma yang cepat dalam melakukan segmentasi citra dokumen dan mengambil signature. Dan dengan penggunaan threshold yang tepat serta metode rekursif, analisa projection

profile dapat melakukan segmentasi citra dokumen pada struktur layout dokumen yang rumit sekalipun dengan hasil yang bisa diterima (*acceptable*).

Sayangnya analisa projection profile tidak bisa digunakan untuk menangani citra dokumen yang memiliki kemiringan atau bingkai. Padahal dalam basis data citra dokumen, banyak sekali citra yang yang memiliki karakteristik seperti itu. Analisa projection profile juga lemah dalam segmentasi karakter. Karena itu penulis menerapkan analisa connected component untuk melakukan proses segmentasi karakter.

Kendala lain dalam penggunaan analisa projection profile adalah penggunaan threshold dalam menentukan posisi pemilahan (*cut*) pada saat proses rekursif. Threshold yang dianggap memberikan hasil yang dapat diterima, biasanya diperoleh melalui *trial and error*. Dan threshold yang tepat untuk citra dokumen dengan jenis typeface dan resolusi tertentu dapat saja memberikan hasil yang tidak diharapkan pada citra dokumen dengan jenis typeface dan resolusi berbeda.

6.1.2 Pengambilan Signature

Proses pengambilan signature dengan menggunakan teknik shape coding cukup akurat untuk citra dokumen yang bersih tanpa kemiringan. Masalahnya adalah apabila cukup banyak karakter atau baris yang saling bersentuhan, sehingga menyebabkan terlewatinya beberapa baris yang mungkin saja merupakan baris representatif. Hal ini dapat terjadi dalam citra yang berasal dari dokumen yang telah difotokopi berulang kali. Banyaknya karakter yang saling bersentuhan juga dapat menyebabkan baris teks yang bersangkutan memiliki jumlah simbol yang kurang dari batas panjang untuk suatu signature. Tapi seperti yang telah ditunjukkan dalam

percobaan pada tabel 5.1, dan 5.2, sekalipun terdapat 10 sampai 15 error dalam signature yang diambil, program masih mampu mendeteksi keberadaan duplikat. Ini tentunya hanya berlaku jika program berhasil menemukan baris representatif yang benar.

Untuk citra yang berasal dari dokumen-dokumen yang terdegradasi, hal yang paling kritis adalah menentukan letak baris representatif yang sama untuk citra-citra yang berasal dari dokumen yang sama. Jika proses penentuan posisi baris representatif ini gagal, maka tidak akan bisa dihasilkan signature yang benar. Pada kasus citra dengan degradasi seperti inilah nampak keuntungan dalam penggunaan teknik shape coding jika dibandingkan dengan metode lain seperti OCR, di mana program tidak memerlukan informasi pada isi (*content level information*) apapun, baik secara apriori atau sebagai bagian dalam proses analisa.

6.2 Saran Pengembangan

Beberapa pengembangan yang bisa dilakukan untuk Tugas Akhir ini, yaitu:

- penambahan modul preprocessing untuk menghilangkan kemiringan citra
- perbaikan metode segmentasi citra dokumen agar bisa menerima bingkai dan mengambil teks yang berada dalam tabel
- penambahan kemampuan untuk menangani karakter yang memisah atau menyambung dalam segmentasi karakter
- penggunaan informasi dasar lain sebagai filter sekunder untuk mengurangi jumlah citra duplikat yang dihasilkan, contohnya seperti jumlah halaman (citra multipage), jumlah baris dalam dokumen, kepadatan dokumen dan lain-lain.

DAFTAR PUSTAKA

[1] David Doermann, Huiping Li, Omid Kia, Kemal Kilic, *The Detection of Duplicates in Document Image Databases*, Language and Media Processing Laboratory, Center for Automation Research, University of Maryland, 1997.

[2] Larry Davis, *Machine Vision Systems*, 1998.

[3] Henry Budgett, *Typesetting and Publishing Glossary*, 1997.

[4] Sargur Srihari, Stephen Lam, Venu Govindaraju, *Document Understanding: Research Directions*, Centre of Excellence for Document Analysis and Recognition, State University of New York, 1992.

[5] William B. Cavnar, *Using An N-Gram-Based Document Representation With a Vector Processing Retrieval Model*, 1994.

[6] N. Shivakumar, H. Garcia-Molina, *SCAM: A Copy Detection Mechanism for Digital Document*, Department of Computer Science Stanford University, 1995.

[7] H. Garcia-Molina, S. Brin, J. Davis, *Copy Detection Mechanisms for Digital Document*, 1995.

[8] Arthur Lee Wilson, *The Digital Library*, 1994.

[9] Jaekyu Ha, Ihsin T. Phillips, Robert M. Haralick, *Document Page Decomposition using Bounding Boxes of Connected Components of Black Pixels*, Department of Electrical Engineering University of Washington and Department of Computer Science Seattle University.