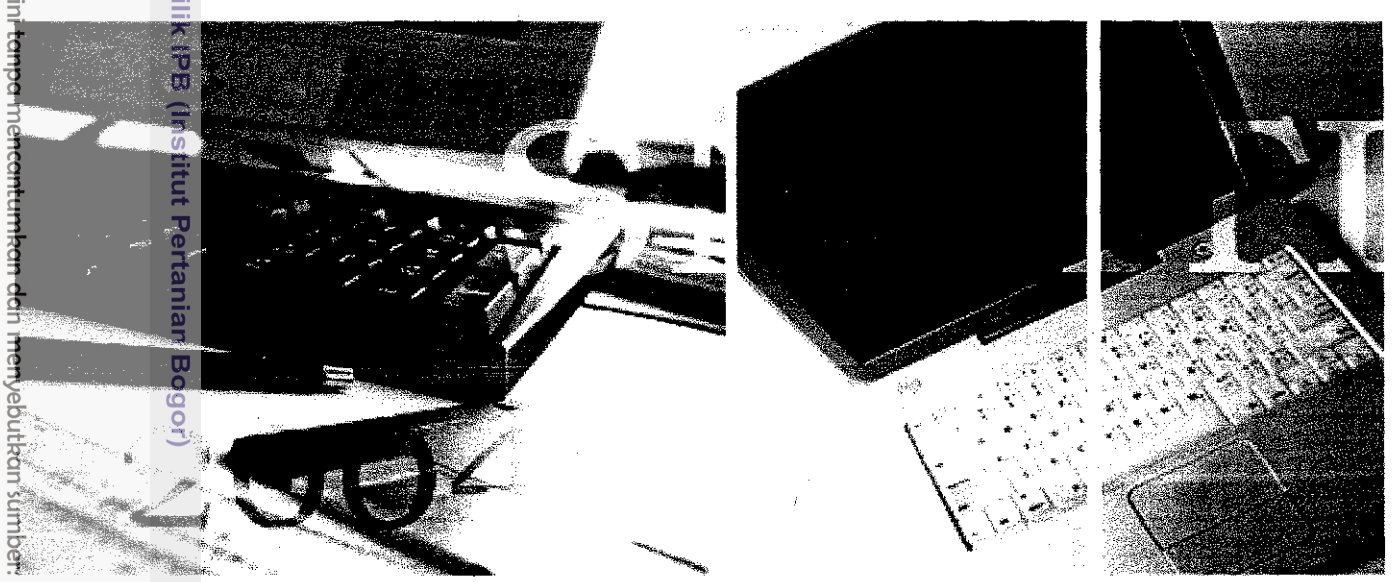
ISSN : 1411-107

P R O S I D I N G

SNIKTI V

Seminar Nasional Ilmu Komputer dan Teknologi Informasi V



Vol. V No. 1 September 2004

Aula Matematika - FMIPA IPB, 2-3 September 2004

DEPARTEMEN ILMU KOMPUTER - FMIPA
INSTITUT PERTANIAN BOGOR

ICIS
INDONESIAN SOCIETY ON COMPUTER AND INFORMATION SCIENCE
MASYARAKAT ILMU KOMPUTER DAN INFORMASI INDONESIA

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)

Bogor Agricultural University

DAFTAR ISI

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)

Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)

KATA PENGANTAR

Dengan mengucap syukur ke hadirat Allah SWT, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam IPB mengadakan Seminar Nasional Ilmu Komputer dan Teknologi Informasi 2004 (SNIKTI-2004) pada tanggal 2 – 3 September 2004 di Kampus IPB Baranangsiang Bogor. Seminar ini merupakan kegiatan rutin dari Indonesian Society on Computer and Information Sciences (ICIS). Pada kesempatan ini bekerja sama dengan Departemen Ilmu Komputer IPB.

Pada seminar ini Panitia menerima 87 makalah penelitian dari beberapa institusi pendidikan tinggi seperti IPB, ITB, ITS, UI, Univ. Petra Surabaya, STT. Telkom Bandung, Univ. Pancasila , dan lain-lain. Setelah dievaluasi, 58 makalah diterima untuk dipresentasikan dan dibukukan dalam bentuk prosiding dengan ISSN : 1411 – 1071.

Panitia menyampaikan penghargaan setinggi-tingginya kepada keynote speaker, para pemakalah dan peserta yang telah hadir dalam acara seminar ini, sehingga dapat memberikan kontribusi yang tidak ternilai. Selanjutnya kepada pihak-pihak yang membantu sehingga acara ini dapat berlangsung dengan lancar kami ucapkan banyak terima kasih. Kemudian bila di dalam proses awal sampai terselenggaranya acara ini terdapat kesalahan atau hal-hal yang kurang berkenan, maka atas nama panitia kami mohon maaf sebesar-besarnya.

Bogor, September 2004

Ir. Agus Buono, M.Si., M.Komp.
Penanggung Jawab SNIKTI V 2004

Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak Cipta Milik IPB (Institut Pertanian Bogor)

SAMBUTAN
DEKAN FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR

Assalamu 'alaikum Wr. Wb.

Pertama-tama marilah kita panjatkan puji syukur ke hadirat Allah SWT – Tuhan Yang Mahakuasa, dimana hanya karena rahmat dan hidayahNya maka Seminar Nasional Ilmu Komputer dan Teknologi Informasi 2004 (SNIKTI-2004) ini dapat terselenggara dengan baik. Saya ucapkan selamat datang kepada semua pemakalah dan peserta SNIKTI-2004 di Institut Pertanian Bogor.

SNIKTI-2004 terselenggara atas kerja sama antara Departemen Ilmu Komputer IPB dengan Masyarakat Ilmu Komputer dan Informasi Indonesia (atau yang lebih deikenal dengen sebutan ICIS – *Indonesian Society on Computer and Information Sciences*) yang berpusat di Fakultas Ilmu Komputer UI. SNIKTI ini merupakan kegiatan seminar nasional ke-5 . Topik yang dipresentasikan mencakup materi yang terkait dengan bidang ilmu komputer dan teknologi informasi beserta aplikasinya. Dengan cakupan materi ini diharapkan partisipan kegiatan SNIKTI berasal dari kalangan peneliti, akademisi dan praktisi.

Forum ini diharapkan dapat dimanfaatkan untuk diseminasi hasil penelitian teknologi informasi. Selain itu seminar ini dapat menjadi wahana untuk berdiskusi dan berkomunikasi serta untuk meningkatkan kerjasama antar peneliti dan praktisi.

Akhirnya, perkenankanlah saya sampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah mendukung suksesnya penyelenggaraan SNIKTI-2004, terutama sekali kepada komite pelaksana dan program, para sponsor, dan pihak-pihak lain yang secara langsung maupun tidak langsung telah ikut memberikan andil bagi terselenggaranya SNIKTI 2004. Semoga bantuan dan amal baiknya akan mendapat ganti yang jauh lebih baik dari Allah SWT. Semoga kegiatan seminar nasional ini akan memberikan manfaat yang sebesar-besarnya bagi semua pihak yang memerlukan.

Wassalamu 'alaikum Wr. Wb.
Dr. Ir. Yonny Koesmaryono, MS.

Bogor Agricultural University

SUSUNAN PANITIA

Penanggung Jawab
- Ir. Agus Buono M.St. Mkomp (IPB)
- Dr. Benyamin Kusumoputro (UI)

Komite Program
- Prof. Marimin (IPB)
- Dr. Kudang Boro Seminar (IPB)
- Dr. Sugi Guritman (IPB)
- Dr. Zainal A. Hasibuan (UI)
- Dr. Aniati Murni (UI)
- Dr. Bobby AA. Nazief (UI)
- Dr. T. Basaruddin (UI)
- Dr. Iping Supriana (ITB)
- Dr. Jazi Eko Istiyanto (UGM)

Ketua Pelaksana
Yeni Herdiyeni, S.Komp

Komite Pelaksana
- Ir. Meuthia Rachmaniah, MSc
- Ir. Julio Adisantoso, Mkom
- Yani Nurhadryani, S.Si, MT
- Rindang Karyadin, S.Si, MKom
- Shelvie Nidya Neyman, S.Komp
- Agus Pudjijono, S.Komp
- Wisnu Ananta, S.Si, MT
- Imas S. Sitanggang, S.Si, Mkomp
- Annisa, S.Komp
- Panji Wasmana, S.Komp
- Firman Ardiansyah, S.Komp
- Ahmad Ridha, S.Komp
- Hari Agung, S.Komp
- Drs. Prabowo
- Gage, A.Md

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

SUSUNAN ACARA

Kamis, 2 September 2004

Waktu	Kegiatan
08.30 – 09.00	Registrasi Peserta
09.00 – 10.00	Pembukaan - Sambutan Penanggung Jawab SNIKTI V - Sambutan Dekan FMIPA IPB
10.00 – 10.15	Rehat
10.30 – 11.30	Keynote Speaker (Dr. Benyamin Kusumuputro)
11.30 – 13.00	Istirahat
13.00 – 15.00	Penyajian Makalah
15.00 – 15.30	Rehat
15.30 – 16.50	Penyajian Makalah

Jum'at, 3 September 2004

Waktu	Kegiatan
09.00 – 10.00	Keynote Speaker (Prof. Marimin)
10.00 – 10.30	Rehat
10.30 – 11.30	Keynote Speaker (Dr. Ir. Ricardus Eko Indrajit, MSc, MBA)
11.30 – 13.30	Istirahat
13.30 – 15.00	Penyajian Makalah
13.00 – 15.00	Penyajian Makalah
15.00 – 15.30	Rehat
15.30 – 16.35	Penyajian Makalah
17.30 – 17.00	Penutupan

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB, Institut Pertanian Bogor
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

JADWAL PENYAJIAN MAKALAH

KAMIS, 2 SEPTEMBER 2004

WAKTU	NO PAPER	PEMAKALAH	JUDUL	HAL PAPER
SESI 1				
13.00 - 13.20	35	Agus Pudjijono, Julio Adisantoso , Yani Nurhadryani , Ahmad Ridha	ALGORITME SEGMENTED DYNAMIC TIME WARPING PADA PENCARIAN AUDIO	203
13.20 - 13.40	54	Ahmad Ridha, Julio Adisantoso, Fahren	PENGINDEKSAN OTOMATIS DENGAN ISTILAH TUNGGAL UNTUK DOKUMEN BERBAHASA INDONESIA	328
13.40 - 14.00	24	Bayu Wicaksana Wahyuardi , Julio Adisantoso dan Fahren Bukhari	TEMU KEMBALI INFORMASI MUSIKAL PADA BASIS DATA AUDIO MENGGUNAKAN ALGORITMA KESAMAAN STRING BAEZA YATES – PERLEBERG	134
14.00 - 14.20	25	Gatot Wahyudi dan Indra Budi	INDONESIAN NAMED ENTITY RECOGNIZER (InNER)	141
14.20 - 14.40	10	Muhammad Erwin Ashari Hariyono	QUERY EXPANSION MENGGUNAKAN FUZZY JACCARD COEFICIENT SEBAGAI FUNGSI FITNESS PADA PROSES RELEVANCE FEEDBACK	61
14.40 - 15.00	37	Zainal Arifin Hasibuan,Phd, Erizal	PENDEKATAN MODEL PERLUASAN BOOLEAN PADA SISTEM TEMU KEMBALI INFORMASI DENGAN MENGGUNAKAN DOKUMEN XML	220
BREAK				
15.30 - 15.50	52	Beni Irawan dan Zainal Hasibuan	PENGARUH PEMROGRAMAN GENETIKA TERHADAP EFEKTIVITAS TEMU KEMBALI INFORMASI	312

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

16.30 - 16.50	49	Teguh Pribadi Arsyad, B. Kusumoputro	PENGEMBANGAN METODE PENENTUAN BOBOT AWAL JARINGAN NEURAL FUZZY-LVQ DALAM RUANG EIGEN UNTUK PENINGKATAN DERAJAT PENGENALAN AROMA 3 CAMPURAN	293
SESI 3				
13.00 - 13.20	48	Harry B. Santoso dan Indra Budi	COMPUTER-MEDIATED LEARNING DENGAN PENDEKATAN COLLABORATIVE LEARNING DAN PROBLEM BASED LEARNING: STUDI KASUS CML UNIVERSITAS INDONESIA	288
13.20 - 13.40	26	Teguh Prihatmono	DISTRIBUSI QUERY PADA DBMS MYSQL MEMANFAATKAN SISTEM REPLIKASI	149
13.40 - 14.00	39	Budi Sutedjo Dharma Octomo & . R. Gunawan Santosa	KESIAPAN PERGURUAN TINGGI DI DAERAH ISTIMEWA YOGYAKARTA UNTUK MENYELENGGARAKAN LAYANAN E-EDUCATION	233
14.00 - 14.20	30	Taufik Djatna dan Albertus Reinandang	SISTEM INFORMASI BERBASIS SMS/GSM UNTUK PENJADWALAN, PENENTUAN JALUR DAN PENGELOLAAN SAMPAH PADAT PERKOTAAN	172
14.20 - 14.40	47	Resmana Lim, Arthur Franklin, Agustinus N	SISTEM AKSES FASILITAS PENDAFTARAN RENCANA STUDI DAN JADWAL UJIAN MAHASISWA VIA SMS	279
14.40 – 15.00	58	AbdullahEmbong, Zulikha Jamaluddin	MANAGEMENT OF INFORMATION THROUGH AN INTERACTIVE VOICE RESPONSE SYSTEM THE PROSPECT OF RELIGIOUS DOMAIN	351
BREAK				
15.30 - 15.50	16	Wikan Danar Sunindyo, Rakhmat Aji Jauhari, Isman Hidayat Suryaman	VIRTUAL OFFICE SEBAGAI PENERAPAN COMPUTER-SUPPORTED COOPERATIVE WORK (CSCW)	94
15.50 - 16.10	19	Dr. Muhammad Zarlis, M. Kom dan Rahmat Widia Sembiring, SE. M.Sc.IT	MEMBANGUN TRUST SEBAGAI KUNCI KEBERHASILAN E-COMMERCE	110
16.10 - 16.30	55	Hira Laksmiwati, Ir. MSc.	EKSPLORASI METODOLOGI PEMBANGUNAN DATA MART	336

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

SESI 4				
13.00 - 13.20	23	Jazi Eko Istiyanto dan Nurhayati Masthurah	KONFIGURASI SECURITY POLICY SELINUX	130
13.20 - 13.40	9	Denny Restria Widaryanto, Fazmah Arif Yulianto, Sri Widowati	SISTEM PENDETEKSI PERANGKAT KERAS KOMPUTER MELALUI JARINGAN	55
13.40 - 14.00	53	Novala Panala, TutunJuhana, Hendrawan	EVALUASI PERFORMANCE SCTP SEBAGAI PROTOKOL TRANSPORT SS7OVER IP	318
14.00 - 14.20	14	Hany Ferdinando	MEMBANGUN SISTEM FAULT TOLERANT PADA FIELDBUS UNTUK APLIKASI SISTEM TERDISTRIBUSI	84
14.20 - 14.40	45	Fatilah, Iwan Syarif	APLIKASI ALGORITMA NAIVE BAYES UNTUK PENDETEKSIAN INTRUSI PADA SISTEM JARINGAN KOMPUTER	267
BREAK				
15.30 - 15.50	50	Wisnu M Suryaningrat, Rizal F Aji	KEBUTUHAN INFRASTRUKTUR PERPUSTAKAAN DIJITAL	301
15.50 - 16.10	42	Dwi Handoko	IMPLEMENTASI BLOCK MATCHING MOTION VECTOR ESTIMATION DALAM KOMPUTER KLASTER	252
16.10 - 16.30	34	Heru Suhartanto dan Jimmy	PENERJEMAH KODE NUMERIK FORTRAN KE JAVA	199

JUMAT, 3 SEPTEMBER 2004

WAKTU	NO PAPER	PEMAKALAH	JUDUL	HAL PAPER
SESI 1				
13.30 - 13.50	3	Muhammad Arief	CONFLICT RESOLUTION DALAM FORMALISASI BUREAUCRATIC RULE	13
13.50 - 14.10	13	Arna Fariza, Achmad Basuki	PENGEMBANGAN HYBRID ALGORITMA GENETIKA SIMULATED ANNEALING PADA PERAMALAN TIME SERIES DENGAN KLASIFIKASI DATA BERDASARKAN TREND	79

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

14.10 - 14.30	46	Suyanto, Andhika Dwi Putera, Fazmah Arief Yulianto	KONVERSI CITRA RASTER KE CITRA VEKTOR MENGGUNAKAN TRANSFORMASI WAVELET	273
14.30 - 14.50	4	Andrias Hardinata, Suyanto, Fazmah Arief Yulianto	STEGANOGRAPHY PADA CITRA BINER MENGGUNAKAN DETEKSI SISI MINIMUM DISTANCE	19
BREAK				
15.30 - 15.50	56	Sani Muhamad Isa	SEBUAH TINJAUAN MENGENAI METODE ANALISIS CITRA HIPEKSPEKTRAL DAN APLIKASINYA	340
15.50 - 16.15	7	Veronica S. Moertini,	PENANGANAN ATRIBUT CITRA DENGAN WAVELET UNTUK PENGEMBANGAN ALGORITMA C4.5	43
SESI 4				
13.30 - 13.50	28	Rahmat hidayati, Achmad Basuki, Nana Ramadijanti, Arna Fariza	IMPLEMENTASI METODA GIS DALAM REVISI RENCANA KAWASAN PERMUKIMAN MENGGUNAKAN GIS - GRASS (STUDI KASUS : KABUPATEN MOJOKERTO 2013)	160
14.10 - 14.30	15	Eko Sediyono dan Daniel Yohanes	PENYEMBUNYIAN PESAN DALAM DOKUMEN CITRA DUA WARNA DENGAN METODE STEGANOGRAFI	89
14.30 - 14.50	41	G.A.Putri Saptawati , dan H.B. Rachmat, D.Laksana	KONSTRUKSI STRUKTUR MODEL BAYESIAN NETWORK DARI DATA DENGAN ALGORITMA K2 DAN B	244
BREAK				
15.30 - 15.50	17	Muliyadi, Indrawati, Arna Fariza, Ali Ridho Barakbah	RANCANG BANGUN PERANGKAT LUNAK PEMODELAN ANALYTIC HIERARKI MODEL	98
15.50 - 16.15	11	*Susany Soplanit*	METODE PEMBANGKITAN CHAOS UNTUK ENKRIPSI CITRA DIGITAL	67

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

DAFTAR MAKALAH

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber.
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
©Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

TEMU KEMBALI INFORMASI MUSIKAL PADA BASIS DATA AUDIO MENGGUNAKAN ALGORITMA KESAMAAN STRING BAEZA YATES – PERLEBERG

Julio Adisantoso *, Fahren Bukhari †, dan Bayu Wicaksana Wahyuardi *
* Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor
Jl. Raya Pajajaran, Bogor, Indonesia
email : julio@fmipa.ipb.ac.id
bayu@ilkomerz35.com
† Departemen Matematika, FMIPA, Institut Pertanian Bogor
Jl. Raya Pajajaran, Bogor, Indonesia

ABSTRAK

Adanya basis data dengan tipe data audio membuat orang membutuhkan sebuah metode baru untuk menemukembalikan informasi tentang keberadaan sebuah lagu pada basis data, penelitian ini bertujuan untuk mempelajari dan menerapkan algoritma pencocokan string Baeza-Yates dan Perleberg pada sebuah sistem temu kembali. Percobaan dilakukan menggunakan basis data yang memiliki jumlah koleksi lagu berbeda (30, 40 dan 50) yang terdiri dari berbagai jenis aliran musik. Seluruh koleksi memiliki frekuensi 8 KHz. Percobaan yang dilakukan ada 5, yaitu pengukuran waktu, perpedaan panjang *input*, perbedaan posisi potongan lagu terhadap lagu asal, perubahan amplitudo *input*, dan perbedaan frekuensi *input*. Secara umum tujuan percobaan adalah untuk mengetahui waktu pencarian dan pengaruh berbagai macam perlakuan pada *input* terhadap hasil. Untuk pengukuran waktu, dapat disimpulkan bahwa makin banyak jumlah koleksi maka makin lama pula waktu yang dibutuhkan untuk melakukan pencarian. Untuk perbedaan panjang *input*, dapat disimpulkan makin panjang durasi *input* maka makin lama waktu pencariannya. Sedangkan untuk perbedaan panjang *input*, *input* dengan panjang 30 detik memiliki persentase terambil pada urutan pertama sebesar 70%, persentase kecocokan tertinggi sebesar 96.68% dan persentase terambil sebesar 90%. *Input* berposisi diakhir lagu memiliki persentase terambil sebesar 91,67%, persentase terambil pada urutan pertama sebesar 83,33% dan persentase kecocokan tertinggi sebesar 96,154%. Perbedaan amplitudo *input* tidak memberikan pengaruh yang ekstrim pada hasil yang diperoleh, karena perubahan amplitudo tidak mengubah bentuk suara. Perbedaan frekuensi *input*, mengakibatkan tidak ada satu lagu pun yang terambil, karena perubahan frekuensi mengubah bentuk suara.

Kata kunci : Baeza Yates-Perleberg, waktu pencarian, durasi, posisi input, amplitude, frekuensi.

1. PENDAHULUAN

Banyak orang yang mengidentifikasikan dirinya dengan musik bukan dengan gambar. Hal ini dapat dilihat dari banyaknya orang yang mengatakan "ini laguku!" dan bukan "ini gambarku!" (Francu & Nevill-Manning, 2000). Oleh karena itu musik tidak dapat dipisahkan dari kehidupan seseorang.

Sesuai dengan perkembangan teknologi dewasa ini, sebuah lagu tidak hanya berbentuk kaset atau piringan hitam saja, tetapi sudah dapat dijumpai dalam bentuk berkas komputer. Lagu dalam bentuk berkas komputer ada yang berdiri sendiri dan ada pula yang dikumpulkan dalam sebuah basis data. Dengan adanya basis data audio, maka dibutuhkan sebuah sistem temu kembali informasi yang dapat digunakan pada basis data ini. Menurut Ghias *et al.*(1995) cara yang paling efektif dan lazim untuk mencari keberadaan sebuah lagu pada basis data audio adalah dengan menyenandungkan nada-nada sebuah lagu, tetapi pada penelitian ini *input* yang digunakan adalah potongan lagu.

Untuk proses pencarian pada basis data, penelitian ini menggunakan algotitma kesamaan string yang dikembangkan oleh Baeza-Yates dan Perleberg (1992) sebagaimana yang pernah dilakukan oleh Ghias *et al.* (1995). Meskipun demikian terdapat beberapa perbedaan antara penelitian yang pernah dilakukan oleh Ghias *et al.* (1995) dengan penelitian ini, antara lain pada penelitian sebelumnya *input* yang digunakan adalah senandung nada sebuah lagu sedangkan pada penelitian ini *input* yang digunakan adalah potongan lagu. Untuk koleksi lagu pada

penelitian sebelumnya lagu koleksi dikonversi dari MIDI sedangkan pada penelitian ini lagu koleksi dikonversi dari MP3. Perbedaan juga terdapat pada percobaan yang dilakukan, pada penelitian sebelumnya percobaan hanya dilakukan terhadap 1 macam *input* sedang pada penelitian ini *input* memperoleh berbagai macam perlakuan. Sehingga penelitian ini tidak bertujuan untuk melakukan perbaikan pada penelitian yang telah dilakukan sebelumnya, akan tetapi mencoba menerapkan hal yang sama pada kondisi yang berbeda.

2. SISTEM TEMU KEMBALI INFORMASI MUSIKAL

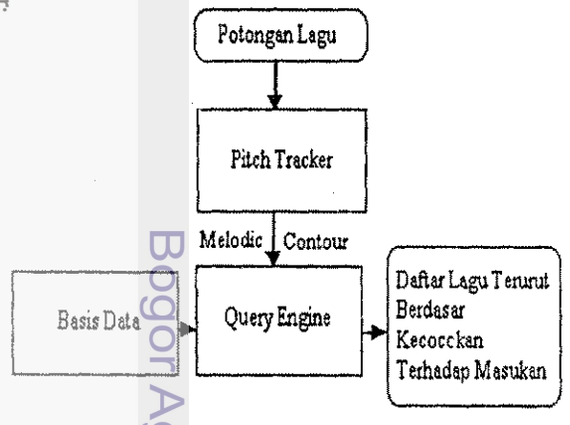
2.1. Arsitektur Sistem

Ada tiga komponen utama dalam sistem temu kembali informasi musikal pada basis data audio yang dikembangkan oleh Ghias *et al.*(1995) (Gambar 1), yaitu:

1. *Pitch Tracker*
2. Basis Data
3. *Query Engine*

Sedangkan proses yang terjadi dalam sistem ini adalah sebagai berikut :

1. *Input* yang berupa potongan lagu berformat WAV dimasukkan ke dalam *pitch tracker* untuk diproses.
2. Hasil pemrosesan di *pitch tracker* yang berupa *melodic contour* dimasukkan ke dalam *query engine*.
3. *Query engine* menghasilkan daftar lagu yang diurutkan berdasarkan kecocokannya terhadap *input* yang diberikan.



Gambar 1. Arsitektur Sistem.

2.2. Bahan Percobaan

Bahan yang digunakan untuk penelitian ini terdiri dari 50 buah berkas lagu berformat WAV yang merupakan hasil perekaman dari berkas berformat mp3. Ke-50 buah berkas lagu berformat WAV tersebut direkam dengan frequensi 8 KHz. Berkas-berkas tersebut terdiri dari berbagai macam aliran musik mulai dari musik klasik sampai musik rock bahkan terdapat pula lagu tradisional

2.3. Pembuatan Program

Untuk membangun program digunakan perangkat lunak MATLAB versi 6.1 dan bahasa pemrograman Microsoft Visual C++. Selain kedua perangkat lunak tersebut, perangkat lunak lain yang digunakan adalah Creative Sound Recorder yang digunakan untuk merekam lagu kedalam format WAV dengan freqkuensi berbeda-beda dan perangkat lunak Ahead Nero Wave Editor untuk memotong lagu. Sedangkan system operasi yang digunakan adalah Windows 98 Second Edition. Untuk basis datanya digunakan Microsoft Access 2000.

Perangkat keras yang digunakan adalah komputer dengan *processor* AMD Athlon 900 MHz, memori sebesar 256 MB dan kapasitas *hardisk* sebesar 20 Gb.

2.4. Tujuan Percobaan

Pada penelitian ini percobaan yang dilakukan adalah :

1. Pengukuran waktu

Percobaan ini mengamati waktu pencarian pada basis data, dengan parameter berupa jumlah lagu yang ada pada basis data dan durasi *input* yang diberikan.

Untuk pengukuran waktu terhadap jumlah lagu yang ada pada basis data, dipakai basis data yang memiliki jumlah koleksi lagu sebanyak 30 buah, 40 buah dan 50 buah dengan *input* yang memiliki durasi berbeda-beda yaitu 10, 20 dan 30 detik. Kemudian diamati bagaimana hubungan antara jumlah lagu yang ada pada basis data dengan waktu pencarian.

Sedangkan untuk faktor perbedaan durasi *input*, potongan lagu yang diberikan sebagai *input* memiliki durasi 10 detik, 20 detik dan 30 detik (masing-masing 10 judul lagu), yang nantinya akan diamati hubungan antara perubahan durasi *input* dengan waktu pencarian. Percobaan ini dicobakan

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak cipta dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

terhadap basis data dengan jumlah koleksi 50 buah lagu.

2. Perbedaan panjang *input*

Percobaan ini dilakukan untuk mengetahui bagaimana pengaruh perubahan durasi potongan lagu (masing-masing 10 detik, 20 detik dan 30 detik) terhadap hasil temu kembali yang diperoleh. Percobaan ini dicobakan kepada basis data dengan koleksi lagu sebanyak 50.

3. Perbedaan posisi potongan lagu terhadap lagu asal

Percobaan ini dilakukan untuk mengetahui bagaimana pengaruh posis atau letak potongan lagu terhadap lagu asal (awal, tengah dan akhir lagu) terhadap hasil temu kembali yang diperoleh. Percobaan ini dicobakan kepada basis data dengan koleksi lagu sebanyak 50.

4. Perubahan amplitudo pada *input*

Percobaan ini dilakukan untuk mengetahui bagaimana pengaruh adanya perubahan amplitudo *input* terhadap hasil temu kembali yang diperoleh. Percobaan ini dicobakan pada basis data yang memiliki koleksi 50 judul lagu.

5. Perbedaan frekuensi *input*

Percobaan ini dilakukan untuk mengetahui bagaimana pengaruh perubahan frekuensi *input* terhadap hasil temu kembali yang diperoleh. Untuk penelitian ini potongan lagu yang diberikan sebagai *input* memiliki frekuensi 11 KHz, 16 KHz, 22 KHz, 24 KHz, 32 KHz dan 44 KHz (masing-masing 10 judul lagu). Percobaan ini dicobakan pada basis data dengan jumlah koleksi sebanyak 50 judul lagu.

2.5. Asumsi

Asumsi-asumsi yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Lagu yang relevan adalah lagu yang memiliki kecocokan lebih dari 75 %.
2. Untuk percobaan pengukuran waktu dan pengaruh perbedaan panjang *input*, posisi potongan lagu terhadap lagu asal tidak diperhatikan.
3. Untuk percobaan pengaruh perbedaan frekuensi *input* dan pengaruh perbedaan amplitudo *input* terhadap hasil yang didapatkan, posisi dan durasi *input* tidak diperhatikan.
4. Untuk percobaan pengaruh posisi *input* panjang atau durasi *input* tidak diperhatikan.

Untuk pembulatan angka, jika angka dibelakang koma lebih besar atau sama dengan 5 maka akan dibulatkan keatas sedangkan untuk angka lebih kecil dari 5 akan dibulatkan kebawah

3. HASIL EKSPERIMEN

Pengukuran waktu

Pengukuran waktu pada penelitian ini hanya dilakukan pada saat proses pencocokan string *input* dengan teks yang ada di basis data. Pada percobaan, pengukuran waktu dicatat dalam satuan mili detik.

1. Hubungan Waktu dan Jumlah Koleksi Basis Data

Hasil percobaan pengukuran waktu pencarian terhadap basis data yang memiliki jumlah koleksi yang berbeda-beda ditunjukkan oleh Tabel 1.

Pengukuran dilakukan terhadap basis data yang memiliki jumlah koleksi sebanyak 30, 40 dan 50 buah judul lagu dengan *input* berdurasi 10, 20 dan 30 detik (masing-masing durasi 10 kali ulangan kemudian dirata-ratakan).

Untuk *input* yang memiliki durasi 10 detik, pencarian pada basis data yang memiliki jumlah koleksi sebanyak 30 judul lagu membutuhkan waktu sekitar 371 mili detik, sedangkan pada basis data dengan jumlah koleksi sebesar 40 judul waktu yang dibutuhkan sekitar 461 mili detik dan untuk basis data dengan koleksi sebanyak 50 judul diperlukan waktu sekitar 548 mili detik. Untuk *input* dengan durasi 20 detik waktu yang dibutuhkan untuk pencarian pada masing-masing basis data adalah 407 mili detik, 504 mili detik, dan 600.33 mili detik, sedangkan untuk *input* yang berdurasi 30 detik waktu yang dibutuhkan untuk melakukan pencarian pada masing-masing basis data sebesar 430 milik detik, 543,67 mili detik dan 644,67 mili detik.

Dari data yang diperoleh menunjukkan makin banyak koleksi lagu, makin besar pula waktu pencarian yang dibutuhkan (Gambar 2).

Kejadian seperti ini adalah hal yang umum pada temu kembali informasi. Semakin banyak jumlah koleksi maka makin banyakjumlah perbandingan yang dilakukan pada saat melakukan pencarian.

Tabel 1. Hasil pengukuran waktu pencarian pada basis data dengan jumlah koleksi dan durasi *input* beragam

Jumlah Koleksi Basis Data	Durasi *Input* (detik)	Waktu (mili detik)		
		Rata-rata	Maksimum	Minimum
30	10	371	440	270
	20	407	500	330
	30	430	490	380
40	10	461	550	380
	20	504	610	440
	30	543.67	610	490
50	10	548	660	490
	20	600.33	710	550
	30	644.67	710	600

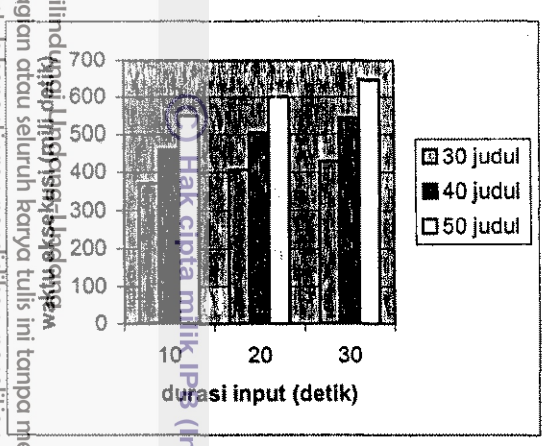
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

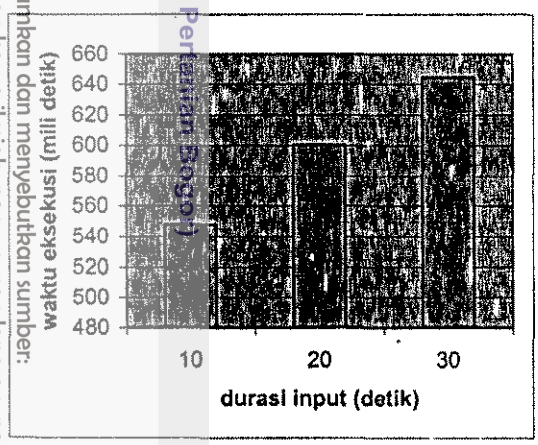
Hak Cipta Dilindungi Undang-Undang
(C) Hak cipta milik IPB (Institut Pertanian Bogor)

2. Hubungan Waktu dan Durasi *Input*

Pada percobaan ini akan diamati bagaimana pengaruh perubahan durasi *input* terhadap waktu pencarian. Durasi *input* yang digunakan adalah 10, 20 dan 30 detik. *Input* tersebut dicobakan pada basis data yang memiliki jumlah koleksi lagu sebanyak 50 judul (Tabel 1 dan Gambar 3).



Gambar 2. Grafik hubungan waktu pencarian dengan jumlah koleksi basis data dan durasi *input* yang beragam



Gambar 3. Grafik hubungan perubahan durasi *input* dan waktu pencarian

Panjang *Input*

Pada percobaan ini akan diamati hubungan antara panjang (durasi) *input* terhadap hasil temu kembali. Durasi *input* yang digunakan adalah 10, 20 dan 30 detik.

Dari hasil percobaan dapat disimpulkan bahwa makin panjang *input* maka akan makin baik pula *output* yang didapatkan. Hal ini terjadi karena makin panjangnya *input* akan membuat lebih banyak lagi nada pada *input* yang dapat dibandingkan dengan nada pada lagu di basis data. Dengan makin banyaknya nada yang dibandingkan maka akan memperkecil kemungkinan lagu-lagu yang tidak relevan terambil, sebab bisa saja ada beberapa lagu yang memiliki nada yang sama pada 10 nada petamanya, akan tetapi sangat berbeda pada nada-nada berikutnya, sehingga *input* dengan durasi yang lebih panjang akan lebih spesifik menunjuk ke lagu yang sesuai.

Posisi *Input*

Pada percobaan ini akan diamati bagaimana hubungan antara posisi *input* terhadap hasil temu kembali. Posisi *input* yang digunakan adalah awal, pertengahan dan akhir lagu.

Dari percobaan yang dilakukan ternyata *input* dengan posisi diakhir lagu memiliki persentase yang lebih baik jika dibandingkan *input* dengan posisi diawal dan dipertengahan lagu. Untuk persentase terambil, *input* dengan posisi diakhir lagu memiliki persentase sebesar 91,67% sedangkan *input* dengan posisi diawal dan tengah lagu masing-masing memiliki persentase sebesar 66,67% dan 83,33% (Tabel 2). Untuk persentase terambil pada posisi pertama *input* berposisi diakhir lagu memiliki persentase sebesar 83,33% (Tabel 2), persentase ini lebih besar jika dibandingkan dengan *input* berposisi diawal dan ditengah lagu. Untuk nilai persentase kecocokan yang paling tinggi, *input* dengan posisi diakhir lagu memiliki nilai terbesar yaitu sebesar 96,154%. Meskipun demikian *input* dengan posisi diawal dan tengah lagu memiliki nilai diatas 90%. (Tabel 2).

Tabel 2. Hasil percobaan pengaruh perubahan posisi *input* terhadap hasil

yang Diamati	Posisi *Input*		
	Depan	Tengah	Belakang
Jumlah Terambil	8	10	11
Jumlah Tidak Terambil	4	2	1
Persentase Terambil	66,667	83,33	91,6667
Persentase Tidak Terambil	33,333	16,67	8,33333
Jumlah Terambil pada Urutan 1	7	8	10
Jumlah Terambil Bukan pada Urutan 1	1	2	1
Jumlah Tidak Terambil	4	2	1
Persentase Terambil pada Urutan 1	58,333	66,67	83,3333
Persentase Terambil Bukan pada Urutan 1	8,3333	16,67	8,33333
persentase Tidak Terambil	33,333	16,67	8,33333
Persentase Kecocokan Tertinggi	90,833	94,65	96,154

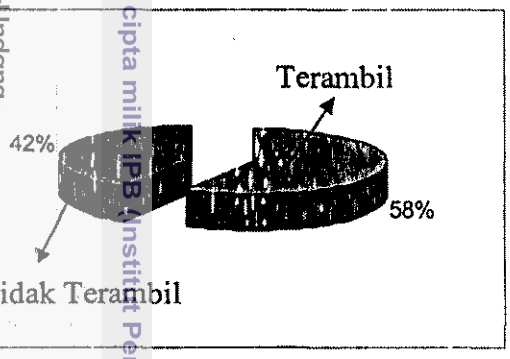
Bogor Agricultural University

1. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak cipta milik IPB (Institut Pertanian Bogor)
© Hak Cipta Dilindungi Undang-Undang
Bogor Agricultural University

Perubahan Amplitudo *Input*

Percobaan ini bertujuan untuk mengamati hubungan antara perubahan amplitudo pada frekuensi-frekuensi tengah (*center frequencies*) yang membentuk suara *input* dengan hasil temu kembali yang dihasilkan. *input* dibagi kedalam dua golongan yaitu *input full* bass dan *input full* treble.

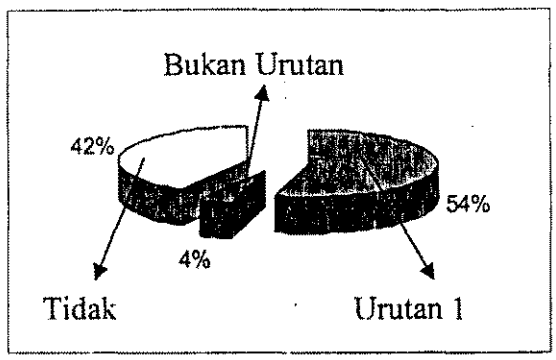
Secara keseluruhan persentase terambil yang dihasilkan oleh kedua macam *input* tersebut sekitar 58% (Gambar 4). Sedangkan persentase terambil pada urutan pertama yang diperoleh dari percobaan sekitar 54% (Gambar 5).



Gambar 4. Persentase terambil atau tidaknya sebuah lagu pada basis data, dimana terjadi perubahan bass dan treble pada *input*

Untuk persentase kecocokan antara *input* dengan lagu asal, nilai tertinggi yang didapat adalah sebesar 94.527%. Sedangkan persentase kecocokan antara *input* dan lagu asal, nilai tertinggi untuk masing-masing jenis *input* adalah sebesar 84.08% untuk *input full* bass dan 94.527% untuk *input full* treble.

Hasil yang diperoleh dari percobaan menunjukkan bahwa meskipun terjadi perubahan amplitudo (dalam hal ini perubahan pada frekuensi tengah penyusun bass dan treble) tetapi hasil temu kembali yang diperoleh masih cukup baik. Hal ini menunjukkan bahwa adanya perubahan amplitudo pada frekuensi-frekuensi tengah tidak memberikan pengaruh yang berarti pada hasil temu kembali karena perubahan amplitudo tidak mengakibatkan berubahnya frekuensi.



Gambar 5. Persentase terambil pada urutan pertama, selain urutan pertama dan tidak terambil sebuah lagu pada basis data, dimana terjadi perubahan bass dan treble pada *input*

Perbedaan Frekuensi *Input*

Pemberian *input* yang memiliki frekuensi berbeda-beda bertujuan untuk mengetahui seberapa besar pengaruh perubahan frekuensi terhadap hasil temu kembali yang didapat. Frekuensi *input* yang digunakan sebagai *input* adalah 11 KHz, 16 KHz, 22 KHz, 24 KHz, 32 KHz dan 44 KHz, sedangkan berkas-berkas lagu yang hendak dicari memiliki frekuensi 8 KHz.

Dari hasil percobaan dapat disimpulkan bahwa untuk frekuensi *input* yang berbeda-beda didapatkan hasil yang sama yaitu tidak ada satu pun judul lagu yang terambil dari basis data. Hal ini menunjukkan bahwa sekecil apapun perubahan frekuensi dari sebuah lagu akan mengakibatkan ketidakcocokan antara *input* dengan lagu yang dicari.

Hal tersebut diatas terjadi karena banyaknya gelombang atau getaran yang dihasilkan dalam waktu 1 detik berbeda, frekuensi adalah banyaknya gelombang atau getaran yang terjadi dalam waktu 1 detik, banyaknya gelombang yang dihasilkan oleh koleksi lagu-lagu yang ada di basis data adalah 8000 gelombang per detik (8KHz), sedangkan banyaknya gelombang yang dihasilkan oleh *input* lebih besar, yaitu antara 11000 sampai 44000 gelombang per detik (11 KHz sampai 44 KHz). Hal ini mengakibatkan string S, D dan U yang dihasilkan oleh lagu-lagu pada basis data berbeda dengan yang dihasilkan oleh *input*. Karena string S, D dan U yang dihasilkan sangat berbeda maka tidak akan pernah ditemukan kecocokan antara *input* dengan lagu yang dicari.

Jadi sekecil apapun perbedaan frekuensi antara *input* dengan lagu yang ada dalam basis data akan mengakibatkan tidak ada satu lagupun yang ditemukembalikan oleh sistem. Hal ini sangat berbeda jika dibandingkan dengan hasil percobaan pengaruh perubahan

amplitudo, karena adanya perubahan frekuensi akan mengakibatkan perubahan bentuk suara, sementara adanya perubahan amplitudo tidak berpengaruh pada bentuk suara karena perubahan amplitudo tidak merubah frekuensi.

Sistem

Untuk penelitian ini sistem yang digunakan untuk melakukan proses temu kembali bukanlah sebuah sitem temu kembali informasi yang utuh dan terintegrasi dengan baik. Sistem dalam penelitian ini terbagi menjadi tiga buah modul yaitu modul 1 yang merupakan elemen terpenting sebab di modul inilah terdapat proses pencocokan string dengan menggunakan algoritma yang dikembangkan oleh Baeza-Yates dan Perleberg (1992), modul 2 adalah sebuah modul yang berfungsi untuk mengubah atau mengkonversi vektor yang diperoleh dari proses *pitch tracking* menjadi string S, D dan U yang akan disimpan sebagai berkas teks yang nantinya akan digunakan oleh modul 1 sebagai *input*, modul 2 tidak hanya menyediakan *input* bagi modul 1, tetapi modul ini juga menyediakan hasil konversinya untuk dimasukkan kedalam basis data, sedangkan modul 3 adalah modul yang digunakan untuk melakukan proses *pitch tracking* atau dengan kata lain modul 3 berfungsi sebagai *pitch tracker*, output yang dihasilkan oleh modul ini disimpan dalam bentuk berkas biner yang digunakan sebagai *input* oleh modul 2. Ketiga modul tersebut dibangun dalam lingkungan bahasa pemrograman yang berbeda . Untuk pembuatan modul 1 dan 2 digunakan bahasa pemrograman Microsoft Visual C++, sedangkan untuk membuat modul 3 digunakan bahasa program yang ada di MATLAB.

Meskipun demikian bukan berarti sistem yang belum terintegrasi ini tidak dapat diintegrasikan dengan baik sebab dengan menggunakan MATLAB *routine* yang dibuat dengan menggunakan bahasa C dapat digunakan secara langsung oleh MATLAB. Selain itu dengan menggunakan MATLAB dapat dibuat *interface* yang menarik.

4. KESIMPULAN

Dari hasil-hasil yang diperoleh dari percobaan-percobaan yang dilakukan dapat ditarik beberapa kesimpulan, yaitu :

1. Jumlah koleksi lagu pada sebuah basis data akan mempengaruhi waktu pencarian. Makin banyak koleksi lagu yang ada pada suatu basis data maka akan semakin lama pula waktu yang diperlukan.
2. Panjang pendeknya sebuah potongan lagu yang digunakan sebagai *input* akan mempengaruhi waktu pencarian, sebab makin panjang durasi *input* maka akan makin lama pula waktu yang dibutuhkan. Hal ini disebabkan oleh penggunaan algoritma Baeza-Yates dan Perleberg sebagai algoritma pencocokan string.
3. Untuk percobaan perubahan panjang (durasi) *input*, hasil percobaan menunjukkan bahwa *input* dengan durasi 30 detik memiliki rata-rata persentase terambil pada urutan pertama paling baik, serta persentase terambil dan persentase kecocokan antara *input* dengan lagu asal yang baik pula. Secara umum dapat disimpulkan bahwa makin panjang durasi *input* maka akan makin baik pula hasil yang akan diperoleh.
4. *Input* dengan posisi diakhir lagu memiliki persentase yang paling baik untuk tiga katagori yang diamati. Hal ini terjadi karena hamper tiap lagu memiliki kecenderungan untuk terus menurun diakhir lagu.
5. Adanya perubahan amplitudo pada frekuensi tengah tidak memberikan pengaruh yang sangat ekstrim, tidak ada yang terambil, terhadap hasil yang diperoleh sebab perubahan amplitudo tidak merubah frekuensi sehingga bentuk suara pun tidak berubah.
6. Frekuensi *input* yang berbeda dengan frekuensi lagu asal akan mengakibatkan tidak terambilnya lagu pada basis data. Hasil percobaan menunjukkan tidak ada satu lagupun yang ditemukembalikan oleh sistem.

REFERENSI

[1] **Baeza-Yates, R. A. & C. H. Perleberg.** 1992. *Fast and Practical Aproximate String Matching.* http://citiseer.nj.nec.com/baeza-yates92fast.html. [16 Juli 2002].

[2] **Bainbridge, D., C. G. Nevill-Manning, I. H. Witten, L. A. Smith & R. J. McNab.** 1999. *Toward a Digital Library of Popular Music.* http://craig.nevill-manning.com/~nevill/publications/DL99.pdf.[11 Juni 2002].

[3] **Ghias, A., J. logan, D. Chamberlin & B. C. Smith.** 1995. *Query by Humming : Musical Information Retrieval in an Audio Database.* http://www.cs.cornel.edu/info/faculty/bsmith/query_by_humming.htm. [11 Juni 2002].

[4] **McNab, R. J., L. A. Smith, I. H. Witten, C. L. Handerson & S. J. Cunningham.** 1996. *Towards The Digital Music Library : Tune Retrieval from Acoustic Input.* http://www.cs.waikato.ac.nz/~ihw/papers/96RJM_LAS_IHW_CLH_SJC.pdf.[11 Juni 2002].

[5] **Nevill-Manning, C. G. & C. Francu. 2000.** *Distance Metrics and Indexing Stratgies for a Digital Library of Popular Music.* http://craig.nevill-

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

manning.com/~nevill/publications/ICME00.pdf.[11 Juni 2002].

[6] **Part-Enader, E.** 1995. *The Matlab Handbook. Addison-wesley*. Canada.

[7] **Salton, G.** 1989. *Automatic Text Processing : The Transformation, Analysis and Retrieval of Information by Computer*. Addison-wesley. Canada.

[8] **Sapp, C., A. Master & P. de la Cuadra.** 2001. *Efficient Pitch Detection Techniques for Interactive Music*. http://ccrma-www.stanford.edu/~craig/papers/01/ICMC01_pitch.pdf. [16 Juli 2002].

[9] **Uitenboogerd, A. & J. Zobel.** 1999. *Melodic Matching Techniques for Large Music Databasess*. http://www.kom.e-technik.tu-darmstadt.de/~cmmn99/ep/uitdenboogerd/Melodic Matching Techniques for Large Music Databases.htm[11 Juni 2002].

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

PERBANDINGAN ALGORITMA HUFFMAN STATIK DENGAN ALGORITMA HUFFMAN ADAPTIF PADA KOMPRESI DATA TEKS

Bib Paruhum Silalahi[1], Julio Adisantoso[2], Danny Dimas Sulistio[3]
[1]Departemen Matematika, FMIPA, Institut Pertanian Bogor
Jl. Raya Pajajaran Bogor, Indonesia
[2]Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor
Jl. Raya Pajajaran Bogor, Indonesia
[3]Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor
Jl. Raya Pajajaran Bogor, Indonesia
Email: ediary@plasa.com

ABSTRAK

Penelitian ini bertujuan untuk mempelajari dan membandingkan unjuk kerja dari algoritma Huffman Statik dan algoritma Huffman Adaptif pada kompresi data. Ruang lingkup penelitian hanya terbatas pada kompresi data teks (.txt) yang dilakukan pada tiga buah percobaan yaitu percobaan dengan menggunakan file teks yang berasal dari potongan artikel, percobaan dengan menggunakan file teks dengan satu variasi karakter, dan percobaan menggunakan file teks dengan lima dan 256 variasi karakter. Berbagai kriteria yang digunakan dalam perbandingan kedua algoritma diantaranya rasio kompresi, lamanya waktu yang diperlukan untuk mengkompresi file, dan lamanya waktu untuk mendekompresi file menjadi seperti semula.Kompresi menggunakan algoritma Huffman Statik memiliki kompleksitas sebesar O(n lg m), sedangkan algoritma Huffman Adaptif memiliki kompleksitas sebesar O(n.m), dengan nilai n adalah banyaknya karakter dan m adalah besarnya variasi karakter. Dari percobaan potongan artikel menunjukan waktu iterasi yang diperlukan oleh algoritma Huffman Statik untuk melakukan kompresi dan dekompresi adalah cenderung lebih kecil dibandingkan dengan yang dilakukan oleh algoritma Huffman Adaptif. Namun untuk hasil kompresi terlihat unjuk kerja Huffman Adaptif adalah lebih baik dibandingkan Huffman Statik.*

Kata kunci: algoritma, Huffman Statik, Huffman Adaptif, kompleksitas, kompresi data.

PENDAHULUAN

Pada saat ini kebutuhan akan informasi sangat diperlukan oleh masyarakat umum. Setiap informasi yang beredar dapat direpresentasikan dalam bentuk citra, suara maupun teks secara digital. Dengan semakin banyaknya informasi yang perlu disimpan secara digital, secara automatis akan meningkatkan keperluan untuk menyediakan media penyimpanan data yang lebih besar lagi. Oleh karena itu diperlukan suatu alternatif mekanisme penyimpanan data sehingga dengan media penyimpanan yang ada, dapat menyimpan data sebanyak-banyaknya.

Pemampatan data, atau yang lebih dikenal dengan istilah kompresi data merupakan salah satu metode untuk memperkecil kebutuhan penyimpanan data pada suatu media penyimpanan data. Dengan melakukan metode kompresi dapat memperkecil ukuran data, sehingga kebutuhan akan media penyimpanan data dapat lebih efektif dan ukuran data yang disimpan dapat optimal. Selain berguna dalam penyimpanan data, kompresi data dapat membantu memperkecil ukuran data yang ditransmisikan di dalam suatu media jaringan, seperti internet. Sehingga dengan ukuran data yang lebih kecil, maka waktu yang diperlukan untuk mentransfer data tersebut dapat diperkecil. Namun dibalik kelebihan yang ditawarkan oleh metode ini, harus diperhatikan pula lamanya waktu yang diperlukan untuk membuat data menjadi termampatkan, dan lamanya waktu untuk mengembalikan data tersebut seperti sediakala.

Pada penelitian kali ini akan dipaparkan suatu kajian algoritma Huffman melalui metode statik dan metode adaptif. Algoritma Huffman merupakan salah satu pelopor lahirnya kompresi data, sehingga ukuran data yang perlu disimpan menjadi lebih kecil dibandingkan dengan ukuran data sebenarnya.

Penelitian mengenai kompresi terhadap data teks menggunakan algoritma Huffman telah dilakukan sebelumnya oleh Hutasoit [3] mengenai pengaruh n-gram dalam pembentukan kode Huffman pada *file* teks berbahasa Indonesia dan Layungsari [4] mengenai implementasi kompresi multi tahap menggunakan algoritma Huffman pada *file* teks. Kesimpulan dari penelitian Hutasoit [3] adalah pengaruh n-gram dapat menghasilkan rasio kompresi yang lebih baik yang ditunjukkan dengan rasio kompresi untuk *tree* yang monogram lebih kecil dibandingkan dengan *tree* digram. Dari penelitian Layungsari [4] dapat disimpulkan bahwa hasil kompresi *file* teks menggunakan kompresi multi tahap Huffman menunjukan hasil yang tidak memuaskan. Hal ini dikarenakan *file* hasil kompresi oleh kompresi tahap pertama telah menghasilkan kode prefiks yang optimal.

Output dari penelitian ini adalah sebuah sistem yang dapat membandingkan hasil kompresi yang dilakukan oleh metode Huffman Statik dengan hasil kompresi yang dilakukan oleh metode Huffman Adaptif. Selain itu diharapkan dengan

© Collection of IPB (Bogor Agricultural University)

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

pengembangan metode ini dapat dijadikan acuan lebih lanjut dalam penelitian mahasiswa mengenai metode pemampatan *file*.

TINJAUAN PUSTAKA

Kompresi Data

Kompresi data dapat dilihat sebagai kumpulan teori informasi dengan tujuan utamanya adalah untuk memperkecil ukuran data yang akan ditransmisikan [5]. Secara sederhana, karakteristik dari kompresi data dapat dianalogikan sebagai sebuah proses untuk mengubah sebuah *string* yang merupakan kumpulan karakter menjadi sebuah *string* yang baru dengan informasi yang sama namun dengan lebar atau ukuran yang lebih kecil. Suatu cara untuk mengembalikan data yang telah terkompresi menjadi seperti sediakala dikenal dengan istilah dekompresi data.

Secara garis besar kompresi data dapat dikelompokkan ke dalam dua metode yaitu metode kompresi *lossy* dan metode kompresi *lossless*. Metode kompresi *lossy* adalah suatu metode kompresi data dengan data yang telah terkompresi apabila dikembalikan ke dalam bentuk semula akan terdapat beberapa informasi yang hilang. Contoh dari metode ini adalah metode yang digunakan pada pemampatan data gambar dan data suara. Sedangkan untuk metode kompresi *lossless* merupakan suatu metode kompresi data dengan data yang telah terkompresi akan dapat dikembalikan seperti semula tanpa ada informasi yang hilang. Implementasi dari metode ini yaitu pada pemampatan data teks atau dokumen ASCII.

Pada metode kompresi *lossless* terdapat dua buah model utama yaitu metode *lossless* dengan menggunakan model statistik dan model kamus. Pada model statistik, kompresi data diawali dengan melakukan perhitungan setiap karakter yang ada di dalam *file*, kemudian dengan statistik karakter yang ada akan dilakukan pengkodean karakter dengan representasi lain. Representasi tersebut apabila dibandingkan dengan karakter asli diharapkan akan lebih kecil ukurannya. Model ini digunakan pada algoritma Huffman dan algoritma Aritmatik.

Sedangkan untuk model kamus, kompresi data diawali dengan melakukan perhitungan setiap *string* yang terdapat di dalam *file*. Kumpulan *string* tersebut akan disusun seperti sebuah indeks dan akan disimbolkan dengan sebuah representasi yang unik. Model ini digunakan pada algoritma Lempel-Ziv dan turunannya (LZW,LZSS, LZRW, dsb).

Algoritma Huffman

Algoritma Huffman diperkenalkan pertama kali oleh D.A. Huffman pada tahun 1950. Ide dasar dari algoritma ini adalah membuat kode dengan representasi *bit* yang lebih pendek untuk karakter ASCII yang sering muncul di dalam *file* dan membuat kode dengan representasi *bit* yang lebih panjang untuk karakter ASCII yang jarang muncul di dalam *file* [2]. Dalam perkembangannya algoritma Huffman terpecah menjadi dua buah kategori yaitu:

- Algoritma Huffman Statik adalah suatu algoritma yang menggunakan kemungkinan kemunculan dari setiap karakter yang telah ditetapkan pada awal pengkodean dan kemungkinan kemunculan karakter tersebut juga dapat diketahui oleh baik oleh enkoder maupun dekoder.
- Algoritma Huffman Adaptif adalah suatu algoritma dengan kemungkinan kemunculan dari setiap simbol tidak dapat ditentukan dengan pasti selama pengkodean. Hal ini disebabkan oleh perubahan pengkodean secara dinamis berdasarkan frekuensi dari simbol yang telah diolah sebelumnya.

Algoritma Huffman Statik

Cara kerja dari algoritma Huffman Statik untuk mengkompresi data yaitu dengan cara sebagai berikut:

1. Hitung jumlah karakter yang muncul di dalam *file*, sehingga masing-masing karakter yang ada akan memiliki bobot sesuai dengan banyaknya karakter tersebut di dalam *file*. Kemudian susunlah suatu pohon biner yang dibangun berdasarkan bobot karakter yang ada. Inti dari pembangunan pohon biner adalah menggabungkan dua buah karakter dengan tingkat kemunculan (bobot) yang lebih kecil, kemudian membangkitkan satu buah node *parent* yang memiliki bobot gabungan dari kedua karakter tersebut.
2. Lakukan pengkodean (*encoding*) karakter yang ada di dalam *file* menjadi suatu representasi *bit* sesuai dengan urutan pada pohon biner yang telah dibangun.

Algoritma dekompresi Huffman Statik dapat dijabarkan sebagai berikut:

```
1. Susun pohon biner.
2. X=""
3. Lakukan sampai karakter terakhir
   {selama X bukan leaf_Tree
      {X ← X + bit_selanjutnya}
    Kirim X ke buffer_file_dekompresi
    X=""
    }
4. Simpan buffer_file_dekompresi ke
   dalam file tujuan
```

Algoritma Huffman Adaptif

Algoritma Huffman Adaptif merupakan pengembangan lebih lanjut dari algoritma Huffman. Ide dasar dari algoritma ini adalah meringkas tahapan algoritma Huffman tanpa perlu menghitung jumlah karakter keseluruhan dalam membangun pohon biner. Algoritma ini dikembangkan oleh trio Faller, Gallager, dan Knuth dan kemudian

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang

dikembangkan lebih lanjut oleh Vitter, sehingga tercipta dua buah algoritma Huffman Adaptif yaitu algoritma FGK dan algoritma V. Secara umum algoritma Huffman Adaptif adalah sebagai berikut:

1. Prosedur enkoder

```
initialize_model();
do{
      c=getc(input);
      encode(c,output);
      update_model(c);
  }while(c!=eof);
```

2. Prosedur dekoder

```
initialize_model();
while((c=decode(input))!=eof);
  {
      putc(c,output);
      update_model(c);
  }
```

Dapat dilihat dari di atas, pada saat data akan dikompresi, pertama kali akan diinisialisasi sebuah pohon biner dengan sebuah node yang dikenal dengan *Not-Yet-Transmitted* (NYT) atau kode *escape*. Kemudian akan dilakukan pembacaan karakter satu persatu dari awal *file* sampai akhir. Selama pembacaan karakter, akan dikirim kode NYT setiap terdapat node (karakter) baru yang belum ada di dalam pohon. Hal ini akan memudahkan dekoder untuk membedakan antara sebuah kode dengan sebuah karakter baru. Setelah kode tersebut dikirimkan bersama kode ASCII karakter baru, maka diperlukan penambahan sebuah node baru untuk karakter baru, dan sebuah kode NYT dari kode NYT yang lama. Dan terakhir akan dilakukan peng-*update*-an pohon.

Sedangkan untuk proses dekompresi pada prosedur dekoder, tidak jauh berbeda dengan proses kompresi data. Hal yang berbeda hanya pada saat membaca data dari *file* tidak dilakukan perkarakter, namun dibaca per-*bit*, hal ini untuk mengidentifikasi keberadaan karakter yang diinisialisasi pada pohon, apakah sudah ada atau belum. Jika belum ada akan dilakukan penambahan node baru, namun apabila sudah ada akan dilakukan perubahan bobot karakter. Setelah itu diakhiri dengan peng-*update*-an pohon.

Salah satu kelebihan dari algoritma kompresi Huffman Adaptif ini adalah kemampuannya untuk melakukan kompresi *file* tanpa perlu mengetahui jumlah frekuensi dari masing-masing karakter sehingga tidak menyimpan *tree* pada *file* hasil kompresi.

METODOLOGI

Pengumpulan Bahan

Data yang digunakan adalah data teks dengan ekstensi *.txt dengan berbagai rentang ukuran *file*. Berbagai rentang ukuran *file* yang digunakan yaitu < 20.000 *byte*, antara 20.000 sampai 40.000 *byte* dan *file* dengan ukuran > 40.000 *byte*. *File* teks tersebut berisi potongan tajuk rencana harian sebuah media cetak *online* (Media Indonesia) yang beredar antara bulan Juni sampai November 2003. Selain itu digunakan sebuah mekanisme untuk membangkitkan karakter dengan rentang bervariasi dari satu buah variasi karakter.

Struktur Percobaan

Dalam penelitian ini dilakukan beberapa pengujian yang antara lain:

1. Melihat pola rasio pemampatan dan lamanya kompresi dan dekompresi algoritma dengan menggunakan *file* yang berasal dari potongan artikel.
2. Melakukan simulasi dengan menggunakan *file* teks yang hanya memiliki satu buah variasi karakter kemudian dilihat pola rasio pemampatan dan lamanya eksekusi dalam rentang 1.000 *byte*, 2.000 *byte*, 3.000 *byte* sampai 100.000 *byte*.
3. Mengamati rasio dan lamanya eksekusi algoritma pada *file* yang memiliki lima buah variasi karakter yang berbeda, dan 256 variasi karakter yang berbeda.
4. Penarikan kesimpulan menggunakan uji Analisis Ragam (ANOVA).

Analisis

Percobaan yang telah dirancang selanjutnya diuji coba dan dilakukan analisis terhadap kinerja algoritma dengan berbagai kriteria berikut:

1. Kebutuhan tempat penyimpanan

   Seluruh *file* yang diuji perlu dicatat ukuran *file* semula dan ukuran *file* setelah dikompresi. Hal ini diperlukan untuk mendapatkan rasio pemampatan yang dilakukan oleh algoritma tertentu. Perhitungan rasio pemampatan dapat didefinisikan sebagai berikut:

$$\text{Rasio Pemampatan} = \frac{\text{loss}}{\text{ukuran file asli}} \qquad (1)$$

   dengan:
   loss =ukuran *file* awal–ukuran *file* kompresi
   =ukuran *file* yang hilang akibat termampatkan.
2. Pengamatan *running time* program

   Kompleksitas dari suatu algoritma dapat diestimasi dengan melakukan perhitungan *running time*. Dengan melihat *flowchart* masing-masing algoritma, sehingga dapat dihitung kompleksitas dari masing-masing algoritma tersebut.

Perancangan Sistem

- Sistem dapat melakukan kompresi dan dekompresi *file* pada kedua algoritma.
- Dapat menampilkan statistik hasil kompresi dan dekompresi dalam bentuk data dan grafik.

© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

Desain Sistem

Sebelum mengimplementasikan rancangan sistem, perlu ditentukan dahulu alur sistem sehingga dapat menunjang kinerja sistem yaitu:

1. Desain masukan

   Pada masing-masing rentang *file* akan berisi 10 buah *file* teks sumber (artikel) yang akan digunakan pada saat pengujian.
2. Desain keluaran

   Untuk membedakan antara *file* teks dengan *file* hasil kompresi dan dekompresi dari masing-masing algoritma, maka diberikan ekstensi *file* yang berbeda untuk setiap operasi yaitu:
   - Algoritma Huffman Statik

     File hasil kompresi menggunakan ekstensi *.huf, sedangkan *file* hasil dekompresi menggunakan ekstensi *.new.
   - Algoritma Huffman Adaptif

     File hasil kompresi menggunakan ekstensi *.ada, sedangkan *file* hasil dekompresi menggunakan ekstensi *.now.
3. Desain Proses

   File artikel yang telah dikelompokkan ke dalam beberapa rentang akan dipisahkan ke dalam beberapa direktori (misal 1,2, dan 3). Setiap *file* tersebut akan diujikan sebanyak 10 kali ulangan. Pada saat pengujian dan simulasi, hasil dari kompresi dan dekompresi akan dimasukkan ke dalam sebuah *file* teks (*.txt) secara automatis yang kemudian akan diolah dan digabungkan ke dalam *file* Excell (*.xls) secara manual. Kemudian diuji dengan ANOVA menggunakan program SAS versi 8.

HASIL DAN PEMBAHASAN

Untuk mempemudah pengaksesan data, rentang *file* yang telah didefinisikan sebelumnya kemudian direpresentasikan sebagai tiga buah sub direktori (1, 2, dan 3) dengan masing-masing sub direktori tersebut memiliki 10 buah *file* teks yang diberi nama 1.txt, 2.txt, 3.txt sampai 10.txt.

Untuk mendapatkan hasil kompresi dilakukan dengan cara memilih menu simulasi kompresi dan untuk mendapatkan hasil dekompresi dilakukan dengan memilih menu simulasi dekompresi. Kedua menu ini dipecah berdasarkan masing-masing algoritma.

Setelah simulasi kompresi dan dekompresi dijalankan, pada setiap direktori yang memiliki sub direktori akan dihasilkan *file* teks yang berisi rangkuman rasio pemampatan, waktu eksekusi dan identitas *file* yang diuji. Dengan bantuan aplikasi Ms Excell penulis mengolah data pada *file* teks tersebut, kemudian mencari rata-rata dari masing-masing kriteria uji.

Representasi Struktur Data

1. Pada *file* kompresi Huffman Statik

   File kompresi (dengan ekstensi *.huf) diawali dengan *header file* Huffman dengan panjang 4 *byte* yang menunjukan *file* tesebut dikompresi atau tidak. Kemudian disambung 1 *byte* sebagai kode CRC sederhana. Lalu disambung dengan 4 *byte* untuk menyimpan informasi ukuran *file* sumber, 2 *byte* untuk menyimpan banyaknya variasi karakter, yang kemudian disambung dengan pasangan-pasangan 2 *byte* yang berisi karakter dan panjang representasi karakter tersebut pada pohon Huffman. Dan pada akhirnya akan dituliskan representasi pohon biner dan hasil *encoding file* sumber menggunakan pohon biner.
2. Pada *file* kompresi Huffman Adaptif

   File kompresi (dengan ekstensi *.ada) diawali dengan kode *escape*, disambung representasi kode *escape* pada pohon biner. Setiap kali ada karakter baru, akan dikirimkan terlebih dahulu representasi kode *escape* yang disambung dengan kode ASCII yang bersangkutan. Sedangkan untuk karakter yang telah didefinisikan, maka akan disimpan representasi karakter yang bersangkutan.
3. Pada *file* statistik simulasi artikel

   Diawali dengan judul untuk kolom-kolom yang ada seperti rasio pemampatan, lama eksekusi, alamat *file* sumber, ukuran *file* sumber, alamat *file* tujuan, ukuran *file* tujuan dan pada baris baru akan disambung dengan data *file* yang berada pada sub direktori yang sedang diuji.

Kompleksitas Algoritma Huffman Statik

Secara garis besar alur algoritma Huffman Statik pada saat mengkompresi *file* adalah:

```
1. Hitung banyak karakter yang ada
   pada file.
2. Lakukan sorting jumlah_Karakter
   yang ada.
3. Bentuk pohon Huffman.
4. Ubah seluruh karakter yang ada di
   dalam file sesuai dengan
   representasi bit pada pohon.
```

Kompleksitas pada iterasi 1 prosedur kompresi Huffman di atas adalah O(n) dengan n adalah besarnya ukuran dari *file* yang merepresentasikan banyaknya karakter yang ada dalam *file* tersebut. Sedangkan pada iterasi 2 memiliki kompleksitas sebesar O(n lg n) dengan n adalah banyaknya karakter yang muncul. Misal terdapat sebuah *string* aabccca maka n untuk iterasi 2 adalah tiga (karakter yang muncul a, b, dan c).

Pada iterasi 3, pembentukan pohon Huffman yaitu dengan cara mengambil 2 buah node dari array yang sudah terurut pada iterasi 2 dengan bobot kemunculan terkecil, kemudian dibuat sebuah node dengan bobot gabungan dari kedua node tersebut. Lalu keluarkan kedua node dari *array*, dan masukkan node baru ke dalam *array*. Prosedur ini terus berulang sampai tersisa hanya 1 buah node pada

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang

array yang memiliki bobot yang merepresentasikan jumlah seluruh karakter yang ada di dalam *file*. Oleh karena pembentukan pohon ini menggunakan kaidah algoritma Heap [1], sehingga kompleksitas dari iterasi 3 ini adalah O(n lg n). Kode semu untuk prosedur pembentukan pohon Huffman adalah sebagai berikut:

```
Untuk semua node dalam Array
Cari 2 buah node dengan bobot terkecil
dari Array;
  {
    node1=lowest(Array);
    node2=2nd_lowest(Array);
  }
Buat node baru dengan bobot gabungan,
dan node baru tersebut menjadi parent
dari kedua node terkecil
  {
    NewNode.weight=node1.weight+
                   node2.weight;
    NewNode.LeftChild =node1;
    NewNode.RightChild =node2;
    node1.Parent=NewNode;
    node2.Parent=NewNode;
  }
```

Pada iterasi 4, akan dilakukan proses transformasi seluruh karakter yang ada di dalam *file*. Sebelum mengubah karakter, terlebih dahulu didefinisikan nilai biner dari cabang pohon Huffman yang telah terbentuk, seperti berikut:

```
Create_Bit_Sequence(From_Top          to
Every_Leaf)
    {
  Untuk Node yang berada di sebelah
  kiri diberi nilai 0 dan untuk Node
  yang berada disebelah kanan diberi
  nilai 1. Pemberian nilai ini
  dilakukan mulai dari cabang paling
  atas sampai pada seluruh ujung daun.
  }
```

Setelah itu barulah dilakukan proses *encoding* seluruh karakter *file* sumber, seperti berikut:

```
Lakukan untuk seluruh byte yang ada
di dalam file
   { Baca 1 byte, representasikan
   dalam bentuk karakter ASCII. Baca
   representasi karakter dalam pohon
   Huffman.      Masukan setiap bit
   representasi satu persatu kedalam
   buffer array yang berukuran 8.
   Jika buffer array terisi penuh,
   bentuk sebuah karakter yang sesuai
   nilai array, lalu set array
   menjadi kosong. }
  Untuk byte terakhir, jika array
  buffer masih ada nilai, bentuk
  sebuah karakter yang sesuai nilai
  array, lalu set array menjadi
  kosong.
```

Kompleksitas algoritma untuk prosedur di atas adalah O(n) dengan n merupakan besarnya variasi karakter, karena pada iterasi ini dilakukan kunjungan dari node yang paling atas sampai seluruh node dilewati, sedangkan pada Gambar 8 memiliki kompleksitas O(n lg m) dengan n adalah besar ukuran *file* dan m adalah besar variasi karakter. Dari uraian di atas dapat disimpulkan bahwa untuk kompresi menggunakan algoritma Huffman Statik memiliki kompleksitas sebesar O(n lg m), dengan nilai n adalah banyaknya karakter dan m adalah besarnya variasi karakter.

Kompleksitas Algoritma Huffman Adaptif

Algoritma Huffman Adaptif secara umum dapat dituliskan sebagai berikut:

```
BEGIN
 1. Inisialisasi tree
 'untuk seluruh karakter yang ada di
  dalam file
    {
    2.Baca karakter satu-satu
    3.Ambil representasi karakter
      dari pohon huffman
    4.Jika representasi tersedia,
      kirim bit ke stream
    5.Jika belum tersedia, tambahkan
      karakter yang bersangkutan ke
      stream
    6.Lakukan update bobot karakter
      yang terkirim pada pohon
      Huffman
    }
 7. Kirim bit ke stream untuk escape
    code
 8. Jika stream masih memiliki nilai,
    tambahkan digit 0 sebanyak sisa
    ruang yang masih kosong
 9. Bentuk karakter dari stream
    terakhir
END
```

Pada iterasi 1, dilakukan inisialisasi *tree* dengan masing-masing node belum memiliki bobot dan belum menunjuk ke mana-mana. Kompleksitasi iterasi ini adalah O(n) dengan n sebesar 512. Hal ini didasari bahwa maksimum banyaknya node pada *tree* adalah sebesar 2 x (256) - 1 yaitu sebanyak 511 node. Adapun prosedur inisialisasi *tree* sebagai berikut:

```
BEGIN
    'set node tidak memiliki hubungan
    satu dengan yang lain dan belum
    memiliki bobot
    'set posisi node belum ada di
    dalam tree
    'set node yang terkirim = 0
    'Update_Tree(EscapeCode)
END
```

Setelah sebuah karakter dibaca oleh mesin enkoder, akan dilakukan pengecekan tentang sudah ada atau belum karakter tersebut di dalam *tree*. Jika representasi karakter sudah ada, maka representasi *bit* karakter tersebut akan dimasukan ke dalam buffer. Namun jika belum, maka nilai representasi *bit* dari node NYT yang akan disimpan. Prosedur untuk penyimpanan *bit* adalah:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

```
BEGIN
    'input representasi karakter
    kedalam buffer dalam satuan bit
    'jika buffer penuh,masukkan nilai
    buffer kedalam senarai data, lalu
    set buffer=0
END
```

Kompleksitas untuk prosedur di atas adalah sebesar O(lg n), dengan n adalah besar variasi karakter yang sudah diolah oleh mesin enkoder. Setelah penyimpanan nilai *bit* karakter, akan dilakukan pengubahan pohon Huffman, dengan cara menambahkan bobot karakter yang dibaca, yang disambung dengan penambahan nilai bobot *parent* sampai node yang paling atas, seperti berikut:

```
BEGIN
    'baca posisi karakter di dalam
    tree
    'jika karakter belum ada, maka
    tambahkan node baru kedalam tree,
    lalu set:
        'bobot=1,      parentnode=NYT,
        lalu set NYT berada pada
        RightNode NYT lama
    'jika karakter sudah ada, maka
    tambahkan bobot karakter pada
    tree
    'lakukan pertukaran node jika
    diperlukan mulai dari posisi
    karakter sampai ke root
END
```

Kompleksitas untuk proses *update tree* di atas adalah sebesar O(n) dengan n adalah besar variasi karakter yang terbaca. Dari uraian di atas dapat disimpulkan bahwa untuk kompresi menggunakan algoritma Huffman Adaptif memiliki kompleksitas sebesar O(nm), dengan nilai n adalah banyaknya karakter dan m adalah besarnya variasi karakter.

Percobaan Artikel Editorial

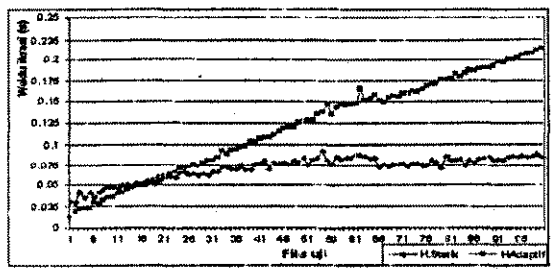
Tujuan dari percobaan artikel ini adalah untuk mendapatkan rasio dan lamanya waktu eksekusi dari kedua algoritma pada tiga buah rentang *file* potongan artikel yaitu percobaan pada rentang *file* < 20.000 *byte*, rentang *file* antara 20.000 sampai dengan 40.000 *byte*, dan percobaan pada rentang *file* > 40.000 *byte*. Statistik rata-rata lamanya iterasi kompresi, rasio kompresi, dan lamanya iterasi dekompresi yang dilakukan oleh masing-masing algoritma dapat dirangkum pada Tabel 1. Dapat dilihat bahwa waktu iterasi untuk kompresi dan dekompresi Huffman Statik lebih cepat dibandingkan waktu iterasi kompresi dan dekompresi Huffman Adaptif. Namun rasio pemampatan Huffman Adaptif lebih besar dibandingkan rasio pemampatan Huffman Statik. Hal ini menjadikan *file* kompresi yang dihasilkan oleh Huffman Adaptif akan lebih kecil ukurannya dibandingkan dengan *file* kompresi yang dihasilkan oleh Huffman Statik. Selain itu dapat terlihat bahwa semakin besar ukuran *file* yang akan dikompresi berimplikasi pada lama iterasi yang semakin lama.

Tabel 1. Rata-rata lama dan rasio kompresi *file* artikel

	Ukuran *file*	Kompresi		Dekompresi	
		Lama(s)	Rasio(%)	Lama(s)	Rasio(%)
Huffman Statik	< 20 KB	0,0564441	42,6445	0,03934281	-74,363
	20 s/d 40 KB	0,0656619	43,1081	0,04878656	-75,777
	> 40 KB	0,0763654	43,2218	0,05559875	-76,13
Huffman Adaptif	< 20 KB	0,1235163	43,3365	0,11193219	-76,493
	20 s/d 40 KB	0,2234394	43,4659	0,18995875	-76,689
	> 40 KB	0,3226191	43,4665	0,26700437	-76,893

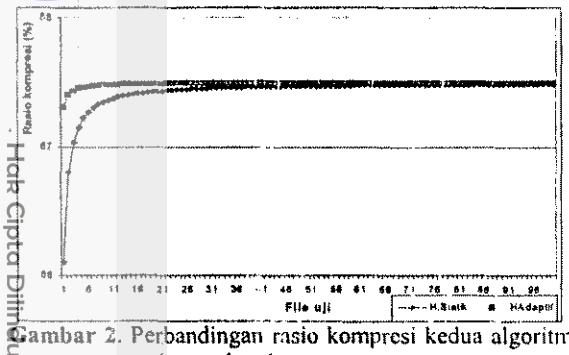
Percobaan Satu Variasi Karakter

Tujuan dari percobaan ini untuk mendapatkan rasio dan lamanya waktu eksekusi dari kedua algoritma untuk *file* yang memiliki karakter dengan peluang kemunculan adalah satu Hasil iterasi kedua algoritma pada saat ditampilkan akan dibedakan dengan warna, pada hasil iterasi algoritma Huffman Statik ditunjukkan dengan warna hitam dan untuk hasil algoritma Huffman Adaptif ditunjukkan dengan warna abu tua. Dapat dilihat dari Gambar 1 dan 3 bahwa lama iterasi untuk kompresi dan dekompresi Huffman Statik untuk satu variasi karakter cenderung lebih cepat dibandingkan dengan Huffman Adaptif, walaupun pada rentang *file* 1.000 *byte* sampai 20.000 *byte* masih terjadi fluktuasi disebabkan faktor ukuran *file* yang terlalu kecil dan iterasi yang cukup cepat. Pada Gambar 2 dapat terlihat bahwa rasio kompresi Huffman Adaptif lebih baik dibandingkan rasio kompresi Huffman Statik, dengan kecenderungan semakin stabil seiring besarnya ukuran *file*.

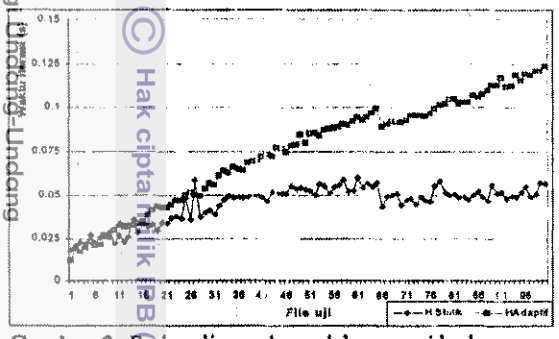


Gambar 1. Perbandingan lama kompresi pada satu karakter.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University



Gambar 2. Perbandingan rasio kompresi kedua algoritma pada satu karakter.



Gambar 3. Perbandingan lama dekompresi kedua algoritma pada satu karakter.

Percobaan Lima dan 256 Variasi Karakter

Tujuan dari percobaan ini adalah mengamati rasio dan lamanya eksekusi algoritma pada *file* dengan lima buah variasi karakter yang berbeda, dan 256 buah variasi karakter yang berbeda. Pada kompresi *file* yang berisi lima karakter terjadi suatu anomali dapat dilihat pada Tabel 2, menunjukan hasil kompresi yang seharusnya mengecil menjadi membesar. Pada algoritma Huffman Statik disebabkan oleh penambahan karakter diawal *file*, berupa *header* terkompresi tidaknya *file* (4 *byte*), 1 *byte* kode CRC, 4 *byte* untuk menyimpan ukuran *file* sumber, dan 2 *byte* untuk menyimpan banyaknya karakter. Ditambah kebutuhan untuk menyimpan *tree* berupa 2 *byte* untuk masing-masing karakter, disambung representasi *tree* dan representasi hasil kompresi. Sedangkan pada algoritma Huffman Adaptif mengalami pembesaran ukuran *file* karena selain menyimpan setiap variasi karakter yang muncul, akan disimpan pula hasil pengkodean berdasarkan *tree* . Karena *tree* yang dibentuk pada Huffman Adaptif tidak perlu disimpan, menjadikan rasio kompresi Huffman Adaptif lebih baik dibandingkan rasio kompresi Huffman Statik.

Tabel 2. Rata-rata lama dan rasio kompresi *file* berisi lima buah karakter (abcde = 5 *byte*)

Huffman Statik	Lama iterasi (detik)	0,01328
	Rasio pampat (%)	-400
	Ukuran hasil (*byte*)	25
Huffman Adaptif	Lama iterasi (detik)	0,008969
	Rasio pampat (%)	-40
	Ukuran hasil (*byte*)	7

Pada percobaan dengan *file* berisi 256 karakter terjadi pula anomali seperti halnya yang terjadi pada percobaan 5 karakter. Seperti yang dapat dilihat pada Tabel 3, rasio kompresi kedua algoritma menunjukan nilai negatif yang berarti hasil kompresi mengalami pembengkakan ukuran. Pada algoritma Huffman Statik lebih disebabkan karena besarnya tempat yang diperlukan untuk menyimpan statistik karakter.

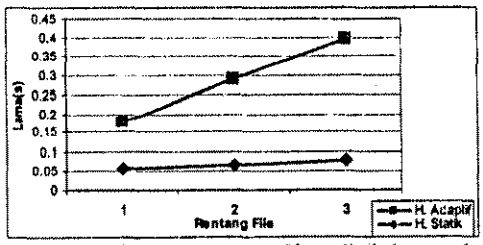
Karena pada percobaan ini digunakan 256 karakter yang berbeda, maka secara automatis akan diperlukan minimal 512 *byte* untuk menyimpan informasi karakter. Hal ini belum termasuk penyimpanan *header*, dan penyimpanan hasil pengkodean *file* sumber. Sedangkan pada algoritma Huffman Statik selain menyimpan 256 *byte* yang merepresentasikan karakter yang berbeda, perlu juga disimpan hasil pengkodean *file* sumber.

Tabel 3. Rata-rata lama dan rasio kompresi *file* berisi 256 buah karakter berbeda

Huffman Statik	Lama iterasi (detik)	0,025
	Rasio pampat (%)	-304,297
	Ukuran hasil (*byte*)	1035
Huffman Adaptif	Lama iterasi (detik)	0,011344
	Rasio pampat (%)	-100,781
	Ukuran hasil (*byte*)	514

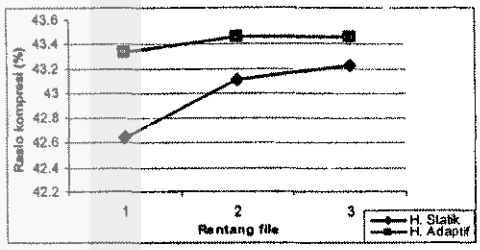
Pengujian Menggunakan ANOVA

Tujuan dari pengujian ini adalah untuk membantu dalam penarikan kesimpulan akhir dari data-data hasil percobaan dengan menggunakan pendekatan ragam. Kriteria yang diuji diantaranya, lama iterasi kompresi, rasio kompresi, dan lama iterasi dekompresi kedua algoritma. Adapun percobaan yang akan diujikan dengan Anova ini hanyalah percobaan dengan menggunakan potongan artikel. Hal ini dikarenakan representasi karakter yang ada di dalam *file* artikel tersebut biasa digunakan sehari-hari. Dari data pada Tabel 1, dapat ditarik suatu hubungan antara rentang *file* dengan algoritma pada lamanya iterasi kompresi, rasio kompresi dan lamanya iterasi dekompresi dengan menggunakan analisis secara deskriptif seperti pada Gambar 4 sampai Gambar 6.

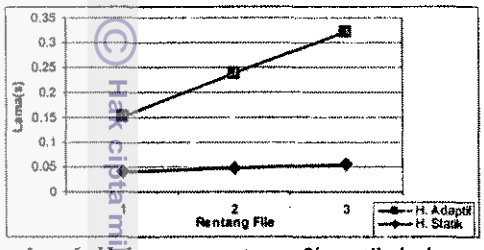


Gambar 4. Hubungan rentang *file* artikel dengan lamanya iterasi kompresi.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.



Gambar 5. Hubungan rentang *file* artikel dengan rasio kompresi.



Gambar 6. Hubungan rentang *file* artikel dengan lama iterasi dekompresi.

Dari Gambar 5 dan 6 dapat disimpulkan bahwa lamanya iterasi yang dilakukan oleh algoritma Huffman Adaptif sangat dipengaruhi oleh besar ukuran *file* (rentang *file*). Hal ini berbeda sekali jika dibandingkan dengan algoritma Huffman Statik yang relatif stabil. Dari ketiga gambar di atas dapat ditarik kesimpulan sebagai berikut:

- Iterasi kompresi Huffman Statik lebih cepat 2,19 kali pada rentang 1, lebih cepat 3,4 kali pada rentang 2, dan lebih cepat 4,22 kali pada rentang 3 jika dibandingkan dengan iterasi kompresi Huffman Adaptif.
- Rasio kompresi Huffman Adaptif lebih tinggi 0,69% pada rentang 1, lebih tinggi 0,36% pada rentang 2, dan lebih tinggi 0,24% pada rentang 3 jika dibandingkan dengan rasio kompresi Huffman Statik.
- Iterasi dekompresi Huffman Statik lebih cepat 2,84 kali pada rentang 1, lebih cepat 3,89 kali pada rentang 2, dan lebih cepat 4,8 kali pada rentang 3 jika dibandingkan dengan iterasi kompresi Huffman Adaptif.

Dari perhitungan ANOVA untuk lama iterasi kompresi, rasio kompresi dan lama iterasi dekompresi pada tiga buah rentang *file* artikel didapatkan nilai P yang berada dibawah taraf nyata ($\sigma <= 0,05$ pada selang kepercayaan 95%), sehingga dapat disimpulkan bahwa lamanya waktu proses dan besarnya rasio kompresi dipengaruhi oleh rentang *file* yang ada dan algoritma kompresi yang digunakan. Dari tabel ANOVA tersebut pula dapat disimpulkan bahwa perbedaan waktu dan rasio kompresi pada kedua algoritma berbeda nyata (P-value <= 0,05).

KESIMPULAN

- Semakin variatif karakter yang muncul, akan memperkecil rasio kompresi yang dihasilkan, baik oleh algoritma Huffman Statik, maupun pada algoritma Huffman Adaptif.
- Rasio kompresi terbaik terjadi pada *file* yang memiliki karakter dengan peluang kemunculan mendekati nilai satu (bobot karakter hampir sebesar ukuran *file*). Rasio terbaik yang didapat selama penelitian yaitu sebesar 87,489%.
- Rasio kompresi terburuk terjadi pada *file* yang memiliki variasi karakter besar dan pada *file* yang berukuran kecil. Hal ini ditandai dengan terjadinya pembesaran ukuran *file* hasil dibandingkan ukuran *file* awal.
- Dengan menggunakan metode kompresi apapun memiliki *trade off*, pengguna akan diberi pilihan hasil maksimum dengan iterasi yang lama atau iterasi yang cepat namun dengan hasil yang biasa.
- Hasil percobaan yang terdapat pada pembahasan menunjukan bahwa waktu iterasi yang diperlukan oleh algoritma Huffman Statik untuk melakukan kompresi dan dekompresi adalah cenderung lebih kecil dibandingkan dengan yang dilakukan oleh algoritma Huffman Adaptif. Namun untuk hasil kompresi terlihat unjuk kerja Huffman Adaptif adalah lebih baik dibandingkan Huffman Statik.

SARAN

Pada saat mengimplementasikan algoritma Huffman Adaptif, penulis menggunakan prosedur yang menjadikan kompleksitas algoritma Huffman Adaptif menjadi O(n.m). Seharusnya kompleksitas Huffman Adaptif, hampir sama dengan algoritma Huffman Statis yaitu O(n lg m). Pengembangan selanjutnya untuk perbandingan kompresi Huffman Statik dan Huffman Adaptif ini dapat diterapkan pada mekanisme pengiriman paket data terkompresi pada media jaringan.

DAFTAR PUSTAKA

[1] Cormen, Thomas H., C. E. Leiserson., & R. L. Rivest. 1990. *Introduction to Algorithms*. Mc Graw Hill Company.

[2] Huffman, D. A. 1952. *A Method for the Construction of Minimum-Redundancy Codes**. Proc. IRE, vol. 40, pp. 1098–1101, Sept. 1952.

[3] Hutasoit, Yenny E. 2001. *Huffman Coding Untuk Kompresi Data Teks Berbahasa Indonesia*. Skripsi. Jurusan Ilmu Komputer Fakultas MIPA IPB. Bogor, Indonesia.

[4] Layungsari. 2003. *Penyempurnaan dan Implementasi Software Kompresi Multi Tahap Menggunakan Huffman Coding*. Skripsi. Jurusan Ilmu Komputer Fakultas MIPA IPB. Bogor, Indonesia.

[5] Lelewer, D. A. & Hirschberg, D. S. 1987. *Data Compression*. ACM Computing Surveys 19(1987) 261-296.

[6] Moore, David S. 1994. *The Basic Practice of Statistics*. ISBN 0-7167-2628-9. W.H. Freeman and Company. New York.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)

Bogor Agricultural University

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hak cipta milik IPB (Institut Pertanian Bogor)
Intellectual Property Rights belong to Bogor Agricultural University

PENGINDEKSAN OTOMATIS DENGAN ISTILAH TUNGGAL UNTUK DOKUMEN BERBAHASA INDONESIA

Ahmad Ridha*, Ir. Julio Adisantoso, M.Komp.**, Ir. Fahren Bukhari, M.Sc.***
Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor
Jl. Raya Pajajaran, Bogor 16127, Indonesia
*ridha@ilkom.fmipa.ipb.ac.id
**julio@bima.ipb.ac.id
***fahren@ciam.or.id

ABSTRAK

Pengindeksan memiliki peranan penting dalam sistem temukembali informasi untuk menyediakan pemrosesan kueri yang lebih cepat dan daftar terurut dokumen hasil temu-kembali. Tujuan penelitian ini adalah mengembangkan metode pengindeksan dengan istilah tunggal untuk dokumen-dokumen dalam Bahasa Indonesia. Tercakup di dalamnya proses *parsing*, *stemming*, dan pembentukan *inverted index* dengan pembobotan istilah menggunakan fungsi *tf·idf*. Pengujian menggunakan 422 dokumen dan sepuluh kueri (yang masing-masing diungkapkan dalam tiga bentuk) dan diukur dengan 10-pt *average precision*. Ternyata penggunaan daftar kata buang dan *stemming* tidak berpengaruh signifikan terhadap kinerja temu-kembali. Akan tetapi daftar kata buang dan *stemming* bermanfaat dalam sistem temu-kembali untuk mengurangi ukuran indeks yang akan menjadikan operasi pencarian lebih efisien. Penggunaan daftar kata buang, *stemming*, dan keduanya menghasilkan penurunan lebih dari 25%, 10%, dan 30% masing-masingnya.

Kata kunci: sistem temu kembali informasi, indeks, *stemming*.

1. PENDAHULUAN

Penyimpanan dokumen secara digital berkembang dengan pesat seiring meningkatnya penggunaan komputer. Kondisi tersebut memunculkan masalah untuk mengakses informasi yang diinginkan secara akurat dan cepat sehingga pencarian terhadap seluruh isi dokumen yang tersimpan bukanlah solusi yang tepat mengingat pertumbuhan ukuran data yang tersimpan umumnya sangat tinggi. Dalam pencarian informasi di Internet para pengguna di Inggris menjadi frustrasi dalam dua belas menit jika tidak menemukan yang diinginkannya [1].

Untuk memenuhi tuntutan tersebut dibutuhkan *information retrieval system* (IRS) yang menggunakan suatu pengganti yang dapat merepresentasikan kumpulan dokumen dalam bentuk dan ukuran yang lebih mudah untuk pencarian. Struktur data yang populer dan telah lama digunakan untuk keperluan tersebut adalah sebuah indeks, yakni gugus kata atau konsep terpilih sebagai penunjuk ke informasi (atau dokumen) terkait. Indeks, dalam berbagai bentuk, merupakan inti setiap IRS modern karena menyediakan akses yang lebih cepat ke data dan juga mempercepat pemrosesan kueri [2].

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan pengindeksan dengan istilah tunggal untuk digunakan dalam IRS untuk dokumen-dokumen teks berbahasa Indonesia.

Penelitian ini terbatas pada pemrosesan dokumen teks berbahasa Indonesia menjadi indeks istilah tunggal meliputi proses *parsing* serta *stemming*. Struktur data serta metode penyimpanan indeks tidak termasuk dalam penelitian ini.

2. PENGINDEKSAN OTOMATIS DENGAN ISTILAH TUNGGAL

Proses pengindeksan dilakukan dalam struktur *inverted index* dan menggunakan pembobotan *tf·idf*. Langkah-langkah dalam pengindeksan adalah sebagai berikut:

1. *inverted index* dikosongkan
2. dokumen diproses hingga menjadi *document terms*
3. untuk tiap *stem* pada *document terms*, tambahkan *posting list node* pada *posting list* yang bersesuaian dalam kamus istilah.
4. simpan informasi panjang dokumen (jumlah kata) pada kamus dokumen
5. proses dokumen berikutnya hingga seluruh koleksi telah ditambahkan pada indeks
6. lakukan pembobotan untuk seluruh isi kamus istilah dan hitung faktor normalisasi tiap dokumen untuk pembobotan *tf·idf*.

simpan faktor normalisasi untuk setiap dokumen dalam kamus dokumen.

Untuk penambahan dokumen tunggal dilakukan langkah 2-7 karena terjadi perubahan frekuensi istilah dan dengan sendirinya perubahan bobot istilah dan faktor normalisasi secara keseluruhan.

2.1 *Tokenizer*

Tokenizer menerima masukan berupa rangkaian karakter dan memilahnya menjadi *token* dengan aturan sebagai berikut:

- Suatu *token* dimulai oleh huruf atau angka
- *Token* dipisahkan oleh karakter *whitespace*

Karakter-karakter khusus yang mengikuti huruf atau angka dianggap bagian dari *token* (misalnya tanda persen dalam 125%) namun dianggap sebagai pemisah *token* jika tidak.

2.2 *Stemming*

Stemming merupakan bagian yang sangat memerlukan pengetahuan bahasa karena penentuan *stem* suatu kata berbeda tergantung tata bahasa yang digunakan. Oleh karena itu perlu dikembangkan algoritme *stemming* tersendiri untuk Bahasa Indonesia.

Sistem pemotong sufiks untuk Bahasa Indonesia yang berdasarkan algoritme Porter [3] telah dikembangkan dalam [4] namun pengindeksan memerlukan sistem yang mampu memotong prefiks dan sufiks yang banyak digunakan. Infiks tidak dihilangkan karena prosesnya lebih kompleks dan tidak lagi produktif dalam Bahasa Indonesia.

Sebagaimana algoritme Porter, digunakan suatu fungsi penghitung ukuran kata untuk mencegah *stemming* menghasilkan *stem* yang terlalu pendek. Diasumsikan minimal *stem* hasil berukuran dua kecuali jika *token* berukuran kurang dari dua.

Akan tetapi fungsi ukuran kata pada algoritme Porter tidak dapat digunakan pada Bahasa Indonesia. Sebagaimana dalam [4], jumlah vokal dalam kata akan digunakan sebagai penentu ukuran kata kecuali kata-kata tanpa vokal yang terdiri dari tiga karakter atau lebih dianggap memiliki ukuran dua untuk mengakomodasi singkatan yang hanya terdiri dari konsonan.

Vokal didefinisikan sebagai huruf-huruf A, I, U, E, dan O. Huruf-huruf selain itu merupakan konsonan.

Aturan pemotongan diungkapkan sebagai:

P1 (kondisi) S1→ P2 S2

yang artinya jika sebuah kata berprefiks P1 dan bersufiks S1, dan bagian kata setelah P1 dan sebelum S1 memenuhi kondisi yang diberikan, maka P1 dan S1 akan diganti dengan P2 dan S2.

Kondisi dapat menggunakan operator *AND*, *OR*, atau *NOT* untuk menyatakan aturan yang kompleks.

Beberapa notasi juga digunakan untuk membantu, yakni:

- *W*, seluruh kata termasuk P1 dan S1
- *M*, ukuran kata
- *L*, jumlah karakter dalam kata
- *V*, huruf vokal
- *C*, huruf konsonan
- *V**, kata diawali vokal
- *C**, kata diawali konsonan
- **CC*, kata diakhiri konsonan ganda
- *x**, kata diawali huruf atau kumpulan huruf *x*
- **x*, kata diakhiri huruf atau kumpulan huruf *x*
- *V(x)*, huruf ke-*x* adalah vokal
- *C(x)*, huruf ke-*x* adalah konsonan

Sebagai contoh, dalam aturan:

(*M* > 1) *wan* →

S1 adalah *wan* dan S2 adalah *null* (kata kosong). Sehingga kata *dermawan* dipotong menjadi *derma* karena *derma* berukuran 2 (*M* > 1).

Stemming dilakukan terhadap elemen-elemen berikut:

- prefiks: *meng-*, *di-*, *per-*, *ber-*, *ter-*, *peng-*, *pe-*, *per-*, *se-*
- sufiks: *-an*, *-kan*, *-i*, *-nya*, *-ik*, *-is*, *-if*, *-al*, *-(is)asi*, *-at*, *-iah*, *-wi*, *-wiah*, *-isme*, *-sionis*
- konfiks: *ke-an*, *ke-i*
- partikel: *-kah*, *-lah*
- kata ganti: *ku-*, *kau-*, *-mu*, *-nya*

Walaupun partikel dan kata ganti tidak termasuk afiks namun diperlakukan sama sehingga partikel dianggap sebagai sufiks dan kata ganti dianggap sebagai prefiks atau sufiks sesuai posisinya.

Selanjutnya walaupun *stemming* umumnya hanya digunakan untuk memotong afiks suatu kata namun dalam sistem ini fungsi serupa juga diterapkan pada angka. *Token* berupa angka dikelompokkan ke dalam bentuk yang lebih umum misalnya 800.000 dan 796.352 menjadi bentuk yang sama yakni 800000.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

Hak Cipta Dilindungi Undang-Undang
Penggunaan tesis ini tanpa mencantumkan sumber

2.4. Koleksi Pengujian

Koleksi pengujian menggunakan artikel-artikel utama, nasional (politik dan keamanan serta umm pada Media Indonesia) dan internasional harian Kompas, harian Media Indonesia, dan harian Koran Tempo (http://www.kompas.com, http://www.media-indonesia.com, dan http://www.tempo.co.id) terbitan 6 April 2002 dan 8-12 April 2002. Terbitan tanggal 7 April 2002 tidak disertakan karena merupakan harian Minggu yang memiliki susunan berita berbeda. Tiga media massa digunakan untuk menguji sistem pada dokumen-dokumen yang memiliki topik serupa namun dengan penyampaian yang berbeda-beda.

Tabel 1. Jumlah dokumen yang digunakan berdasarkan tanggal.

Sumber	Tanggal						Σ
	6*	8	9	10	11	12	
Kompas	31	23	23	24	28	27	156
Koran Tempo	21	28	18	24	18	16	125
Media Indonesia	25	25	23	21	24	23	141
Total	77	76	64	69	70	66	422

April 2002

Untuk pengindeksan, dokumen-dokumen tersebut diubah susunannya menjadi terdiri dari:

1. judul (satu baris)
2. isi dokumen.

Sedangkan isi dokumen hanya mengalami perubahan dalam penggantian tanda *ampersand* (&) menjadi kata *dan* serta penggantian karakter dengan kode ASCII 173 menjadi tanda hubung (-). Kesalahan ejaan dan kesalahan tata bahasa tidak diperbaiki.

2.5. Evaluasi Sistem

Evaluasi pengindeksan otomatis dilakukan dengan menentukan kinerjanya dalam *recall* dan *precision*. Hal ini dilakukan dengan menggunakan koleksi pengujian beserta gugus kueri dan penilaian relevansinya (gugus jawaban) [5]. Dari hasil evaluasi tersebut dapat diperoleh nilai *average precision* (AVP).

Relevansi dokumen umumnya ditentukan oleh manusia, sebaiknya oleh orang yang sama yang memberikan kueri. Walaupun penilaian relevansi tersebut akan berbeda-beda bagi pemeriksa yang berbeda namun [6] menunjukkan bahwa kurva *recall* dan *precision* yang dihasilkan hampir identik. Metode serupa juga digunakan dalam TREC [7].

Sistem mengembalikan daftar dokumen terurut menurun berdasarkan besar hasil fungsi kesamaan kueri dan dokumen.

Selanjutnya dari hasil kueri dihitung banyak dokumen yang diperoleh untuk mencapai tingkat *recall* tertentu dan selanjutnya nilai *precision* dihitung. Tingkat *recall* yang digunakan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan 1.0 untuk menghitung AVP.

Untuk melihat pengaruh penggunaan *stemming* dan daftar kata buang dilakukan empat kali pengindeksan yakni:

1. IDX_A: menggunakan *stemming* dan daftar kata buang
2. IDX_B: hanya menggunakan *stemming*
3. IDX_C: hanya menggunakan daftar kata buang
4. IDX_D: tidak menggunakan *stemming* dan daftar kata buang.

Untuk membandingkan algoritme *stemming*, juga dilakukan pengindeksan $IDX_{A'}$ yang menggunakan *stemming* sufiks dalam [4] dan daftar kata buang.

Selanjutnya beberapa kueri beserta gugus jawaban dibuat berdasarkan koleksi pengujian. Kueri yang sama diungkapkan dalam bentuk-bentuk berikut:

1. Bentuk menengah, mengungkapkan topik yang diinginkan dalam kalimat
2. Bentuk pendek, hanya terdiri dari sebanyak-banyaknya lima kata yang merupakan inti dari bentuk menengah
3. Bentuk panjang, paragraf yang diambil dari salah satu dokumen yang relevan.

Pengujian dilakukan dengan uji *t* berpasangan dengan selang kepercayaan 95% terhadap:

1. IDX_A - IDX_C dan IDX_B - IDX_D: untuk pengaruh *stemming*
2. IDX_A - IDX_B dan IDX_C - IDX_D: untuk pengaruh daftar kata buang
3. IDX_A - IDX_D: untuk pengaruh keduanya
4. IDX_A - $IDX_{A'}$: untuk pengaruh perbedaan algoritme *stemming*

Selain kemampuan tenu-kembali, juga dilihat ukuran indeks yang dihasilkan. Digunakan dua jenis ukuran yakni yang menyertakan keterangan posisi suatu istilah (S_P) dan yang tidak (S_{NP}).

2.6. Asumsi-asumsi

Asumsi-asumsi yang digunakan dalam penelitian ini adalah sebagai berikut:

1. indeks dapat termuat dalam memori utama
2. dokumen dan teks kueri menggunakan ASCII *character set*.

© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

Hak Cipta Dilindungi Undang-Undang

© Perpustakaan Pertanian Bogor (Bogor Agricultural University)

seluruh koleksi telah terindeks sebelum digunakan oleh untuk pemrosesan kueri.

3. HASIL EKSPERIMEN

3.1. Algoritme *Tokenizer*

Teks masukan diproses secara sekuensial per karakter dari awal dan menghasilkan sebuah *token* serta posisinya atau keterangan bahwa teks telah selesai diproses. Algoritme yang digunakan sebagai berikut:

Jika sudah mencapai akhir teks maka proses berakhir selainnya karakter yang diperoleh dibandingkan terhadap tabel jenis karakter. Sebuah karakter dapat memiliki salah satu di antara tiga jenis berikut:

a. *whitespace*, berarti karakter ini merupakan karakter pemisah *token*
b. *alphanumeric*, berarti karakter ini merupakan huruf atau angka
c. *other*, berarti karakter ini tidak termasuk jenis-jenis di atas.

Jika karakter yang ditemukan merupakan huruf atau angka maka karakter berikut menjadi karakter pertama dari *token* sedangkan selainnya kembali ke langkah pertama.

Karakter-karakter selanjutnya menjadi bagian dari *token* hingga ditemukan karakter *whitespace* atau akhir dari teks.

3.2. Masalah-masalah Afiksasi

Proses afiksasi terutama prefiks dalam Bahasa Indonesia mengalami proses morfofonemik sesuai dengan fonem yang mengikutinya. Misalnya prefiks *meng-* dapat menjadi *meng-*, *me-*, *men-*, *mem-*, *meny-* atau *menge-*. Selain itu beberapa fonem juga mengalami peluluhan. Hal ini menyulitkan proses *stemming* karena satu bentuk perubahan dapat ditimbulkan oleh beberapa fonem, sehingga pemotongan afiks secara sempurna tanpa pengetahuan kata yang lengkap sangat sulit dilakukan.

Untuk mengatasi masalah di atas, dilakukan perubahan-perubahan prefiks tertentu terhadap *stem* sebagai berikut:

1. $ny\,(M > 0) \rightarrow s$
2. $V\,(M > 0) \rightarrow ng$
3. $k\,(M > 0) \rightarrow ng$
4. $p\,(M > 0) \rightarrow m$
5. $t\,(M > 0) \rightarrow n$

Selanjutnya masalah afiksasi pada sufiks *-an* dan *-kan*. Algoritme *stemming* yang dikembangkan berdasarkan algoritme Porter yang bersifat *iterative longest match* sehingga dapat terjadi kesalahan pemotongan seperti contoh berikut:

hentakan → *henta* + *-kan*

seharusnya

hentakan → *hentak* + *-an*

Sebagaimana masalah proses morfofonemik di atas, dilakukan perubahan pada akhir *stem* yakni:

$(M > 1)\ k \rightarrow$

sehingga bentuk-bentuk *hentak*, *hentakan*, dan *hentakkan* menghasilkan *stem* yang sama yakni *henta*.

Walaupun cara-cara di atas dapat dikatakan *merusak* bentuk *stem* namun karena tujuan *stemming* di sini adalah untuk sedapat mungkin menjadikan bentuk-bentuk kata yang diturunkan dari kata dasar yang sama ke dalam satu *stem* dan hasil *stemming* ditujukan untuk sistem (tidak untuk pengguna) maka bentuk yang *aneh* tersebut dapat digunakan. Asumsinya adalah tidak banyak kata-kata yang memiliki bentuk yang sama dengan hasil perubahan. Kesalahan terjadi dalam kasus seperti berikut:

pakan → *makan*
galak → *gala*

padahal terdapat kata *makan* dan *gala* yang memiliki makna yang berbeda dengan *pakan* dan *gala*.

Selain itu masih ada kasus yang belum tertangani yakni proses morfofonemik pada prefiks *ber-*, *ter-*, dan *per-* terhadap fonem vokal dan */r/* sehingga misalnya bentuk-bentuk *merebut*, dan *terebut*, menghasilkan *stem* yang berbeda yakni *rebut* dan *ebut* (atau *ngebut* dengan perubahan di atas).

3.3. Algoritme *Stemming*

Beberapa fungsi pendukung yang digunakan dalam *stemming* antara lain:

1. *Valid* (*x*), memeriksa kevalidan suatu *token* masukan untuk diproses lebih lanjut atau kevalidan suatu *stem* untuk digunakan. *Token* atau *stem* *x* valid jika memiliki:

- panjang lebih dari dua karakter dan memiliki huruf (misalnya *IBM* dan *M16*), atau
- panjang lebih dari tiga karakter dan tidak memiliki huruf namun memiliki angka (misalnya *2002* dan *1987-1992*).

ReduceRep (*x*), menangani kata *x* yang memiliki tanda ulang (-) dengan cara:

- jika kata sebelum tanda ulang tersebut termasuk morfem terikat tertentu.
- jika ukuran kata sebelum tanda ulang lebih dari satu atau kata sebelum tanda ulang sama dengan kata setelah tanda ulang maka yang digunakan hanyalah bagian sebelum tanda ulang (misalnya *lalu-lalang* → *lalu*, *hak-hak* → *hak*).
- jika tidak termasuk dalam kedua bentuk di atas maka tanda ulang dipertahankan (misalnya *F-15* → *F-15*).

ValidDblConsonant (*x*), memeriksa kevalidan konsonan ganda yang mengawali kata *x*. Dalam Bahasa Indonesia konsonan ganda yang mengawali suatu suku kata terbatas pada *pl*, *bl*, *kl*, *gl*, *fl*, *sl*, *pr*, *br*, *tr*, *dr*, *kr*, *gr*, *fr*, *sr*, *ps*, *sw*, *kw*, *sp*, *sm*, *sn*, *sk*, *pt*, *ts*, *st*, *ng*, *ny*, *str*, *spr*, *skr*, dan *skl* (Alwi *et al*, 1998).

AdjustHead (*x*) dan *AdjustTail* (*x*), mengubah huruf awal dan akhir dari *stem x* untuk menangani masalah-masalah afiksasi.

Right(*x*, *y*), mengembalikan *y* karakter paling kanan dari kata *x*

Left(*x*, *y*), mengembalikan *y* karakter paling kiri dari kata *x*

Mid(*x*, *y*, *z*), mengembalikan *z* karakter dari kata *x* mulai posisi *y*. Jika *z* tidak diberikan maka semua karakter dari *y* hingga akhir dikembalikan.

StripPrefix (*x*, *y*), melakukan pemotongan prefiks terhadap kata *x* pada posisi *y* dengan aturan:

- jika pada posisi *y* terdapat tanda ulang (-) maka dilakukan pemotongan *y* karakter, selainnya
- jika (*V*(*y*) *OR* (*C*(*y*) *AND* *V*(*y*+1)) *OR* *ValidDblConsonant* (*Mid*(*x*, *y*))) maka dilakukan pemotongan (*y*-1) karakter, selainnya
- tidak dilakukan pemotongan terhadap *x*

Aturan-aturan *stemming* yang digunakan adalah sebagai berikut:

1. PreS1
   se (*M* > 1) →
2. PreS2
   mem (*M* > 1 *AND* (*b** *OR* *p** *OR* *f**)) →
   mem (*M* > 1) → *m*
   meng (*M* > 1 *AND* (*g** *OR* *h** *OR* *kh**)) →
   meny (*M* > 1 *AND* *V**) → *s*
   men (*M* > 1 *AND* *V**) → *n*
   men (*M* > 1) →
   me (*M* > 1) → *StripPrefix*(*W*, 3)

   di (*M* > 1) → *StripPrefix*(*W*, 3)
3. PreS3
   ber (*M* > 1) →
   be (*M* > 1 *AND* *Cer**) →
4. PreS4
   pem (*M* > 1 *AND* *b**) →
   peng (*M* > 1 *AND* (*g** *OR* *h** *OR* *kh**)) →
   peny (*M* > 1 *AND* *V**) → *s*
   pen (*M* > 1 *AND* *V**) → *n*
   pen (*M* > 1) →
   per (*M* > 1 *AND* *C**) → *StripPrefix*(*W*, 3)
   pe (*M* > 1) → *StripPrefix*(*W*, 3)

   ter (*M* > 1) →
   te (*M* > 1 *AND* *Cer**) →
5. SufS1
   (*seni OR budi*) *man* →
   (*M* > 1) *wan* →
   (*M* > 1) *wati* →
6. SufS2
   (*L* > 4) *-kan* →
   (*M* > 1) *kan* →
7. SufS3
   (*L* > 3) *-an* →
   (*M* > 1) *an* →
8. SufS4
   (*M* > 1 *AND NOT*(**i*) *AND*
   (**ng* *OR* *NOT* (**CC*))) *i* →
9. FSufS1
   (*M* > 1) *sionis* → *si*
   (*L* > 5) *-isme* → *is*
   (*M* > 0) *isme* → *is*
   (*M* > 1) *itas* →
   (*M* > 1) *asi* →
   (*M* > 1 *AND* **C*) *si* → *t*
   (*M* > 1) *or* →
   (*M* > 1) *er* →
10. FSufS2
    (*M* > 1) *if* →
    (*M* > 1) *ik* →
    (*M* > 1) *is* →

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik Perpustakaan Bogor)
Bogor Agricultural University

1. Dilarang menyebarluaskan dan mendistribusikan tulisan ini dalam bentuk apapun tanpa izin IPB.
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

© Hakcipta milik IPB (Institut Pertanian Bogor)

Bogor Agricultural University

FSuf3
(M > 1) at →
(M > 1) wi →
(M > 1) wiah →
(M > 1) iah →
(M > 1) al →

FSuf4
(M > 0 AND *V) v → f
(M > 0 AND *V) pt → p
(M > 0 AND *V) kt → k
(M > 0 AND *V) nt → n

ConS
Ke (M > 1) an →
Ke (Tahu) i →

ParS
(M > 1) –kah →
(M > 1) –lah →

ProS
(M > 1) -ku →
(M > 1) -mu →
(M > 1) ku →
(M > 1) mu →
(M > 1) -nya →
(M > 1) nya →
Ku (M > 1) →
Kau (M > 1) →

Dengan menggunakan fungsi-fungsi dan aturan-aturan tersebut *stemming* terhadap *token t* dilakukan dengan algoritme berikut:

1. *t* diubah menjadi *lower case*
2. *StripPunct*(*t*)
3. jika *t* diawali oleh tanda rupiah (*rp*) dan bersifat numerik maka *rp* dihilangkan
4. jika *Valid*(*t*) tidak benar maka proses berakhir
5. *ReduceRep*(*t*)
6. jika ukuran *t* kurang dari dua dan tidak mengandung tanda ulang maka proses berakhir
7. aturan-aturan *stemming* digunakan dengan urutan:
   - PreS1
   - ParS
   - ProS
   - ConS
   - PreS2
   - ConS
   - SufS1 – SufS3
   - PreS3 – PreS4
   - FSuf1 - FSuf4
   - SufS4
8. *AdjustHead*(*t*)
9. *AdjustTail*(*t*)
10. jika ada, hilangkan tanda ulang di sisi kiri *t*
11. *StripPunct*(*t*)
12. panjang *t* dibatasi maksimal 15 karakter dan jika lebih diambil 15 karakter paling kiri
13. jika *t* bersifat numerik maka diubah dengan pembulatan kemudian selain dua digit paling kiri diganti dengan nol
14. jika *Valid*(*t*) benar maka hasil stemming dikembalikan

3.4. Pemrosesan Dokumen dan Teks Kueri

Selain *tokenizer* dan *stemming*, digunakan fungsi *StripPunct* untuk membuang karakter-karakter tanda baca berikut dari sisi kanan: . , ? ! - : ;)] } > serta menghilangkan semua kemunculan tanda kutip satu (') dan tanda kutip dua (") dari *token*. Misalnya:

"Serang!" → *Serang*
Ma'ruf → *Maruf*

Baik dokumen maupun teks kueri diproses dengan langkah-langkah berikut:

1. dengan menggunakan *tokenizer*, *token* dikenali dan dimasukkan ke fungsi *StripPunct*.
2. *token* diubah ke *lower case*, jika *token* termasuk dalam daftar kata buang maka *token* tersebut dilewati dan langsung ke langkah 4. Jika tidak maka *token tersebut* mengalami *stemming*
3. *stem* dimasukkan dalam kamus istilah dan informasinya disesuaikan
4. jika teks belum berakhir kembali ke langkah 1.

Perbedaan pemrosesan dokumen dan teks kueri berada pada informasi yang disertakan dalam kamus istilah. Daftar istilah teks kueri hanya menyertakan informasi frekuensi istilah namun daftar istilah dokumen menyertakan informasi berupa *posting list node* (tetapi informasi bobot tentunya belum tersedia). Daftar istilah dokumen selanjutnya dirujuk sebagai *document terms* dan daftar istilah kueri disebut *query terms*.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.
Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University

3.5. Pemrosesan kueri

Pembobotan kueri menggunakan VSM dengan vektor dokumen ternormalisasi. Bobot istilah kueri w_{qj} diperoleh dengan cara serupa dengan pembobotan istilah dokumen namun frekuensi yang digunakan adalah frekuensi istilah j dalam kueri.

$$w_{qj} = tf_{qj} \cdot \log \frac{N}{df_j}$$

sehingga istilah yang tidak terdapat dalam kueri ($tf_{qj} = 0$) memiliki bobot nol.

3.6. Ukuran Indeks dan Jumlah Istilah

Deskripsi koleksi yang diproses dapat dilihat pada Tabel 2.

Tabel 2. Deskripsi koleksi pengujian.

Deskripsi	Total	Rata-rata
Ukuran	1.558.884	3.694,038
Total *token*	210.622	499,104

dalam *byte*

Tokenizer menghasilkan 16.442 *token* unik dengan frekuensi total sebesar 210.622. Penggunaan daftar kata buang mengurangi 250 *token*, yakni hanya 1,522% dari jumlah *token* unik, namun dengan frekuensi total sebesar 69.106, mencapai 32,81% dari frekuensi total.

Hal ini berdampak langsung terhadap S_P yang mengalami penurunan sebesar 25,80% ($IDX_B \rightarrow IDX_A$) dan 26,20% ($IDX_D \rightarrow IDX_C$) serta S_{NP} yang mengalami penurunan sebesar 21,27% ($IDX_B \rightarrow IDX_A$) dan 21,84% ($IDX_D \rightarrow IDX_C$).

Stemming turut mengurangi jumlah istilah indeks sebesar 41,671% (IDX_A) dan sebesar 41,68% (IDX_B) yang selanjutnya menurunkan ukuran indeks S_P sebesar 10,40% ($IDX_C \rightarrow IDX_A$) dan 10,89% ($IDX_D \rightarrow IDX_B$) serta S_{NP} yang mengalami penurunan sebesar 16,29% ($IDX_C \rightarrow IDX_A$) dan 16,89% ($IDX_D \rightarrow IDX_B$).

Sebagai perbandingan, $IDX_{A'}$ mengurangi jumlah istilah indeks sebesar 14,995% serta menurunkan S_P sebesar 4,625% ($IDX_C \rightarrow IDX_{A'}$) dan S_{NP} sebesar 6,249% ($IDX_C \rightarrow IDX_{A'}$). Hal ini menunjukkan tingginya penggunaan prefiks dalam afiksasi.

Secara keseluruhan, penggunaan daftar kata buang dan *stemming* ($IDX_D \rightarrow IDX_A$) menurunkan ukuran indeks S_P sebesar 33,88% dan S_{NP} sebesar 34,57%.

3.7. Kinerja Temu-Kembali

Kinerja temu-kembali masing-masing indeks dapat dilihat pada Tabel 3.

Tabel 3. Ringkasan kinerja temu-kembali.

Indeks	AVP
IDX_A	0,775
IDX_B	0,776
IDX_C	0,771
IDX_D	0,771
$IDX_{A'}$	0,773

Hasil uji menunjukkan bahwa tidak ditemukan perbedaan yang signifikan dalam kinerja temu-kembali. Perbedaan algoritme *stemming* ($IDX_A - IDX_{A'}$) juga tidak menghasilkan perbedaan kinerja yang signifikan.

Kinerjanya secara umum dapat dikatakan baik karena dengan AVP sekitar 0,77 berarti secara rata-rata pada tiap *recall point*, 77% hasil temu-kembali relevan dengan kueri. Akan tetapi hal ini dapat juga ditimbulkan oleh kecilnya ukuran koleksi sehingga memiliki *noise* yang rendah.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa:

1. Daftar kata buang dan *stemming* berperan untuk memperkecil ukuran indeks sehingga meningkatkan efisiensi operasi temu-kembali.
2. Penggunaan *stemming* prefiks dan sufiks penting bagi IRS Bahasa Indonesia karena tingginya penggunaan prefiks walaupun dari segi kinerja temu-kembali tidak signifikan.

Sistem ini dapat dikembangkan lebih lanjut untuk menjadikannya sebuah IRS yang lengkap untuk Bahasa Indonesia. Beberapa alternatif pengembangan antara lain:

1. penggunaan kompresi untuk memperkecil ruang penyimpanan serta mempercepat proses pencarian teks.
2. penggunaan *thesaurus* untuk mengelompokkan istilah-istilah yang berhubungan.
3. penggunaan frase-frase yang diturunkan dari istilah-istilah yang muncul bersamaan dalam koleksi contoh.
4. penggunaan teknik *relevance feedback* untuk penyesuaian bobot istilah.
5. penggunaan koleksi yang lebih besar untuk lebih mendekati penggunaan sesungguhnya.

penggunaan teknik kedua hingga keempat diperkirakan meningkatkan *precision* sebesar 10% - 20%, 5% - 10% dan 3% - 60% masing-masingnya [8].

REFERENSI

[1] Nua Internet Surveys, *Net users' patience only lasts 12 minutes*, http://www.nua.ie/surveys/index.cgi?f=VS&art_id=905356650&rel=true [7 Maret 2002]

[2] Baeza-Yates, R. and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.

[3] Porter, M.F., An Algorithm for Suffix Stripping. *Program*, 14(3), 1980, 130-137.

[4] Arifin, C.H., *Algoritme Pemotong Sufiks Baku untuk Kata dalam Bahasa Indonesia Berbasis Algoritme Porter*. Bogor; Jurusan Ilmu Komputer IPB, 2002.

[5] Lancaster, F. and A. Warner, *Information Retrieval Today*. Arlington; Information Resources Press, 1993.

[6] Salton, G., *Automatic Text Analysis*. Technical Report No. 69-36, Department of Computer Science. Ithaca; Cornell University, 1969.

[7] Vorhees, E.M. and D. Harman, Overview of TREC 2001. *Proc. of the 10th Text REtrieval Conference*, 2001.

[8] Salton, G, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

Hak Cipta Dilindungi Undang-Undang
© Hak cipta milik IPB (Institut Pertanian Bogor)
Bogor Agricultural University