PENERAPAN METODE HYBRID DARI TERM WEIGHTING DALAM MELAKUKAN PENCARIAN PASAL KITAB UNDANG-UNDANG HUKUM PIDANA DENGAN KASUS PIDANA UMUM

Ignasia Rosa W [1], Lucia D. Krisnawati[2], Wimmie Handiwidjojo [3]

Abstrak:

Pencarian pasal dalam KUHP membutuhkan waktu yang lama apabila dilakukan secara manual. Selain itu, warga sipil akan mengalami kesulitan untuk menemukan pasal yang dapat dikenakan kepada seorang tersangka pidana umum karena tidak semua orang mempelajari KUHP. Artikel ini akan membahas mengenai sistem pencarian pasal KUHP dengan menerapkan metode *hybrid* dari *Term Weighting*. Sistem ini akan melakukan pencarian berdasarkan kemunculan kata kunci dalam pasal kemudian hasil pencarian akan diurutkan berdasarkan bobotnya. Pasal yang memiliki bobot terbesar dianggap sebagai pasal yang relevan dengan kata kunci yang telah dimasukkan pengguna.

Kata Kunci: Term Weighting, KUHP, Information Retrieval, Local Weight, Global Weight, Normalization Factor.

1. Pendahuluan

Kitab Undang-undang Hukum Pidana (KUHP) digunakan dalam dunia peradilan untuk menentukan hukuman yang dapat dikenakan kepada seorang tersangka tindak pidana umum. Namun, tidak semua orang mengerti mengenai KUHP. Hal tersebut menjadi salah satu alasan dibuatnya sistem pencarian pasal KUHP ini.

Karya tulis ini akan membahas mengenai pembuatan sistem pencarian pasal KUHP berdasarkan kata kunci pengguna dengan menerapkan metode *hybrid* dari *term weighting*. Proses yang akan terjadi adalah proses pembobotan *term* baik *term* pada *query* maupun *term* pada database. Untuk proses pembobotan *term* pada database akan dilakukan dalam tiga tahap. Yaitu proses pembobotan lokal, proses pembobotan global, dan proses menghitung faktor normalisasinya. Dengan menggunakan nilai hasil dari ketiga proses tersebut, maka akan didapat hasil akhir berupa bobot total suatu *term* ada di dalam database. Sedangkan alur sistem dalam mengambil keputusan terhadap kata kunci masukan pengguna dilakukan dengan cara membandingkan bobot *term* pada *query* dan bobot *term* di database dalam rentang pencarian yang telah ditentukan. Pasal yang mengandung *term* yang memiliki bobot yang sama dengan bobot *term* pada *query* diasumsikan menjadi pasal yang relevan. Sehingga pasal tersebut ditampilkan menjadi keluaran dari sistem dan dianggap sesuai dengan kata kunci yang telah dimasukkan oleh pengguna sebelumnya.

Dengan dibuatnya sistem pencarian ini, diharapkan dapat membantu orang awam untuk dapat mencari dengan cepat pasal yang berkenaan dengan suatu tindak pidana umum. Selain itu, sistem ini diharapkan juga dapat membantu penyelenggara pengadilan untuk dapat mencari pasal-pasal KUHP dengan cepat.

[1] *Ignasia Rosa W , Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
Email : rignasia@yahoo.com
[2] *Lucia D. Krisnawati , Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*
[3] *Drs. Wimmie Handiwidjojo, MIT , Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

b. Perumusan Masalah

1) Apakah sistem dapat melakukan pencarian dalam database dengan menggunakan metode yang ada sesuai dengan masukan pengguna ?
2) Apakah pasal-pasal dalam KUHP dapat disimpan dalam database dan diklasifikasikan sesuai dengan klasifikasi dalam KUHP ?

c. Metode Penelitian

Tahapan Metode penulisan yang digunakan dalam menyelesaikan karya tulis ini adalah :

1) Pengumpulan Data

Studi pustaka mengenai KUHP dan bidang studi, serta pengumpulan informasi pasal KUHP dari file html kemudian dimasukkan ke dalam database dengan cara *copy paste*.

2) Implementasi Sistem

Metode yang akan digunakan adalah metode *Hybrid* dari *Term Weighting*. Proses yang akan dilakukan terbagi dalam tiga tahap :

a) Pra Pemrosesan

Isi dari buku kedua dan buku ketiga KUHP diklasifikasikan menjadi 40 kelas, kemudian disimpan dalam 1 tabel. Kemudian dilakukan proses normalisasi kata dan tokenisasi berdasarkan spasi dan token atau kata yang termasuk *stoplist* akan diabaikan. Hasil akhir pra-pemrosesan akan tersimpan dalam tabel bobot dan selanjutnya akan diproses sesuai dengan rumus yang ada. Sedangkan isi dari buku pertama KUHP, akan ditampilkan dalam bentuk teks dan tidak diikutsertakan dalam proses pencarian.

b) Pemrosesan

Pada tahap ini, sistem akan melakukan proses pembobotan *term* pada *query* dan *term* di database. Proses yang dilakukan untuk menghitung bobot *term* dalam database, dilakukan dalam tiga proses. Proses pembobotan lokal (*Local Weighting*), proses pembobotan global (*Global Weighting*), dan proses perhitungan faktor normalisasi (*Normalization Factor*). Pada akhir pemrosesan, sistem akan memutuskan pasal mana dianggap sesuai dengan masukan pengguna

c) Pascapemrosesan

Pada tahap ini, dilakukan evaluasi kinerja sistem dengan melakukan perhitungan nilai *precision* dan *recall*. Proses ini dapat digunakan untuk membuktikan apakah sistem ini dapat dikatakan berhasil dalam menghasilkan keluaran sesuai dengan *query* yang diinput pengguna.

2. Landasan Teori

a. Proses Pembobotan

Berdasarkan artikelnya, Popescu memberikan beberapa skema dengan kombinasi rumus yang berbeda dari tiap tahap proses pembobotan di database.

Kombinasi rumus yang akan digunakan dalam sistem pencarian ini adalah campuran dari beberapa rumus yang berbeda dari yang telah ada. Kombinasi rumus campuran inilah yang disebut dengan *hybrid*.

1) Pembobotan *term* di database

a) Pembobotan Lokal (*Local Weight*)

Pembobotan lokal adalah proses untuk menghitung berapa kali *term* muncul dalam setiap dokumen. Semakin sering *term* itu muncul dalam satu dokumen, maka akan memberikan pengertian bahwa *term* itu penting dalam dokumen yang bersangkutan.

Rumus yang akan digunakan : Augmented Normalized Term Frequency (ATF1)

$$\begin{cases} 0.5 + 0.5\,(f_{ij} / x_j) & \text{if } f_{ij} > 0 \\ \text{Diabaikan} & \text{if } f_{ij} = 0 \end{cases} \quad [2.1]$$

f_{ij} = frekuensi *term* i dalam dokumen j.

x_j = frekuensi maksimum dari *term* apa aja yang muncul di dokumen j.

b) Pembobotan Global (*Global Weight*)

Pembobotan global adalah proses yang menghitung berapa kali *term* muncul dalam koleksi dokumen secara keseluruhan (database). *Term* yang muncul di beberapa dokumen saja akan lebih baik daripada *term* yang muncul di banyak dokumen.

Rumus yang akan digunakan : Inverse Dokument Frequency (IDF)

$$\text{Log}\,(N / n_i) \quad [2.2]$$

N = banyaknya dokumen dalam database

n_i = banyaknya dokumen dimana *term* itu muncul.

c) Faktor Normalisasi (*Normalization Factor*)

Proses yang ketiga ini mengkompensasi untuk ketidakcocokan dalam ukuran suatu dokumen. *Term* yang muncul dengan frekuensi kemunculan yang sama di dokumen yang pendek dan dokumen yang panjang, maka dokumen yang akan dipilih adalah dokumen yang pendek.

Rumus yang digunakan : Cosine

$$\frac{1}{\sqrt{\sum_{i=0}^{lj} (G_i L_{ij})^2}} \quad [2.3]$$

G_i = bobot global *term* i

L_{ij} = bobot local *term* i dalam dokumen j

L_j = banyak *term* dalam suatu dokumen

d) Bobot Total tiap *term*

Bobot total dari suatu *term* didapat dengan mengalikan hasil dari ketiga proses diatas. Adapun rumusnya adalah

$$D_{ij} = L_{ij} * G_i * N_j \quad [2.4]$$

D_{ij} = bobot total *term* i

L_{ij} = Bobot local *term* i

G_{ij} = Bobot global *term* i

N_{ij} = Nilai faktor normalisasi

2) Pembobotan *term* pada *query*

Pembobotan *term* pada *query* hanya dilakukan satu kali. Rumus yang digunakan :

$$W_{i,j} = (0.5 + (0.5 * tf_{i,j}) / (\max_j tf_{k,j})) * idf_i * n_j \quad [2.5]$$

Penjelasan :

$tf_{i,j}$ = banyaknya frekuensi *term* i dalam dokumen j.

$max_j tf_{k,j}$ = menunjukkan frekuensi yang paling besar dari *term* yang paling sering muncul di dokumen j.

idf_i = inverse dokumen frequency.

n_j = nilai faktor normalisasi dokumen dimana *term* ada

$W_{i,j}$ = bobot tiap *term* dari *query*.

3) Proses pemilihan pasal berdasarkan *query*

Tiap *term* dalam tabel tampung, diasumsikan sebagai kata kunci atau *index* dari dokumen bersangkutan. Dengan demikian, bila *term* x terpilih, maka dokumen dimana *term* tersebut muncul, akan ditampilkan sebagai dokumen yang dianggap relevan dengan *query* pengguna.

Untuk memutuskan dokumen yang relevan dengan *query* pengguna adalah dengan membandingkan bobot *term* pada *query* dengan bobot *term* di database. *Term* pada database yang memiliki bobot yang sama dengan bobot *term* pada *query* itulah yang akan dipilih. Dengan demikan, pasal yang berkaitan dengan *term* tersebut yang akan dipilih sistem untuk ditampilkan sebagai keluaran sistem.

4) Tahap Evaluasi

Tahap ini digunakan untuk mengetahui apakah sistem pencarian yang telah dibangun adalah sistem yang efektif dalam melakukan pencarian sesuai dengan *query* pengguna.

Proses yang akan dilakukan untuk mengevaluasi sistem yaitu proses *recall* dan *precision*. Proses *Recall* adalah langkah yang dilakukan untuk menentukan seberapa besar sistem dapat memberikan informasi yang relevan dengan *query* yang diinputkan.

$$\text{Rumus : } \frac{\text{\# jawaban yang relevan (sistem)}}{\text{Total \# jawaban yang relevan (database)}} \qquad [2.5]$$

Sedangkan proses *Precision* adalah langkah yang dilakukan untuk menentukan seberapa besar sistem dapat memberikan informasi yang tepat dan akurat sesuai dengan *query* yang diberikan.

$$\text{Rumus : } \frac{\text{\# jawaban yang akurat (sistem)}}{\text{\# jawaban yang dihasilkan sistem}} \qquad [2.6]$$

3. Pembahasan

a. Perancangan Sistem

1) Pra Pemrosesan

KUHP terdiri dari tiga bagian yaitu buku pertama berisi tentang aturan umum, buku kedua membahas tentang kasus kejahatan, dan buku ketiga membahas tentang kasus pelanggaran. Buku kedua dan buku ketiga KUHP yang akan menjadi objek pencarian sistem akan disimpan dalam satu tabel dengan struktur tabel :

Field	Tipe Data	Ukuran	Keterangan
Id	Longinteger	-	PK
Bab	Teks	3	-
Pasal	Teks	25	-
Isi	Memo	-	-
Buku	Teks	3	-

Tabel 3.1 Struktur Tabel KUHP

Dengan menggunakan data dalam tabel KUHP, akan dilakukan proses normalisasi dan tokenisasi. Proses normalisasi adalah proses menghilangkan semua tanda baca, karakter khusus, penomoran pasal. Sedangkan proses tokenisasi adalah proses memecah string menjadi token berdasarkan spasi. Proses ini dilakukan per satu record pasal dalam tabel KUHP, sehingga asal pasal dari suatu token dapat disimpan dalam tabel terpisah dan dapat digunakan kelak untuk memanggil kembali pasal dari token bersangkutan.

Hasil dari proses tokenisasi akan ditampung dalam tabel bobot dan token yang disimpan adalah token yang telah disaring berdasarkan daftar *stoplist*. Token yang termasuk dalam daftar *stoplist* tidak akan disimpan dalam tabel bobot dan tidak diikutsertakan dalam objek pencarian. Tabel bobot itu sendiri akan menyimpan data token beserta frekuensi kemunculan, bobot lokal, bobot global serta bobot totalnya. Sedangkan nilai faktor normalisasi akan disimpan dalam tabel bab dan disimpan berdasarkan banyaknya bab yang ada.

2) Pemrosesan

Sebelum masuk ke proses perhitungan bobot lokal, dihitung terlebih dahulu variabel-variabel pendukungnya seperti frekuensi kemunculan, variabel n_i yang menampung frekuensi kemunculan dokumen dimana token itu muncul, dan variabel f_{maks} yang akan digunakan untuk menampung nilai frekuensi maksimal dari tiap bab yang ada. Nilai variabel n_i untuk tiap token akan disimpan dalam tabel n_i sedangkan nilai f_{maks} akan disimpan dalam tabel bab. Di dalam tabel n_i berisikan data mengenai token, dan nilai n_i. Sedangkan dalam tabel bab, berisikan nilai faktor normalisasi dan f_{maks} tiap bab. Setelah didapatkan nilai variabel untuk tiap token, maka selanjutnya adalah proses pembobotan untuk tiap token yang ada di database.

a) Pembobotan Lokal

Proses pembobotan lokal dilakukan untuk setiap token yang ada di tabel bobot. Pembobotan lokal ini dilakukan dengan menerapkan rumus ATF1 (persamaan [2.1]). Rumus ATF1 memberikan bobot standard 0.5 untuk setiap token yang muncul dalam bab. Sehingga tidak akan terdapat nilai 0 sebagai bobot lokal.

b) Pembobotan Global

Proses pembobotan global dilakukan dengan menerapkan rumus IDF dengan persamaan [2.2]. Rumus ini membutuhkan nilai N, yaitu banyaknya dokumen yang ada dalam database. Dalam kasus ini, bab yang menjadi klasifikasi tiap buku diasumsikan menjadi dokumen. Dengan demikian, rumus ini dapat diterapkan dan nilai N adalah nilai tetap yaitu 40.

c) Faktor Normalisasi

Untuk menghitung nilai faktor normalisasi tiap bab, akan menggunakan rumus cosine dengan persamaan [2.3]. Untuk mempermudah dalam penerapannya ke dalam sistem, maka rumus tersebut dilakukan dalam beberapa langkah. Langkah pertama yaitu hitung nilai n_j yaitu (bobot lokal * bobot global)^2 untuk setiap tokennya pada tabel bobot. Kemudian, jumlahkan n_j untuk tiap babnya dan hitung nilai faktor normalisasinya.

d) Bobot total

Setelah ketiga proses diatas dilakukan, dan didapat bobot lokal dan bobot global untuk tiap token, dan nilai faktor normalisasi per babnya. Maka untuk mendapatkan

bobot totalnya adalah dengan mengalikan ketiga faktor tersebut seperti pada persamaan [2.4].

Apabila tiap token dalam database telah memiliki bobotnya, maka sistem pencarian sudah dapat dijalankan. Sistem pencarian dilakukan dengan cara memasukkan kata kunci yang mengarah ke tindak pidana umum. Pada saat pengguna telah memasukkan kata kunci dan menekan tombol cari, saat itu sistem langsung melakukan pengecekan. Apakah kata kunci yang dimasukkan terdapat di dalam database. Apabila kata tersebut ada di dalam database, maka akan dilakukan proses pembobotan dengan menggunakan persamaan [2.5]. Namun bila kata tersebut tidak terdapat dalam database, maka sistem akan memberi peringatan bahwa kata tersebut tidak ada dalam database. Untuk setiap kata yang dicari, dilakukan proses pembobotan dalam rentang pencarian tertentu.

Rentang pencarian ditentukan dari asal bab dimana kata kunci itu berada. Misal kata kunci yang dimasukkan adalah kata pencurian. Dalam database, kata pencurian berada di bab 22. Berarti pencarian *term* dalam database difokuskan di bab 22 saja. Kemudian pasal pada tabel kuhp yang berhubungan dengan kata tersebut akan dipanggil dan kemudian ditampilkan kepada pengguna. Penampilan hasil keluaran akan diurutkan berdasarkan bobot *term* dimulai dari bobot yang terbesar. Untuk menampilkan pasal yang berhubungan dengan masukan pengguna, digunakan satu tabel temporari. Pengambilan keputusan yang dilakukan sistem dilakukan dengan membandingkan bobot *term* pada *query* dan bobot *term* di database.

3) Pasca Pemrosesan

Setiap dilakukan proses pencarian, dilakukan proses evaluasi dengan menggunakan rumus presisi (2.1.4).

b. Analisa Sistem

Pada tahap ini, akan dijelaskan beberapa point mengenai hasil yang didapat dari sistem pencarian ini :

1) Hasil Pencarian

Tabel 3.1 Hasil pencarian dengan kata kunci yang berbeda
***Angka Tebal** : menunjukkan bahwa pasal tersebut yang mengandung kata kunci*

No	Kata Kunci		presisi	jumlah yang dihasilkan	jml relevan	pasal yang dihasilkan (Pasal)	bobot yang dicari	Term lain yang muncul
	Term	Bobot						
1	Makar (bab 3, buku 2)	frek = 4, Iw = 0.600000, gw = 1.301030,nj = 0.202617, ni = 2, bobot = 0.158160	66.67	9	6	144,142,141 **,140,139b,139a,107,106 ,104**	0.066056 atau 0.158166	memerintah
2	Penculikan (bab 18, buku 2)	frek = 1, Iw = 0.521739, gw = 1.602060, nj = 0.155353, ni = 1, bobot = 0.129853	11.11	9	1	337, 336, 334, 332, 331, **328,** 327, 326, 325	0.129853	Perniagaan,awak, mengangkutkan, kediamannya, pengusutan, tuanya, diteruskannya, pembakaran, 324
3	pencurian (buku 2, bab 22)	frek = 10, Iw = 0.916667, gw = 1.602060, nj = 0.158183, ni = 1, bobot = 0.232301	100	4	4	**362, 363, 364, 365**	0.232301	(Tidak ada)

Tabel 3.1 adalah contoh dari tiga percobaan dengan menggunakan tiga kata kunci yang berbeda. Dapat dilihat bahwa, untuk dua kali percobaan ternyata ada pasal lain yang tidak berkenaan dengan kata kunci ikut serta ditampilkan sebagai keluaran sistem. Seperti misalnya percobaan pertama menghasilkan 9 pasal dimana 3 diantaranya yaitu pasal 144, 142, 141 bukan pasal yang berkenaan dengan kata kunci. Hal ini disebabkan karena pasal tersebut mengandung *term* yang memiliki bobot sama dengan bobot *term* pada *query*. Kesamaan bobot antar *term* yang berbeda disebabkan karena *term* memiliki jumlah frekuensi yang sama dan berada pada bab dan buku yang sama dengan *term* pada *query*.

Hal ini didukung pula dengan data-data yang terdapat dalam tabel bobot yang berisi tentang *term* beserta bobot totalnya. Jika diteliti bobot *term* berdasarkan asal babnya, masih terdapat beberapa *term* yang memiliki bobot yang sama dan tentu saja dipengaruhi jumlah frekuensi yang sama. Hal inilah yang membuat sistem ikut menyaring pasal lain meski pasal itu tidak mengandung kata kunci yang telah diinputkan.

Sebagai tambahan, pada sistem ini disediakan fasilitas penyaringan terhadap keluaran sistem yang masih menampilkan pasal lain yang tidak mengandung masukan pengguna. Proses penyaringan ini dilakukan dengan menggunakan *string matching*. Dengan menggunakan asal pasal *term* pada *query* yang tersimpan dalam tabel link, maka dapat diketahui pasal mana saja yang memiliki *term* sesuai dengan masukan pengguna. Dengan demikian, dapat dilihat hasil pencarian hanya menampilkan pasal yang mengandung kata kunci dari *query*.

2) Evaluasi Sistem

Tabel 3.2 Hasil evaluasi sistem

No	Token	Jumlah Relevan	Hasil Cari	Presisi (%)
1	makar	6	9	66.67
2	presiden	8	33	24.24
3	penculikan	1	9	11.11
4	pencurian	4	4	100
5	perkawinan	17	21	66.67
6	perceraian	1	20	5
7	perkelahian	6	6	100
8	dicuri	2	4	50
9	mati	40	47	85.11
10	asuransi	3	16	18.75
11	membunuh	5	29	17.24
12	memperkosa	1	17	5.88
13	mengancam	2	12	16.67
14	merusak	26	35	74.29
15	perkosaan	2	14	14.29
16	menggelapkan	3	12	25
17	menyerang	5	31	12.9
18	penipuan	5	20	25
19	hamil	1	15	6.67
20	mencemarkan	1	10	10

Pada tabel 3.2, penulis melakukan proses pencarian dengan kata kunci acak. Berdasarkan hasil pencarian yang dihasilkan oleh sistem menunjukkan nilai presisi sangat jarang mencapai 100% sedangkan nilai presisi terendah adalah 5%. Bila kita lihat kembali tabel 3.1. Hasil dari ketiga proses pencarian tersebut, tetap ada pasal yang ditampilkan meskipun dalam pasal tersebut tidak terdapat *term* yang sama dengan *query*. Bobot yang samalah yang menjadi alasan mengapa hasil pencarian tidak selalu 100% relevan seperti yang telah dijelaskan pada point 1). Sedangkan besar kecilnya bobot

term tidak bersangkutan dengan semakin presisinya sistem pencarian ini. Namun, besar kecilnya bobot *term* akan menentukan rangking kerelevanan suatu pasal terhadap kata kunci yang telah diinputkan.

4. Kesimpulan

Berdasarkan hasil analisa pada sistem pencarian pasal KUHP dengan kasus pidana umum, menghasilkan kesimpulan sebagai berikut :

a. Sistem dapat memberikan hasil pencarian berupa pasal yang mengandung kata kunci sesuai dengan masukan pengguna. Namun, sistem juga menampilkan pasal lain karena proses pengambilan keputusannya dilakukan dengan cara melakukan perbandingan bobot antara bobot *term* pada database dan bobot *term* pada *query*. Sedangkan di dalam database sistem ini, terdapat banyak *term* memiliki bobot yang sama.
b. Berdasarkan proses evaluasi sistem yang seringkali memberikan nilai persentase yang kecil, menunjukkan bahwa sistem pencarian dengan metode *hybrid* dari *term weighting* belum optimal dalam memberikan hasil pencarian dengan data KUHP.
c. Sistem dapat mengklasifikasikan bab dalam database sesuai dengan klasifikasi bab pada KUHP yang sesungguhnya. Pada sistem ini, bab yang ada diasumsikan sebagai dokumen. Oleh sebab itu, metode ini dapat diterapkan pada data KUHP.
d. Berdasarkan percobaan yang dilakukan, dapat pula disimpulkan bahwa besar kecilnya bobot suatu *term* tidak berpengaruh terhadap sistem dalam memutuskan *term* yang akan dipilih. Namun besar kecilnya bobot suatu *term* akan menentukan seberapa relevan *term* tersebut dari bab dimana *term* berasal. Keakuratan sistem dapat terjadi apabila bobot tiap *term* berbeda atau unik.

5. Daftar Pustaka

Daniel, Jurafsky., dan James H. Martin, *Speech and Language Processing*, Prentice Hall, 2000.

Frakes, William B dan Ricardo Baeza Yates, "*Information Retrieval*", Prentice Hall, New Jersey, 1992

Grossman, David A dan Ophir Frieder, "*Information Retrieval Algorithm and Heuristics*", second edition, Springer, 2004.

Harpiandi, "*Pemrograman Database dengan ADO Menggunakan Visual Basic 6.0*", PT Elex Media Computindo, Jakarta, 2003.

Madcoms, "*Panduan Pemrograman dan referensi kamus visual basic 6.0*", Penerbit Andi, 2006.

Mori, Tatsunori & Miwa Kikuchi & Kazufumi Yoshida, http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings2/morisumm.pdf, (diakses 18 November 2006)

Popescu, Andrei, http://www.cs.helsinki.fi/u/popescu/docs/ir.pdf, (diakses 18 November 2006)

Siswoutomo, Wiwit, " Tip dan trik canggih Visual Basic 6", PT Elex Media Computindo, Jakarta, 2002.

Soerrodibroto, R. Soenarto, "*KUHAP dan KUHP dilengkapi yurisprudensi mahkamah agung dan hoge raad*", Cetakan kelima, PT RajaGrafindo, Persada Jakarta, 2006.

Yates, Ricardo Baeza dan Berthier Riberiro, "*Modern Information Retrieval*", Addison Wesley, 1999.