

IMPLEMENTASI MODEL INFORMATION RETRIEVAL UNTUK PENCARIAN KONTEN PADA KUHP BERDASARKAN TINGKATAN HUKUMAN TERBERAT DI INDONESIA

Rifky Permana¹, Richardus Eko Indrajit², Riska Aryanti³, Andika Bayu Hasta Yanto⁴

¹ STMIK Nusa Mandiri Jakarta
<http://www.nusamandiri.ac.id>
rifkyhatake@gmail.com

² ABFI Institute Perbanas Jakarta
<https://perbanas.id/>
indrajit@post.harvard.edu

³ STMIK Nusa Mandiri Jakarta
<http://www.nusamandiri.ac.id>
riskaary22@gmail.com

⁴ Sistem Informasi Akuntansi
AMIK BSI Tangerang
<http://www.bsi.ac.id>
andika.akx@bsi.ac.id

Abstract—*The Criminal Code is the law of law applicable as the legal basis in Indonesia. The Criminal Code always changes according to the dynamics of the times. The numerous articles in the Criminal Code make enough difficulty in searching the data to determine the right article for criminal law stakeholders. Using the information retrieval query wildcard method and the binary tree applied in the search program will make it easier to get the information needed, especially in terms of searching the Criminal Code based on the penalty level. This paper shows the conceptual model used in developing prototype of information retrieval computer program.*

Keywords: *Criminal Code, information retrieval, wildcard query, binary tree, prototype.*

Intisari—KUHP adalah kitab undang-undang hukum yang berlaku sebagai dasar hukum di Indonesia. KUHP selalu mengalami perubahan sesuai dengan dinamika perkembangan jaman. Banyaknya pasal di dalam KUHP cukup membuat kesulitan dalam pencarian data untuk menentukan pasal yang tepat bagi para pemangku kepentingan hukum pidana. Dengan menggunakan metoda *information retrieval query wildcard* dan pohon biner yang diterapkan di dalam program pencarian akan mempermudah mendapatkan informasi yang dibutuhkan, khususnya dalam hal pencarian pasal KUHP berdasarkan tingkat hukuman. Tulisan ini memperlihatkan model konseptual yang

dipergunakan dalam mengembangkan prototipa program komputer berbasis *information retrieval*

Kata Kunci: *KUHP, information retrieval, query wildcard, pohon biner, prototipe.*

PENDAHULUAN

KUHP merupakan singkatan dari Kitab Undang Undang Hukum Pidana yaitu kitab undang-undang hukum yang merupakan landasan dasar hukum di Indonesia. KUHP di Indonesia saat ini terbagi menjadi dua sumber hukum yaitu sumber hukum pidana materiil serta sumber hukum pidana formil. Definisi dari sumber hukum materiil adalah sumber hukum yang berasal dari faktor-faktor masyarakat yang mempengaruhi pembentukan hukum (pengaruh terhadap pembuat UU, pengaruh terhadap keputusan hakim, dan sebagainya), Sedangkan, definisi sumber hukum formil merupakan dasar kekuatan mengikatnya peraturan-peraturan agar ditaati oleh semua lapisan masyarakat maupun oleh lapisan penegak hukum. KUHP sejak dari awal sampai saat ini selalu mengalami perubahan sesuai dengan dinamika perkembangan jaman. KUHP yang saat ini berlaku di Indonesia bersumber dari hukum kolonial belanda di masa lampau. Usaha pemerintah Indonesia untuk melakukan pembaruan hukum pidana masih berjalan sampai saat ini yang dilakukan dengan membentuk Lembaga Pembinaan Hukum Nasional sebagai salah satu upaya pemerintah untuk membentuk KUHP nasional yang baru

sampai saat ini. Dikarenakan selalu ada pembaruan di dalam KUHP yang mengakibatkan terjadinya penumpukan data KUHP maka cukup membuat kesulitan melakukan pencarian data (Fiarni, Sipayung, & Martiana, 2015) bagi para pemangku kepentingan hukum pidana, baik data KUHP terbaru ataupun data KUHP yang sudah lama. Oleh karena itu di buatlah implementasi model *Information Retrieval* untuk pencarian konten di dalam KUHP berdasarkan tingkatan hukuman terberat sampai teringan agar para pemangku kepentingan hukum pidana dapat melakukan tugasnya dengan cepat untuk memutuskan tindakan yang tepat bagi para pelaku pidana di Indonesia. Berdasarkan hal tersebut rumusan masalahnya adalah Bagaimana membangun sistem informasi dengan implementasi model implementasi retrieval dalam pencarian konten pada KUHP berdasarkan tingkatan hukuman terberat di Indonesia agar dapat memudahkan kinerja para pemangku kepentingan dalam mencari konten untuk mengambil keputusan yang tepat dan cepat.

BAHAN DAN METODE

Dalam penulisan ini, penulis mengumpulkan data-data dan informasi yang diperlukan dengan menggunakan metode pengumpulan data sebagai berikut:

1. Observasi
Observasi adalah pengamatan langsung suatu kegiatan yang sedang dilakukan. Untuk mendapatkan data yang diperlukan penulis menggunakan cara pengamatan langsung pada situs pencari konten KUHP
2. Studi Pustaka
Pada metode ini penulis mendapat banyak bahan masukan tentang bagaimana merancang atau mengembangkan suatu sistem informasi menurut para ahlinya. Juga pada metode ini penulis membuka, mengambil dan mengutip dari beberapa kutipan para ahli yang berdasarkan dari jurnal ilmiah nasional.

Metode Pencarian Dengan *Binary Search*

Pencarian pada data yang terurut menunjukkan kinerja yang lebih baik daripada pencarian pada data yang belum terurut. Hal ini, sudah kita bicarakan pada metode pencarian beruntun untuk data yang sudah terurut. Data yang terurut banyak ditemukan didalam kehidupan kita sehari-hari. Data nomor telepon di dalam buku telepon misalnya, sudah terurut berdasarkan nama/instansi pelanggan telepon dari A sampai Z. Data karyawan dirurut berdasarkan nomor induknya dari nomor kecil ke nomor besar. Data

mahasiswa diurut berdasarkan NIM (Nomor Induk Mahasiswa), kata-kata (entry) di dalam kamus bahasa Inggris/Indonesia telah diurut dari A sampai Z, dan sebagainya

Proses yang terjadi pada pencarian dengan metode ini adalah sebagai berikut :

1. Membaca Array data
2. Apabila array belum terurut maka array diurutkan terlebih dahulu
3. Menentukan data yang akan dicari
4. Menentukan elemen tengah dari array
5. Jika nilai elemen tengah sama dengan data yang dicari, maka pencarian berhenti
6. Jika elemen tengah tidak sama dengan data yang dicari maka :
 - a. Jika nilai elemen tengah > data yang dicari maka pencarian dilakukan pada setengah array pertama.
 - b. Jika nilai elemen tengah lebih kecil dari pada data yang dicari maka pencarian pada setengah array berikutnya (Anggraeni, Sukmaaji, & Sutanto, 2016)

Tabel 1. Metode pencarian dengan *Query Wildcard*

| Karakter | Deskripsi | Contoh |
|----------|---|--|
| * | Mencocokkan berapa pun jumlah karakter bisa menggunakan tanda bintang (*) di mana pun dalam string karakter. | wh* mencari what, white, dan why, tetapi bukan awhile atau watch. |
| ? | Mencocokkan alfabet tunggal dalam posisi khusus. | b?ll menemukan ball, bell, dan bill. |
| [] | Mencocokkan karakter di dalam kurung. | b[ae]ll menemukan ball dan bell, namun bukan bill. |
| ! | Tidak mencakup karakter di dalam tanda kurung. | b[!ae]ll menemukan bill dan bull, tetapi bukan ball atau bell. Like "[!a]*" menemukan semua item yang tidak dimulai dengan huruf a. |
| - | Mencocokkan rentang karakter. Ingat untuk menentukan karakter dalam urutan naik (A sampai Z, bukan Z sampai A). | b[a-c]d menemukan bad, bbd, dan bcd. |
| # | Mencocokkan setiap karakter numerik tunggal. | 1#3 123.m |

Sumber: (Burns, Jackson, Sheehy, Finkel, & Quezado, 2017)

HASIL DAN PEMBAHASAN

Information Retrieval atau yang biasa disebut dengan sistem temu balik informasi merupakan bagian dari ilmu komputer yang ditujukan untuk pengambilan informasi dari dokumen-dokumen yang mengacu pada isi dan konteks dokumen-dokumen itu sendiri. Menurut Gerald Kowalski, sistem temu balik informasi adalah suatu sistem yang mampu melakukan penyimpanan, pencarian, dan pencarian informasi. Informasi dalam konteks ini dapat terdiri dari teks (termasuk data numerik dan tanggal), gambar, audio, video, dan objek multimedia lainnya. Definisi Information Retrieval adalah bagaimana cara menemukan sebuah dokumen dari dokumen-dokumen tidak terstruktur untuk memberikan informasi yang dibutuhkan dari koleksi dokumen besar yang tersimpan dalam komputer (Pardede, 2014). Information Retrieval dalam kehidupan sehari-hari sangat sering digunakan terutama dalam ruang lingkup tindak pidana yang selalu terjadi setiap harinya untuk memudahkan para pemangku kepentingan dalam penerimaan informasi konten tentang tingkatan hukum pidana yang tepat sesuai dengan tindak pidana yang sudah terjadi.

Menurut Martiman Prodjohamidjojo Hukum Pidana adalah bagian dari keseluruhan hukum yang berlaku di suatu negara, yang mengadakan dasar - dasar dan aturan - aturan untuk menentukan perbuatan - perbuatan mana yang tidak boleh dilakukan, yang dilarang, dengan disertai ancaman atau sanksi pidana tertentu bagi siapa saja yang melanggarnya. Hal ini untuk memutuskan dalam hal apa dan kapan kepada orang-orang yang telah melakukan hal yang terlarang dapat dijatuhi pidana seperti yang telah dicantumkan. Menegaskan dengan cara bagaimana pengenaan pidana itu dapat dilakukan apabila orang yang diduga telah melanggar ketentuan tersebut (Baskoro, Ridok, & Furqon, 2015).

Kitab Undang Hukum Pidana tidak dapat berjalan dengan sendirinya. Hukum hanya dapat bekerja melalui orang. Karena itu memungkinkan undang- undang pidana itu dilakukan. Pedoman untuk melaksanakan pidana ini adalah Hukum Acara Pidana (KUHP) yang sebelumnya bernama HIR dan undang- undang lain (UU Kepolisian, UU Kejaksaan, dan UU Mahkamah Agung). KUHP bertujuan untuk menyatukan penyidikan (penyidik berada di satu tangan) berdasarkan Pasal 284 ayat (2) KUHP. Di Indonesia sampai saat ini pelaksanaan hukum pidana masih fragmentasi dan instansi centris. Karena itu harus dirubah jika mau melaksanakan hukum pidana secara tepat sasaran dan berdaya guna (Abdulah, Hukum, & Mada, 2015)

Penjatuhan hukuman pidana dapat dilakukan kepada pelaku yang telah melakukan perbuatan pidana. Tindakan pidana merupakan hal yang oleh suatu aturan norma dilarang dan dapat dikenakan pidana, hanya saja dalam pada itu diingat bahwa larangan ditujukan kepada perbuatan, yaitu suatu kejadian yang dihasilkan oleh perbuatan orang, sedangkan ancaman pidananya ditujukan kepada orang yang menimbulkan kejadian itu (Helmi, 2016).

Hukum pidana di Indonesia sangat banyak tingkatannya karena terus di perbarui setiap waktu dan juga harus tepat pengenaannya kepada para terpidana, oleh karena itu diperlukan algoritma pencarian yang nantinya bisa memudahkan para pemangku kepentingan pidana dalam menjatuhkan hukuman yang tepat kepada para terpidana.

Algoritma pencarian adalah algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah sistem pencarian dilakukan maka dapat diperoleh salah satu dari beberapa kemungkinan, yaitu data yang dicari telah ditemukan (successful) atau tidak dapat ditemukan (unsuccessful). Program dalam mencari data bisa dilakukan dengan dua cara yaitu pencarian internal dan pencarian eksternal. Dalam pencarian internal, semua rekaman yang diketahui berada dalam pengingat komputer sedangkan pada pencarian eksternal, tidak semua rekaman yang diketahui berada dalam pengingat komputer, tetapi ada sejumlah rekaman yang tersimpan dalam penyimpan luar misalnya pita atau cakram magnetis. (Wahdaniah & L, 2017).

Wildcard adalah fasilitas untuk searching yang sangat berguna ketika hendak melakukan pencarian dalam sebuah field. Wildcard bisa berperan dalam menggantikan satu atau lebih karakter saat mencari data dalam sebuah database. Dalam pencarian dengan wildcard harus berhati-hati saat menggunakannya. Digunakan apabila sangat memerlukan dan gunakan awalan atau postfix wildcard atau akhiran prefix wildcard saat melakukan pencarian data. Karena full wildcard akan menampilkan kemungkinan yang lebih banyak apabila dibandingkan dengan postfix wildcard.

```
#Full wildcard
SELECT * FROM table WHERE COLUMN LIKE '%halo%';
#Postfix wildcard
SELECT * FROM TABEL WHERE COLUMN LIKE 'halo%';
#Prefix wildcard
SELECT * FROM TABEL WHERE COLUMN LIKE 'hello%';
```

Contoh wildcard query :

Yogya* : menemukan semua dokumen yang berisi sembarang kata berawalan yogya.

Dengan B+Tree dapat cukup dilakukan dengan mencari semua node yang berada dibawah node yogya.

$Yogya \leq w < yogyb$

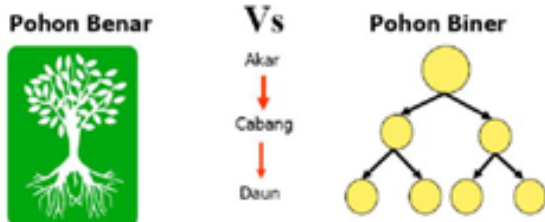
Bagaimana dengan query: *arta ?

Perlu dipelihara B+Tree tambahan yang menyimpan backward dari semua kata

Bagaimana dengan query: yo*arta?

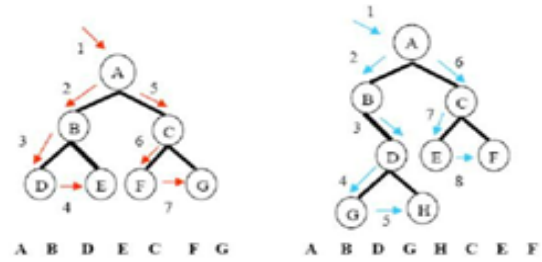
Pohon (tree) merupakan struktur data nonlinier yang sering digunakan dalam aplikasi sehari-hari. Misalnya aplikasi pohon yang bisa kita lihat sehari-hari adalah pengelolaan file dalam direktori penyimpanan. Pohon merupakan struktur data yang memiliki suatu struktur hirarki pada sekumpulan elemen, dan memiliki hubungan satu ke banyak (one to many relationship) seperti yang kita lihat dalam struktur organisasi sebuah perusahaan atau daftar isi sebuah buku (Wijaya & Feter, 2015)

Binary tree atau yang biasa di sebut pohon biner dapat deskripsikan sebagai suatu daftar simpul atau node dengan salah satu elemen khusus yang disebut root dan dua sub pohon yang disebut subtree kiri dan subtree kanan.



Sumber: (Widiastono, Banyal, & Angriani, 2016)
Gambar 1. Pohon Binary Tree

Salah satu contoh binary tree yaitu dengan menggunakan double linked list. Double linked list merupakan rangkaian dari node/simpul yang saling terhubung dengan menggunakan dua pointer dalam satu elemen dan list dapat melintas baik di depan atau belakang. Node atau simpul pada double linked list terdiri dari satu bagian yang menyimpan data, dan dua bagian yang menyimpan pointer menuju ke subtree Kiri dan subtree kanan (Widiastono et al., 2016)



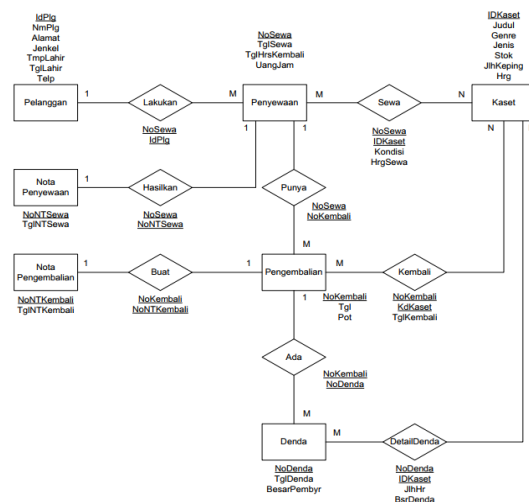
Sumber: (Widiastono et al., 2016)

Gambar 2. Proses Penelusuran Binary Tree

Populasi data yang digunakan di ambil berdasarkan keseluruhan kitab undang undang hukum pidana yang ada di Indonesia dari pertama dibentuk sampai saat ini. Karena selalu terjadi pembaharuan setiap waktunya agar bisa tepat mengenai sasaran untuk kepada para pelaku tindak pidana. Karena selalu di perbaharui setiap waktu maka data sampel yang ada cukup banyak mulai dari pertama kali dibuatnya KUHP sampai saat ini.

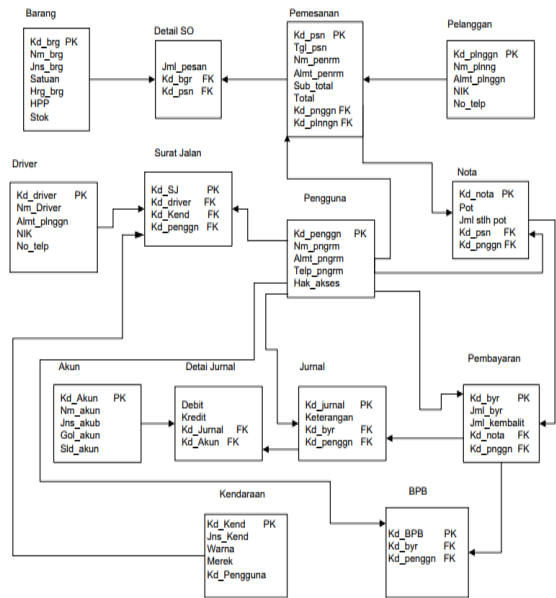
Format dan karakteristik informasi yang di ambil dalam kitab undang hukum pidana merupakan data yang terstruktur karena di dalam KUHP tersebut ada jenis pidana, tahun pembuatan, ketentuan pidana, pasal pidana, anak pasal pidana, dan jenis hukuman. Informasi yang nanti akan di dapat harus berhubungan dengan masalah dan pencarian, harus bebas dari kesalahan-kesalahan, harus ada ketika di butuhkan dan tidak boleh terlambat serta harus mempunyai manfaat bagi para pemakai dan pencari informasi.

Media penyimpanan yang digunakan yaitu dengan penyimpanan data informasi KUHP di dalam sebuah database. Pada tahap implementasi dan hasil nanti akan di implementasikan ke dalam bentuk entity relationship diagram (ERD) dan Logical Record Structure (LRS).



Sumber: (Mayasari, 2015)

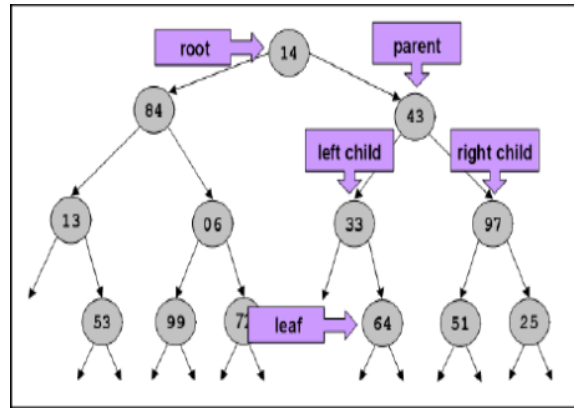
Gambar 3. Organisasi Penyimpanan Informasi Bentuk ERD



Sumber: (Harjunawati, 2017)
 Gambar 4. Organisasi Penyimpanan Informasi Bentuk LRS

Pencarian yang digunakan bisa berbentuk menu, yaitu berdasarkan tahun pembuatan, berdasarkan tingkatan hukuman, berdasarkan tindak/jenis pidana yang dilakukan, berdasarkan tahun pembuatan serta bisa juga berdasarkan pasal pasal pidana yang keseluruhannya tercantum di dalam KUHP dengan menggunakan metode binary tree dan query wildcard.

Algoritma pencarian dilakukan dengan menggunakan pencarian pohon biner (binary tree) dan query wildcard, dikarenakan pencarian pohon biner dan query wildcard paling efisien dilakukan apabila data yang dikumpulkan sudah terurut. Karena data yang dicari cukup banyak dan bisa menghasilkan waktu yang cepat. B-tree merupakan pohon cari keseimbangan yang dibuat untuk penyimpanan pada magnitic disk karena operasi yang sangat lambat dibandingkan dengan RAM. B- tree mulai di perkenalkan pada tahun 1960 tujuan dikembangkan metode ini adalah sebagai file system yang disebut dengan metode akses yang digunakan untuk beberapa mesin sperti Sperry Univac Corporation. Orang yang pertama kali menciptakan adalah Rudolf Bayer dan Ed McCreight tidak menjelaskan huruf B pada B-tree namun yang paling diyakini adalah B merupakan kependekan dari Balance yang artinya seimbang, karena semua simpul daun pada pohon berada pada tingkat atau level yang salam, namun ada sebagian yang mengartikan B merupakan Bayer nama depan dari penciptanya atau Boeing kerana merka bekerja untuk Boeing Scientific Research Labs.



Sumber: (Rosmala & Kresna, 2012)
 Gambar 5. Full Binary Tree

Setiap pohon biner memiliki operasi traversal yaitu kunjungan pada sebuah simpul tepat satu kali, agar diperoleh urutan informasi secara linier yang tersimpan di dalam pohon biner maka harus melakukan kunjungan lengkap kepada pohon biner tersebut. Terdapat beberapa cara untuk mengunjungi pohon biner, berikut cara tersebut :

1.Preorder

Urutan kunjungan ini sebagai berikut : cetak isi simpul yang di kunjungi – kunjungi cabang kiri – kunjungi cabang kanan. Berikut coding program untuk binary tree dengan cara preorder :

```

Prosedur PreOrder (P;B-Tree)
If P≠Nil then
    Output(Info (P))
    PreOrder (kiri (P))
    PreOrder (kanan(P))
endif
    
```

2.Inorder

Urutan kunjungan ini mempunyai urutan sebagai berikut : Kunjungi cabang kiri – cetak isi simpul yang ingin di kunjungi – kunjungi cabang kanan. Berikut coding program untuk binary tree dengan cara inorder :

```

Prosedur InOrder (P:BTtree)
If P≠Nil then
    InOrder (Kiri(P))
    Output(Info (P))
    InOrder (Kanan (P))
endif
    
```

3.Postorder

urutan kunjungan ini mempunyai urutan sebagai berikut : kunjungi cabang kiri – kunjungi cabang kanan – cetak simpul yang ingin di kunjungi.

```

#include<stdio.h>//header file
#include<conio.h>
    
```

```

/* Deklarasi struct */
typedef struct Node{
int data; //data pada tree
Node *kiri; //penunjuk node anak kiri
Node *kanan; //penunjuk node anak kanan
};
    
```

Contoh Bahasa Program Binary Tree

```

Kamus
Conts N :
Type t=array[0 ... N] of integer
val, left, right, mid : Integer
    
```

```

Algoritma
int binarySearch(LIST t[], int n, int val)
{
int left, right, mid;
left = 0; right = n-1;
while(left<=right = n-1;
mid=(left+right)/2;
if (val<t[mid].key)
right=mid-1;
else if (val>t[mid].key)
left=mid+1;
else
return mid; /* found */
}
return -1; /* not found */
}
    
```

Untuk kecepatan pencarian dengan metode *wildcard* pencarian bisa menggunakan *dynaset* dan snapshot berdasarkan catatan yang cocok dengan kriteria pencarian. Bisa dengan menggunakan metode *MoveFirst*, *MoveLast*, *MoveNext* dan *MovePrevious*. Contoh metode pencarian cepat dengan *wildcard*:

```

Dim MyDS As Dynaset, MyDb As database, SQLx As String
SQLx = "Select * from Authors Where Author Like 'b*' "
Set MyDb = OpenDatabase("BIBLIO.MDB")
'Open a database.
Set MyDs = MyDb.CreateDynaset(SQLx)
'Create Dynaset using SQLx.
While Not MyDs.EOF
Print MyDs!Author
MyDs.MoveNext
Wend
    
```

```

Sub Cmdcari_Click()
Dim SQLX As String, SearchText As String

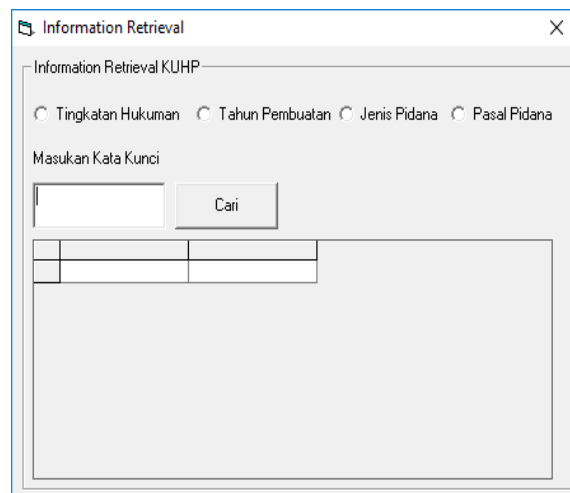
If Right$(text2.Text,1) <> "*" Then text2.Text =
text2.Text & "*"
If Left$(text2.Text,1) <> "*" Then text2.Text = "*" &
text2.Text
    
```

```
SearchText = text2.Text
```

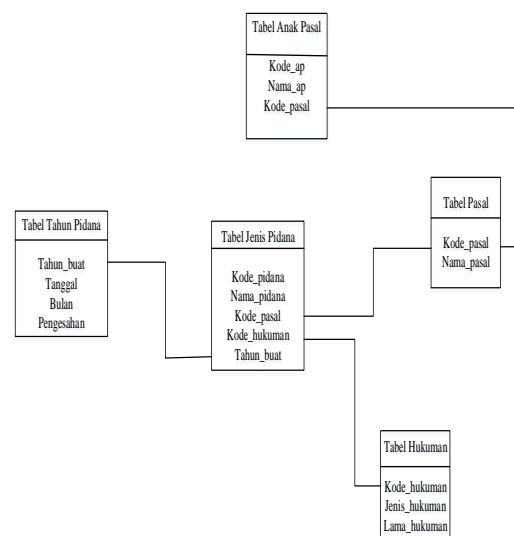
```

SQLx = "Select * From Authors Where Author Like '" & SearchText
SQLx = SQLx & "' Order By Au_id"
Data1.DatabaseName = "kuhp.mdb"
Data1.RecordSource = SQLx ' Data1 control will
use SQLx query string.
Data1.Refresh 'Update the data control with
results of SQL query.
End Sub
    
```

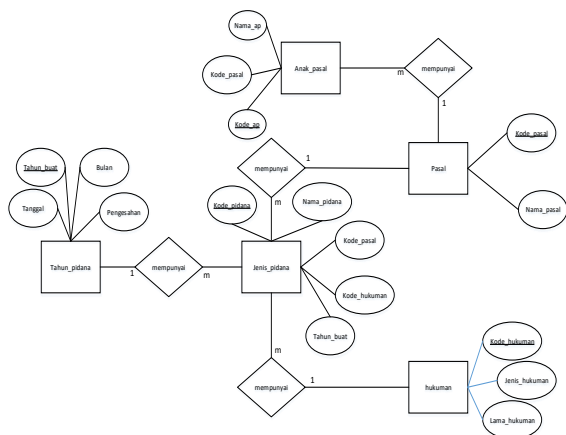
Implementasi pohon biner yang di terapkan di program *visual basic* ini untuk mendapat hasil yang di inginkan dengan ketepatan dan keakusarian pencarian data yang sangat baik. Berikut rincian tampilan untuk information retrieval dalam pencarian data KUHP



Sumber: (Permana et al., 2017)
Gambar 6. Tampilan Rancangan Program



Sumber: (Permana et al., 2017)
Gambar 7. Logical Record Structure (LRS)



Sumber: Sumber: (Permana et al., 2017)
Gambar 8. Entity Relationship Diagram (ERD)

Pencarian akan menghasilkan tingkat akurasi yang baik dan sangat cepat apabila dimasukkan keyword dengan lengkap karena menggunakan binary search yang datanya sudah terurut dan juga *query wildcard*, dimana dengan algoritma pohon kita mendapatkan struktur penyimpanan data dan pencarian data yang relatif lebih baik dan efisien daripada stuktur penyimpanan lainnya dan juga *query wildcard* yang sangat efisien dalam pencarian data sesuai dengan kriteria yang dimasukkan.

KESIMPULAN

Algoritma biner sangat baik dan efisien dalam melakukan pencarian data apabila data yang ingin dicari dan jumlahnya banyak sudah dilakukan proses sorting terlebih dahulu. Sedangkan kekurangan dalam metode ini yaitu data yang ada harus di sorting terlebih dahulu agar lebih tepat dalam pencarian, dan juga algoritma lebih rumit sehingga tidak baik untuk data yang terbengkalai Metode Query Wildcard dapat melakukan pencarian data dengan sangat cepat dan sesuai dengan kriteria pencarian yang dimasukkan akan tetapi metode ini membaca semua field yang termasuk field-field yang seharusnya tidak diperlukan. Ini tentu saja akan membebani memori yang akan digunakan nantinya. Untuk kedepannya riset bisa dilakukan untuk data yang sudah di urutkan dan sebisa mungkin menghindari atau merapikan data yang sudah lama terbengkalai agar dapat lebih akurat dan efisien dalam melakukan pencarian serta lebih efisien dalam penggunaan memori penyimpanan yang digunakan.

REFERENSI

Abdulah, R. H., Hukum, F., & Mada, U. G. (2015). URGENSI HUKUM ADAT DALAM

PEMBAHARUAN Urgency of Customary Law in the Renewable of National Criminal Law, 9(2), 168–181.

Anggraeni, A., Sukmaaji, A., & Sutanto, T. (2016). Rancang Bangun Aplikasi Kamus Indonesia - Korea Menggunakan Metode Algoritma Binary Issn 2338-137X, 5(3), 1–10.

Baskoro, S. Y., Ridok, A., & Furqon, M. T. (2015). Pencarian Pasal Pada Kitab Undang-Undang Hukum Pidana (KUHP) Berdasarkan Kasus Menggunakan Metode Cosine Similarity Dan Latent Semantic Indexing (Lsi), 2(2), 83–88.

Burns, J., Jackson, K., Sheehy, K. A., Finkel, J. C., & Quezado, Z. M. (2017). The Use of Dexmedetomidine in Pediatric Palliative Care: A Preliminary Study. *Journal of Palliative Medicine*, 20(7), 779–783. <https://doi.org/10.1089/jpm.2016.0419>

Fiarni, C., Sipayung, E. M., & Martiana, Y. (2015). PERANCANGAN APLIKASI PEMBUATAN BERKAS PERKARA PIDANA DAN PENGELOLAAN BERKAS PADA SISTEM INFORMASI DIREKTORAT RESERSE KRIMINAL UMUM. In *Seminar Nasional Sistem Informasi Indonesia (SESINDO)* (pp. 429–434). Surabaya: Departemen Sistem Informasi, ITS. Retrieved from <http://is.its.ac.id/pubs/oajis/index.php/home/detail/1592/PERANCANGAN-APLIKASI-PEMBUATAN-BERKAS-PERKARA-PIDANA-DAN-PENGELOLAAN-BERKAS-PADA-SISTEM-INFORMASI-DIREKTORAT-RESERSE-KRIMINAL-UMUM>

Harjunawati, S. (2017). SISTEM INFORMASI AKUNTANSI PENJUALAN BERBASIS WATERFALL MODEL UNTUK PERUSAHAAN DAGANG. *Jurnal Ilmu Pengetahuan Dan Teknologi Komputer*, 2(2), 131–138.

Helmi, M. (2016). Ketiadaan Daluwarsa Penuntutan dalam Hukum Pidana Islam dan Pembaruan Hukum Pidana di Indonesia. *Mazahib*, 15(2), 196–207. Retrieved from <http://journal.iain-samarinda.ac.id/index.php/mazahib/article/view/643>

Mayasari, M. S. (2015). Perancangan Aplikasi Sistem Informasi Penyewaan Kaset (CD), 6(1), 47–56.

Pardede, J. (2014). IMPLEMENTASI MULTITHREADING UNTUK

MENINGKATKAN KINERJA INFORMATION
RETRIEVAL DENGAN METODE GVSM.

Permana, R., Indrajit, R. E., Aryanti, R., & Yanto, A. B. H. (2017). *Laporan Akhir Penelitian Mandiri*. Jakarta.

Rosmala, D., & Kresna, G. (2012). Implementasi Algoritma Binary Tree Pada Sistem Informasi Multilevel Marketing, *7(2)*, 31-43.

Wahdaniah, I., & L, H. (2017). APLIKASI PENCARIAN GAMBAR DENGAN KONTEN BERDASARKAN FITUR BENTUK BERBASIS WEB MENGGUNAKAN METODE GRADIEN VEKTOR FLOW SNAKE, *9(April)*, 49-56.

Widiastono, A., Banyal, N. A., & Angriani, L. (2016). SISTEM INFORMASI TROUBLESHOOTING PADA PERSONAL COMPUTER (PC). *ILKOM Jurnal Ilmiah*, *8(Agustus)*, 103-110.

Wijaya, A., & Feter, N. (2015). Aplikasi Simulasi Pengurutan Data Menggunakan Algoritma Heap Sort. *Jurnal Pseudocode*, *2(2)*, 81-88.