

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menguraikan hasil proses analisis dan perancangan sistem yang dikembangkan yaitu mengenai analisis kebutuhan sistem yang dikembangkan, kemudian dilanjutkan dengan penjelasan mengenai perancangan sistem yang memenuhi hal-hal yang telah dianalisis sebelumnya untuk diwujudkan menjadi sebuah sistem yang dapat digunakan dengan baik.

3.1 Analisis Kebutuhan

Bagian analisis ini tersusun atas berbagai permasalahan yang melatarbelakangi pengembangan sistem ini beserta dengan kebutuhan yang harus dapat diatasi oleh sistem ini. Setelah itu akan dijelaskan juga mengenai gambaran umum dari sistem yang akan dikembangkan.

Sesuai dengan yang telah disampaikan pada bagian latar belakang dari laporan tugas akhir ini, maka sistem yang dikembangkan ini dapat melakukan pencarian keterkaitan antar dokumen untuk mengetahui sejarah perubahan undang-undang, sejarah perubahan undang-undang tersebut memiliki dua jenis yaitu sejarah perubahan berdasarkan judul undang-undang dan sejarah perubahan berdasarkan pasal. Dengan memproses dokumen undang-undang yang telah memiliki format XML yang dihasilkan dari sistem *Violina* [VIO08], maka penulis membuat *rules* atau aturan-aturan untuk mengenali bagian-bagian dari dokumen XML tersebut.

Berdasarkan hal-hal yang telah disebutkan sebelumnya, maka penulis akan mengidentifikasi kebutuhan-kebutuhan yang perlu disediakan solusinya dalam sistem yang akan dikembangkan. Kebutuhan-kebutuhan tersebut antara lain:

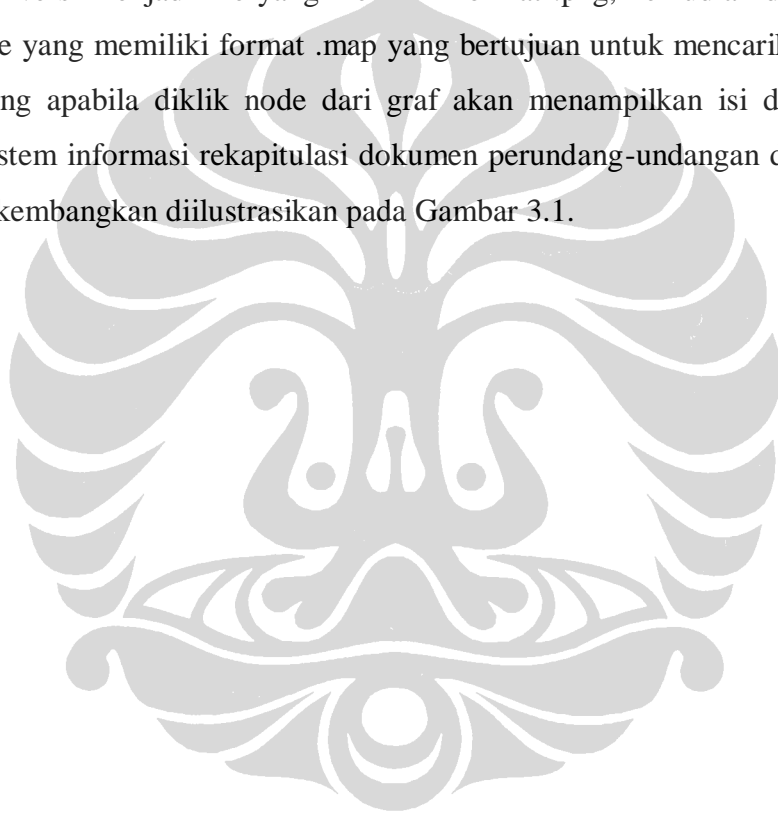
- a. Sistem yang dibangun adalah sistem yang dapat menerima data masukan berupa dokumen undang-undang yang memiliki format XML.

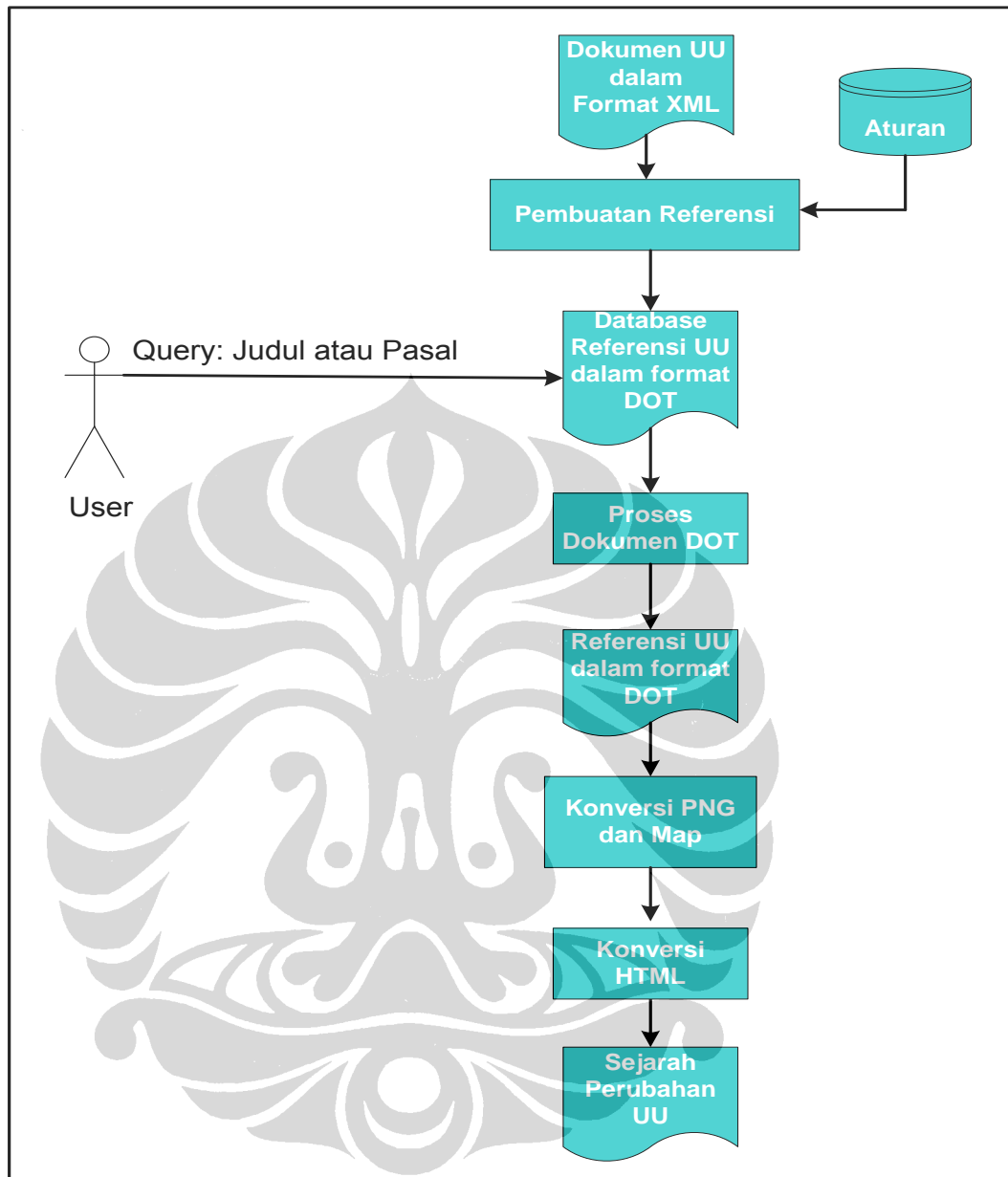
- b. Sistem harus memiliki fitur untuk melakukan pencarian keterkaitan antar dokumen secara otomatis sehingga pengguna bisa dengan mudah mengetahui apakah dokumen UU yang dia temukan merupakan versi terakhir atau bukan.
- c. Keterkaitan antar dokumen tersebut akan direpresentasikan dalam bentuk graf berarah
- d. Graf berarah yang menyatakan keterkaitan antar dokumen harus memiliki sifat informatif terhadap pengguna, yang artinya setiap node yang terbentuk harus bisa diklik dan akan menampilkan informasi dari undang-undang tersebut.

Sistem informasi rekapitulasi dokumen perundang-undangan di Indonesia yang dikembangkan oleh penulis ini, akan memproses semua dokumen undang-undang XML yang dihasilkan dari Sistem Extraction Information (SEI) untuk dilakukan pemrosesan agar keterkaitan antar dokumen dapat dibuat, melalui aturan atau rule yang telah dibuat oleh penulis maka akan didapatkan referensi dari setiap dokumen yang menyatakan keterkaitan antar dokumen. Setelah semua referensi yang dimiliki dokumen ditemukan maka proses selanjutnya adalah pembuatan dokumen DOT. Pembuatan dokumen ini dilakukan dengan cara menyatukan seluruh referensi yang ditemukan sebelumnya dalam satu berkas (*file*) dan menyimpannya dalam format *.dot*. Namun untuk memenuhi syarat pembentukan dokumen DOT, seluruh referensi tersebut diletakkan di antara tanda kurung kurawal dan di bagian awal dokumen ditambahkan kata kunci *digraph* diikuti nama graf. Selain itu dapat juga dilakukan penambahan atribut pada *node* dan *edge* dari graf untuk memperindah tampilan graf.

Apabila pengguna sistem memasukkan input berupa kata kunci atau *query* pada sistem perolehan informasi dokumen legal yang telah dikembangkan oleh [PRA09], maka dari *query* tersebut sistem akan menampilkan list undang-undang yang terkait dengan *query* tersebut. Kemudian bila list undang-undang tersebut diklik

oleh pengguna maka sistem akan mencari keterkaitan antar dokumen tersebut kedalam dokumen yang sudah dalam format *.dot*, dimana pada dokumen ini berisi semua keterkaitan antar dokumen, dari sana sistem akan melakukan pemrosesan pada dokumen *.dot* tersebut untuk dicari keterkaitan antar dokumen sesuai dengan list undang-undang yang diklik pengguna tersebut. Kemudian dilakukan pembuatan dokumen dengan cara menyatukan seluruh referensi yang ditemukan dan menyimpannya dalam format *.dot*. Selanjutnya, dari dokumen *.dot* tersebut dilakukan konversi menjadi file yang memiliki format *.png*, kemudian dikonversi lagi menjadi file yang memiliki format *.map* yang bertujuan untuk mencari referensi dokumen yang apabila diklik node dari graf akan menampilkan isi dari dokumen tersebut. Sistem informasi rekapitulasi dokumen perundang-undangan di Indonesia yang akan dikembangkan diilustrasikan pada Gambar 3.1.





Gambar 3.1: Arsitektur Sistem Rekapitulasi Dokumen Perundang-undangan di Indonesia

Dari arsitektur sistem tersebut terlihat bahwa pengembangan sistem ini terbagi menjadi tiga subsistem yang mendukungnya, yaitu Sistem Ekstraksi Informasi (SEI) yang telah dikembangkan oleh [VIO08], Sistem Perolehan Informasi Dokumen Legal

yang dikembangkan oleh [PRA09], dan sistem informasi rekapitulasi dokumen legal yang dikembangkan oleh penulis. Melalui SEI akan didapat dokumen undang-undang yang memiliki format XML, dimana dokumen ini yang akan menjadi objek penelitian dari penulis. Dari dokumen undang-undang XML ini maka penulis membuat suatu aturan untuk mengenali bagian-bagian dari dokumen untuk mencari keterkaitan antar dokumen. Melalui sistem perolehan informasi dokumen legal yang telah dikembangkan oleh [PRA09], maka akan diperoleh suatu *list* undang-undang dan pasal, apabila *list* ini diklik oleh pengguna maka sistem akan mencarikan dan menampilkan keterkaitan dokumen tersebut dengan dokumen yang lain yang direpresentasikan dalam bentuk *graf* berarah yang informatif.

3.2 Dokumen Masukan

Pada sistem ini akan menerima dokumen masukan berupa dokumen undang-undang dalam format XML yang didapat dengan melakukan ekstraksi 472 dokumen UU dalam format TXT melalui sistem ekstraksi informasi (SEI) yang telah dikembangkan oleh [VIO08]. Dokumen ini akan diproses dengan menggunakan bahasa pemrograman PERL untuk mencari keterkaitan antar dokumen.

Keluaran dari sistem rekapitulasi ini adalah sebuah sejarah perubahan undang-undang yang disajikan dalam bentuk *graf* berarah yang terdiri dari elemen *node* dan *edge*, dimana *node* menggambarkan dokumen legal dan *edge* menggambarkan hubungan antar dokumen yang satu dengan yang lainnya.

3.3 Aturan (*Rule*)

Aturan atau *rule* merupakan sekumpulan aturan-aturan yang dibuat penulis untuk melakukan pemrosesan setiap dokumen, aturan ini digunakan untuk memproses semua dokumen undang-undang yang memiliki format XML. Dari dokumen XML ini akan dilakukan pemrosesan pada bagian judul dan batang tubuh untuk mencari keterkaitan antar dokumen.

Proses pembuatan aturan dibantu dengan informasi kontekstual yang terdiri dari informasi kata-kata kunci, informasi kata hubung, dan informasi frase khusus. Informasi kata kunci adalah kata-kata yang dapat menunjukkan secara langsung bagian-bagian atau subbagian-subbagian dalam bagian tersebut. Contohnya adalah kata “menimbang” yang dapat menunjukkan sub bagian konsiderans. Informasi kata hubung adalah kata-kata hubung yang melekat atau berdampingan dengan kata-kata kunci dalam dokumen. Informasi frase khusus, yaitu gabungan kata-kata yang memiliki arti tertentu. Contohnya frase “Dengan Rahmat Tuhan Yang Maha Esa” menunjukkan sub bagian frase syukur dalam bagian pembukaan.

Sesuai dengan proses pembuatan aturan dalam pendekatan *knowledge engineering*, proses pengembangan aturan-aturan dilakukan secara iteratif. Proses iteratif dilakukan dengan membuat aturan yang kemudian diimplementasikan dalam sistem. Kemudian sistem tersebut dijalankan dengan memasukkan *input* berupa dokumen observasi yang sudah dalam format XML.

Output atau keluaran sistem dianalisis untuk menentukan kecukupan aturan-aturan yang telah dibuat. Proses analisis dilakukan dengan cara membandingkan keterkaitan antar dokumen yang dihasilkan oleh sistem dengan keterkaitan antar dokumen yang telah dibuat secara manual oleh penulis.

3.4 Pembuatan Referensi Dokumen Legal

Dokumen XML yang dihasilkan oleh fitur standardisasi dokumen akan menjadi input dari fitur pembuatan graf referensi dokumen. Fitur tersebut terdiri atas dua proses, yaitu proses pencarian referensi dan proses pembuatan dokumen DOT. Pada subbab ini dibahas proses yang pertama, yakni proses pencarian referensi.

Proses pencarian referensi diawali dengan penguraian elemen XML menjadi token-token. Token yang dihasilkan berupa elemen-elemen XML tanpa tanda baca ‘<’ dan ‘>’ pada *tag* pembuka dan *tag* penutupnya. Setelah itu dilakukan pengambilan ID dari dokumen yang sedang diproses. ID diambil dari isi atribut ID yang terdapat pada elemen akar DOKUMEN. Sebagai *node* pertama dari graf dibuat

node dengan nama sesuai ID yang diperoleh tersebut. *Node* ini akan memiliki hubungan dengan *node* lain yang menggambarkan keterkaitan dokumen yang diproses dengan dokumen lain.

Node dari graf dibuat berdasarkan elemen kosong REF yang ditemukan dalam dokumen yang sedang diproses. Pencarian elemen REF dilakukan dengan cara mencari token yang diawali dengan kata 'REF ID='. Selanjutnya, proses substring dilakukan pada setiap token yang cocok sehingga didapatkan isi atribut ID. Setiap ID yang ditemukan dibuatkan *node* nya dan *node* tersebut memiliki hubungan dengan *node* yang bernama sesuai ID dokumen yang diproses.

Target referensi suatu dokumen dapat berupa referensi terhadap dokumen berdasarkan judul dan pasal. Berikut aturan pencarian ID dari setiap dokumen yang menjadi target referensi tersebut.

Tabel 3.1 Aturan Pencarian Referensi Judul

Aturan untuk mengetahui Judul dari UU	
JIKA	Jika pada baris ditemukan tag <DOKUMEN>
MAKA	ID yang terdapat dalam tag tersebut merupakan judul dari UU
Aturan untuk menemukan jenis UU perubahan Pertama	
JIKA	Jika pada baris ditemukan tag <NAMA_PERATURAN> dan suatu string "PERUBAHAN ATAS"
MAKA	Undang undang tersebut merupakan undang-undang perubahan pertama
Aturan untuk menemukan jenis UU perubahan yang telah diubah lebih dari satu kali	
JIKA	Jika baris ditemukan tag <NAMA_PERATURAN> dan suatu string "PERUBAHAN ATAS" dan diantara kata PERUBAHAN dan kata ATAS disisipkan keterangan yang menunjukkan berapa kali perubahan tersebut telah dilakukan, kedua, ketiga, dan seterusnya.

MAKA	Undang undang tersebut merupakan undang-undang perubahan yang telah dilakukan lebih dari satu kali
Kemudian akan dibuat graf berarah dengan node disebelah kiri menyatakan UU yang diubah dan node yang disebelah kanan menyatakan node UU yang mengubah	

Tabel 3.2: Aturan Pencarian Referensi Pasal

Aturan untuk mengetahui pasal Inti pada UU perubahan	
JIKA	Didalam undang-undang perubahan barisnya ditemukan tag <PASAL>, yang diikuti nomor urut berupa angka Romawi (I, II, III, dan seterusnya)
MAKA	ID dari Pasal tersebut merupakan Pasal inti dari undang-undang perubahan
Aturan untuk mengetahui pasal yang diubah yang terdapat pada pasal Inti UU perubahan	
JIKA	Didalam undang-undang perubahan barisnya ditemukan tag <PASAL>, yang diikuti nomor urut berupa angka Arab(1,2,3, dan seterusnya)
MAKA	ID dari Pasal tersebut merupakan Pasal perubahan dari undang-undang perubahan
Kemudian akan dibuat graf berarah dengan node disebelah kiri menyatakan pasal UU yang diubah dan node yang disebelah kanan menyatakan node pasal UU yang mengubah yaitu pasal inti dari UU perubahan	

3.5 Pembuatan Dokumen DOT

Setelah semua referensi yang dimiliki dokumen ditemukan maka proses selanjutnya adalah pembuatan dokumen DOT. Pembuatan dokumen ini dilakukan dengan cara menyatukan seluruh referensi yang ditemukan sebelumnya dalam satu berkas (*file*) dan menyimpannya dalam format *.dot*. Namun untuk memenuhi syarat pembentukan dokumen DOT, seluruh referensi tersebut diletakkan di antara tanda kurung kurawal dan di bagian awal dokumen ditambahkan kata kunci *digraph* diikuti nama graf. Selain itu dapat juga dilakukan penambahan atribut pada *node* dan *edge* dari graf untuk memperindah tampilan graf.

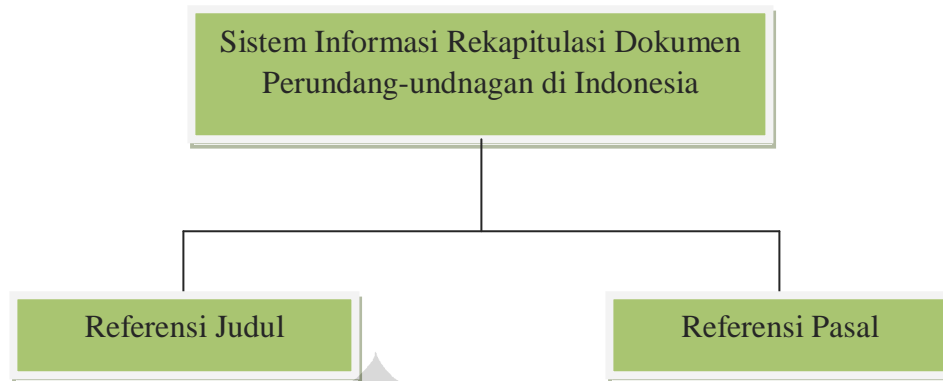
Dokumen DOT yang dihasilkan berisi pendefinisian properti yang dimiliki suatu graf referensi. Dokumen tersebut hanya terbatas pada pendefinisian hubungan antar-*node* yang terdapat dalam graf. Untuk memvisualisasikan keterhubungan tersebut, maka *file* DOT dapat diproses lebih lanjut dengan menggunakan aplikasi *Graphviz*.

3.6 Rancangan Sistem

Bagian ini akan menjelaskan tentang perancangan sistem yang akan dikembangkan berdasarkan hasil analisis kebutuhan yang telah dilakukan sebelumnya. Rancangan sistem terbagi menjadi dua bagian yang terdiri dari desain modul dan desain antar muka.

3.6.1 Desain Modul Referensi

Berdasarkan kebutuhan dari sistem yang telah dianalisis sebelumnya, maka diperoleh dua modul yang akan menyusun sistem rekapitulasi dokumen perundang-undangan ini, yaitu modul referensi judul dan modul referensi pasal. Desain modul tersebut diilustrasikan pada Gambar 3.2.



Gambar 3.2 : Desain Modul

Berikut adalah penjelasan dari masing-masing modul yang digambarkan diatas.

1. Modul Referensi Judul

Modul ini berguna untuk melakukan pencarian sejarah perubahan undang-undang yang berdasarkan judul. Untuk data masukan berupa undang-undang yang sudah dalam format XML yang didapat dari sistem yang telah dikembangkan oleh Violina[VIO08], modul ini melakukan pemrosesan dokumen XML tersebut untuk mendapatkan sejarah perubahan judul dari setiap dokumen.

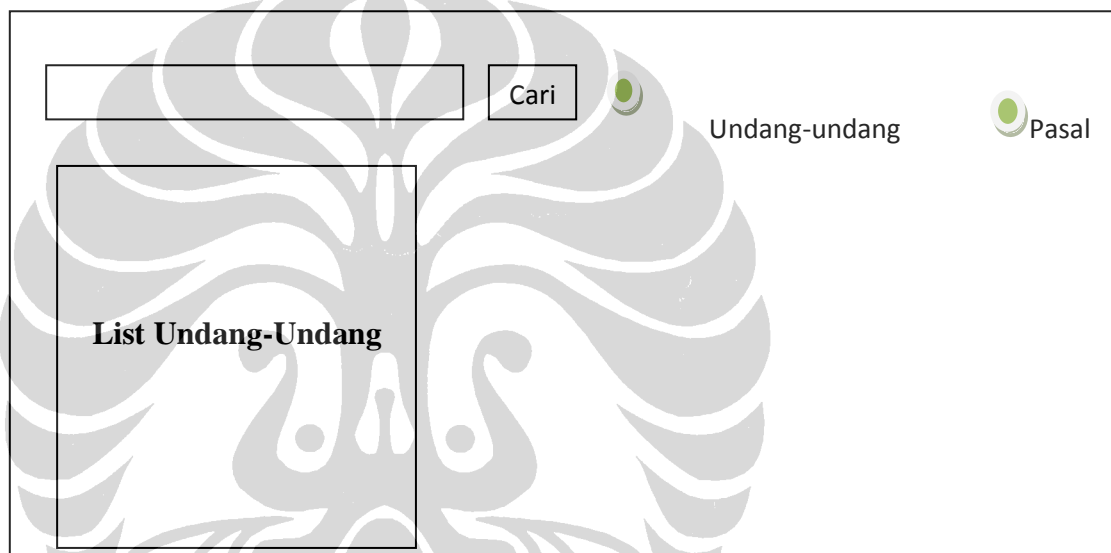
2. Modul Referensi Pasal

Modul ini berguna untuk melakukan pencarian sejarah perubahan undang-undang berdasarkan pasal yang ada disetiap dokumen legal. Untuk dokumen masukan berupa pasal dari undang-undang yang sudah memiliki format XML yang didapat dengan menjalankan salah satu program pada sistem perolehan informasi dokumen legal yang memecah dokumen undang-undang menjadi pasal yang telah dikembangkan oleh [PRA09], modul ini melakukan

pemrosesan dokumen XML tersebut untuk mendapatkan sejarah perubahan pasal dari setiap dokumen.

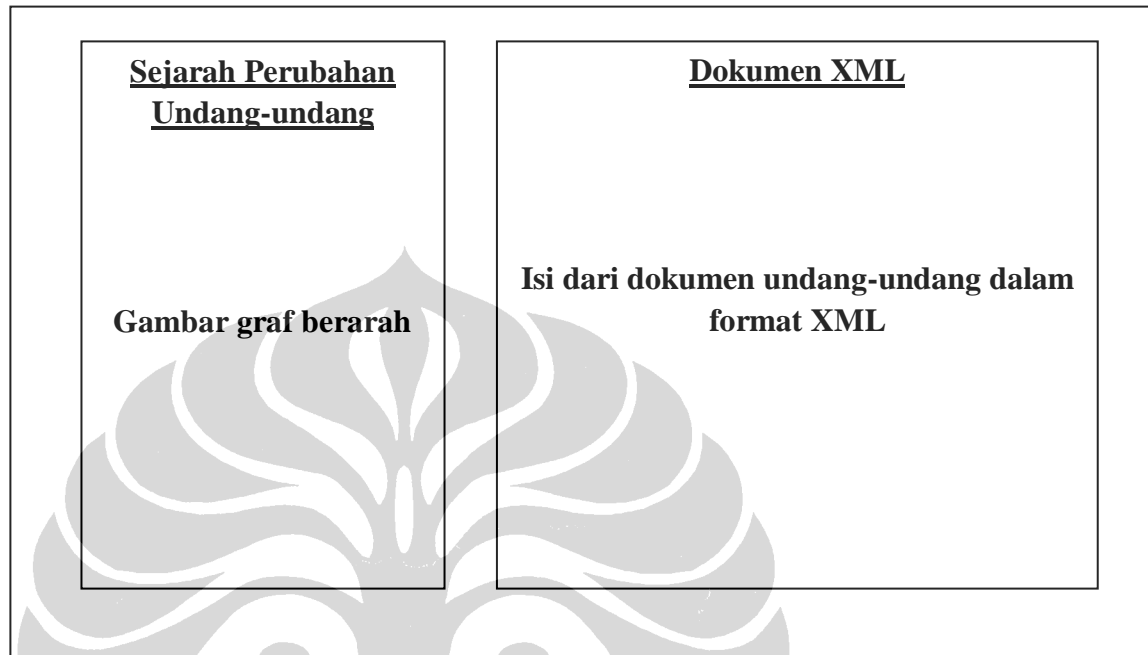
3.6.2 Desain Antar Muka

Sistem yang dikembangkan merupakan sistem yang membutuhkan interaksi dengan pengguna, oleh sebab itu sistem ini akan berbasis *Graphical User Interface* (GUI). Berdasarkan kebutuhan yang telah diidentifikasi sebelumnya, maka desain GUI yang dapat memfasilitasi kebutuhan tersebut akan digambarkan pada Gambar 3.3.



Gambar 3.3: Desain Antar Muka Sistem Perolehan Informasi Dokumen Legal

Dari Gambar 3.3 terlihat halaman depan dari sistem perolehan informasi dokumen legal, dihalaman ini pengguna sistem bisa memasukan kata kunci atau *query* untuk mencari dokumen undang-undang, hasil pencarian bisa ditampilkan berdasarkan undang-undang (judul undang-undang) dan berdasarkan pasal dari suatu undang-undang. Berikut halaman sistem informasi rekapitulasi dokumen perundang-undang di Indonesia dapat dilihat pada Gambar 3.4.



Gambar 3.4: Antar Muka Sistem Rekapitulasi Dokumen Perundang-undangan di Indonesia

Dari Gambar 3.4 terlihat bahwa rancangan dari desain antar muka sistem, yang akan menampilkan sejarah dari undang-undang berdasarkan input masukan dari sistem, sistem akan menampilkan sejarah undang-undang dalam bentuk graf berarah, dimana pada setiap node dari graf ini dapat informatif dengan pengguna, yakni setiap node bisa diklik dan menampilkan isi dari undang-undang tersebut.

BAB 4

IMPLEMENTASI

Bab ini menjelaskan tentang implementasi dari sistem yang dikembangkan. Penjelasan yang diberikan meliputi komponen perangkat keras, komponen perangkat lunak, dan struktur implementasi dari pengembangan sistem serta antar muka dari sistem yang telah dikembangkan.

4.1 Spesifikasi Sistem

Pada bagian spesifikasi sistem ini dijelaskan mengenai komponen-komponen yang digunakan untuk membangun sistem. Komponen-komponen yang dijelaskan meliputi komponen perangkat keras dan perangkat lunak.

4.1.1 Perangkat Keras

Sistem informasi rekapitulasi dokumen perundang-undangan di Indonesia ini dikembangkan pada *notebook* dengan spesifikasi perangkat keras sebagai berikut:

Prosesor	: Intel(R) Core(TM) 2 Duo CPU T5250 @1.50 GHz 1.50GHz
Memori	: 1024 MB
Harddisk	: 160 GB

4.1.2 Perangkat Lunak

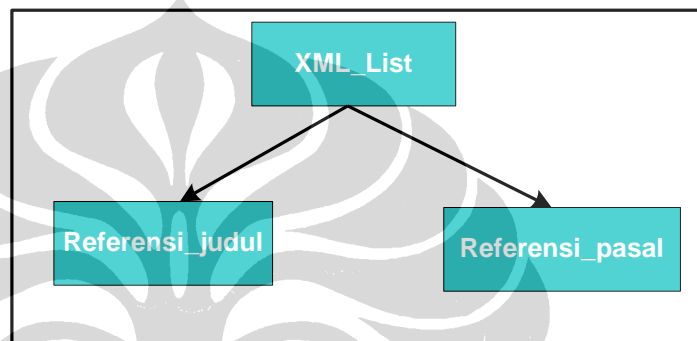
Perangkat lunak yang digunakan untuk mengembangkan sistem ini adalah sebagai berikut:

Sistem Operasi	: Microsoft Windows Vista
Internet Browser	: Mozilla Firefox 2.0, Internet Explorer
Web Server	: Apache
Text Editor	: Scite Text Editor
Dokumentasi	: MS. Word, MS. Visio
Library	: ActivePerl 5.10.0 , PHP, Graphviz-2.18

4.2 Struktur Implementasi Sistem Informasi Rekapitulasi Dokumen

Perundang-undangan di Indonesia

Sistem informasi rekapitulasi dokumen perundang-undangan di Indonesia yang dikembangkan dalam penelitian ini diimplementasikan dengan menggunakan bahasa pemrograman PERL dan PHP. Hasil implementasinya dapat dilihat pada Gambar 4.1.



Gambar 4.1: Keterhubungan Antar-Kelas

Pada Gambar 4.1 dapat dilihat bahwa hubungan langsung antara kelas XML_List dengan kelas Referensi_judul dan kelas Referensi_pasal. Penjelasan masing-masing kelas dapat dilihat pada subbab 4.2.1 sampai 4.2.3.

4.2.1 Kelas XML_List

Kelasa XML_list bertugas untuk melakukan pemanggilan terhadap semua kelas berdasarkan *query* yang dimasukkan oleh pengguna, dikelas ini akan dipanggil kelas tertentu untuk menampilkan sejarah perubahan undang-undang. Dibawan ini akan diberikan potongan program dengan menggunakan bahasa pemrograman PHP untuk melakukan proses pemanggilan kelas-kelas.

```

<table>
<tr>
<td style="width:30%;background:#E8EDFC;vertical-align:top">
  <h3><center>Sejarah Undang-undang</center></h3>
  <div>
  <?php

    if (eregi("dok-uu", $file)) {
      include "proses_dok_dot_Referensi_judul.php";
      include "proses_dok_dot_Referensi_judul_html.php";
    }
    else if (eregi("dok-pasal", $file))
    {
      include "proses_dok_dot_Referensi_pasal.php";
      include "proses_dok_dot_Referensi_pasal_html.php";
    }
  <?>
  </div>
</td>
</tr>
</table>

```

Gambar 4.2: Program untuk Kelas XML_List

Dari Gambar 4.2 terlihat bahwa pada kelas Indeks terdapat pemanggilan kelas lain, dimana pemanggilan kelas ini berdasarkan input atau *query* masukkan dari pengguna.

4.2.2 Kelas Referensi Judul

Kelas Referensi Judul merupakan kelas yang bertugas untuk membuat graf referensi dokumen undang-undang. Proses dalam kelas ini diawali dengan penguraian elemen-elemen XML menjadi token-token. Token-token tersebut kemudian diperiksa untuk mencari elemen REF yang menunjukkan referensi yang dimiliki dokumen ke dokumen legal lain. Seluruh elemen REF yang ditemukan disatukan dalam satu dokumen dalam format DOT dengan terlebih dahulu mengubahnya dalam format keterhubungan antar-*node*.

Dalam kelas Referensi_judul untuk mengetahui judul dari suatu dokumen undang-undang dapat dilihat dari potongan program pada gambar 4.3. Dalam kelas ini juga kita bisa mengetahui suatu undang-undang perubahan sudah diubah berapa kali.

```

if ($line =~ /<NAMA_PERATURAN>.*PERUBAHAN(| KEDUA| KETIGA)
    ATAS.*<REF.*ID="(.)".*\|>/)
{
    $perubahan = 1;
    #UU perubahan pertama
    if ($1 eq "")
    {
        $n = "perubahan pertama";
    }
    #UU perubahan kedua
    elsif ($1 eq " KEDUA")
    {
        $n = "perubahan kedua";
    }
    #UU perubahan ketiga
    elsif ($1 eq " KETIGA")
    {
        $n = "perubahan ketiga";
    }
    #UU perubahan keempat
    elsif ($1 eq " KEEMPAT")
    {
        $n = "perubahan keempat";
    }
}

```

Gambar 4.3: Program untuk Mengetahui UU Perubahan

Untuk membuat referensi antar dokumen maka seluruh elemen REF yang ditemukan disatukan dalam satu dokumen dalam format DOT dengan terlebih dahulu mengubahnya dalam format keterhubungan antar-*node*, proses pembuatan referensi dan graf berarah dapat dilihat pada Gambar 4.4.

```

print "$dokid [label=\"$dokstr ($n)\"] URL=\"$dok-uu/$dokfile\"; \n";
print "$revid [label=\"$revstr\", URL=\"$dok-uu/$revfile\"; \n$revid
-> $dokid; \n"; #mencetak referensi dokumen

print "digraph {\nrankdir=LR; \n"; //mencetak graf berarah

```

Gambar 4.4: Mencetak Referensi dari UU

Contoh dokumen yang memiliki format .dot yang dihasilkan dari potongan program pada Gambar 4.3 dan pada gambar 4.4 dapat dilihat pada Gambar 4.5.


```

digraph {
rankdir=LR;
uu_4_2000 [label="uu-4-2000 ( perubahan pertama)" URL="dok-uu/uu-2000-004.xml"];
uu_3_1999 [label="uu-3-1999", URL="dok-uu/uu-1999-003.xml"];
uu_3_1999 -> uu_4_2000;
uu_9_2000 [label="uu-9-2000 ( perubahan pertama)" URL="dok-uu/uu-2000-009.xml"];
uu_49_1999 [label="uu-49-1999", URL="dok-uu/uu-1999-049.xml"];
uu_49_1999 -> uu_9_2000;
uu_10_2006 [label="uu-10-2006 ( perubahan kedua)" URL="dok-uu/uu-2006-010.xml"];
uu_12_2003 [label="uu-12-2003", URL="dok-uu/uu-2003-012.xml"];
uu_12_2003 -> uu_10_2006;
}

```

Gambar 4.5: Dokumen UU dalam Format .dot

Pada Gambar 4.5 dapat dilihat hasil dokumen yang memiliki format .dot, dimana pada dokumen tersebut diawali dengan “digraph” yang menyatakan bahwa dokumen yang terbentuk merupakan graf berarah (*directed graf*). Nama setiap node dilambangkan dengan label dan referensi dokumen tersebut terdapat pada URL di file dok-uu. Keterkaitan antar dokumen atau keterkaitan antar node dilambangkan dengan tanda panah “->”.

4.2.2.1 Kelas Proses dokumen judul dalam format DOT

Kelas ini melakukan pemrosesan dokumen yang sudah memiliki format .dot yang dihasilkan dari kelas Referensi_judul dimana pada kelas dokumen .dot tersebut terdapat keterkaitan semua dokumen undang-undang yang menjadi input yang memiliki format XML. Kelas ini akan mencarikan sejarah perubahan undang-undang berdasarkan input undang-undang yang masuk kedalam sistem rekapitulasi dokumen perundang-undangan di Indonesia.

Kelas ini akan menghasilkan *output* berupa dokumen yang memiliki format .dot juga namun pada dokumen ini hanya terdapat referensi yang terkait dengan undang-undang yang menjadi input dalam sistem rekapitulasi. Program pada kelas ini dapat dilihat pada Gambar 4.6.

```

<?
$filename="Referensi_judul.dot";
$output="proses_dok_dot_Referensi_judul.dot";
$handle = fopen($filename, "r"); //open file
$content = fread($handle, filesize($filename));
$line = preg_split('/\n/', $content);
fclose($handle);
preg_match('/(..)(dok-.*)(uu.*)(.xml)/i', $file, $matches);
$handle = fopen("$output", "w");
if (preg_match('/uu-(.*)-00(./i', $matches[3], $match))
{
    $uu = 'uu_'. $match[2].'_'. $match[1];
}
else if (preg_match('/uu-(.*)-0(..)/i', $matches[3], $match))
{
    $uu = 'uu_'. $match[2].'_'. $match[1];
}
fwrite($handle, $line[0]); //tulis digraph { ke file
fwrite($handle, $line[1]); //tulis rankdir=LR; ke file
$ada_history = 0;
for ($ii = 2; $ii < count($line); )
{
    $a = $line[$ii];$ii++;
    $b = $line[$ii];$ii++;
    $c = $line[$ii];$ii++;
    if (eregi($matches[3], $a) or eregi($matches[3], $b))
    {
        $ada_history = 1;
        fwrite($handle, $a);
        fwrite($handle, $b);
        fwrite($handle, $c)
    }
    elseif (eregi("}", $a) and $ada_history == 1)
    {
        fwrite($handle, "$uu [color=blue fontcolor=blue
fontsize=14];");
    }
}
fwrite($handle, "}"); fclose($handle);
$cmd = 'dot proses_dok_dot_Referensi_judul.dot -Tpng -
oproses_dok_dot_Referensi_judul.png -Tcmapx -
oproses_dok_dot_Referensi_judul.map';
$output = shell_exec($cmd." 2>&1"); //system call
$cmd = 'perl konversi_html_Referensi_judul.pl';
$output = shell_exec($cmd." 2>&1"); //system call
?>

```

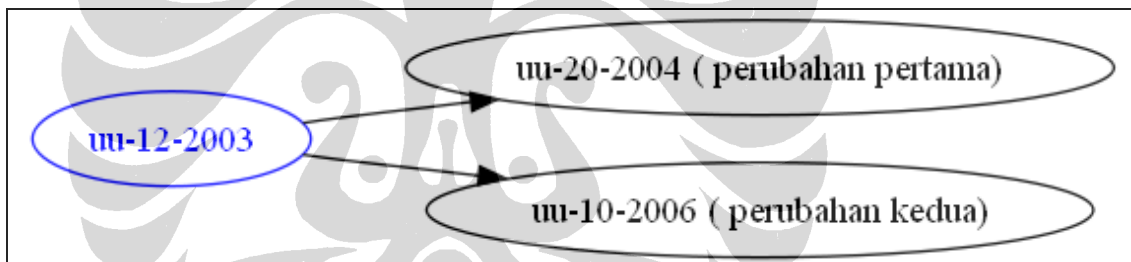
Gambar 4.6: Program untuk Proses Dokumen Judul UU dalam Format .dot

Contoh dokumen .dot yang dihasilkan dari program diatas adalah sejarah perubahan dari UU.

```
digraph {
rankdir=LR;
uu_10_2006 [label="uu-10-2006 ( perubahan kedua)" URL="dok-uu/uu-2006-010.xml"];
uu_12_2003 [label="uu-12-2003", URL="dok-uu/uu-2003-012.xml"];
uu_12_2003 -> uu_10_2006;
uu_10_2006 [color=blue fontcolor=blue fontsize=14];}
```

Gambar 4.7: Dokumen Sejarah Perubahan UU dalam Format .dot

File .png yang dihasilkan oleh program diatas adalah gambar graf berarah yang menyatakan sejarah perubahan dari UU, dimana node sebelah kiri menyatakan UU yang diubah dan node yang disebelah kanan menyatakan UU yang mengubah dan dicantumkan juga UU tersebut merupakan perubahan keberapa, perubahan tersebut bisa pertama, kedua, ketiga dan keempat.



Gambar 4.8: Sejarah Perubahan UU dalam Bentuk Graf Berarah

File .map yang dihasilkan oleh program diatas adalah:

```
<map id="_anonymous_0" name="_anonymous_0">
<area shape="poly" href="dok&#45;uu/uu&#45;2004&#45;020.xml"
title="uu&#45;20&#45;2004 ( perubahan pertama)" alt=""
coords="195,30 204,37 229,44 269,49 319,52 375,54 430,52 480,49
520,44 546,37 554,30 546,22 520,16 480,10 430,7 375,6 319,7 269,10
229,16 204,22"/>

<area shape="poly" href="dok&#45;uu/uu&#45;2003&#45;012.xml"
title="uu&#45;12&#45;2003" alt="" coords="5,66 9,73 19,80 34,85
54,88 76,90 98,88 118,85 133,80 143,73 147,66 143,58 133,52 118,46
98,43 76,42 54,43 34,46 19,52 9,58"/>
```

```

<area shape="poly" href="dok&#45;uu/uu&#45;2006&#45;010.xml"
title="uu&#45;10&#45;2006 ( perubahan kedua)" alt=""
coords="205,102 213,109 237,116 275,121 322,124 375,126 427,124
475,121 512,116 536,109 545,102 536,94 512,88 475,82 427,79 375,78
322,79 275,82 237,88 213,94"/>
</map>

```

Gambar 4.9: File Keterkaitan Judul yang Memiliki Format .map

4.2.2.2 Kelas Konversi HTML judul

Kelas ini melakukan pemrosesan pada dokumen yang dihasilkan dari kelas proses dokumen judul, dimana dokumen yang dihasilkan dalam format .map dan .png.

```

open(IN, "proses_dok_dot_Referensi_judul.map");
open (OUT, ">proses_dok_dot_Referensi_judul_html.php");

print OUT "<html>\n";
print OUT "<head />\n";
print OUT "<body bgcolor=\"silver\">\n";
print OUT "<center>\n";
print OUT "<img src=\"proses_dok_dot_Referensi_judul.png\"
usemap=\"#testmap\" /> </center>\n";
print OUT "<map id=\"testmap\" name=\"testmap\">\n";
while ($baris=<IN>)
{
if ($baris =~ /\<map id="(.)" name="(.)"/>/)
{
}
elseif ($baris =~ s/href="dok&#45;uu/href="xml-
parser.php?file=./dok-uu/)
{
print OUT $baris."\n";
}
else
{
print OUT $baris."\n";
}
}
}

```

Gambar 4.10: Program Kelas Konversi HTML Judul

File yang dihasilkan dari program pada Gambar 4.10 adalah:

```
<html>
<head />
<body bgcolor="silver">
<center>

</center>
<map id="testmap" name="testmap">
<area shape="poly" href="xml-parser.php?file=./dok-
uu/uu&#45;2006&#45;010.xml" title="uu&#45;10&#45;2006 ( perubahan
kedua)" alt="" coords="195,30 204,37 228,44 265,49 313,52 365,54
418,52 465,49 503,44 527,37 535,30 527,22 503,16 465,10 418,7
365,6 313,7 265,10 228,16 204,22"/>
<area shape="poly" href="xml-parser.php?file=./dok-
uu/uu&#45;2003&#45;012.xml" title="uu&#45;12&#45;2003" alt=""
coords="5,30 9,37 19,44 34,49 54,52 76,54 98,52 118,49 133,44
143,37 147,30 143,22 133,16 118,10 98,7 76,6 54,7 34,10 19,16
9,22"/>
</map>
```

Gambar 4.11: Output Kelas Konversi HTML Judul

4.2.3 Kelas Referensi Pasal

Kelas ini melakukan pemrosesan pada setiap dokumen undang-undang untuk mengetahui apakah undang-undang tersebut merupakan undang-undang perubahan atau bukan, dikelas ini juga akan diketahui merupakan perubahan yang keberapa kali apabila undang-undang tersebut merupakan undang-undang perubahan, perubahan tersebut bisa perubahan pertama, kedua, ketiga , dan keempat.

```
if ($line =~ /<NAMA_PERATURAN>.*PERUBAHAN(| KEDUA| KETIGA)
    ATAS.*<REF.*ID="(.)".*\|>/)
{
    $perubahan = 1;

    if ($1 eq "")
    {
        $n = " perubahan pertama";
    }
    elsif ($1 eq " KEDUA")
    {
        $n = " perubahan kedua";
    }
    elsif ($1 eq " KETIGA")
    {
```

```

        $n = " perubahan ketiga";
    }
    elsif ($1 eq " KEEMPAT")
    {
        $n = " perubahan keempat";
    }
}

```

Gambar 4.12: Program untuk Mengetahui UU Perubahan

Potongan program pada Gambar 4.13 merupakan program untuk mengetahui pasal Inti dan pasal perubahan dari undang-undang perubahan.

```

elsif ($perubahan == 1 and $line =~ /\<PASAL NO="I*"
ID="(.*)"\>/)
{
    $dokid=$1;
    $dokstr = $1;
    $dokfile="$1.xml";
    $dokid =~ s/-/_/g;
    $dokid =~ s/\./_/g;
    $dokstr = $dokid;
}
elsif ($perubahan == 1 and $line =~ /\<PASAL NO="[1-9]*"
ID="(.*)"\>/)
{
    $revid=$1;
    $revstr=$1;
    $revfile="$1.xml";
    $revid =~ s/-/_/g;
    $revid =~ s/\./_/g;
    $revstr = $revid;
    #pasal yg dirubah
    print "$dokid [label=\"\$dokstr ($n)\"] URL=\"dok
    pasal\/\$dokfile\"";\n";
    print "$revid [label=\"\$revstr\", URL=\"dok-
    pasal\/\$revfile\"";\n\$revid -> $dokid;\n";
}

```

Gambar 4.13: Program Pasal Inti dan Pasal Perubahan dari UU Perubahan

4.2.3.1 Kelas Proses pasal dalam format DOT

Kelas ini melakukan pemrosesan dokumen yang sudah memiliki format .dot yang dihasilkan dari kelas Referensi_pasal dimana pada kelas dokumen .dot tersebut terdapat keterkaitan semua dokumen pasal undang-undang yang menjadi input yang memiliki format XML. Kelas ini akan mencarikan sejarah perubahan dari pasal undang-undang berdasarkan input pasal undang-undang yang masuk kedalam sistem rekapitulasi dokumen perundang-undangan di Indonesia.

Kelas ini akan menghasilkan *output* berupa dokumen yang memiliki format .dot juga namun pada dokumen ini hanya terdapat referensi yang terkait dengan pasal undang-undang yang menjadi input dalam sistem rekapitulasi dokumen undang-undang. Program pada kelas ini dapat dilihat pada Gambar 4.14.

```
<?
$filename="Referensi_pasal.dot";
$output="proses_dok_dot_Referensi_pasal.dot";
$handle = fopen($filename, "r"); //open file
$content = fread($handle, filesize($filename));
$line = preg_split('/\n/', $content);
fclose($handle);
global $file;
preg_match('/(..)(dok-.*)(uu.*.xml)/i', $file, $matches);
$handle = fopen("$output", "w");
if (preg_match('/uu-(.*)-(....).ps-(.*)xml/i', $matches[3],
$match))
{
    $pasal = 'uu_'. $match[1].'_'. $match[2].'_ps_'. $match[3];
}

fwrite($handle, $line[0]); //tuliskan digraph { ke file
fwrite($handle, $line[1]); //tuliskan rankdir=LR; ke file
$ada_history = 0;
//baca dari baris ketiga, empat, lima
for ($i = 2; $i < count($line); )
{
    $a = $line[$i];$i++;
    $b = $line[$i];$i++;
    $c = $line[$i];$i++;
    if (eregi($matches[3], $a) or eregi($matches[3], $b))
    {
        $ada_history = 1;
        fwrite($handle, $a);
        fwrite($handle, $b);
        fwrite($handle, $c);
    }
}
```

```

    }
    elseif (eregi(")", $a) and $ada_history == 1)
    {
        fwrite($handle, "$pasal [color=blue fontcolor=blue
fontsize=14];");
    }
}
fwrite($handle, "");
fclose($handle);
$cmd = 'dot proses_dok_dot_Referensi_pasal.dot -Tpng -
oproces_dok_dot_Referensi_pasal.png -Tcmapx -
oproces_dok_dot_Referensi_pasal.map';
$output = shell_exec($cmd." 2>&1"); //system call
$cmd = 'perl konversi_html_Referensi_pasal.pl';
$output = shell_exec($cmd." 2>&1"); //system call
?>

```

Gambar 4.14: Program untuk Proses Dokumen Pasal UU Dalam Format .dot

Contoh dokumen .dot yang dihasilkan dari program diatas adalah sejarah perubahan dari pasal UU, dapat dilihat pada Gambar 4.15.

```

digraph {
rankdir=LR;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_26 [label="uu_32_2004_ps_26", URL="dok-pasal/uu-32-2004.ps-
26.xml"];
uu_32_2004_ps_26 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_42 [label="uu_32_2004_ps_42", URL="dok-pasal/uu-32-2004.ps-
42.xml"];
uu_32_2004_ps_42 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_56 [label="uu_32_2004_ps_56", URL="dok-pasal/uu-32-2004.ps-
56.xml"];
uu_32_2004_ps_56 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_58 [label="uu_32_2004_ps_58", URL="dok-pasal/uu-32-2004.ps-
58.xml"];
uu_32_2004_ps_58 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_59 [label="uu_32_2004_ps_59", URL="dok-pasal/uu-32-2004.ps-
59.xml"];
uu_32_2004_ps_59 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_62 [label="uu_32_2004_ps_62", URL="dok-pasal/uu-32-2004.ps-

```



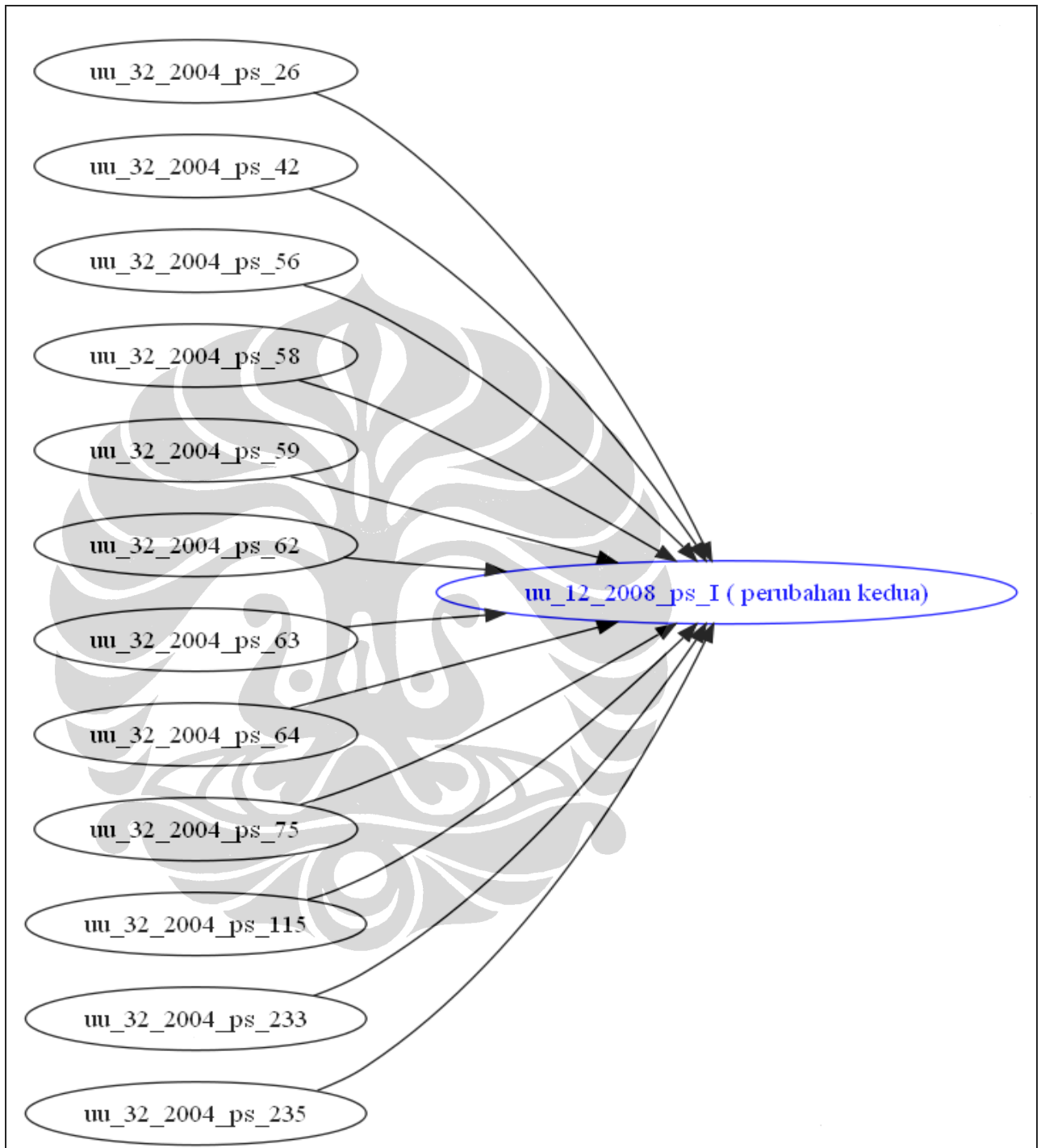
```

62.xml"];
uu_32_2004_ps_62 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_63 [label="uu_32_2004_ps_63", URL="dok-pasal/uu-32-2004.ps-
63.xml"];
uu_32_2004_ps_63 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_64 [label="uu_32_2004_ps_64", URL="dok-pasal/uu-32-2004.ps-
64.xml"];
uu_32_2004_ps_64 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_75 [label="uu_32_2004_ps_75", URL="dok-pasal/uu-32-2004.ps-
75.xml"];
uu_32_2004_ps_75 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_115 [label="uu_32_2004_ps_115", URL="dok-pasal/uu-32-
2004.ps-115.xml"];
uu_32_2004_ps_115 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [label="uu_12_2008_ps_I ( perubahan kedua)" URL="dok-
pasal/uu-12-2008.ps-I.xml"];
uu_32_2004_ps_235 [label="uu_32_2004_ps_235", URL="dok-pasal/uu-32-
2004.ps-235.xml"];
uu_32_2004_ps_235 -> uu_12_2008_ps_I;
uu_12_2008_ps_I [color=blue fontcolor=blue fontsize=14];}

```

Gambar 4.15: File Keterkaitan Pasal yang Memiliki Format .map

File .png yang dihasilkan oleh program pada Gambar 4.14 adalah:



Gambar 4.16: Sejarah Perubahan Pasal UU Dalam Bentuk Graf Berarah

4.2.3.2 Kelas Konversi HTML Pasal

Kelas ini melakukan pemrosesan pada dokumen yang dihasilkan dari kelas proses dokumen pasal, dimana dokumen yang dihasilkan dalam format .map dan .png.

```

open (IN, "proses_dok_dot_Referensi_pasal.map");
open (OUT, ">proses_dok_dot_Referensi_pasal_html.php");

print OUT "<html>\n";
print OUT "<head />\n";
print OUT "<body bgcolor=\"silver\">\n";
print OUT "<center>\n";
print OUT "<img src=\"proses_dok_dot_Referensi_pasal.png\"
usemap=\"#testmap\" /> </center>\n";
print OUT "<map id=\"testmap\" name=\"testmap\">\n";

while ($baris=<IN>)
{
    if ($baris =~ /\<map id="(.)" name="(.)" \>/)
    {;
    }
    elsif ($baris =~ s/href="dok&#45;pasal/href="xml-
parser.php?file=./dok-pasal/)
    {
        print OUT $baris."\\n";
    }
    else
    {
        print OUT $baris."\\n";
    }
}

```

Gambar 4.17: Program Kelas Konversi HTML Pasal

File yang dihasilkan dari program pada Gambar 4.17 adalah:

```

<html>
<head />
<body bgcolor="silver">
<center>

</center>
<map id="testmap" name="testmap">
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;12&#45;2008.ps&#45;I.xml" title="uu 12 2008 ps I (

```

```

perubahan kedua)" alt="" coords="294,426 304,433 333,440 378,445
435,448 499,450 562,448 619,445 664,440 693,433 703,426 693,418
664,412 619,406 562,403 499,402 435,403 378,406 333,412 304,418"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;26.xml" title="uu_32_2004_ps_26"
alt="" coords="12,30 17,37 33,44 59,49 90,52 125,54 160,52 192,49
217,44 233,37 239,30 233,22 217,16 192,10 160,7 125,6 90,7 59,10
33,16 17,22"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;42.xml" title="uu_32_2004_ps_42"
alt="" coords="12,102 17,109 33,116 59,121 90,124 125,126 160,124
192,121 217,116 233,109 239,102 233,94 217,88 192,82 160,79 125,78
90,79 59,82 33,88 17,94"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;62.xml" title="uu_32_2004_ps_62"
alt="" coords="12,390 17,397 33,404 59,409 90,412 125,414 160,412
192,409 217,404 233,397 239,390 233,382 217,376 192,370 160,367
125,366 90,367 59,370 33,376 17,382"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;63.xml" title="uu_32_2004_ps_63"
alt="" coords="12,462 17,469 33,476 59,481 90,484 125,486 160,484
192,481 217,476 233,469 239,462 233,454 217,448 192,442 160,439
125,438 90,439 59,442 33,448 17,454"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;64.xml" title="uu_32_2004_ps_64"
alt="" coords="12,534 17,541 33,548 59,553 90,556 125,558 160,556
192,553 217,548 233,541 239,534 233,526 217,520 192,514 160,511
125,510 90,511 59,514 33,520 17,526"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;75.xml" title="uu_32_2004_ps_75"
alt="" coords="12,606 17,613 33,620 59,625 90,628 125,630 160,628
192,625 217,620 233,613 239,606 233,598 217,592 192,586 160,583
125,582 90,583 59,586 33,592 17,598"/>
<area shape="poly" href="xml-parser.php?file=./dok-
pasal/uu&#45;32&#45;2004.ps&#45;233.xml" title="uu_32_2004_ps_233"
alt="" coords="5,750 11,757 28,764 55,769 88,772 125,774 162,772
196,769 222,764 239,757 245,750 239,742 222,736 196,730 162,727
125,726 88,727 55,730 28,736 11,742"/>
</map>

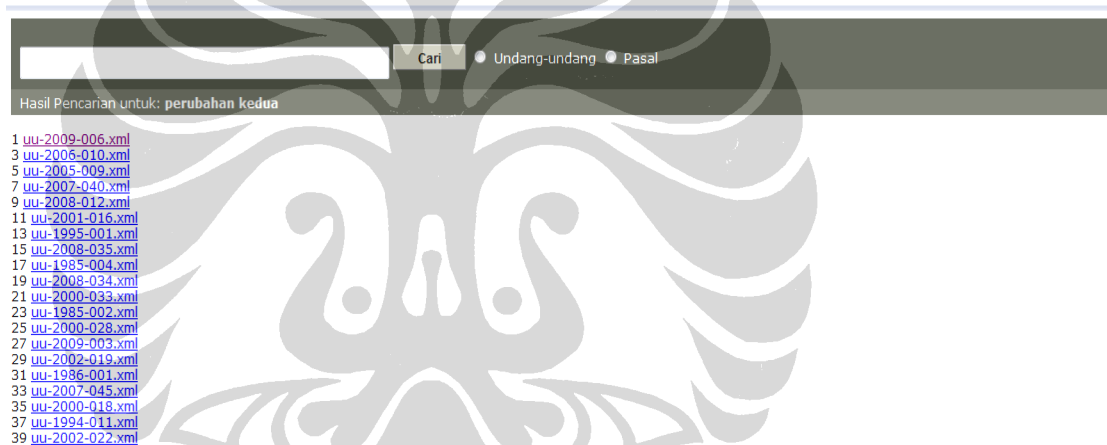
```

Gambar 4.18: Output Kelas Konversi HTML Pasal

4.3 Antar Muka

Antar muka dari sistem informasi rekapitulasi dokumen perundang-undangan di Indonesia ini dibuat dengan menggunakan HTML dan PHP. Berikut tampilan antar muka dari sistem yang telah dikembangkan yang berbasis *Graphical User Interface* (GUI).

Pada Gambar 4.19 merupakan halaman awal dari sistem perolehan informasi dokumen legal yang telah dibuat oleh [PRA09], yang menampilkan list undang-undang. Hasil atau output dari sistem ini merupakan input bagi sistem informasi rekapitulasi dokumen perundang-undangan di Indonesia yang telah penulis kembangkan.



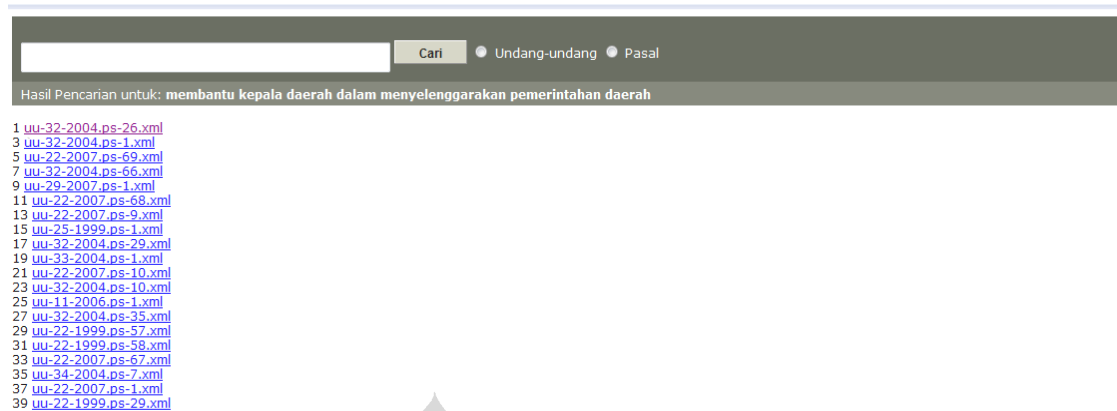
Gambar 4.19 : Halaman Output Sistem Perolehan Informasi Dokumen Legal

Pada Gambar 4.20 merupakan halaman sejarah perubahan Undang-undang yang ditampilkan dalam bentuk graf berarah, di mana *node* sebelah kiri merupakan undang-undang non perubahan dan *node* yang disebelah kanan merupakan undang-undang perubahan. Apabila *node* tersebut diklik oleh pengguna maka sistem akan menampilkan isi dari undang-undang tersebut dalam format XML.

The screenshot displays a web interface for tracking legislative changes. On the left, a diagram shows the evolution of UU No. 23 of 1999 through two amendments: UU No. 3 of 2004 (first amendment) and UU No. 6 of 2009 (second amendment). The main content area on the right provides detailed XML metadata for the document, including its title, year, and specific clauses (huruf) that have been amended or added. The interface is viewed through a Mozilla Firefox browser.

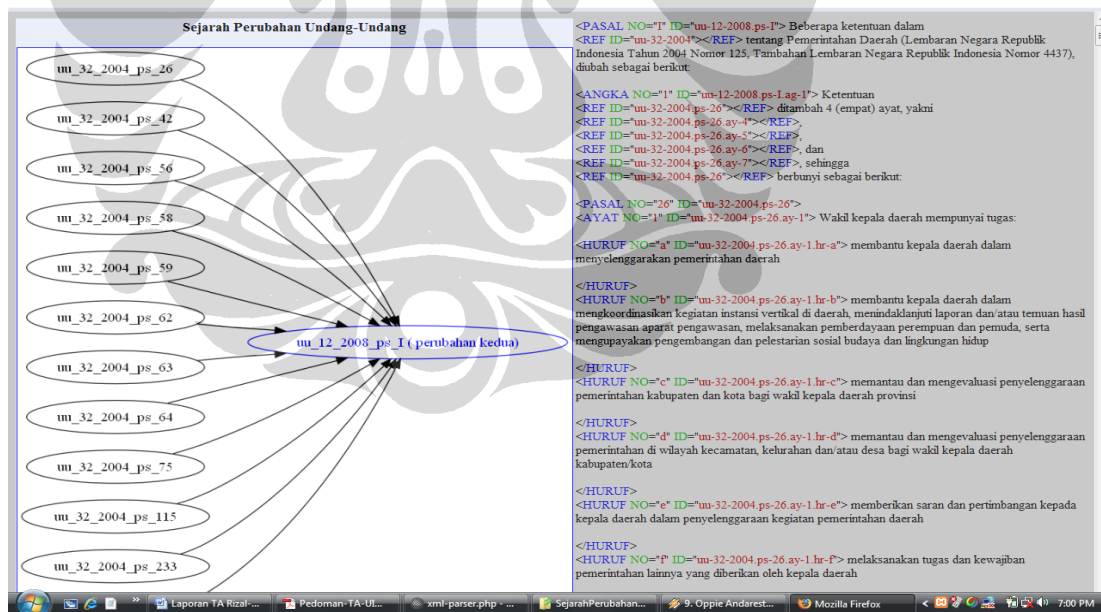
Gambar 4.20: Hasil Pencarian dari Sistem Rekapitulasi Dokumen Perundang-undangan di Indonesia

Pada Gambar 4.21 dibawah ini merupakan tampilan halaman *output* sistem perolehan informasi dokumen legal yang telah dikembangkan oleh [PRA09] yang menampilkan list dari pasal, dimana list pasal ini merupakan input bagi sistem rekapitulasi dokumen perundang-undangan di Indonesia yang telah penulis kembangkan.



Gambar 4.21: Halaman Output Sistem Perolehan Informasi Dokumen Legal

Pada Gambar 4.22 merupakan halaman sejarah perubahan Undang-undang yang ditampilkan dalam bentuk graf berarah, di mana *node* sebelah kiri merupakan pasal undang-undang non perubahan dan *node* yang disebelah kanan merupakan pasal undang-undang perubahan yang dibuat berdasarkan pasal inti dari undang-undang perubahan. Apabila *node* tersebut diklik oleh pengguna maka sistem akan menampilkan isi dari undang-undang tersebut dalam format XML



Gambar 4.22: Hasil Pencarian dari Sistem Rekapitulasi Dokumen Perundang-undangan di Indonesia

BAB 5

UJI COBA DAN ANALISIS

Bab ini menjelaskan uji coba terhadap sistem yang telah dikembangkan dalam penelitian ini. Subbab 5.1 menjelaskan lingkungan uji coba yang dilanjutkan penjelasan karakteristik dokumen, selanjutnya dijelaskan skenario uji coba dan hasil uji coba serta analisis dari hasil analisis uji coba.

5.1 Lingkungan Uji Coba

Sistem informasi rekapitulasi dokumen perundang-undangan yang dikembangkan ini telah dilakukan Uji coba pada komputer dengan *processor* Intel(R) Core(TM) 2 Duo CPU T5250 @1.50 GHz 1.50 GHz, *memory* 1024 MB, dan *harddisk* 160 GB. Sistem operasi yang digunakan adalah Microsoft Windows Vista Home Premium.

5.2 Karakteristik Dokumen

Dokumen yang digunakan dalam uji coba ini adalah semua jenis dokumen UU yang sudah memiliki format XML yang diperoleh dengan melakukan ekstraksi terhadap 472 dokumen TXT menggunakan Sistem Ekstraksi Informasi (SEI) Undang-undang yang telah dikembangkan oleh Violina [VIO08]. Dokumen yang diuji coba terdiri dari dua bagian yaitu dokumen UU dan pasal, dokumen UU terdiri dari 472 dokumen dan dokumen pasal terdiri dari 784 dokumen. Karakteristik dokumen yang diujicobakan adalah sebagai berikut:

- Rentang tahun undang-undang adalah dari tahun 1983 sampai tahun 2009
- Ukuran berkas undang-undang berkisar antara 3 KB sampai 200 KB
- Ukuran berkas pasal berkisar antara 1 KB sampai 125 KB
- Ada delapan jenis dokumen undang-undang yang diujicobakan pada sistem, yaitu dapat dilihat pada Tabel 5.1.

Tabel 5.1: Dokumen Uji Coba

Jenis Dokumen UU	Jumlah yang diujicobakan
Undang-undang perubahan (UUP)	68
Undang-undang non-perubahan yang tidak pernah dirubah oleh undang-undang lain (UUNP-TPU)	346
Undang-undang perubahan yang pernah dirubah oleh undang-undang lain (UUP-PU)	2
Undang-undang non-perubahan yang pernah dirubah oleh undang-undang lain (UUNP-PU)	56
Pasal undang-undang non perubahan yang tidak pernah dirubah oleh pasal undang-undang lain (p-UUNP-TPU)	100
Pasal undang-undang non perubahan yang pernah dirubah oleh pasal undang-undang lain (p-UUNP-PU)	618
Pasal undang-undang perubahan (p-UUP)	60
Pasal undang-undang perubahan yang pernah dirubah oleh pasal undang-undang lain (p-UUP-PU)	6

5.3 Hasil Uji Coba

Pada subbab ini dijelaskan hasil uji coba yang telah dilakukan. Hasil uji coba dapat dilihat pada Tabel 5.2.

Tabel 5.2: Hasil Uji Coba UU

Jenis UU	Jumlah Dokumen UU	Jumlah Sukses	Jumlah gagal	Prosentase
UUP	68	68	-	100
UUNP-TPU	346	346	-	100
UUP-PU	2	2	-	100
UUNP-PU	56	54	2	96
Jumlah	472	470	2	99.6

Tabel 5.3: Hasil Uji Coba Pasal-UU

Jenis Pasal-UU	Jumlah Dokumen UU	Jumlah Sukses	Jumlah gagal	Prosentase
p-UUNP-TPU	100	100	-	100
p-UUNP-PU	618	618	-	100
p-UUP	60	60	-	100
p-UUP-PU	6	-	6	0
Jumlah	784	778	6	99.2

5.5 Analisis Hasil Uji Coba

Dari hasil uji coba yang telah dilakukan penulis, maka hasil uji coba terhadap dokumen UUP, UUNP-TPU, UUP-PU, p-UUNP-TPU, p-UUNP-PU, dan p-UUP memberikan hasil yang sempurna, hal ini dikarenakan aturan atau *rule* yang telah dibuat penulis dapat melakukan pemrosesan yang baik terhadap dokumen undang-undang yang memiliki format XML.

Pada pengujian dokumen untuk sejarah perubahan undang-undang berdasarkan judul terdapat dua dokumen yang gagal, hal tersebut dikarenakan kurangnya koleksi dokumen undang-undang sehingga sejarah perubahan dari undang-undang tersebut tidak ditampilkan. Namun demikian dari hasil uji coba terhadap 472 dokumen undang-undang maka keakuratan sistem ini adalah 99.6 %, hasil tersebut sudah cukup optimal.

Pada pengujian dokumen untuk sejarah perubahan undang-undang berdasarkan pasal terdapat enam dokumen pasal yang gagal menampilkan sejarah perubahan pasal tersebut, hal ini dikarenakan pasal tersebut merubah pasal dari undang-undang perubahan, sedangkan dikoleksi pasal undang-undang tidak terdapat pasal undang-undang perubahan yang ada hanya pasal inti dari undang-undang perubahan. Dari hasil uji coba terhadap 784 dokumen pasal maka keakuratan sistem ini adalah 99.2%.

Dari hasil uji coba yang telah dilakukan menunjukkan bahwa aturan (*rule*) yang telah dibuat penulis sudah cukup baik untuk diimplementasikan didalam Sistem Rekapitulasi Dokumen Perundang-undangan di Indonesia.