

Dasar-Dasar Pemrograman 1

Tugas Pemrograman 02

Deadline: 9 Oktober 2023 23:55
Waktu SCell



Pencarian Dokumen Putusan Pengadilan

Anda diberikan sebuah folder yang berisi **22.630 dokumen** putusan pengadilan (ukuran total **~1,1 GB**). Silakan Anda lihat folder **dataset** yang diberikan bersamaan dengan deskripsi tugas pemrograman 2 ini. Sebuah dokumen merupakan sebuah file teks yang berformat **XML**.

XML adalah singkatan dari *Extensible Markup Language*. Ini adalah bahasa markup yang mendefinisikan seperangkat aturan untuk mengkodekan dokumen dalam format yang dapat dibaca manusia dan mesin. XML digunakan untuk menyimpan dan mengangkut data, dan banyak digunakan dalam berbagai aplikasi. Dokumen XML mengikuti struktur hirarkis dengan elemen-elemen yang dianotasikan dengan tag pembuka (**<tag>**) dan tag penutup (**</tag>**).

Dalam kasus ini, elemen utama adalah **<putusan>** yang memiliki atribut seperti **klasifikasi**, **lembaga_peradilan**, **provinsi**, **sub_klasifikasi**, dan lain-lain seperti contoh berikut:

```
<putusan      klasifikasi="..."      lembaga_peradilan="..."      provinsi="..."
sub_klasifikasi="..." ...>
<kepala_putusan>
...
</kepala_putusan>
<identitas>
...
</identitas>
<riwayat_penahanan>
terhadap terdakwa telah dilakukan penangkapan berdasarkan surat perintah
penangkapan ...
</riwayat_penahanan>
<riwayat_perkara>
pengadilan negeri tersebut setelah membaca ...
</riwayat_perkara>
<riwayat_tuntutan>
setelah mendengar pembacaan tuntutan pidana yang diajukan oleh ...
```

```
</riwayat_tuntutan>
<riwayat_dakwaan>
menimbang bahwa terdakwa diajukan ke persidangan oleh penuntut umum ...
</riwayat_dakwaan>
<fakta>
menimbang bahwa untuk membuktikan dakwaannya penuntut umum telah
mengajukan saksi saksi ...
</fakta>
<fakta_hukum>
menimbang bahwa berdasarkan alat bukti yang diajukan berupa keterangan ...
</fakta_hukum>
<pertimbangan_hukum>
mengingat pasal 351 ayat 3 jo pasal 55 ayat 1 ke 1 kuh pidana dan ...
</pertimbangan_hukum>
<amar_putusan>
mengadili
1 menyatakan terdakwa ... telah terbukti secara sah dan meyakinkan bersalah ...
</amar_putusan>
<penutup>
demikianlah diputuskan dalam sidang permusyawaratan majelis hakim pengadilan
negeri ...
</penutup>
</putusan>
```

Sebuah dokumen putusan, terdiri dari 11 buah *section*: **kepala_putusan**, **identitas**, **riwayat_penahanan**, **riwayat_perkara**, **riwayat_tuntutan**, **riwayat_dakwaan**, **fakta**, **fakta_umum**, **pertimbangan_hukum**, **amar_putusan**, **penutup**, dimana setiap *section* diawali dengan string **<section>** dan diakhiri string **</section>**. Setiap dokumen sudah diproses sehingga semua huruf *lowercase*, pemisah kata berupa spasi tunggal, dan tidak mengandung tanda baca seperti koma (,) dan titik (.). Detail masing-masing *section* dapat dilihat pada link asli dataset yang dapat diakses pada:

<https://github.com/ir-nlp-csui/indo-law/tree/main>

Koleksi dokumen ini merupakan hasil tesis dari mahasiswa magister ilmu komputer, Fasilkom UI.

Anda dapat membaca mengenai dataset tersebut lebih lanjut pada karya tesis berikut:

Nuranti, E. Q., Yulianti, E., & Husin, H. S. (2022). [Predicting the Category and the Length of Punishment in Indonesian Courts Based on Previous Court Decision Documents](#). Computers, 11(6), 88.

Tugas Anda adalah membuat program Python yang menerima **sebuah atau beberapa kata**, dan mengembalikan informasi daftar nama file yang mengandung kata-kata kunci tersebut. Secara spesifik, program Python Anda menerima kata kunci dengan menggunakan argumen program:

```
>> python search.py [section] [kata kunci 1]  
atau
```

```
>> python search.py [section] [kata kunci 1] [AND/OR/ANDNOT] [kata kunci 2]
```

[**section**] bisa bernilai salah satu dari 11 *section* yang sudah disebutkan sebelumnya, atau bernilai **all** jika proses pencarian kata kunci dilakukan pada semua *section* pada setiap file. Sebuah kata kunci dapat berupa satu kata, seperti **narkotika**, atau dapat berupa frasa seperti "**kebakaran hutan**". Jika berupa frasa, perlu dituliskan diantara **double quotation**. Kemudian, kita juga bisa mencari dokumen dengan dua kata kunci yang dihubungkan dengan operator logika **AND**, **OR**, atau **ANDNOT**. Anda bisa mempelajari situs berikut untuk mengetahui bagaimana caranya agar program Python dapat menerima beberapa argumen:

<https://www.geeksforgeeks.org/how-to-use-sys-argv-in-python/>

Sebagai contoh, misal, kita ingin mencari daftar nama file putusan pengadilan yang pada *section* **fakta** mengandung frasa "**kebakaran hutan**" tetapi tidak mengandung frasa "**pohon acasia**":

```
>> python search.py fakta "kebakaran hutan" ANDNOT "pohon acasia"
```

Pada contoh berikutnya, kita ingin mencari semua nama dokumen yang mengandung kata **penghinaan** tanpa peduli pada *section* mana kata tersebut berada:

```
>> python search.py all penghinaan
```

Format output adalah daftar yang terdiri dari 5 kolom:

- Nama file XML dari dokumen
- Nama provinsi dimana pengadilan yang memutuskan berada (space 15 karakter dan **right-justified**)
- Klasifikasi putusan (space 15 karakter dan **right-justified**)
- Sub-klasifikasi putusan (space 30 karakter dan **right-justified**)
- Lembaga peradilan (space 20 karakter dan **right-justified**)

Informasi nama provinsi, klasifikasi, sub-klasifikasi, dan Lembaga peradilan perlu Anda ekstraksi dari atribut-atribut yang ada pada tag **putusan** pada bagian awal file (biasanya baris pertama). Jika Panjang karakter dari informasi tertentu lebih besar dari space yang ditetapkan, string informasi tersebut perlu dipotong. Selain itu, Program Anda perlu menampilkan informasi banyaknya dokumen yang ditemukan dan total waktu pencarian pada bagian akhir output.



Anda dapat menggunakan modul time pada python untuk mendapatkan total waktu pencarian.

Contoh penggunaan:

```
>> import time
>> st = time.time()
>> ... # program anda
>> end = time.time()
>> end - st
```

Pada contoh tersebut kita menggunakan time.time() untuk mendapatkan waktu sebelum dan sesudah mengerjakan dan mencari selisihnya untuk mendapatkan total durasi dari program berjalan.

Berikut adalah beberapa contoh tampilan dan juga sekaligus bisa menjadi beberapa test-case untuk Anda:

Contoh 1:

```
>> python search.py all "barang bukti berupa 1 satu buah laptop"
```

08d6fc0758dc22e4f4475a5d89e9d810.xml	yogya	pidana-umum	pencurian	pn-wonosari
0ae384f35acb5157aac5ea44fa9dcbb9.xml	jabar	pidana-umum	pencurian	pn-ciamis
1351a0681feea945472206bbbed8a6092.xml	jabar	pidana-umum	pencurian	pn-bale-bandung
1e2a311d3c2b5b16ceeee8a8600c84e6.xml	jateng	pidana-umum	tidak-diketahui	pn-brebes
2949e107cf0506c6da88482ede1e33bc.xml	jateng	pidana-umum	tidak-diketahui	pn-brebes
332cd8a0fdc58e4a4050be5409fa839a.xml	jateng	pidana-umum	pencurian	pn-salatiga
4809e2e01277dc2fc4f53a11a7fc2b6d.xml	jabar	pidana-umum	penggelapan	pn-depok
48732956685bf1b7f537436130710997.xml	jateng	pidana-umum	pencurian	pn-magelang

4a982e9ba077931d175a3cd25dd0094e.xml	jabar	pidana-umum	pencurian	pn-depok
5a071b947bead3c6a3b89d9f5151791f.xml	yogya	pidana-khusus	lain-lain	pn-sleman
5ade2cca07dd0596c78c31b67fb08242.xml	jateng	pidana-umum	pencurian	pn-semarang
63562cabb27228fae7fe2291e605a85b.xml	jateng	pidana-umum	penadahan	pn-kabupaten-tegal-d
6497a8498634bd0e109d7352eacd87d6.xml	jabar	pidana-umum	pencurian	pn-ciamis
6f2af792d9a691203f80fd0282f1fa16.xml	jateng	pidana-umum	tidak-diketahui	pn-surakarta
75ab3613fb8537ff1596140bf19fa08c.xml	jatim	pidana-umum	pencurian	pn-tulungagung
a2e5c3c3fc81e280df174761ce5606e0.xml	jateng	pidana-umum	pencurian	pn-purbalingga
a9fc26a15a25033a701c7a9188b7212b.xml	yogya	pidana-umum	pencurian	pn-wonosari
ad0bbc89a934303fde8a11c6e880cab4.xml	jateng	pidana-umum	kejahatan-terhadap-keamanan-ne	pn-kab-semarang
af11ef1854efd79184acc8975744f6e7.xml	jatim	pidana-umum	penggelapan	pn-blitar
b1e8bab1a30260acd0ebc4e721f0dec9.xml	jabar	pidana-umum	pencurian	pn-bale-bandung
b67608e957a27ab4398d65d52ebb01fb.xml	jateng	pidana-umum	pencurian	pn-banjarnegara
bfc1a1fc520283033d5efb8a268df38.xml	yogya	pidana-umum	pencurian	pn-bantul
c05c0f6e680b7af5ad3a0040e16181f1.xml	jateng	pidana-umum	pencurian	pn-kabupaten-magelan
c0a5ef35ddd3c4b64100fc2b066dc587.xml	jatim	pidana-khusus	tidak-diketahui	pn-madiun
c12904aa64900ef192289e584502f6fd.xml	jateng	pidana-umum	pencurian	pn-banjarnegara
cdff5975ada1c4973e97d52111254a0f.xml	jatim	pidana-umum	pencurian	pn-malang
e125424d869b0c6ef70d239dd308f48d.xml	yogya	pidana-umum	penghinaan	pn-yogyakarta
e9e4588999741edd854bd501d6c1155b.xml	jatim	pidana-umum	penghinaan	pn-lamongan
eb8e2f685ea6e7d3e5ab8185539316b0.xml	jatim	pidana-umum	pencurian	pn-pamekasan
f362d249cdcc4c1e3980fe21033252b7.xml	jatim	pidana-umum	kejahatan-terhadap-keamanan-ne	pn-bangil
fb9337c9c997b60bfec5f5b74e043b1f.xml	jateng	pidana-umum	pencurian	pn-purbalingga
Banyaknya dokumen yang ditemukan = 31				
Total waktu pencarian = 6.503 detik				

Contoh 2:

>> python search.py fakta "kebakaran hutan"				
11e6d59c5c9be7cda2f00bc36fcf8e2c.xml	jabar	pidana-umum	tidak-diketahui	pn-cianjur
67ba7ead1831ccf13c08909b5868de7a.xml	jabar	pidana-umum	tidak-diketahui	pn-cianjur
694cc7747372d39151deaf6a59b053af.xml	jabar	pidana-khusus	pertambangan	pn-garut
6ce60d5107633d6664d0a47c48c9664a.xml	jateng	pidana-khusus	lingkungan-hidup	pn-temanggung
6ea6a2a91c0c4705df57df2dd3139779.xml	jateng	pidana-khusus	lingkungan-hidup	pn-wonosobo
b0766c501c40baf9c94fa80c73410f.xml	jateng	pidana-khusus	korupsi	pn-semarang

Banyaknya dokumen yang ditemukan = 6
Total waktu pencarian = 6.052 detik

Contoh 3:

>> python search.py fakta "kebakaran hutan" AND "pohon acasia"

6ce60d5107633d6664d0a47c48c9664a.xml	jateng	pidana-khusus	lingkungan-hidup	pn-temanggung
6ea6a2a91c0c4705df57df2dd3139779.xml	jateng	pidana-khusus	lingkungan-hidup	pn-wonosobo

Banyaknya dokumen yang ditemukan = 2
Total waktu pencarian = 6.410 detik

Contoh 4:

>> python search.py fakta "kebakaran hutan" ANDNOT "pohon acasia"

11e6d59c5c9be7cda2f00bc36fcf8e2c.xml	jabar	pidana-umum	tidak-diketahui	pn-cianjur
67ba7ead1831ccf13c08909b5868de7a.xml	jabar	pidana-umum	tidak-diketahui	pn-cianjur
694cc7747372d39151deaf6a59b053af.xml	jabar	pidana-khusus	pertambangan	pn-garut
b0766c501c40bafd9cf94fa80c73410f.xml	jateng	pidana-khusus	korupsi	pn-semarang

Banyaknya dokumen yang ditemukan = 4
Total waktu pencarian = 6.450 detik

Contoh 5:

>> python search.py fakta "kebakaran hutan" OR "narkotika gol onga"

11e6d59c5c9be7cda2f00bc36fcf8e2c.xml	jabar	pidana-umum	tidak-diketahui	pn-cianjur
2bb42e475c2e77003ccc9bfe59c9c5e9.xml	jabar	pidana-umum	kejahatan-terhadap-keamanan-ne	pn-bale-bandung
67ba7ead1831ccf13c08909b5868de7a.xml	jabar	pidana-umum	tidak-diketahui	pn-cianjur
694cc7747372d39151deaf6a59b053af.xml	jabar	pidana-khusus	pertambangan	pn-garut
6ce60d5107633d6664d0a47c48c9664a.xml	jateng	pidana-khusus	lingkungan-hidup	pn-temanggung
6ea6a2a91c0c4705df57df2dd3139779.xml	jateng	pidana-khusus	lingkungan-hidup	pn-wonosobo
b0766c501c40bafd9cf94fa80c73410f.xml	jateng	pidana-khusus	korupsi	pn-semarang

Banyaknya dokumen yang ditemukan = 7 Total waktu pencarian = 6.397 detik

Pesan Kesalahan

Program akan menampilkan pesan kesalahan jika argumen program tidak benar. Jika kesalahan argumen ada pada penulisan operator logika (jika ada 2 kata kunci), pesan kesalahan yang ditampilkan adalah

Operator harus berupa AND, OR atau ANDNOT.

Selain itu, kesalahan pada argumen akan menyebabkan program menyampaikan pesan umum:

Argumen program tidak benar.

Contoh 6:

>> python search.py fakta "kebakaran hutan" ORXX "narkotika gol onga"

Mode harus berupa AND, OR atau ANDNOT.
--

Contoh 7:

>> python search.py all

Argumen program tidak benar.

Bonus (10 poin)

Anda akan mendapatkan nilai bonus jika menggunakan kreatifitas Anda untuk menambahkan fitur lain yang lebih kompleks dan berguna.

Rubrik Penilaian

Penilaian akan mengacu pada [Rubrik Penilaian TP DDP1](#)

Plagiarisme dan keterlambatan pengumpulan tidak akan ditoleransi. Anda diperbolehkan berdiskusi dengan teman terkait ide implementasi tugas ini. Harap menuliskan kolaborator jika berdiskusi dengan mahasiswa lain. Perlu diingat bahwa **implementasi kode dilakukan secara individu**. Tim pengajar akan melakukan *code similarity checking* pada implementasi kode mahasiswa, dan jika terbukti melakukan kecurangan/plagiarisme akan mendapat **sanksi berupa nilai 0** pada TP ini.

Kebenaran Program 70%

Fungsionalitas	55 poin
Kenyamanan Penggunaan Program	5 poin
Validasi Input	5 poin
Standar Penulisan Kode	10 poin
Dokumentasi (Komentar)	10 poin
Efisiensi	10 poin
Persyaratan Submisi	5 poin

Demo 30%

Presentasi	50 poin
Qna	50 poin

Silahkan tanyakan ke asdos masing-masing terkait teknis dan pelaksanaan demo

Pengumpulan

Berkas yang perlu dikumpulkan adalah .zip dengan format penamaan sebagai berikut:

- [Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap]_TP02.zip

Contoh:

A_XYZ_2306123456_NamaSaya_TP02.zip

Good luck and happy coding ^^