

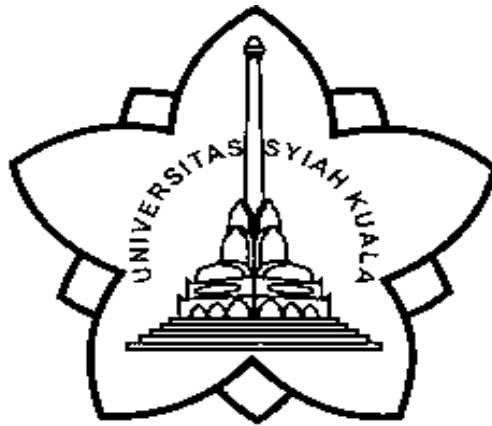
FINAL

## **FINAL LAB PI**

Disusun untuk memenuhi  
tugas matakuliah penelusuran informasi

Oleh:

**M. KHAIRUL RAMADHAN**  
**(1708107010006)**



**PROGRAM STUDI INFORMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS SYIAH KUALA  
DARUSSALAM, BANDA ACEH  
2020**

## Requirement:

- Install Lamp-Server
- Install datetime
- Install PySastrawi
- Install tqdm
- Install beautifulsoup4

## 1. CRAWLING

Langkah pertama dalam membangun search engine ialah mengumpulkan data/berkas yang nantinya akan ditampilkan sebagai content di search engine yang akan kita buat. Crawling adalah proses mengunduh atau mengumpulkan informasi dari halaman di website. Pada tahapan awal ini saya menggunakan code python untuk melakukan crawling dari website kompas, sebelum menjalankan semua file program yang tersedia maka kita harus membuat beberapa direktori terlebih dahulu, yaitu folder data dan didalam folder data terdapat folder cleaning,crawling,indexing dan link, berikut strukturnya:

- Data
  - Cleaning
  - Crawling
  - Indexing
  - Link

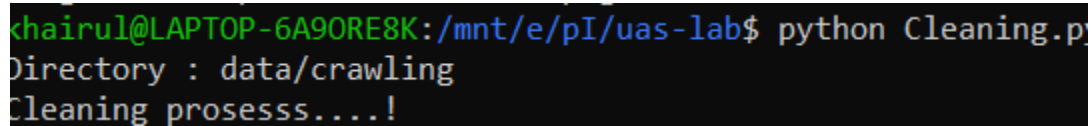
```
khairul@LAPTOP-6A90RE8K:/mnt/e/pI/uas-lab$ python Crawling.py -p 200
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2020-01-03
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2020-01-02
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2020-01-01
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2019-12-31
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2019-12-30
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2019-12-29
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2019-12-28
Url found : 25
get url : http://indeks.kompas.com/terpopuler/?site=all&date=2019-12-27
Url found : 25
Program mencapai batas max 200 page
```

### Gambar 1 hasil Crawling

Gambar diatas merupakan hasil dari menjalankan file crawling.py, pada file program tersebut saya memberikan batas ketika menjalankan programnya yaitu 200 halaman dari website kompas. Ketika file program crawling selesai dijalankan maka akan terdapat hasil sebanyak 200 data didalam folder data => crawling.

## 2. CLEANING

Setelah kita berhasil melakukan crawling, kemudian kita akan melakukan cleaning, cleaning ini bermaksud untuk membersihkan tag-tag html dari file yang kita kumpulkan/crawling sebelumnya. Disini file yang dihasilkan akan bebas dari tag html sekaligus membagi antara title dan content. Proses berjalannya cleaning ini yaitu program akan membuka direktori data => crawling dan akan membaca satu persatu berkas yang ada didalam folder tersebut , kemudia jika program mendeteksi adanya text html atau tanda baca, maka program akan menghapus tanda tersebut dari file yang kita cleaning , kemudian file tersebut akan ditulis kembali dan dimasukan ke data => cleaning.



```
khairul@LAPTOP-6A90RE8K:/mnt/e/pI/uas-lab$ python Cleaning.py
Directory : data/crawling
Cleaning prosessss....!
```

Gambar 2 Hasil dari Cleaning.py

Hasil dari menjalankan code program Cleaning yaitu akan menghasilkan file yang sudah bersih dari tag html dan file tersebut dimasukan didalam folder data => cleaning.

## 3. PREPROSESING

Preprocessing memiliki 6 tahapan yaitu :

- Parsing dokumen  
Parsing dokumen adalah proses memilah-milah dokumen yaitu seperti memilah apakah document tersebut berbahasa Indonesia atau inggris.
- Tokenisasi  
Yaitu proses memotong-motong dokumen menjadi perkata,
- Eliminasi Stopword  
Stopword merupakan kata yang sering muncul didalam sebuah dokumen, namun dia tidak bias berdiri sendiri atau tidak memiliki makna, oleh karena itu kata-kata tersebut dianggap kurang berguna dan sebaiknya dihapus dari indexing untuk menghemat ukuran.
- Normalisasi kata  
Normalisasi adalah proses menyamakan makna dari 2 atau lebih kata , contohnya seperti USA dan U.S.A , keduanya memiliki makna yang sama.
- Case Folding  
Case folding yaitu proses mengecilkan semua kata yang akan diindex.

- Stemming

Adalah proses mengembalikan sebuah kata menjadi bentuk asalnya(kata dasar).

Semua proses diatas dilakukan didalam code program Score , untuk proses parsing dokumen diabaikan, karena untuk tugas kali ini hanya menggunakan berkas/data dalam Bahasa Indonesia, kemudian untuk tokenisasi dilakukan dengan pemisahannya adalah spasi, lalu eliminasi stopword dilakukan dengan menggunakan library Sastrawi, karean library ini adalah yang paling bagus untuk melakukan eliminasi stopword, kemudian pada tahapan Normalisasi diabaikan untuk tugas kali ini karena sedikit komplek, dan untuk stemming juga dilakukan menggunakan library sastrawi.

Selain dari tahapan preprocessing diatas, code program scoring ini juga melakukan proses pemisahan antara title, content dan url. Hal ini berguna untuk dilakukan pembobotan dan penampilan content nantinya di search engine.

## 4. INDEXING

Proses indexing ini juga terjadi pada code program scoring, sebelum dilakukannya indexing, berkas/data harus melewati preprosesing terlebih dahulu supaya kata(term) yang akan deindex adalah unique, kemudia setelah dilakukannya preprocessing maka data akan dibagi menjadi beberapa bagian yaitu title, content1, content2 dan content3. Setiap bagian tersebut memiliki formula yang berbeda-beda untuk menunjukkan pembobotan yang lebih baik dari search engine nantinya. Untuk title diberikan formula  $(tf*idf)^{4/4}$  yang artinya title akan memiliki tingkat bobot yang lebih besar dari pada content1,2 dan 3. Kemudian untuk content1 diberikan formula  $(tf*idf)^{3/4}$ , content2 diberikan formula  $(tf*idf)^{2/4}$  dan content3 diberikan formula  $(tf*idf)^{1/4}$ . Dari setiap formula tersebut dapat disimpulkan bahwa Title > Content1 > Content2 > Content3.

```
khairul@LAPTOP-6A90RE8K:/mnt/e/pI/uas-lab$ python Scoring.py  
Directory :data/cleaning  
200it [11:06, 3.33s/it]  
unique words : 6130
```

100% | ██████████ | 6130/6130 [00:01<00:00, 4427.70it/s]

```
selesai
```

Gambar 4 Hasil menjalankan file Scoring

Setelah selesai menjalankan code program scoring, maka akan terbentuk sebuah file didalam folder data => indexing yang mana berisi semua bobot dari kata yang dihasilkan untuk setiap dokumen.

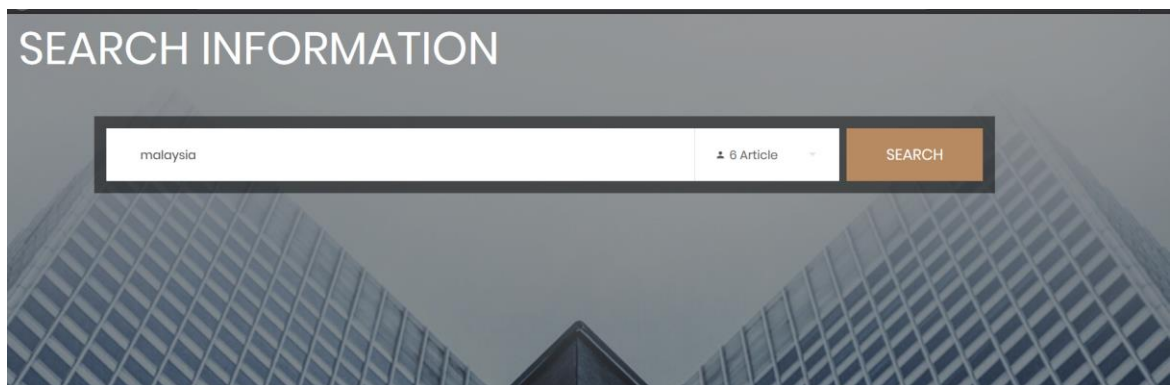
```
C:\xampp\htdocs\uas-lab>bash
khairul@LAPTOP-6A90RE8K:/mnt/c/xampp/htdocs/uas-lab$ python Query.py 2 nasi
[{"doc": "data115.txt", "score": 3.0, "url": "https://nasional.kompas.com/read/2019/12/29/20494401/kong-latip-dalam-kena-
ngan-dan-nasi-liwet-terakhir"}, {"doc": "data86.txt", "score": 1.5, "url": "https://regional.kompas.com/read/2019/12/30/
18111071/cerita-penjual-intip-goreng-depan-masjid-kendal-kaget-yang-beli-jokowi"}]
```

Gambar 4 Contoh dari Query.py

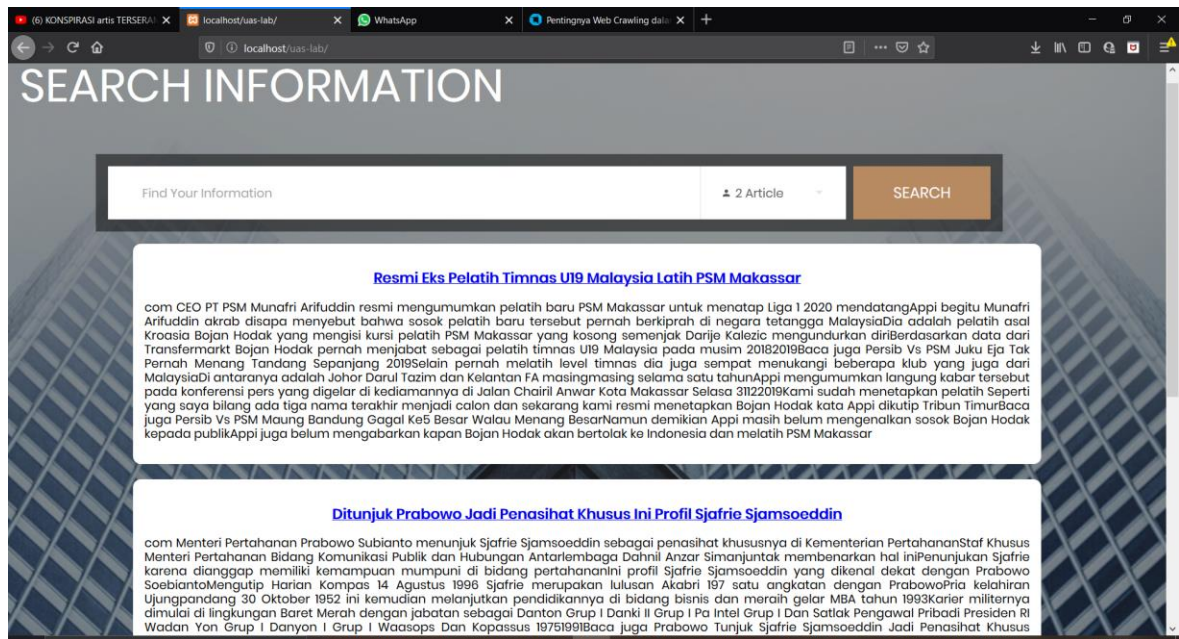
Gambar diatas merupakan hasil dari menjalankan code program query.py dengan kata kunci nasi dan menampilkan 2 buah data.

## 5. HASIL INDEX.PHP

Setelah didapatnya file indexing.txt yang berisi semua pembobotan dari dari kata dan dokumen, maka hasilnya akan ditampilkan web browser, untuk tampilannya dapat dilihat pada gambar dibawah ini. Sebelum itu jalankan **apache**.



Gambar 5 Proses Searching



Gambar 6 Hasil Searching

Dari gambar diatas dapat dilihat, tampilan dari mini search engine yang dihasilkan, pada gambar diatas saya mencoba mencari kata Malaysia dengan 6 artikel yang akan dikeluarkan, kemudian terlihat pada gambar6 hasil dari searching yang dilakukan, disitu terdapat nama judul dokumennya dan isi dari dokumennya, judul dari dokumen tersebut dapat di klik dan akan mengarah ke berita aslinya di web Kompas.com.

Cara untuk menampilkan hasil dari query ke web browser yaitu dengan membuat output dari hasil query kedalam format Json(dapat dilihat pada gambar 4) dan nantinya Json tersebut akan dikembalikan ke bentuk array asosiatif sehingga akan mudah untuk dilakukan pemanggilan menggunakan php.

GitHub : <https://github.com/MKhaifulRamadhan/Simple-Search-Engine>