Laporan Tugas 1
Teks dan Web Mining

Diajukan untuk melengkapi tugas Mata Kuliah Teks dan Web Mining

Oleh:

MUAMMAR ZIKRI AKSANA

1608107010045

JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
MARET, 2019

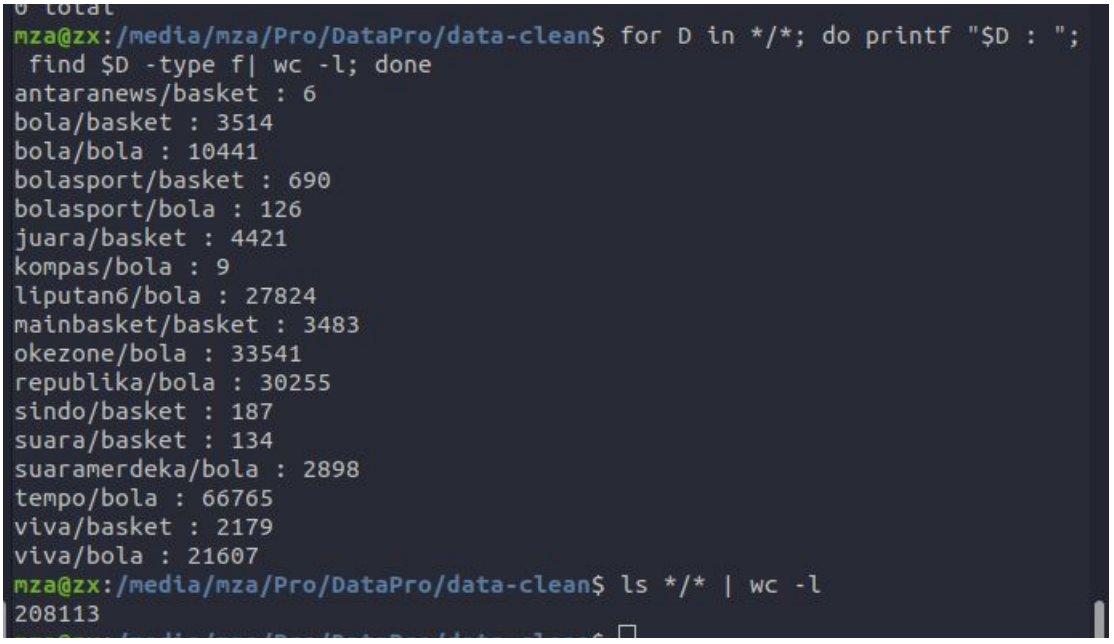
Daftar Isi Halaman

I. Penjelasan Tugas

Pada tugas 1 ini, diberikan case untuk mendownload 25000 file html dengan 2 kelas konten, pada tugas ini diambil dua kelas yaitu sepak bola dan basket. File didownload dari beberapa portal berita antara lain :

- www.bolasport.com
- indeks.kompas.com
- www.tempo.co/indeks/
- index.okezone.com
- republika.co.id/
- www.liputan6.com
- www.bola.com
- www.suaramerdeka.com
- www.viva.co.id
- www.mainbasket.com

dengan file yang berhasil didapat sebanyak 208113 :



II. Proses Pembagian Isi Konten

Proses extract file dari hasil crawling dilakukan dengan beberapa tahapan :

1. Didapat semua path dari file yang ingin di extract dan disimpan pada sebuah array. (extractContent.pl : line 33)

```
my $listFile=`find $ARGV[0] -type f -follow`;
# setiap path file disimpan kedalam array
my @filePaths= split("\n",$listFile);
```

2. Setiap path file yang di dapat, dilakukan looping untuk setiap anggota array dan didapat *path-file* lalu dijalankan proses **extractFile(***path-file***),** sampai semua anggota array selesai dibaca. (extractContent.pl : line 38)

```
foreach my $file (@filePaths){
    # print $filePaths[0];
    print "----------------------------------------\n";
    print "Start at : $startAt\n";
    print "Time     : ".localtime."\n";
    print "countf : ".$countf++." : ".scalar @filePaths."\n";
    print "from : $file\n";
    # ubah kedalam format clean lalu di save
    saveFile($ARGV[1],$file,extractFile($file));
}
```

extractFile(*path-file***)** > Tahapan untuk pembagian isi konten :

1. File dari *path-file* dibaca lalu ditampung pada sebuah variabel
2. Lalu diambil title dan url dengan regex dari variabel file yang telah ditampung sebelumnya
3. Isi variabel dibersihkan dari tag-tag html dan simbol yang tidak diperlukan dan diambil teksnya saja dan ditampung sebagai content (untuk step 2 -3 , extractContent.pl : line 57-73)

```
if( $file =~ /<title.*?>(.*?)<\/title>/){
    $title = $1;
    $title = clean_str($title);
    $title="<title>$title</title>\n";
    # bagian title dihapus dari file agar tidak ikut dalam content
    $file  =~ s/<title.*?>(.*?)<\/title>//;
}
# get URL
if( $file =~ /<url>(.*?)<\/url>/){
    $url = $1;
    $url="<url>$url</url>\n";
    # bagian url dihapus dari file agar tidak ikut dalam content
    $file  =~ s/<url>(.*?)<\/url>//;
}
```

```
$extractor->extract($file);
# get BODY (Content)
my $content = $extractor->as_text;
```

4. Content yang telah didapat dinormalisasi karakter pemisah (tanda bacanya) untuk penanda kalimat.(util.pl : line 113-134)

```
# Fungsi ini untuk normalisasi string
# Misal setelah titik harus ada spasi, tapi untuk url dan titel tidak dan lain sebagai
# Proses ini dilakukan agar tahap pembagian kontent jumlah kata dapat diprediksikan de
# return string
sub simanticToken{
    $var=$_[0];
    my @matches = ( $_[0] =~ m/\s+[a-z]+\.\s*[A-Z]+/g );
    # @varr=split \s+[a-z]+\.\s?[A-Z]+,$var;

    foreach $data (@matches){
        # print $data;
        my $token=simanticTokenSentence($data);
        $var=~ s/$data/$token/g;
    }

    return $var;
}

# Fungsi untuk split dan menambahkan spasi pada sepenggal string
# ex: sekolah.Zikri => sekolah. Zikri
# return string
sub simanticTokenSentence{
    my @token = split "\\.",$_[0];
    # return index 0 dan 1 karena hasil split telah pasti hanya terdari 2 anggota
    return $token[0].". ".$token[1];
}
```

5. Content yang telah dinormalisasi tanda bacanya di-*split* menjadi 3 bagian sesuai case dari tugas 1 , pada kasus ini dipisah dengan titik, tanda seru dan tanda tanya. (util.pl : line 83)

```
# Fungsi untum case-soal dimana content dipecah
# return string
sub fragmentContent{
    my @kalimat=split /(?<=[\.!?])\s+/,$_[0];
    my $splice = int(scalar @kalimat/3);
    my $html="<atas>".iterateFor(@kalimat,0,$splice*1)."</atas>\n";
    $html.="<tengah>".iterateFor(@kalimat,$splice*1+1,$splice*2)."</tengah>\n";
    $html.="<bawah>".iterateFor(@kalimat,$splice*2+1,$splice*3+@kalimat%3)."</bawah>\n";

    return $html;
}
```

6. Lalu file di-*save* dengan isi url - title - dan content yang telah dibagi sebelumnya.

III. Crawling

Pada proses crawling untuk mendapatkan source file yang dibutuhkan untuk menyelesaikan tugas 1, dilakukan 2 tahapan yaitu , proses untuk mendapatkan link dan proses download untuk setiap link yang telah didapat.

a. Mendapatkan Link Dengan Mechanize

Untuk mengambil link pada sebuah halaman web dengan Mechanize dibutuhkan link awal dari web tersebut, pada kasus ini web yang di crawling adalah portal berita yang memiliki indeks berita yang sesuai dengan *class* yang dipilih(pada tugas ini dipilih *class* sepak bola dan basket) portal berita dipilih karena terdapat pola pada setiap url indeksnya sehingga memungkinkan untuk di-crawling dengan skala besar meski kita tidak memiliki list link untuk di crawling dengan hanya merubah sedikit dari pola link sebelumnya.

Pada kasus ini dibuat proses :

1. Input format link dan tentukan jumlah hari yang ingin di-*crawling*
2. Dibuat perulangan dari 0 sampai jumlah hari lalu untuk masing perulangan didapat url berbeda dari hasil *generate-date*. (crawl.pl : line 38)

```
# looping sebanyak jumlah day yang ditentukan
for(my $i=0; $i <= $ARGV[2]; $i++){
    # proses mendapatkan url
    fly(generator($ARGV[1]),$ARGV[4]);
}
```

3. Untuk link dengan date yang telah di-generate, diambil yang ada dari url yang dihasilkan tersebut dengan method *get* dari kelas *mechanize*. (crawl.pl line : 54)

```
my $temp = eval{$mech->get($_[0])};
```

4. List url yang didapat dari hasil *get* sebelumnya di-*hash* untuk setiap anggota list sehingga url dapat dipastikan unik dan url tersebut memenuhi pattern url yang dibutuhkan, setelah perulangan tahap 2 selesai maka lanjut tahap 5. (crawl.pl line :70)

```
# untuk setiap link yang didapat
for my $i ($mech->links()){
    # difilter dengan filer url yang ditentukan
    if($i->url=~$_[1]){
        print "Link get : ".$go++."\n";
        $urls{$i->url}=1;
        print $hashLink $i->url."\n";
        # print $i->url."\n";
    }
}       You, 15 days ago • init project
```

5. Setelah perulangan tahap 2 selesai, maka dilanjutkan ke proses crawling dari url yang terdapat pada hash.

```
# Fungsi untuk melakukan crawling
sub crawl{
    #folder-file-url

    # perintah wget melakukan crawling terhadap url yang diberkan dan hasilnya disimpan pada sebuah scalar
    my $RESULT=`wget -qO- --user-agent="Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
    (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36" $_[0]/$_[1].html $_[2]`;
    # print $RESULT;
    open(my $web,'>>:encoding(UTF-8)',"$_[0]/$_[1].html");

    my $uri="<url>$_[2]</url>\n";
    # concat url dengan hasil crawling sebelumnya, agar data url dari web terdata
    print $web $uri.$RESULT;
    close $web;
    # print $CMDOUT;

}
```

b. Melakukan Crawling Dengan Wget

Jika list url yang akan di-*crawling* telah selesai didapatkan maka dilanjutkan ke proses crawling, dengan tahapan :

1. Key dari hash diambil dan disimpan pada sebuah array, lalu dilakukan perulangan untuk mendapatkan url dari anggota array sebelumnya. untuk setiap anggota array di panggil fungsi **crawling**(path-save,url).

```
# Procedure go untuk melakukan proses each dari url
sub go{
    # static-name - destination
    foreach my $url (keys %urls) {
        print "-------------------------------------------------\n";
        print "Crawl    : $_[0] : $loop : $go\n";
        print "From     : $url\n";
        print "To       : $_[1]\n";
        print "Start at : $startAt\n";
        print "Time     : ".localtime."\n";
        print "-------------------------------------------------\n";
        # proses crawling untuk setiap url dari hash
        crawl($_[1],($_[0]."-".$loop++),$url);
    }
    return 1;
}
```

crawling(path-save,url**)** > Tahapan mendasar untuk pembagian isi konten :

1. Dijalankan perintah wget : (crawl.pl : line 105)
   ```
   wget -qO- --user-agent="Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36" $_[0]/$_[1].html $_[2]
   ```
   - index 0 adalah path-save
   - index 1 adalah nama file
   - index 2 adalah alamat dari web yang akan di-*crawl*
   - diperlukan user agent untuk beberapa situs berita.
2. Setiap hasil wget disimpan pada sebuah variabel (wget menggunakan flag -q untuk no download).
3. Sebelum disimpan kedalam file variabel sebelumnya di concat

   ```
   open(my $web,'>>:encoding(UTF-8)',"$_[0]/$_[1].html");
   my $uri="<url>$_[2]</url>\n";
   # concat url dengan hasil crawling sebelumnya, agar data url dari web terdata
   print $web $uri.$RESULT;
   close $web;
   ```

   dengan url (crawl.pl : line 107)
4. Setelah hasil wget di concat dengan urlnya barulah di-*save*.

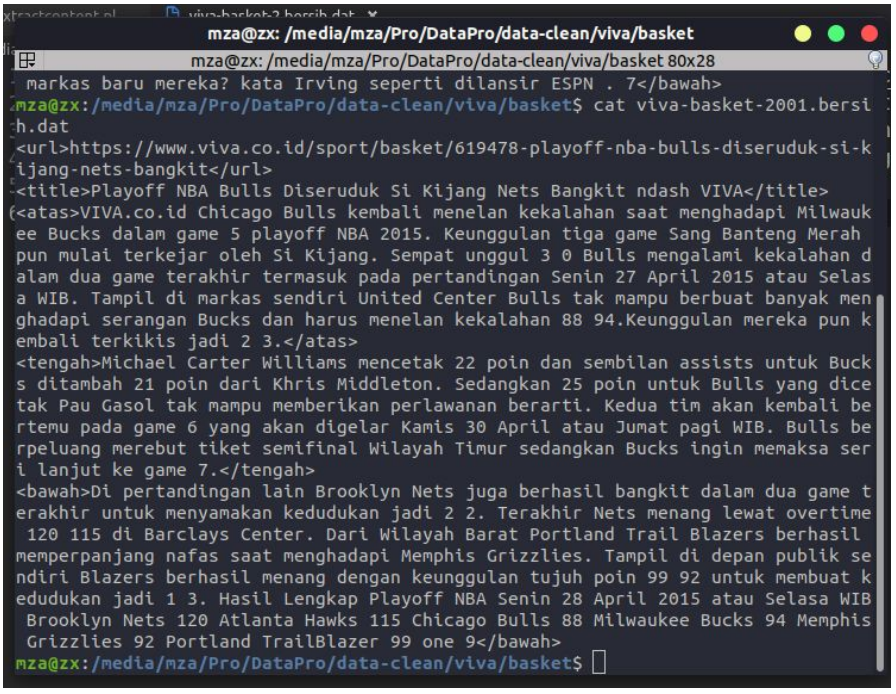
IV. File hasil ekstraksi

File hasil ekstraksi terdiri dari :

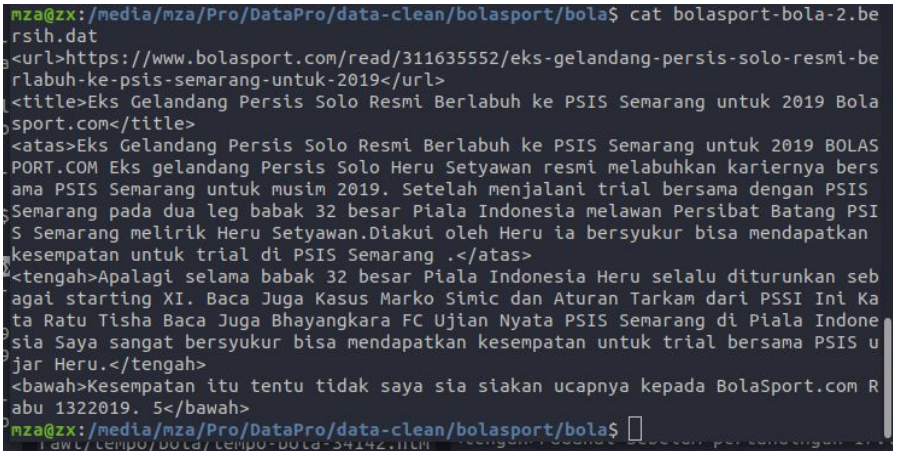
- <url></url>
- <title></title>
- <atas></atas>
- <tengah></tengah>
- <bawah></bawah>

berikut screenshot file beberapa hasil file hasil ekstraksi :

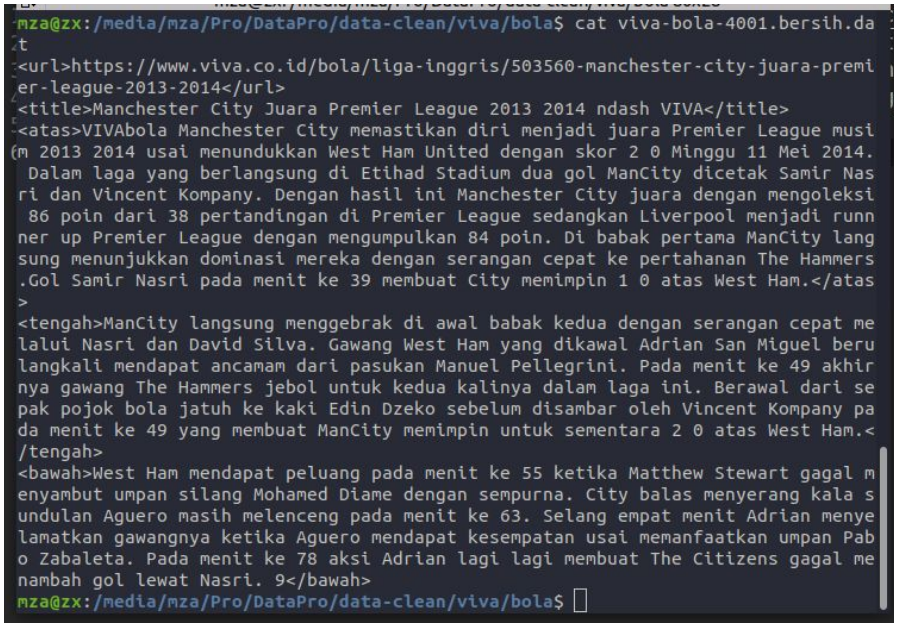
1. File laman dengan kategori basket dari situs viva



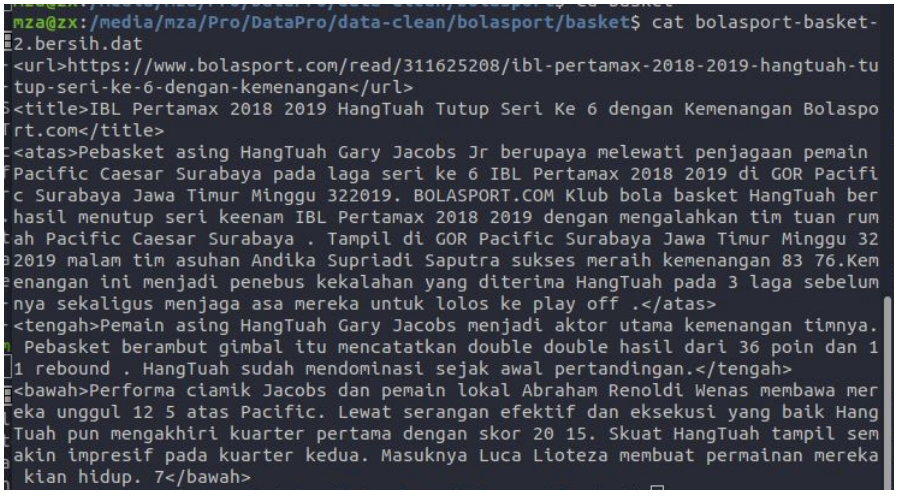
2. File laman dengan kategori bola dari situs bolasport



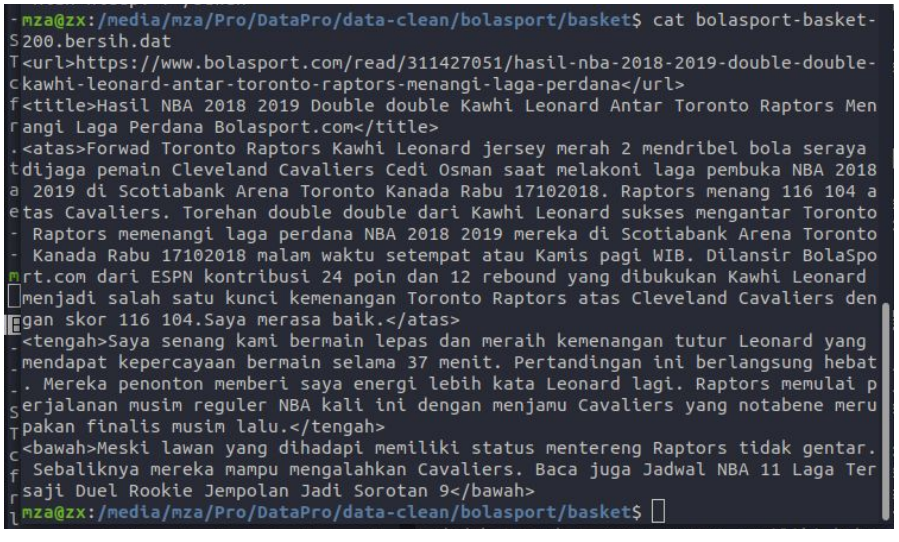
3. File laman dengan kategori bola dari situs viva



4. File laman dengan kategori basket dari situs bolasport



5. File laman dengan kategori basket dari situs bolasport



V. Lampiran

Script untuk menyelesaikan tugas 1 terdapat pada file MuammarZikriAksana_1608107010045_twm.tar.gz yang terdiri dari :

folder :

- link : folder ini berisi file link yang berhasil di-*crawling* dan file link yang gagal pada saat di-*crawling*.
- log : folder ini berisi file yang mencatat file yang gagal di-*extract*
- ref : folder ini berisi referensi link ,syntax dan lainnya pada saat tugas dikerjakan
- src : folder ini berisi script yang digunakan untuk menyelesaikan tugas 1 :
    - crawl.pl , script untu melakukan crawling
    - extractContent.pl, script untuk membersihkan file hasil crawling
    - generateUrl.pl, script untuk meng-*generate* url untuk dicrawling
    - util.pl , script yang berisikan fungsi-fungsi yang diperlukan untuk proses crawling dan extract

* beberapa folder diatas dibutuhkan kan untuk run program meski dibuat sebagai empty folder