



ITS
Institut
Teknologi
Sepuluh Nopember

i

TUGAS AKHIR - KS141501

**AKUISISI DATA LEMBAGA PENKAJIAN PANGAN
OBAT-OBATAN DAN KOSMETIKA MAJELIS ULAMA
INDONESIA DENGAN DATA HALAL NUTRITION FOOD**

**DATA ACQUTION OF LEMBAGA PENKAJIAN
PANGAN OBAT-OBATAN DAN KOSMETIKA MAJELIS
ULAMA INDONESIA WITH DATA OF HALAL
NUTRITION FOOD**

**RIFQY PRATAMA PUTRA BARLIANSYAH
052116 40000 106**

**Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.**

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020**

TUGAS AKHIR - KS141501

AKUISISI DATA LEMBAGA PENGKAJIAN PANGAN OBAT-OBATAN DAN KOSMETIKA MAJELIS ULAMA INDONESIA DENGAN DATA HALAL NUTRITION FOOD

RIFQY PRATAMA PUTRA BARLIANSYAH
052 116 40000 106

Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom., M.Sc. Eng., Ph.D

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

UNDERGRADUATE THESIS - KS141501

***DATA ACQUITION OF LEMBAGA PENGKAJIAN
PANGAN OBAT-OBATAN DAN KOSMETIKA
MAJELIS ULAMA INDONESIA WITH DATA OF
HALAL NUTRITION FOOD***

**Rifqy Pratama Putra Barliansyah
052 116 40000 106**

Supervisor

Nur Aini Rakhmawati, S.Kom., M.Sc. Eng., Ph.D

INFORMATION SYSTEM DEPARTMENT

Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya 2020

LEMBAR PENGESAHAN

**AKUISISI DATA LEMBAGA PENKAJIAN
PANGAN OBAT-OBATAN DAN KOSMETIKA
MAJELIS ULAMA INDONESIA DENGAN DATA
HALAL NUTRITION FOOD**

TUGAS AKHIR

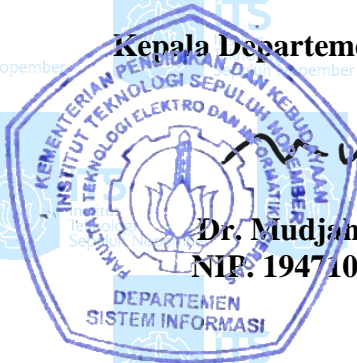
Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
(ELECTICS)
Institut Teknologi Sepuluh Nopember

Oleh:

Rifqy Pratama Putra Barliansyah
NRP. 05211640000106

Surabaya, 15 Agustus 2020

Kepala Departemen Sistem Informasi



Dr. Mudjanidin, ST., MT
NIP. 194710251978031001

LEMBAR PERSETUJUAN

AKUISISI DATA LEMBAGA PENKAJIAN PANGAN OBAT-OBATAN DAN KOSMETIKA MAJELIS ULAMA INDONESIA DENGAN DATA HALAL NUTRITION FOOD

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:

RIFQY PRATAMA PUTRA BARLIANSYAH

NRP. 0521164000106

Disetujui Tim Penguji: Tanggal Ujian: 10 Juli 2020

Periode Wisuda: September 2020

Nur Aini Rakhmawati, S.Kom., M.Sc. (Pembimbing I)
Eng., Ph.D

Nur Aini Rakhmawati
Faizal Johan

Faizal Johan Atletiko, S.Kom , MT. (Penguji I)

Radityo Prasetyanto Wibowo, S.Kom, (Penguji II)
M.Kom.

AKUISISI DATA LEMBAGA PENGAJIAN PANGAN OBAT-OBATAN DAN KOSMETIKA MAJELIS ULAMA INDONESIA DENGAN HALAL NUTRITION FOOD

Nama Mahasiswa : Rifqy Pratama Putra Barliansyah
NRP : 052 116 40000 106
Departemen : Sistem Informasi FTIK-ITS
Pembimbing I : Nur Aini Rakhmawati, S.Kom.,
M.Sc.Eng., Ph.D.

ABSTRAK

Indonesia merupakan salah satu negara dengan populasi penduduk beragama islam terbesar di dunia. Prosentase penduduk muslim di Indonesia mencapai hingga 12,7 persen dari populasi penganut muslim di dunia. Jumlah tersebut berbanding lurus dengan prosentase penganut muslim di Indonesia terhadap jumlah penduduk, yaitu sebesar 88,1 persen. Oleh karena umat muslim sangat memperhatikan hal mengenai makanan apakah makanan tersebut halal atau tidak sehingga, dibutuhkan sesuatu yang dapat membantu umat muslim mengenai hal tersebut. Saat ini telah ada aplikasi Halal Nutrition Food yang mana pengguna dapat melakukan pengecekan produk halal yang terdapat dalam aplikasi tersebut serta hal-hal lainnya yang disediakan aplikasi Halal Nutrition Food. Fitur pencarian pada aplikasi Halal Nutrition Food merupakan bagian penting dari penggunaan aplikasi ini, dimana keakuratan performa pencariannya menjadi komponen utama yang menjadi tujuan pengguna penggunaan aplikasi tersebut. Tugas akhir ini merupakan bagian dari pengembangan website Halal Nutrition Food dimana pengembangan yang dilakukan merupakan mengambil atau mengakuisisi data dari tujuh situs web LPPOM MUI provinsi, yang memiliki daftar produk pada website masing-masing. Data yang di dapat akan di indeks dan diuji menggunakan kueri pencarian pada Elasticsearch dan akan menjadi sumber data produk halal yang baru guna melengkapi data yang sudah ada.

Tugas akhir ini bermanfaat untuk mengetahui performa pencarian dari sistem basis data elasticsearch terhadap produk-produk yang ada pada website Halal Nutrition Food. Jumlah data yang didapatkan pada tugas akhir ini berjumlah 10.649 data produk halal. Berdasarkan pengujian pencarian yang dilakukan menggunakan full text query dengan input yang terpotong didapatkan bahwa indeks yang memiliki autocomplete analyzer dan search analyzer dapat melakukan pencarian dokumen secara tepat. Hasil pencarian yang dilakukan mampu menghasilkan dokumen dengan value yang mengandung semua bagian input query. Sedangkan pada untuk pencarian menggunakan term-level query ditemukan bahwa indeks yang tidak memiliki analyzer dan indeks yang hanya menggunakan analyzer memiliki performa yang sama dalam melakukan pencarian dan dapat menghasilkan pencarian yang lebih akurat karena dapat menampilkan dokumen dengan value yang sama dengan harapan hasil dari input query.

Kata Kunci: Halal Nutrition Food, LPPOM MUI, Scraping, Kueri, Indexing, Analyzer.

**DATA ACQUITION OF LEMBAGA PENGAJIAN
PANGAN OBAT-OBATAN DAN KOSMETIKA
MAJELIS ULAMA INDONESIA WITH DATA OF
HALAL NUTRITION FOOD**

Name : Rifqy Pratama Putra Barliansyah
NRP : 052 116 40000 106
Department : Sistem Informasi FTIK-ITS
Supervisor I : Nur Aini Rakhmawati, S.Kom.,
M.Sc.Eng., Ph.D.

ABSTRACT

Indonesia is one of the countries with the largest Muslim population in the world. The percentage of Muslim population in Indonesia reaches up to 12.7 percent of the world's Muslim population. The number is directly proportional to the percentage of Muslims in Indonesia to the population, which is 88.1 percent. Because Muslims are very concerned about matters of food whether the food is halal or not so, it takes something that can help Muslims about it. Currently there is a Halal Nutrition Food application where users can check halal products contained in the application as well as other things provided by the Halal Nutrition Food application. The search feature on the Halal Nutrition Food application is an important part of using this application, where the accuracy of its search performance is the main component that users are using the application for. This final project is part of the development of the Halal Nutrition Food website where the carried-out development is taking or acquiring data from seven website of the provincial LPPOM MUI, which has a list of products on each website. The data obtained will be tested using a search query on Elasticsearch and will be a new source of halal product data to complement the existing data. This final project is useful to determine the search performance of the elasticsearch database system for products on the Halal Nutrition Food website. The amount of data obtained in this

final project is 10,649 halal product data. Based on testing the search performed using full text query with truncated input it was found that the index that has an autocomplete analyzer and search analyzer can search documents correctly. The results of performed search are able to produce documents with values that contain all parts of the query input. Whereas for searching using term-level query it is found that indexes that do not have an analyzer and indexes that only use analyzers have the same performance in performing searches and can produce more accurate searches because they can display documents with the same value as expected results from input queries

Keywords: Halal Nutrition Food, LPPOM MUI, Scraping, Query, Indexing, Analyzer.

PERNYATAAN BEBAS PLAGIARISME

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Rifqy Pratama Putra Barliansyah
NRP : 0521164000106
Tempat/Tanggal lahir : Bogor, 14 September 1998
Fakultas/Departemen : FTEIC/Sistem Informasi
Nomor Telp/Hp/email : 081280194879/rifqybarliansyah@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul

AKUISISI DATA LEMBAGA PENGKAJIAN PANGAN OBAT-OBATAN DAN KOSMETIKA MAJELIS ULAMA INDONESIA DENGAN DATA HALAL NUTRITION FOOD

Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 13 Agustus 2020



RIFQY PRATAMA PUTRA
BARLIANSYAH
NRP. 0521164000106

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Dengan mengucapkan rasa syukur kepada Allah SWT Yang Maha Pengasih dan Maha Penyayang atas izin-Nya penulis dapat menyelesaikan buku yang sederhana ini dengan judul “Akuisisi Data Lembaga Pengkajian Pangan Obat-Obatan Dan Kosmetika Majelis Ulama Indonesia Dengan Halal Nutrition Food”. Dalam penyelesaian tugas akhir ini, penulis diiringi oleh pihak-pihak yang selalu memberi dukungan, saran, dan doa sehingga penelitian berlangsung dengan lancar. Secara khusus penulis mengucapkan terima kasih dari lubuk hati terdalam kepada:

1. Kedua orang tua penulis, kakak, dan dua adik, yang selalu memberikan doa, dukungan dan semangat untuk menyelesaikan studi S1 dengan sebaik-baiknya.
2. Bapak Dr. Mudjahidin, S.T., M.T. selaku Ketua Departemen Sistem Informasi ITS Surabaya serta seluruh dosen pengajar beserta staf dan karyawan di Departemen Sistem Informasi, FTEIC ITS Surabaya selama penulis menjalani perkuliahan.
3. Ibu Nur Aini Rakhmawati, S.Kom., M.Sc., Eng. Ph. D selaku dosen pembimbing selama pengerjaan tugas akhir yang senantiasa meluangkan waktu, tenaga dan ilmunya serta selalu memberi dorongan yang tidak terkira nilainya untuk menyelesaikan tugas akhir ini.
4. Fadhila Syahira, yang selalu ada untuk penulis.
5. Teman-teman dari VVUE yang menemani penulis selama prose penelitian dan selalu membahas tentang tugas akhir sehingga memicu penulis untuk segera menyelesaikannya.
6. Teman-teman BNN yaitu Jazeed, Jek, dan Kalief yang selalu menemani ketika sedang dilanda rasa bosan.
7. Teman Bersepuluh yang selalu menjadi motivasi penulis untuk segera menyelesaikan penelitian.

8. Akram dan Maro yang sangat berjasa membantu menyelesaikan penelitian ketika penulis sedang menemui kesulitan.
9. Kepada seluruh teman-teman di Lab Akuisisi Data dan Diseminasi Informasi (ADDI) yang telah bersama-sama berjuang mengerjakan tugas akhir, terutama teman-teman seperbimbingan Bu Iin yaitu Rima dan Wulan.
10. Berbagai pihak lainnya yang tidak bisa disebutkan satu persatu yang turut menyukseskan dan mendukung penulis menyelesaikan tugas akhir.

Penyusunan laporan tugas akhir ini masih jauh dari kata sempurna, sehingga penulis menerima segala kritik dan saran yang dapat membangun serta perbaikan untuk menjadi lebih baik lagi di masa yang akan datang. Semoga tugas akhir ini dapat bermanfaat bagi pembaca.

Jakarta, Juli 2020

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERSETUJUAN.....	iii
ABSTRAK.....	v
ABSTRACT.....	vii
PERNYATAAN BEBAS PLAGIARISME.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE.....	xxi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Permasalahan.....	3
1.4. Tujuan.....	3
1.5. Manfaat.....	4
1.5.1. Bagi Pengguna.....	4
1.5.2. Bagi Penulis.....	4
1.6. Relevansi.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Penelitian Sebelumnya.....	5
2.1.1. Linked Open Data for Halal Food Products.....	5
2.1.2. Integrasi Data dan Visualisasi Graf pada Aplikasi Halal Nutrition Food.....	6
2.1.3. Integrasi Data Lembaga Sertifikasi Halal di Dunia dengan Data Halal Nutrition Food.....	6
2.1.4. The Implementation of Crawling News Page Based on Incremental Web Crawler.....	7
2.1.5. Optimization of Scheduler for a Web Scraping System.....	7
2.2. Dasar Teori.....	8
3.2.1 Halal Nutrition Food.....	8
3.2.2 Web Crawler.....	10
3.2.3 Scheduling.....	11

3.2.4	ElasticSearch	11
3.2.5	Scrapy	12
3.2.6	LPPOM MUI.....	15
BAB III	METODOLOGI	19
3.1.	Arsitektur Sistem.....	19
3.1.1.	Data Source – LPPOM MUI Provinsi & KlikIndomaret	19
3.1.2.	Data Collection.....	19
3.1.3.	Scheduling System.....	20
3.1.4.	Data Ingestion	20
3.1.5.	Halal Nutrition Food	21
3.2.	Tahapan Pengerjaan Tugas Akhir	21
3.2.1.	Studi Literatur	22
3.2.2.	Database Design.....	22
3.2.4.	Scheduling.....	23
3.2.5.	Data Ingestion	23
3.2.6.	Pengujian Pencarian.....	23
3.2.7.	Penyusunan Laporan Tugas Akhir	24
3.2.8.	Integrasi Data	24
BAB IV	PERANCANGAN	25
4.1.	Database Design.....	25
4.2.	Data Collection	26
4.2.1.	Melakukan web scraping.....	26
1.	Membuat kode scraper	26
2.	Melakukan web scraping	27
4.2.2.	Menggabungkan data menjadi satu	28
4.2.3	Membersihkan Data.....	28
4.3.	Scheduling.....	31
4.4.	Data Ingestion	31
4.5.	Pengujian Pencarian	32
4.6.	Integrasi Data	33
BAB V	IMPLEMENTASI	35
5.1.	Data Collection	35
5.1.1	Melakukan web scraping.....	35
a.	LPPOM MUI Kepulauan Riau	35
b.	LPPOM MUI Bali	39
c.	LPPOM MUI Jateng.....	42

d. LPPOM MUI Kalsel.....	46
e. LPPOM MUI Kaltim.....	50
f. LPPOM MUI Jogja.....	52
g. LPPOM MUI Aceh.....	57
h. KlikIndomaret.....	61
5.1.2 Menggabungkan data menjadi satu.....	65
5.1.3 Membersihkan data.....	65
a. Pemisahan nama produk sebaris.....	66
b. Penghapusan nama produk tidak valid.....	67
c. Penyeragaman format tanggal.....	68
5.2 Scheduling.....	70
5.3 Data Ingestion.....	71
5.3.1 Membuat Index.....	71
5.3.2 <i>Mapping</i>	73
5.3.3 <i>Analyzer</i>	76
5.4 Pengujian Pencarian.....	79
5.4.1 Full Text Query.....	80
a. Input query “daging ren”.....	81
b. Input query “tepung”.....	81
5.4.2 Term-level Query.....	82
5.5 Integrasi Data.....	84
BAB VI HASIL DAN PEMBAHASAN.....	85
6.1 Hasil Query Pencarian.....	85
6.1.1 Full Text Query.....	85
a. Input query “daging ren”.....	85
b. Input query “tepung”.....	87
6.1.2 Term-level Query.....	90
6.2 Pembahasan.....	92
BAB VII KESIMPULAN DAN SARAN.....	99
7.1. Kesimpulan.....	99
7.2. Saran.....	100
DAFTAR PUSTAKA.....	101
BIODATA PENULIS.....	103

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Halal Food Nutrition	9
Gambar 3. 1 Arsitektur Sistem dalam Pengerjaan Tugas Akhir	19
Gambar 3. 2 Cara Kerja Analyzers Elasticsearch	20
Gambar 3. 3 Metodologi Tugas Akhir	22
Gambar 4. 1 Flowchart Kode Scraper	27
Gambar 4. 2 Contoh hasil file JSON hasil scraping yang diharapkan	28
Gambar 4. 3 Alur proses pembersihan data	29
Gambar 4. 4 Proses pemisahan nama produk sebaris	29
Gambar 4. 5 Proses penghapusan nama produk tidak valid...	30
Gambar 4. 6 Proses penyeragaman format tanggal	30
Gambar 5. 1 Proses Data Collection	35
Gambar 5. 2 Tampilan Halaman Website LPPOM MUI Kepulauan Riau	36
Gambar 5. 3 Tampilan Halaman Website LPPOM MUI Bali	39
Gambar 5. 4 Tampilan produk halal dari website LPPOM MUI Jateng.....	43
Gambar 5. 5 tampilan kategori-kategori produk halal pada website LPPOM MUI Kalsel	47
Gambar 5. 6 tampilan tabel produk halal LPPOM MUI Kalsel	47
Gambar 5. 7 Tampilan Halaman Depan Website LPPOM MUI Kaltim.....	51
Gambar 5. 8 Tampilan file pdf daftar produk halal LPPOM MUI Kaltim.....	51
Gambar 5. 9 Tampilan halaman depan website LPPOM MUI Jogja	52
Gambar 5. 10 Tampilan daftar produk halal LPPOM MUI Jogja	52
Gambar 5. 11 Tampilan website untuk mengubah file PDF ke CSV	53
Gambar 5. 12 Tampilan pratinjau dan opsi untuk mengunduh file dalam format yang diinginkan	54
Gambar 5. 13 tabel daftar produk halal sebelum dibersihkan	54

Gambar 5. 14 tabel daftar produk halal setelah dibersihkan ..	55
Gambar 5. 15 Tampilan hasil file JSON	57
Gambar 5. 16 tampilan halaman depan website MPU ACEH	57
Gambar 5. 17 tampilan daftar tautan untuk mengunduh daftar produk halal.....	58
Gambar 5. 18 Tampilan pratinjau file PDF daftar produk halal sebelum diunduh dalam bentuk CSV	58
Gambar 5. 19 Daftar produk halal MPU Aceh setelah dibersihkan	59
Gambar 5. 20 data sebelum nama produk dipisahkan.....	67
Gambar 5. 21 data setelah nama produk dipisahkan	67
Gambar 5. 22 Data dengan format tanggal yang belum seragam	69
Gambar 5. 23 Data dengan format tanggal yang sudah seragam	70
Gambar 5. 24 Perintah dalam batch file untuk menjalankan web scraper	70
Gambar 5. 25 Scheduler yang sudah dibuat pada Task Scheduler.....	71
Gambar 5. 26 Tampilan awal Kibana.....	71
Gambar 5. 27 Tampilan pratinjau isi dari file yang diunggah	72
Gambar 5. 28 Tampilan statistik data dari tiap kolom file yang diunggah.....	72
Gambar 5. 29 Tampilan pengaturan awal.....	73

DAFTAR TABEL

Tabel 2.1 Tabel LPPOM MUI Nasional	16
Tabel 3. 1 Contoh Language Analyzers	21
Tabel 3. 2 Contoh kerja dari Autocomplete analyzer.....	21
Tabel 4. 1 Contoh Basis Data ProdukHalal.....	25
Tabel 4. 2 Contoh Basis Data KandunganMakanan.....	26
Tabel 6. 1 Hasil pencarian pada index tanpa analyzer	85
Tabel 6. 2 Hasil query pada index dengan language analyzer	86
Tabel 6. 3 Hasil pencarian pada index dengan autocomplete analyzer dan search analyzer.....	87
Tabel 6. 4 Hasil Pencarian pada Index tanpa Analyzer.....	88
Tabel 6. 5 Hasil query pada index dengan language analyzer	88
Tabel 6. 6 Hasil pencarian pada index dengan autocomplete analyzer dan search analyzer.....	89
Tabel 6. 7 Hasil pencarian pada index tanpa analyzer	90
Tabel 6. 8 Hasil pencarian pada index dengan language analyzer	91
Tabel 6. 9 Hasil query pada index dengan autocomplete analyzer dan search analyzer.....	91
Tabel 6. 10 Perbandingan input dengan hasil pencarian	93
Tabel 6. 11 Perbandingan hasil pencarian menggunakan input query "tepung".....	94
Tabel 6. 12 Perbandingan input dengan hasil pencarian	96

Halaman ini sengaja dikosongkan

DAFTAR KODE

Kode 5. 1 Penulisan Tautan LPPOM Kepulauan Riau	37
Kode 5. 2 Pembuatan Selector Item	37
Kode 5. 3 Penulisan Item field pada File Item	38
Kode 5. 4 Penulisan Kode untuk Halaman Berikutnya.....	38
Kode 5. 5 Command untuk Menjalankan Scraper	39
Kode 5. 6 Penulisan Tautan pada Kode Scraper LPPOM MUI Bali	40
Kode 5. 7 Pembuatan Selector	41
Kode 5. 8 Penulisan Field Item pada File Item	42
Kode 5. 9 Command untuk menjalankan scraper pada LPPOM MUI Bali	42
Kode 5. 10 Penulisan Tautan untuk Kode Scraper.....	44
Kode 5. 11 penulisan kode untuk setiap item yang datanya akan diambil.....	45
Kode 5. 12 penulisan field item pada file item.....	46
Kode 5. 13 Command untuk Menjalankan Scraper pada LPPOM MUI Jawa Tengah	46
Kode 5. 14 Penulisan kode untuk tautan LPPOM MUI Kalsel	48
Kode 5. 15 penulisan kode untuk setiap item yang datanya akan diambil.....	49
Kode 5. 16 penulisan field pada file item.....	49
Kode 5. 17 Command untuk menjalankan scraper pada LPPOM MUI Kalsel.....	50
Kode 5. 18 Kode Phyton untuk Mengubah File CSV menjadi JSON	60
Kode 5. 19 Tampilan hasil file JSON	61
Kode 5. 20 penulisan tautan klikindomaret.....	62
Kode 5. 21 pembuatan selector dan item	63
Kode 5. 22 penulisan item field pada file item.....	64
Kode 5. 23 penulisan kode untuk halaman berikutnya	64
Kode 5. 24 Command untuk menjalankan scraper pada KlikIndomaret	65
Kode 5. 25 Kode untuk memisahkan nama produk	66
Kode 5. 26 data setelah nama produk dipisahkan	68

Kode 5. 27 Kode python untuk menetapkan nilai tanggal	69
Kode 5. 28 Kode untuk membuat mapping	74
Kode 5. 29 Kode untuk membuat language analyzer	77
Kode 5. 30 Query untuk membuat autocomplete analyzer	78
Kode 5. 31 Query untuk melakukan pencarian dengan input "daging ren"	81
Kode 5. 32 Query untuk melakukan pencarian dengan input query "tepung"	82
Kode 5. 33 Query untuk menampilkan hasil pencarian menggunakan Fuzzy query	83

BAB I

PENDAHULUAN

Bab ini akan menjelaskan tentang pendahuluan pengerjaan tugas akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat yang akan diperoleh dari penelitian tugas akhir ini.

1.1. Latar Belakang

Indonesia merupakan salah satu negara dengan populasi penduduk beragama islam terbesar di dunia. Prosentase penduduk muslim di Indonesia mencapai hingga 12,7 persen dari populasi penganut muslim di dunia. Jumlah tersebut berbanding lurus dengan prosentase penganut muslim di Indonesia terhadap jumlah penduduk, yaitu sebesar 88,1 persen [1]. Penganut muslim sendiri memiliki panduan atau hal yang mengatur bagaimana menjalani kehidupan sebagai seorang penganut agama islam, diantaranya yaitu mengenai makanan dan minuman yang baik dan boleh dikonsumsi. Setiap makanan dan minuman yang ingin dikonsumsi oleh seorang muslim harus bersifat halal. Namun, status halal dari sebuah hal tidak selalu dapat terdefinisi dengan jelas, sehingga untuk hal ini Indonesia memiliki sebuah lembaga yang salah satu fungsinya adalah untuk melakukan pengawasan terhadap produk-produk yang makanan, minuman, dan kosmetik yang berstatus halal untuk masyarakat penganut Agama Islam di Indonesia. Lembaga ini bernama LPPOM MUI (Lembaga Pengawasan Pangan, Obat-obatan, dan Kosmetika Majelis Ulama Indonesia).

LPPOM MUI pada awalnya didirikan atas mandat pemerintah/negara agar Majelis Ulama Indonesia (MUI) berperan aktif dalam meredakan kasus lemak babi di Indonesia pada tahun 1988. LPPOM MUI didirikan pada tanggal 6 Januari 1989 untuk melakukan pemeriksaan dan sertifikasi halal [2]. Dalam perjalanannya, LPPOM MUI mengalami penguatan sebagai lembaga yang bertugas melakukan pemeriksaan/audit, penetapan fatwa, dan menerbitkan sertifikat halal. Namun,

dalam menetapkan fatwa dan sertifikasi halal, LPPOM MUI kerap menemui kontroversi dengan banyaknya berita yang beredar tentang kadungan suatu makanan yang diragukan status halalnya.

Dengan perkembangan teknologi seperti saat ini, informasi apapun dapat beredar secara cepat dan mempengaruhi persepsi masyarakat terhadap sesuatu. Beredarnya suatu berita yang tidak memiliki validasi atas status benar atau salahnya biasa disebut hoax. Hoax menjadi suatu momok karena masyarakat pun seringkali membagikan suatu berita hanya melihat berdasarkan judulnya saja, tanpa mengetahui lebih jauh tentang isi yang diberitakan. Berita hoax terus meningkat dari tahun 2002 hingga 2016. Dalam kurun waktu tersebut, terdapat 8.617 kasus dengan sekitar 4.600 kasus terjadi di tahun 2016 [3].

Lebih jauh, hoax ini seringkali berhubungan dengan status halal dari sebuah produk makanan. Seringkali kandungan dari suatu produk makanan dikaitkan dengan keberadaan bahan yang mengandung babi, alkohol, ataupun substansi lain yang diharamkan dalam ajaran agama islam. Setiap daerah di Indonesia pun memiliki banyak produk makanan khas daerahnya masing-masing dan dengan komposisi yang sangat beragam. Dengan jumlah produk makanan khas daerah yang berjumlah lebih dari tiga ribu makanan, seringkali menimbulkan pertanyaan apakah kandungan tersebut terjamin status halalnya. Hal inilah yang menjadi tugas dari LPPOM MUI provinsi yang bertanggung jawab terhadap daerahnya masing-masing. Mereka membuat daftar produk halal dari tiap daerah yang mereka naungi. Namun, tidak semua daerah memiliki situs webnya masing masing, dari 33 LPPOM MUI Regional, hanya 7 yang memiliki situs webnya sendiri. Jumlah yang sedikit itu pun masih tersaring lagi dimana tidak semua situs web tersebut menyediakan daftar produk halal yang terdaftar di daerah tersebut.

Pengembangan data ataupun akuisisi data pada Halal Nutrition Food web dari tiap situs LPPOM MUI tersebut pun membutuhkan data yang lengkap dan akurat. Hasil data akan

dicek redundansinya dengan data yang sudah ada pada situs web Halal Nutrition Food.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan diatas, maka dapat diturunkan menjadi rumusan masalah berikut, antara lain:

1. Bagaimana proses *crawling* data produk halal dari situs LPPOM MUI regional
2. Bagaimana cara menyimpan data hasil *crawling* dalam bentuk *document-based*
3. Bagaimana proses pencarian data produk halal LPPOM MUI regional dalam Halal Nutrition Food

1.3. Batasan Permasalahan

Batasan untuk tugas akhir ini yaitu:

1. Data yang akan diambil/*crawl* merupakan data dari situs LPPOM MUI pusat maupun regional.
2. Data yang dikumpulkan berupa data produk makanan beserta nomor sertifikasinya.

1.4. Tujuan

Tujuan dari penelitian ini adalah untuk menambah sumber data dari aplikasi Halal Nutrition Food. Dengan bertambahnya sumber data, maka daftar produk yang terdapat dalam aplikasi ini menjadi semakin beragam. Berikut tujuan lainnya:

1. Mengumpulkan data produk dari LPPOM MUI regional yang tersebar di seluruh Indonesia
2. Mengintegrasikan data dari LPPOM MUI Regional dengan *Halal Nutrition Food*
3. Menguji performa pencarian sistem basis data elasticsearch pada proses pencarian menggunakan data produk Halal Nutrition Food

1.5. Manfaat

Manfaat yang akan diperoleh pada tugas akhir ini adalah sebagai berikut

1.5.1. Bagi Pengguna

1. Memberikan informasi dan data yang dibutuhkan oleh pengguna.
2. Sebagai acuan terhadap penelitian lebih lanjut yang dilakukan di masa mendatang

1.5.2. Bagi Penulis

1. Dapat memiliki data produk lengkap dari LPPOM MUI baik nasional maupun regional yang nantinya dapat dilakukan penambahan data dari lembaga yang tidak tersedia datanya pada web.
2. Membantu penulis dalam mendalami mengenai proses *crawling* dan pembuatan algoritma *scheduling* dan menerapkannya pada penelitian tugas akhir.

1.6. Relevansi

Tugas akhir ini sangat berhubungan dengan metode pengumpulan data atau *crawling data* dari situs web yang kemudian akan diintegrasikan dengan data yang sudah ada pada website *Halal Nutrition Food*. Hal ini sendiri sesuai dengan salah satu aspek dari laboratorium ADDI, yaitu akuisisi data. Tugas akhir ini sendiri berkaitan dengan mata kuliah Pemrograman Berbasis Web yaitu berkaitan pada saat melakukan scrapping pada website.

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir ini.

2.1. Penelitian Sebelumnya

Sub-bab ini berisi penelitian-penelitian sebelumnya yang dijadikan acuan dalam pengerjaan tugas akhir

2.1.1. Linked Open Data for Halal Food Products

Penelitian ini melakukan pengembangan sebuah sistem *Linked Open Data* untuk produk halal berbentuk aplikasi web dan android yang menyediakan rekomendasi status halal dan kandungan nutrisi dari sebuah produk makanan. Data yang digunakan dalam sistem ini sebagian besar bersumber dari LPPOM MUI, dengan data lainnya memiliki sumber yang beragam yaitu World Open Food Facts, DBpedia, MeSH, dan PubChem. Proses pengembangan sendiri mencakup mengusulkan kosakata makanan halal yang dikembangkan dari kosakata dua makanan yang sudah ada, mentransformasikan sekumpulan data makanan halal menjadi sebuah *Linked Data*, mengintegrasikan data ke *Linked Open Data* yang lain, dan pembuatan aplikasi LODHalal itu sendiri. Akses data yang diberikan pada sistem ini beragam, seperti SPARQL endpoint, RDF dump, dan *web service* [4].

Tugas akhir ini melanjutkan pengembangan pencarian dengan cara menambahkan data yang lebih beragam dari LPPOM MUI regional

2.1.2. Integrasi Data dan Visualisasi Graf pada Aplikasi Halal Nutrition Food

Penelitian yang dilakukan oleh Berlian Fajar Yusuf merupakan pengembangan dari *open data* produk makanan berserta informasinya yang berasal dari seluruh dunia. Dalam penelitian ini diperlukan integrasi data antara Open Food Fact dan Halal Nutrition Food. Data yang diperoleh dibersihkan melalui *data cleansing* yang di dalamnya terdapat pengukuran kesamaan bahan makanan menggunakan Levenshtein dan Jaccard distance untuk memperbaiki kesalahan penulisan. Kemudian, data yang sudah dibersihkan diintegrasikan ke Halal Nutrition Food. Untuk mengetahui kelompok makanan apa saja yang ada, data yang sudah dalam bentuk grafik produk dan bahan makanan, dikelompokkan menggunakan METIS *graph partitioning*. Hasil *graph partitioning* divisualisirkan bersama dengan visualisasi grafik hubungan produk dan bahan makanan, visualisasi grafik produk makanan yang berstatus haram dan visualisasi grafik produk makanan yang mengandung MSG [5].

Penelitian ini memiliki relevansi dengan tugas akhir karena terdapat hubungan dalam integrasi antara sumber data dengan Halal Nutrition Food sebagaimana dalam tugas akhir ini integrasi antara data dari Lembaga LPPOM MUI Regional dengan data Halal Nutrition Food.

2.1.3. Integrasi Data Lembaga Sertifikasi Halal di Dunia dengan Data Halal Nutrition Food

Penelitian yang dilakukan oleh Alkautsar ini melakukan pengembangan terhadap aplikasi Halal Nutrition Food dimana pengembangan yang dilakukan merupakan pengumpulan data dari 10 lembaga sertifikasi dari berbagai negara, yang memiliki list produk pada website masing-masing lembaga tersebut [6]. Data yang didapat akan dihitung similaritasnya dan diintegrasikan dengan data yang sudah ada pada website Halal Nutrition Food. Tugas akhir ini bermanfaat untuk meningkatkan kepercayaan pengguna terhadap produk-produk yang ada pada website Halal Nutrition Food.

Tugas akhir ini dapat melihat dasar teknik penelitian diatas mengenai pengumpulan data dan integrasi data Lembaga sertifikasi dengan Halal Nutrition Food.

2.1.4. The Implementation of Crawling News Page Based on Incremental Web Crawler

Penelitian yang dilakukan oleh Zejian Shi, Minyong Shi dan Weiguo Lin ini mengimplementasikan penggunaan teknologi Web crawler, yaitu Python *web crawler* dan *framework crawler*, Scrapy, untuk meng-crawl halaman web berita secara bertahap dan *real-time*, dan menyimpan datanya ke dalam database [7]. Scrapy Engine berperan untuk mengontrol aliran data pada Scrapy. Scrapy Engine mendapatkan URL untuk di proses dari Spider dan menjadwalkannya sebagai *Request* di Scheduler. Kemudian Request di kirim kepada Downloader untuk selanjutnya dikembalikan dalam bentuk Response yang berisi *source code* HTML dan akan ditangani oleh Spider kembali. Metode yang digunakan untuk menghilangkan pengulangan tautan web yang telah diakses adalah Bloom Filter. Bloom Filter dapat digunakan untuk menilai apakah sebuah elemen sudah ada di dalam koleksi. Jadi, algoritma Bloom Filter dapat mengimplementasikan fitur halaman yang pernah dikunjungi, dengan meletakkan URL yang akan diproses kedalam bagian tersendiri, dan menghapus URL yang berduplikat. Hasil penelitian ini menunjukkan bahwa *web crawler* dapat mengawasi situs web berita dengan Bloom Filter dan menyimpan berita yang diperbarui ke dalam database.

Tugas akhir dapat mengacu pada penelitian diatas terkait dengan penggunaan kerangka kerja Scrapy dan arsitektur serta strukturnya, dalam melakukan *scraping* situs web LPPOM MUI Regional untuk diintegrasikan dengan Halal Nutrition Food.

2.1.5. Optimization of Scheduler for a Web Scraping System

Penelitian yang dilakukan oleh Daniel Martinsson ini mendeskripsikan peningkatan yang telah dilakukan untuk penjadwalan atau *scheduling* dalam sistem *web scraping* yang sudah ada [8]. Karakteristik sistem diperiksa untuk membantu

menyajikan model baru tentang bagaimana menggunakan pendekatan optimasi untuk penjadwalan, yang juga akan membantu membuat penjadwalan otomatis sebagai ganti proses manual yang digunakan sebelumnya. Masalah yang ada termasuk jumlah pekerjaan yang perlu dijadwalkan ke sejumlah *node*, masalah yang mirip dengan penjadwalan pekerjaan toko.

Penjadwal yang dihasilkan meningkatkan sistem dalam beberapa aspek, dengan mengotomatiskan proses penjadwalan dan juga membuat lebih mudah untuk memahami pada frekuensi situs tertentu melalui proses *scraping*. *Input* dari deskripsi situs (prioritas dan waktu untuk melakukan *scraping* terhadap situs) menjadi salah satu kunci untuk meningkatkan penjadwalan lebih jauh, karena situs dengan deskripsi serupa dapat dikelompokkan bersama untuk mempercepat penjadwalan. Hasil penelitian ini menyimpulkan bahwa sistem penjadwalan meningkatkan kerja sistem secara keseluruhan, namun diperlukan beberapa pembatasan terhadap *input* untuk menjaga performa sistem.

Tugas akhir ini dapat melihat dasar konsep dan teknik dalam melakukan penjadwalan, penentuan prioritas dan hal-hal terkait lainnya dari penelitian diatas.

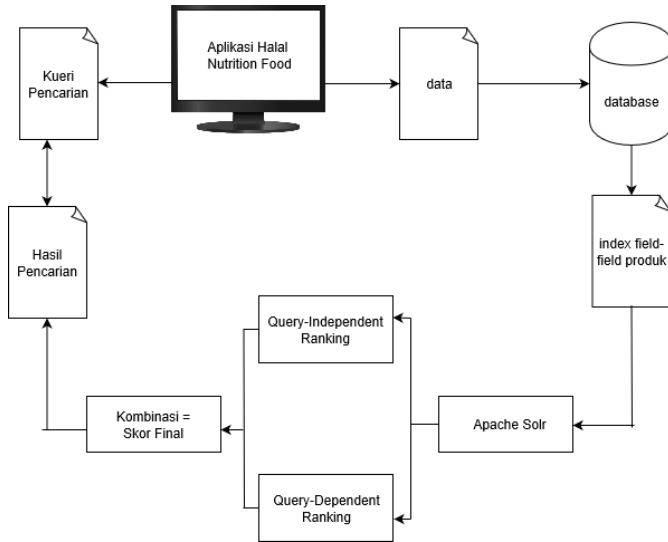
2.2. Dasar Teori

3.2.1 Halal Nutrition Food

Halal Nutrition Food adalah aplikasi yang dibangun untuk membantu pengguna mencari informasi mengenai status halal dari suatu produk makanan, dan informasi nutrisi yang terkandung dalam makanan tersebut. Aplikasi ini berbasis android dan juga web. Halal Nutrition Food menyediakan nutrisi produk halal dimana tidak ditampilkan di situs LPPOM MUI. Halal Nutrition Food memungkinkan pengguna untuk mencari produk makanan halal melalui mesin pencarian dan mengetahui nutrisi berupa fakta nutrisi, komposisi, produsen, sertifikat, dan sebagainya [6]. Pada penelitian ini dilakukan pengembangan pada data yang ada pada aplikasi Halal Nutrition

Food agar lebih relevan dengan persebaran produk makanan saat ini. Berikut mengenai arsitektur Halal Nutrition Food:

Arsitektur sistem dari aplikasi Halal Nutrition Food dibuat melalui *2-layer server* yang saling terkoneksi. Penjelasananya seperti Gambar 2.1



Gambar 2. 1 Arsitektur Halal Food Nutrition

- a. Apache Web Server, di dalamnya terdapat dua sistem yang berjalan yakni basis data dan aplikasi web Halal Nutrition Food [6].
 - Sistem basis data yang digunakan adalah MySQL. Sistem kerjanya adalah ketika pengguna memasukkan data produk halal maka aplikasi web akan menyimpan data-data produk tersebut ke dalam basis data MySQL.
 - Aplikasi android dan web Halal Nutrition Food yang berjalan pada server ini dibangun menggunakan *framework* PHP Laravel yang di dalamnya terdapat fungsi untuk menghubungkan website dengan server Apache Solr, proses

penghubungan keduanya digunakan ketika pengguna mengirimkan kueri pencarian produk melalui halaman aplikasi web kemudian aplikasi web akan mengirimkan kueri tersebut untuk di proses menggunakan metode kombinasi query-independent dan query dependent ranking di dalam server Apache Solr, selanjutnya server Apache Solr akan mengembalikan daftar hasil pencarian ke aplikasi web.

- b. Apache Solr, digunakan untuk memproses pencarian yang dilakukan pengguna dan menyimpan indeks dari dokumen. Dalam Apache Solr akan dilakukan dua proses [6]:
 - Proses *indexing field-field* dari tabel dalam basis data agar pengguna bisa melakukan pencarian berdasarkan data dalam *field* tersebut.
 - *Prosessoring* menggunakan metode kombinasi *query independent* dan *query-dependent ranking* sehingga hasil pencarian akan diurutkan berdasarkan bobot penilaian metode tersebut.

3.2.2 Web Crawler

Web Crawler adalah program yang menelusuri *World Wide Web* dengan cara yang metodis, otomatis dan teratur. Istilah lain untuk web crawler adalah *ant*, *automatic indexer*, *bots*, *web spiders* atau *web robots*. Web Crawler merupakan salah satu bentuk bot atau agen perangkat lunak. Sistem *web crawling* bekerja dengan cara mengunjungi tiap URL yang sudah di *list*, dan melakukan pengambilan data dengan mengikuti struktur web yang saling terhubung. Jika terdapat *hyperlink* di dalam halaman URL, maka akan ditambahkan ke dalam list URL yang akan dikunjungi. Sistem ini disebut *crawl frontier*. URL yang didapat dari *crawl frontier* akan dikunjungi secara rekursif dengan kebijakan tertentu.

3.2.3 Scheduling

Scheduling secara singkat dapat di deskripsikan sebagai daftar tugas yang perlu dikerjakan dalam jangka waktu tertentu. Setiap tugas itu sendiri dapat memiliki jangka waktu dan jumlah pekerjaan yang tetap, ataupun waktu yang bervariasi berdasarkan pada jumlah pekerjaan yang dilakukan. Jangka waktu yang tetap akan mempermudah proses *scheduling* karena waktu telah diketahui sebelum pekerjaan dimulai, dan tidak akan terpengaruh hal lain [8].

Dalam *scraping system*, ada beberapa *nodes* yang kesemuanya bekerja secara independen terhadap situs unik mereka masing-masing. Mereka dapat dilihat sebagai *node* yang mengambil pekerjaan dari daftar pekerjaan yang tersedia.

3.2.4 ElasticSearch

Elasticsearch adalah sebuah mesin pencarian dan analitik yang bersifat *opensource* dan sangat mudah untuk disesuaikan dengan penggunaan-penggunaan khusus, serta memungkinkan pengguna untuk menyimpan, mencari, dan menganalisis data dengan volume yang besar dengan cepat dan mendekati *real time* [9]. Data mentah mengalir ke *Elasticsearch* dari berbagai sumber, termasuk log, metrik sistem, dan aplikasi web. Proses dalam *Elasticsearch* yang paling umum dimanfaatkan adalah *data ingestion*. *Data Ingestion* adalah proses di mana data mentah ini diuraikan, dinormalisasi, dan diperkaya sebelum diindeks dalam *Elasticsearch*. Setelah diindeks di *Elasticsearch*, pengguna dapat menjalankan kueri kompleks terhadap data mereka dan menggunakan agregasi untuk mengambil ringkasan kompleks dari data mereka.

Elasticsearch biasa digunakan dalam proses pembuatan indeks. Indeks *Elasticsearch* adalah kumpulan dokumen yang saling terkait. *Elasticsearch* menyimpan data sebagai dokumen JSON. Setiap dokumen mengkorelasikan seperangkat kunci (nama dari *fields* atau properti) dengan nilai yang sesuai (string, angka, Boolean, tanggal, array nilai, geolokasi, atau tipe data lainnya).

Elasticsearch menggunakan struktur data yang disebut *inverted index* yang dirancang untuk memungkinkan pencarian teks lengkap yang sangat cepat. *Inverted index* mencantumkan setiap kata unik yang muncul di dokumen apa pun dan mengidentifikasi semua dokumen di mana setiap kata tersebut muncul.

Selama proses pengindeksan, *Elasticsearch* menyimpan dokumen dan membuat indeks terbalik untuk membuat data dokumen dapat dicari dalam waktu singkat. Pengindeksan dimulai dengan API indeks, di dokumen JSON dalam indeks tertentu dapat ditambahkan atau diperbarui.

3.2.5 Scrapy

Scrapy atau biasa disebut dengan Web Scrapping merupakan sebuah proses ekstraksi data dari sebuah website. Web scrapping merupakan kegiatan yang dilakukan untuk mengambil data tertentu dari sebuah halaman website sebelum pengambilan data biasanya proses akan menganalisis terlebih dahulu dokumen yang ada sebelum memulai pengumpulan atau pengambilan data. Melakukan scrapping biasanya menggunakan web scrapper, bot, web spider, atau web crawler. Berikut beberapa konsep dan fungsi dasar yang perlu diketahui untuk melakukan scrapy yaitu:

a. Spiders

Spiders adalah kelas yang menentukan bagaimana situs tertentu (atau sekelompok situs) akan di Scrap, termasuk cara melakukan crawl dan cara mengekstraksi data terstruktur dari halaman mereka.

Dengan demikian spiders adalah tempat di mana Anda menentukan perilaku khusus untuk merayapi dan mengurai halaman untuk situs tertentu (atau, dalam beberapa kasus, sekelompok situs).

Konsep ini digunakan pada tugas akhir ketika membuat scrapper untuk tujuh website LPPOM MUI provinsi.

- b. **Item Pipeline**

Setelah item di scrap oleh spiders, item dikirim ke pipeline yang prosesnya melalui beberapa komponen yang di eksekusi secara berurutan. Pipeline ini biasanya sebuah kode python sederhana yang dijalankan secara berurutan dengan fungsi yang bermacam-macam. Untuk kasus tugas akhir ini memungkinkan kita melakukan beberapa proses lainnya seperti membersihkan data HTML, memeriksa data yang duplikat, dan menyimpan data yang di scrap dalam database.
- c. **Selector**

Merupakan mekanisme mengekstraksi data. Disebut selector karena hal ini memilih bagian tertentu dari dokumen html yang telah ditentukan. Pada tugas akhir ini fungsi ini dapat digunakan pada halaman web yang memiliki list produk saja.
- d. **Telnet Console**

Scrapy dilengkapi dengan konsol telnet built-in untuk memeriksa dan mengendalikan proses yang berjalan Scrapy. Konsol telnet hanyalah shell python biasa yang berjalan di dalam proses Scrapy, sehingga kita dapat melakukan apa saja pada telnet console. Apabila dalam tugas akhir nantinya tidak digunakan maka konsol telnet ini dapat di non-aktifkan.
- e. **Settings**

Settings pada scrapy memungkinkan Anda untuk menyesuaikan perilaku semua komponen Scrapy, termasuk inti, ekstensi, pipelines, dan spiders sendiri.
- f. **Link extractors**

Link extractors adalah objek yang tujuannya hanya mengekstrak tautan dari halaman web.

g. Requests and Responses

Biasanya, Request object dihasilkan di spider dan melewati sistem sampai mereka mencapai Downloader, yang mengeksekusi permintaan dan mengembalikan objek Respons yang melakukan perjalanan kembali ke spider yang mengeluarkan permintaan.

h. Feed exports

Fungsi ini merupakan fitur yang sangat penting untuk kasus tugas akhir ini dimana ketika menerapkan scraper bagaimana ia mampu menyimpan data yang ter-scrap dengan benar dan, itu berarti menghasilkan "file ekspor" dengan data yang ter-scrap (biasanya disebut "export feed") untuk dikonsumsi oleh sistem lain.

Dalam tugas akhir ini memungkinkan kita menyimpan data yang dibutuhkan dengan format yang macam-macam terutama CSV.

i. Item

Tujuan utama melakukan scrapy adalah mendapatkan atau mengekstraksi data terstruktur dari sumber yang tidak terstruktur. Pada umumnya spiders mengembalikan data yang telah diekstraksi menjadi python dicts akan tetapi, python dicts tidak memiliki struktur dan mudah terjadi kesalahan ketik maupun data yang tidak konsisten.

Oleh karena itu scrapy menyediakan kelas dengan nama item yang menyediakan kamus dengan sintaks yang mudah untuk mendeklarasikan available fields.

j. Stats Collection

Scrapy juga menyediakan fitur stats collection yang dapat memunculkan statistik dalam bentuk key/value, dimana nilai sering menjadi penghitung.

Jika sewaktu-waktu pada tugas akhir ini membutuhkan modul ini kita dapat mengimport nya dengan menggunakan API nya

3.2.6 LPPOM MUI

LPPOM MUI atau singkatan dari Lembaga Pengkajian Pangan, Obat-obatan, dan Kosmetika Majelis Ulama Indonesia yang bertugas kuat untuk meneliti, mengkaji, menganalisa dan memutuskan apakah produk-produk baik pangan dan turunannya, obat-obatan dan produk kosmetika apakah aman dikonsumsi baik dari sisi kesehatan dan dari sisi pengajaran agama Islam yakni halal atau boleh dan baik untuk dikonsumsi bagi umat Muslim khususnya di wilayah Indonesia, selain itu memberikan rekomendasi, merumuskan ketentuan dan bimbingan kepada layanan masyarakat. Lembaga ini didirikan atas keputusan mendukung Majelis Ulama Indonesia (MUI) berdasarkan surat keputusan perizinan nomor 018/MUI/1989, pada tanggal 26 Jumadil Awal 1409 Hijriah atau 6 Januari 1989. Adapun lembaga ini didirikan bahwa ajaran agama Islam mengatur sedemikian rupa tentang makanan dan minuman Sebagai lembaga otonomi bentukan MUI, LPPOM MUI tidak berjalan sendiri [10].

Keduanya memiliki kaitan erat dalam mengeluarkan keputusan. Sertifikat Halal merupakan langkah yang berhasil dijalankan sampai sekarang. Di dalamnya tertulis fatwa halal MUI yang menyatakan kehalalan suatu produk sesuai dengan syariat Islam dan menjadi syarat pencantuman labelan halal dalam setiap produk makanan minuman, obat-obatan dan kosmetika.

Dalam perjalanannya, LPPOM MUI memiliki 33 perwakilan LPPOM MUI daerah provinsi yang bertanggung jawab untuk menangani masalah terkait produk-produk terkait yang berada pada regional tersebut. Namun, dari ke-33 perwakilan regional tersebut, hanya tujuh yang memiliki halaman web. Berikut tujuh perwakilan daerah yang memiliki website tersebut:

Tabel 2.1 Tabel LPPOM MUI Nasional

No	Nama Lembaga	Daerah	URL	Daftar Prodk	Kandungan Makanan	Perusahaan	Sertifikat	Format Data	API
1	LPPOM MUI Kepulauan Riau	Kepulauan Riau	www.halalmuikpri.com	-	-	-	-	-	-
2	LPPOM MUI Bali	Bali	www.halalmuibali.or.id	ada	-	ada	ada	html	-
3	LPPOM MUI Kalimantan Timur	Kalimantan Timur	halalkaltim.blogspot.co.id	ada	-	ada	ada	pdf	-
4	LPPOM MUI DIY	D.I. Yogyakarta	http://halaljogja.or.id/	ada	-	-	ada	pdf	-

No	Nama Lembaga	Daerah	URL	Daftar Prodk	Kandungan Makanan	Perusahaan	Sertifikat	Format Data	API
5	LPPOM MUI NAD	NAD	mpu.acehprov.go.id	ada	-	ada	ada	pdf	-
6	LPPOM MUI Jateng	Jawa Tengah	www.halaljateng.or.id	ada	-	ada	ada	html	-
7	LPPOM MUI Kalimantan Selatan	Kalimantan Selatan	www.lppommui-kalsel.org	ada	-	ada	ada	html	-

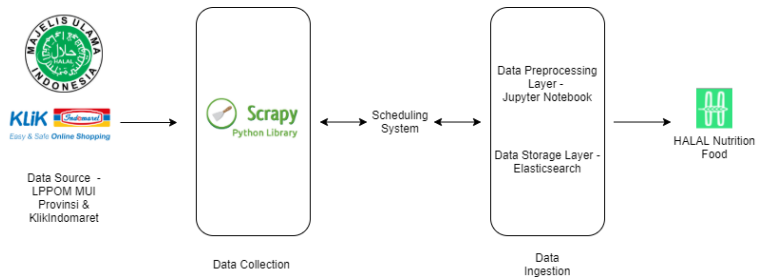
Halaman ini sengaja dikosongkan.

BAB III METODOLOGI

Pada bagian ini dijelaskan metodologi yang akan digunakan sebagai panduan untuk menyelesaikan penelitian tugas akhir ini.

3.1. Arsitektur Sistem

Berikut adalah arsitektur dari sistem yang akan digunakan dalam pengerjaan tugas akhir ini:



Gambar 3. 1 Arsitektur Sistem dalam Pengerjaan Tugas Akhir

3.1.1. Data Source – LPPOM MUI Provinsi & KlikIndomaret
Merupakan sumber data yang terdiri dari tujuh buah website LPPOM MUI provinsi yang akan diambil data produk halalnya. Tipe data yang tersedia dalam *website* yang sudah disiapkan terbagi menjadi dua, yaitu html dan pdf. Sedangkan Klikindomaret [11] adalah website dari indomaret swalayan yang tipe datanya tersimpan dalam bentuk html.

3.1.2. Data Collection

Merupakan tahap dimana proses pengambilan/*scraping* data dilakukan. *Tools* yang digunakan dalam tahap ini adalah *scrapy*. Kemudian data akan dibersihkan dan disimpan dalam sebuah file berformat JSON dimana atribut data akan disesuaikan dengan data yang diperlukan.

3.1.3. Scheduling System

Pada bagian ini, proses penarikan data ke dalam *storage* yang telah ditentukan dibentuk menjadi sebuah proses yang terjadwal secara berkala dengan menggunakan algoritma atau *tools* yang menyediakan fitur yang dibutuhkan.

3.1.4. Data Ingestion

Pada bagian ini data yang telah dikumpulkan akan di simpan dalam sebuah Data Storage Layer yang pada tugas akhir ini menggunakan *Elasticsearch* sebagai medianya. Dalam proses ini, *Elasticsearch* menyediakan fitur indexing yang akan membantu penyajian data ketika ingin diintegrasikan dengan aplikasi Halal Nutrition Food.

Data yang masuk akan melewati serangkaian analisis, yaitu fitur *analyzers* dimana dalam proses ini data yang berbentuk *string* akan dipecah ke dalam bentuk yang lebih mudah untuk proses *indexing*. Cara kerja *analyzers* secara singkat adalah sebagai berikut [12]:



Gambar 3. 2 Cara Kerja Analyzers Elasticsearch

Tipe *analyzer* pada *elasticsearch* yang akan digunakan pada penelitian ini yaitu sebagai berikut, namun tidak terbatas pada:

a. Language Analyzers

Language Analyzers digunakan untuk menganalisis data yang memiliki satu bahasa tertentu. Seluruh data dari LPPOM MUI sendiri memiliki satu bahasa yang sama yaitu bahasa Indonesia. Konfigurasi yang dapat dilakukan dalam *analyzer* tipe ini ada dua, yaitu *stopwords*, sebuah filter yang dapat menyaring kata tertentu yang dikategorikan sebagai *stopwords* agar tidak diikuti dalam hasil analisis, dan *stem exclusion*, sebuah parameter yang memungkinkan pembuatan kumpulan kata-kata *lowercase* yang tidak akan di ambil

dalam proses pemecahan struktur kata. Contoh kerja dari *Language Analyzer* dapat dilihat pada **Tabel 3. 1**

Tabel 3. 1 Contoh Language Analyzers

Sebelum	Sesudah
Jagonya Ayam Kremes Mbak Ning	jago, ayam, kremes, mbak, ning

b. Autocomplete Analyzers

Analyzer ini bekerja dengan memanfaatkan tokenizer *edge_ngram* saat melakukan indexing. *Edge_ngram* bekerja dengan memecah teks menjadi kata-kata setiap kali bertemu salah satu dari daftar karakter yang ditentukan.

Tabel 3. 2 Contoh kerja dari Autocomplete analyzer

Sebelum	Sesudah
Ayam Goreng	a, ay, aya, ayam
Keripik	k, ke, ker, keri, kerip

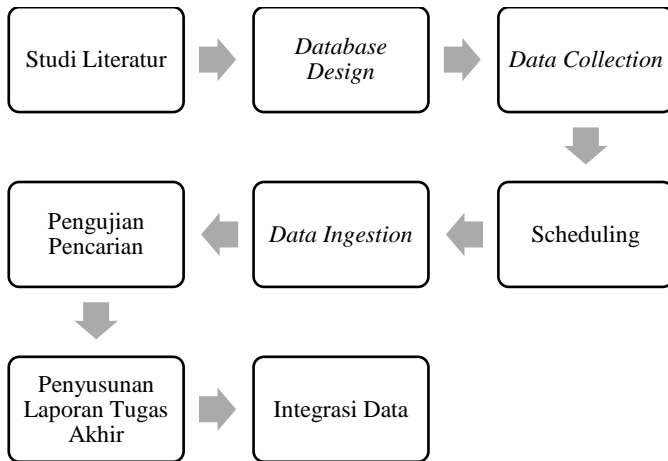
Dari **Tabel 3. 2** menunjukkan bahwa pemisahan dilakukan per karakter dengan penghitungan karakter memiliki parameter batas atas dan bawah yaitu *max_gram* dan *min_gram*.

3.1.5. Halal Nutrition Food

Setelah data melewati proses pengecekan redundansi, maka data akan siap di integrasikan ke dalam Halal Nutrition Fod.

3.2. Tahapan Pengerjaan Tugas Akhir

Pada sub bab ini akan dijelaskan mengenai metodologi dalam pelaksanaan tugas akhir. Berikut gambaran metodologi pelaksanaan tugas akhir.



Gambar 3. 3 Metodologi Tugas Akhir

3.2.1. Studi Literatur

Pada tahap ini sumber literatur yang mendukung pengerjaan tugas akhir dikumpulkan sebanyak mungkin, sebagai acuan dan dasar dalam melakukan penelitian, dalam hal ini *Crawling*. Dasar ilmu yang dimaksud disini adalah konsep, metode, teori, teknik, dan teknologi yang digunakan dalam tugas akhir ini.

3.2.2. Database Design

Desain basis data merupakan tahapan untuk membuat desain untuk mendapatkan sistem basis data yang diperlukan sesuai dengan tujuan penelitian.

3.2.3. Data Collection

Data dapat diperoleh dari tujuh situs web LPPOM MUI Regional yang telah ditetapkan sebelumnya. Data collection dilakukan dengan cara melakukan *web crawling/scraping*. Dalam melakukan proses *crawling*, *crawler* memanfaatkan fitur yang ada pada bahas pemrograman *python* yaitu *Scrapy*. *Scrapy* dilengkapi dengan suatu kerangka kerja tersendiri yang berguna untuk mengeruk atau mengambil data dari situs-situs web yang ada, serta memungkinkan untuk melakukan pengaturan data sesuai kebutuhan. Data yang diperoleh berupa data produk yang

memiliki atribut sebagai berikut: nama produk, nomor sertifikat, dan *Expire date*. Setelah data didapatkan, data dikumpulkan menjadi satu dan dibersihkan agar nilai atau isi dari tiap atribut sesuai dengan ketentuan yang sudah ditentukan dalam tahap Database Design.

3.2.4. Scheduling

Merupakan bagian yang membuat proses pengambilan data menjadi proses yang terjadwal. Data yang sudah dikumpulkan akan mengalami proses integrasi secara rutin yang dibuat dengan algoritma tertentu.

3.2.5. Data Ingestion

Merupakan tahap dimana data disimpan dan diolah sesuai dengan ketentuan yang ditetapkan dalam proses *database design* hingga kemudian siap untuk di implementasikan kepada Halal Nutrition Food.

3.2.6. Pengujian Pencarian

Pada tahap ini dilakukan pengujian terhadap data yang telah diperoleh dalam proses sebelumnya. Dimensi yang akan diuji dari data adalah sebagai berikut [13]:

a. *Completeness*

Memperhitungkan jika set data mencakup semua data yang diperlukan untuk “mewakili setiap keadaan bermakna dari sistem dunia nyata yang diwakili” dan harus mempertimbangkan mengapa suatu nilai hilang.

b. *Accuracy*

Mengevaluasi "sejauh mana data itu benar, dapat diandalkan dan disertifikasi bebas dari kesalahan" dan dapat dihitung sebagai "hasil bagi dari jumlah nilai yang benar dalam suatu sumber dan jumlah keseluruhan nilai"

c. *Consistency*

Mengacu pada "pelanggaran aturan semantik yang didefinisikan atas sekumpulan item data" dan "sejauh mana data selalu disajikan dalam format yang sama dan kompatibel dengan data sebelumnya"

3.2.7. Penyusunan Laporan Tugas Akhir

Penyusunan Laporan Tugas Akhir merupakan, pembuatan dokumentasi terhadap tahap-tahap yang dilakukan dalam tugas akhir ini. Laporan tugas akhir berisikan seluruh tahap pengerjaan tugas akhir mulai dari permasalahan yang diangkat metodologi yang digunakan, perancangan, implementasi, hasil, serta kesimpulan dan saran. Tidak terkecuali pustaka rujukan yang menjadi acuan peneliti. Hasil dari tahap ini berupa buku tugas akhir.

3.2.8. Integrasi Data

Mengimpor hasil akhir dari data yang dikumpulkan yang telah di olah, kedalam basis data Halal Nutrition Food. Untuk data baru yang belum ada pada basis data Halal Nutrition Food, data diimpor secara utuh. Sedangkan untuk data yang sama atau sudah ada dilakukan dengan memperbaharui data produk yang memiliki nama produk sama dengan data Halal Nutrition Food sehingga, hanya dilakukan pembaharuan atau penambahan atribut baru seperti kode produk, sertifikasi, dan lainnya.

BAB IV PERANCANGAN

Pada bab ini akan dijelaskan mengenai rancangan penelitian tugas akhir yang merupakan bagaimana proses penelitian mulai dari merancang basis data hingga pengujian data.

4.1. Database Design

Sebelum data diperoleh dalam tahap *Web Scraping*, sistem basis data dibuat sebagai dasar dari bentuk data yang akan disimpan. Sistem ini akan berisikan atribut-atribut apa saja yang kelak akan disimpan dalam format JSON dan dipergunakan dalam proses Data Ingestion. Berikut contoh gambaran dari sistem basis data yang akan digunakan:

Tabel 4. 1 Contoh Basis Data ProdukHalal

Index		ProdukHalal
Type		_doc
Document	Field	nama_produk
	Field	merek
	Field	nama_perusahaan
	Field	valid_date
	Field	no_sertifikat

Tabel 4. 2 Contoh Basis Data KandunganMakanan

Index		KandunganMakanan
Type		_doc
	Field	nama_produk
Document	Field	komposisi
	Field	daily_value

4.2. Data Collection

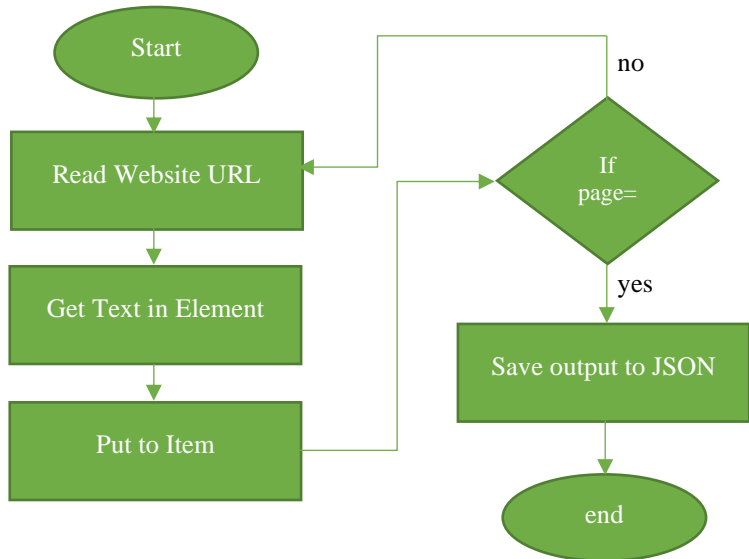
Setelah atribut-atribut yang akan diambil ditentukan dalam tahap *Database Design* selanjutnya dilakukan proses *Data Collection* dengan untuk mendapatkan data yang diinginkan. Dalam proses *data collection* ada beberapa tahap yang dilakukan untuk membuat data siap digunakan.

4.2.1. Melakukan web scraping

Proses *web scraping* dilakukan terhadap website Lembaga LPPOM-MUI dan *website* Klikindomaret yang sudah ditetapkan. Berikut tahap-tahap scraping data yang dilakukan

1. Membuat kode scraper

Pembuatan kode scraper dilakukan terpisah setiap website LPPOM MUI daerah dikarenakan perbedaan struktur html maupun pada *pagination* yang dimiliki oleh masing-masing *website* tersebut. Untuk *flowchart* penulisan kode *scraper* dapat dilihat pada **Gambar 4. 1**



Gambar 4. 1 Flowchart Kode Scraper

Dari **Gambar 4. 1** dapat dilihat pada *flowchart scraper* dimana langkah pertama yang dilakukan adalah pembacaan pada url lalu akan mengambil data dalam elemen html. Selanjutnya *text* dari *element* html dimasukkan pada *item* lalu hal tersebut akan dilakukan seterusnya hingga halaman website tidak lagi tersedia lalu seluruh data dalam *item* akan di impor ke csv yang nantinya akan di gabung menjadi satu.

2. Melakukan web scraping

Melakukan *web scraping* merupakan eksekusi dari kode *scraper* yang telah dibuat pada tahap sebelumnya. Kode *scraper* yang telah dibuat akan dijalankan untuk setiap *website* untuk mengambil data yang ada pada *website-website* yang telah diidentifikasi. Hasil dari tahap ini merupakan data-data dari setiap *website* LPPOM MUI daerah.

4.2.2. Menggabungkan data menjadi satu

Setelah semua kode *scraper* dijalankan pada tiap website LPPOM MUI daerah, hasil dari *scraping* dikumpulkan menjadi satu untuk dilanjutkan ke proses-proses selanjutnya. Berikut contoh hasil data yang diharapkan dari proses *web scraping*:

```

{
  "id_makanan": "1",
  "merek": "Unoe 58",
  "nama_perusahaan": "Unoe 58",
  "nama_produk": "Madu",
  "no_sertifikat": "14120000010119",
  "valid_date": "2022-01-20"
},
{
  "id_makanan": "2",
  "merek": "ARRA",
  "nama_perusahaan": "ARRA Water",
  "nama_produk": "Air Minum Isi Ulang",
  "no_sertifikat": "14120000020119",
  "valid_date": "2022-01-20"
},
{
  "id_makanan": "3",
  "merek": "Cincau Hitam",
  "nama_perusahaan": "UD. Olis Cincau",
  "nama_produk": "Cincau",
  "no_sertifikat": "14190000010119",
  "valid_date": "2022-01-20"
}

```

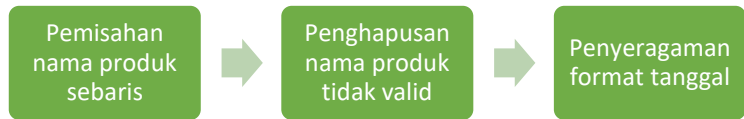
Gambar 4. 2 Contoh hasil file JSON hasil *scraping* yang diharapkan

Gambar 4. 2 merupakan contoh hasil data produk yang diharapkan dari tahap *web scraping*. Meskipun begitu, pada praktiknya pada penelitian tugas akhir ini tidak semua *website* memiliki format pendataan yang sama, sehingga untuk website yang tidak memiliki salah satu atribut yang diharapkan akan diberi nilai 0.

4.2.3 Membersihkan Data

Pembersihan data penelitian ini dilakukan untuk membersihkan dan memisahkan nama produk yang

terkumpul dalam satu baris dan menyeragamkan format *valid date*. Berikut tahap pembersihan data yang dilakukan:



Gambar 4. 3 Alur proses pembersihan data

Pada **Gambar 4. 3** menjelaskan alur proses pembersihan data dari data yang telah dikumpulkan. Berikut penjelasan tahap-tahap dari **Gambar 4. 3**

- a. Pemisahan nama produk sebaris
 Dari data daftar produk halal yang telah didapat, banyak data dari satu merek atau perusahaan yang menyimpan beberapa nama produknya dalam satu baris. Hal ini mengakibatkan jumlah data yang telah discraping tidak menggambarkan jumlah data produk sebenarnya dari semua *website* daftar produk halal yang telah dilakukan *scrape*. Berikut alur proses dari data yang memerlukan pemisahan nama produk:

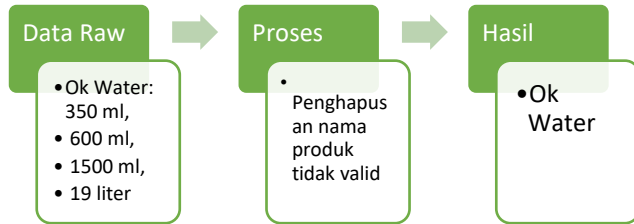


Gambar 4. 4 Proses pemisahan nama produk sebaris

Dari **Gambar 4. 4** menghasilkan data nama produk yang terpisah setiap baris satu. Data seperti diatas dilakukan pemisahan dengan pembatas koma menggunakan *library* Pandas pada Python.

- b. Penghapusan nama produk tidak valid
 Proses pemisahan nama produk yang sebaris memungkinkan data nama produk yang terpisah

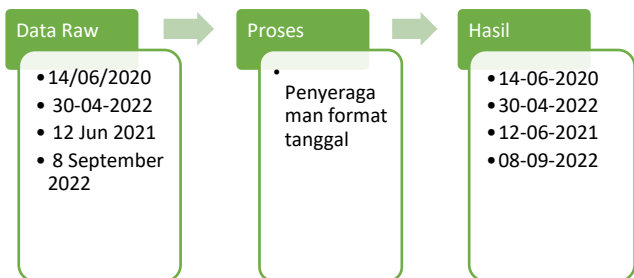
menjadi tidak valid atau hanya berupa angka. Maka dari itu nama produk yang hanya berupa angka dapat dihapus karena pada dasarnya merupakan satu produk yang sama. Berikut gambaran proses untuk menghasilkan data yang diharapkan:



Gambar 4. 5 Proses penghapusan nama produk tidak valid

Gambar 4. 5 menunjukkan proses penghapusan nama produk tidak valid hasil pemisahan produk yang berupa angka seperti 600ml dapat dihapus dan hanya disisakan nama dari produk yaitu Ok Water.

- c. Penyeragaman format tanggal
 Penyeragaman format tanggal dilakukan agar kesemua data tanggal memiliki susunan yang sama yaitu tanggal, bulan, tahun. Penyeragaman dilakukan menggunakan *library* Dateparser Python. Berikut gambaran sederhana data yang ingin dihasilkan:



Gambar 4. 6 Proses penyeragaman format tanggal

Gambar 4. 6 menunjukkan hasil data tanggal yang diinginkan dimana tanggal memiliki format dd-mm-yyyy.

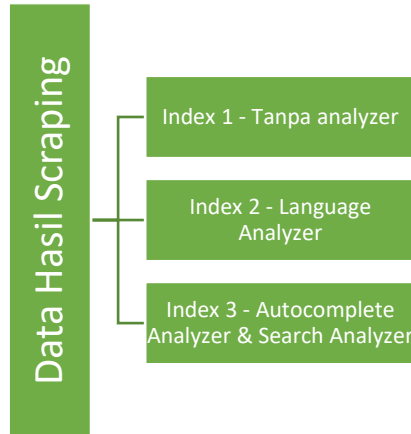
4.3. Scheduling

Merupakan bagian yang membuat proses pengambilan data menjadi proses yang terjadwal. Tahap *web scraping* akan berjalan secara otomatis dengan menggunakan fitur Scheduled Task yang dimiliki oleh Windows. Penjadwalan dilakukan dengan menyesuaikan dengan perubahan atau *update* daftar produk halal dari tiap *website* LPPOM MUI yaitu setiap satu bulan sekali. Proses *web scraping* disimpan dalam bentuk *batch file* dan *file* tersebut dijalankan secara rutin oleh Task Scheduler. Berikut perintah yang diperlukan dalam *batch file* untuk *scheduling*:

1. Alamat untuk mengaktifkan *environment* yang menjalankan Python
2. Perubahan direktori menjadi alamat dimana *project scrapy* berada
3. Perintah untuk menjalankan *file* Scrapy dengan luaran yang ditetapkan

4.4. Data Ingestion

Data yang sudah sesuai dengan format pada *database design* disimpan dalam sebuah sistem penyimpanan yang juga bekerja sebagai *search engine* yaitu Elasticsearch. Elasticsearch memberikan sebuah sistem *indexing* dengan melakukan *mapping*. *Mapping* sendiri berisikan *analyzer*, yaitu sebuah fitur untuk menganalisa data yang dapat membantu kerja dari mesin pencarian. *Analyzer* yang akan digunakan adalah Language Analyzer dimana *analyzer* ini membagi teks kedalam bentuk perkata sebagai *token* dengan acuan bahasa yaitu bahasa Indonesia. Sistem ini memudahkan proses pencarian seperti pada *search engine* yaitu dengan memberikan sebuah *key* pada masing-masing data yang dimiliki dan menghasilkan sugesti pencarian ketika memasukan huruf atau *key* tertentu. Pada tugas akhir ini, data yang dimasukkan kedalam Elasticsearch dimasukkan kedalam tiga buah index berbeda seperti pada gambar berikut:

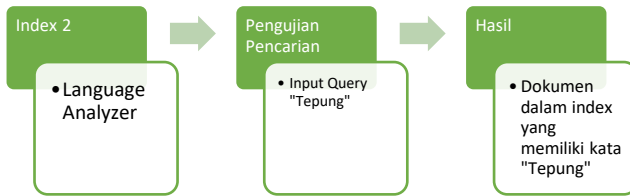


Gambar 4.7 Proses pembuatan tiga indeks

Pada **Gambar 4.7**, Index 1 dibuat tanpa menggunakan analyzer. Kemudian index 2 dibuat dengan menggunakan satu buah analyzer, yaitu Language Analyzer. Lalu Index 3 dibuat menggunakan dua buah analyzer yaitu Autocomplete Analyzer dan Search Analyzer. Perbedaan penggunaan analyzer pada tiap index berguna untuk menguji apakah akan menampilkan hasil pencarian yang berbeda untuk input query yang sama. Kemudian *query* yang akan digunakan adalah *Match Query* dan *Fuzzy Query*.

4.5. Pengujian Pencarian

Menguji ketepatan dan kesesuaian data yang berada dalam *index* ketika dilakukan pencarian. Proses ini menggunakan beberapa *query* yang disediakan oleh Elasticsearch *Search API* dan dengan tiga buah *input query* yang berbeda. Ketiga input query akan dicoba pada tiap index, untuk menguji akurasi dari tiap index yang memiliki analyzer yang berbeda-beda. Berikut gambaran sederhana dari pengujian pencarian pada salah satu index:



Gambar 4.8 Proses pencarian terhadap data yang ada di dalam indeks

Gambar 4.8 menunjukkan alur pengujian pada index yang menggunakan Language Analyzer. Kemudian pada index ini dilakukan pencarian dengan contoh input query “Tepung”. Hasil pencarian yang diharapkan adalah dokumen-dokumen pada index yang memiliki kata “Tepung” sebagai salah satu value-nya.

4.6. Integrasi Data

Mengimpor hasil akhir dari data yang dikumpulkan yang telah di olah, kedalam basis data Halal Nutrition Food. Untuk data baru yang belum ada pada basis data Halal Nutrition Food, data diimpor secara utuh. Sedangkan untuk data yang sama atau sudah ada dilakukan dengan memperbaharui data produk yang memiliki nama produk sama dengan data Halal Nutrition Food sehingga, hanya dilakukan pembaharuan atau penambahan atribut baru seperti kode produk, sertifikasi, dan lainnya.

Halaman ini sengaja dikosongkan

BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai implementasi tugas akhir yang merupakan bagaimana proses penelitian mulai dari merancang basis data hingga pengujian data.

5.1. Data Collection

Seperti yang telah dijelaskan pada sub bab 4.2 *data collection* memiliki tiga tahap agar data dapat dikumpulkan berikut dengan penjelasan *scraping data* untuk setiap *website* LPPOM MUI daerah. Dari tujuh *website* LPPOM MUI daerah yang sudah ditetapkan, *website* yang memiliki data dalam bentuk *html* sendiri hanya berjumlah empat. Sedangkan, untuk tiga *website* sisanya data yang tersedia berbentuk tautan untuk mengunduh *file* berformat *pdf*, sehingga tidak dilakukan *web scraping*, dan hanya dilakukan *convert* atau perubahan format *file*. Untuk tahap *scraping data* untuk semua *website* cenderung sama, yang membedakannya hanya struktur, elemen *html* dan *pagination* yang dimiliki. Berikut tahap *crawling data* untuk seluruh *website*:



Gambar 5. 1 Proses Data Collection

Pada gambar 5.1 terdapat alur proses *data collection* yang selanjutnya akan dijelaskan satu persatu setiap LPPOM MUI daerah.

5.1.1 Melakukan web scraping

a. LPPOM MUI Kepulauan Riau

Berikut gambar tampilan dari halaman *website* LPPOM MUI Kepulauan Riau yang menyediakan daftar produk halal:



NO.	PRODUSEN	KATEGORI	MERKEK DAGANG	NO. SERTIFIKAT	EXPIRED DATE	LOKASI	STATUS
1	Cv. Sai Abadi	Katering	Sai Abadi	04-33-00286-0718	08 Jul 2020	Batam	AKTIF
2	PT. Wahana Tirta Mienta	Minuman dan Bahan Minuman	Mienty, Fineson, Agalia	04-12-00051-0106	08 Jul 2020	Batam	AKTIF
3	UD. Kopi 20	Minuman dan Bahan Minuman	Kopi 20	04-12-00282-0718	08 Jul 2020	Batam	AKTIF
4	M. Ong Bakery	Roti dan Kue (Bakery)	M. Ong Bakery	04-20-00285-0718	08 Jul 2020	Batam	AKTIF
5	M. Ong Bakery	Minuman dan Bahan Minuman	M. Ong Bakery	04-12-00285-0718	08 Jul 2020	Batam	AKTIF
6	Asam Bandung	Restoran	Asam Bandung	04-16-00283-0718	08 Jul 2020	Batam	AKTIF
7	PT. Sri Iwan Bangun Perkasa (Sriwan Hotel)	Restoran	Restaurant Campaka	04-16-00159-1113	08 Jul 2020	Batam	AKTIF
8	PT. Mitra Petala Nusantara	Restoran	Pharis Coffee Shop	04-16-00284-0718	08 Jul 2020	Batam	AKTIF
9	Batam Sehat Soua & Tahu	Tumbuhan dan Produk Tumbuhan Lain-lain	Batam Sehat Soua & Tahu	04-16-00288-0718	28 Jul 2020	Batam	AKTIF

Gambar 5. 2 Tampilan Halaman Website LPPOM MUI Kepulauan Riau

Gambar 5. 2 dimana nantinya data yang ada pada tabel-tabel tersebut akan di ambil datanya. Pada *website* ini terdapat perbedaan pada struktur *html* dan *pagination* dibandingkan dengan *website-website* lainnya. Untuk itu terdapat beberapa hal yang akan dilakukan agar data dari *website* ini dapat di ambil:

1. Menulis Kode *Scraper*

i. Penulisan tautan

Pada bagian ini cukup menuliskan tautan yang akan digunakan untuk pengambilan data lalu dituliskan pada bagian awal kode *scraper*, apabila terdapat tautan yang berbeda-beda untuk beberapa produk maka dapat langsung ditambahkan setelah penulisan tautan pertama dimana harus dipisahkan dengan tanda koma. Apabila terdapat sangat banyak tautan seperti *next_page* maka dalam Scrapy dapat memanfaatkan fitur *pagination* pada tautan yang di ambil datanya. Berikut penulisan tautan pada kode *scraper* LPPOMMUI kepulauan riau:


```

1. import scrapy
2. from ..items import LppomriauItem
3.
4. class crawlriau(scrapy.Spider):
5.     name = 'lppomriau'
6.     page_number = 2
7.     start_urls = ['https://www.halal-
    kepri.com/tampil_act.php?pageNum_tampil=0&to
    talRows_tampil=558']

```

Kode 5. 1 Penulisan Tautan LPPOM Kepulauan Riau

Pada **Kode 5. 1** baris ke-7 objek `start_urls` merupakan penulisan tautan *website* yang akan di lakukan pengambilan data.

ii. Penulisan *selector*

```

1. def parse(self, response):
2.     for row in response.xpath('//*[@class="table"]/tbody/tr'):
3.         item = LppomriauItem()
4.
5.         kategori_produk = row.xpath("td[3]//text()").extract_first()
6.         merek           = row.xpath("td[4]//text()").extract_first()
7.
8.         no_sertifikat   = row.xpath("td[5]//text()").extract_first()
9.
10.        valid_date       = row.xpath("td[6]//text()").extract_first()
11.
12.        nama_perusahaan  = row.xpath("td[2]//text()").extract_first()
13.
14.        item['kategori_produk'] = kategori_produk
15.        item['merek'] = merek
16.        item['no_sertifikat'] = no_sertifikat
17.        item['valid_date'] = valid_date
18.        item['nama_perusahaan'] = nama_perusahaan
19.
20.        yield item

```

Kode 5. 2 Pembuatan Selector Item

Kode 5. 2 diatas menjelaskan cara iterasi untuk melakukan pengambilan data yang telah di pilah-pilah sesuai dengan posisinya pada tag html. Serta dibuatkan *item* agar data yang di dapat hasilnya terstruktur dengan baik.

```

1. import scrapy
2.
3. class LppomriauItem(scrapy.Item):
4.     # define the fields for your item here like
5.     :
6.     # name = scrapy.Field()
7.     kategori_produk = scrapy.Field()
8.     merek = scrapy.Field()
9.     no_sertifikat = scrapy.Field()
10.    valid_date = scrapy.Field()
11.    nama_perusahaan = scrapy.Field()
12. pass

```

Kode 5. 3 Penulisan Item field pada File Item

Kode 5. 3 merupakan data file item yang ditulis untuk mendefinisikan *item field*. Penulisan *item field* sesuai dengan kode *crawl* yang di buat pada *file scraper/crawl*.

iii. Tahap *pagination*

Pada kasus *website* dari LPPOM MUI kepri data pada tabel ternyata tidak ditampilkan langsung dalam satu halaman, melainkan terbagi ke dalam 37 halaman yang dimana tiap halamannya hanya menampilkan 15 data. Sehingga pada tugas akhir ini dilakukan perulangan untuk tautan LPPOM MUI kepri karena setelah dilihat tautannya hanya menambah angka di setiap *pagenya*. Jadi:

```

1. next_page = 'https://www.halal-
   kepri.com/tampil_act.php?pageNum_tampil=' + str(crawlriau
   .page_number) + '&totalRows_tampil=558'
2.     if crawlriau.page_number < 38:
3.         crawlriau.page_number += 1
4.     yield response.follow(next_page, callback = self.parse)

```

Kode 5. 4 Penulisan Kode untuk Halaman Berikutnya

Dari **Kode 5. 4** *page_number* sebelumnya telah di tulis pada **Kode 5. 1** sehingga, langsung dapat digunakan. Selain itu disini hanya dilakukan perulangan lalu di lakukan *callback* agar tautan utamanya muncul lalu ditambah hingga *page* ke 37.

2. Melakukan Web Scraping

Melakukan Web Scraping berarti menjalankan kode program yang telah ditulis sebelumnya. *Command* untuk menjalankan *scrapper* yang telah dibuat adalah sebagai berikut:

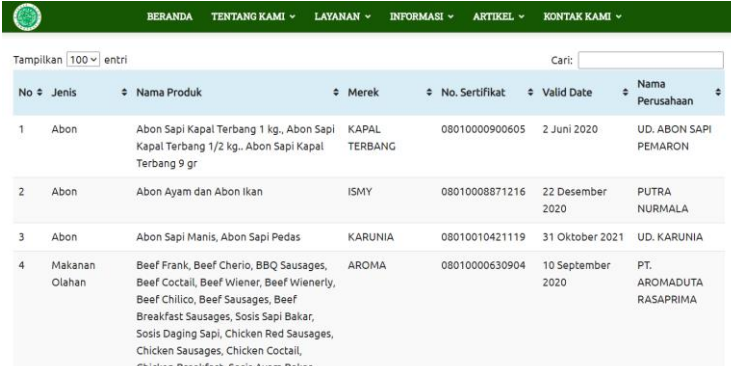
```
scrapy crawl lppomriau -o items.json
```

Kode 5. 5 Command untuk Menjalankan Scrapper

Dari **Kode 5. 5**, terdapat perintah `-o` yang berarti menetapkan format dari *output* yang dihasilkan oleh *scrapper*. Pada tugas akhir ini kode program dijalankan secara terpisah karena struktur, elemen html untuk setiap website berbeda-beda. Setelah kode program untuk LPPOM MUI Kepri dijalankan terdapat 543 data yang berhasil di *scraping*. Seluruh data tersebut nantinya akan dikumpulkan menjadi satu setelah seluruh kode program selesai dijalankan.

b. LPPOM MUI Bali

Berikut tampilan halaman website LPPOM MUI Bali yang terdapat produk dan propertinya:



No	Jenis	Nama Produk	Merek	No. Sertifikat	Valid Date	Nama Perusahaan
1	Abon	Abon Sapi Kapal Terbang 1 kg., Abon Sapi Kapal Terbang 1/2 kg., Abon Sapi Kapal Terbang 9 gr	KAPAL TERBANG	08010000900605	2 Juni 2020	UD. ABON SAPI PEMARON
2	Abon	Abon Ayam dan Abon Ikan	ISMY	08010008871216	22 Desember 2020	PUTRA NURMALA
3	Abon	Abon Sapi Manis, Abon Sapi Pedas	KARUNIA	08010010421119	31 Oktober 2021	UD. KARUNIA
4	Makanan Olahan	Beef Frank, Beef Cherio, BBQ Sausages, Beef Cocktail, Beef Wiener, Beef Wienerly, Beef Chilico, Beef Sausages, Beef Breakfast Sausages, Sosis Sapi Bakar, Sosis Daging Sapi, Chicken Red Sausages, Chicken Sausages, Chicken Cocktail, Chicken Breakfast, Sosis Ayam Bakar,	AROMA	08010000630904	10 September 2020	PT. AROMADUTA RASAPRIMA

Gambar 5. 3 Tampilan Halaman Website LPPOM MUI Bali

Gambar 5. 3 diatas mirip dengan tabel dari LPPOM MUI Kepri akan tetapi pada halaman ini properti dari suatu produk lebih

banyak dibandingkan data dari LPPOM MUI sebelumnya. Pada dasarnya semua kode *scraper* dibuat mirip akan tetapi, yang membedakannya tergantung pada struktur html yang dimiliki website dimana tabel produk tersedia. Perbedaan tersebut dapat berupa dari selectornya maupun pada halamannya.

Pada website LPPOM MUI Bali tidak terlalu memiliki perbedaan dari *website-website* LPPOM MUI lainnya jika dilihat dari struktur html nya. Untuk itu terdapat beberapa hal yang akan dilakukan agar data dari website ini dapat di ambil:

1. Menulis Kode *Scraper*

i. Penulisan tautan

Penulisan tautan untuk kode scraper LPPOM MUI Bali sama halnya dengan LPPOM MUI sebelumnya dan begitu pula untuk website-website lainnya. Untuk LPPOM MUI Bali sendiri penulisan kode untuk tautannya yaitu:

```

1. import scrapy
2. from ..items import LppombaliItem
3.
4. class crawlbali(scrapy.Spider):
5.     name = 'lppombali'
6.     start_urls = ['http://www.halalmuibali.or.id/daftar-penerima-sertifikat-halal-lppom-mui-prov-bali-periode-mar-2020/']

```

Kode 5. 6 Penulisan Tautan pada Kode Scraper LPPOM MUI Bali

Kode 5. 6 penulisan tautan terdapat pada objek `start_urls`. Tautan tersebut merupakan halaman dimana data produk tersedia.

ii. Membuat *selector*

```

1. def parse(self, response):
2. for row in response.xpath('//*[@class="tablepress tablepress-id-
   28"]//tbody/tr'):
3.     item = LppombaliItem()
4.
5.     nama_produk = row.xpath("td[3]//text()").extract_first()
6.     merek = row.xpath("td[4]//text()").extract_first()
7.     no_sertifikat = row.xpath("td[5]//text()").extract_first()
8.     valid_date = row.xpath("td[6]//text()").extract_first()
9.     nama_perusahaan= row.xpath("td[7e]//text()").extract_first()
10.
11.     item['nama_produk'] = nama_produk
12.     item['merek'] = merek
13.     item['no_sertifikat'] = no_sertifikat
14.     item['valid_date'] = valid_date
15.     item['nama_perusahaan'] = nama_perusahaan
16.
17.     yield item

```

Kode 5.7 Pembuatan Selector

Kode 5.7 merupakan penulisan selector pada baris ke-2 menggunakan xpath untuk mendapatkan data yang dibutuhkan. Data produk yang akan di ambil disini terdapat lima item, masing-masing berupa nama produk, merek produk, nomor sertifikat, *valid date*, dan nama perusahaan. Kelima atribut tersebut harus di inisiasi per item agar ketika data di scrape, datanya menjadi teratur. Dalam scrapy ketika menulis program item seperti diatas maka harus membuat field item sesuai jumlah yang di inisiasi pada kode crawler yang dibuat. Berikut penulisan field pada file item:

```

1. import scrapy
2.
3. class LppombaliItem(scrapy.Item):
4.     # define the fields for your item here like
5.     :
6.     # name = scrapy.Field()
7.     nama_produk      = scrapy.Field()
8.     merek            = scrapy.Field()
9.     no_sertifikat    = scrapy.Field()
10.    valid_date       = scrapy.Field()
11.    nama_perusahaan  = scrapy.Field()
12.    pass

```

Kode 5. 8 Penulisan Field Item pada File Item

Penulisan *field item* pada sama dengan yang terdapat pada kode *selector* di file scraper.

2. Melakukan Web Scraping

Melakukan Web Scraping berarti menjalankan kode program yang telah ditulis sebelumnya. *Command* untuk menjalankan scraper yang telah dibuat adalah sebagai berikut:

```
scrapy crawl lppombali -o items.json
```

Kode 5. 9 Command untuk menjalankan scraper pada LPPOM MUI Bali

Setelah kode program untuk LPPOM MUI Bali dijalankan terdapat 268 data yang berhasil di scraping. Seluruh data tersebut nantinya akan dikumpulkan menjadi satu setelah seluruh kode program selesai dijalankan.

c. LPPOM MUI Jateng

Berikut tampilan website yang memuat daftar produk halal dari LPPOM MUI Jateng:



Pencarian Produk Halal !

Input Search

ID	Nama Produk	Nama Perusahaan	No Sertifikat	Masa Aktif	Masa Berlaku
2	Semprong egg roll (Variasi Terlampir)	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku
3	Telur Blanak Keju	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku
4	Krupuk Pare	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku
5	Pangsit	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku
6	Kripik selederi	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku
7	Onde Ketawa	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku
8	Silik (Variasi terlampir)	RULY SNACK	15100028980218	2020-02-10	Sertifikat sudah tidak berlaku

Gambar 5. 4 Tampilan produk halal dari website LPPOM MUI Jateng

Pada website LPPOM MUI Jateng terdapat sedikit perbedaan dimana tabel produk halal yang dimiliki disimpan dalam bentuk sebuah html form. Untuk itu terdapat beberapa hal yang akan dilakukan agar data dari website ini dapat diambil:

1. Menulis Kode *Scraper*

i. Penulisan tautan

Penulisan tautan untuk LPPOM MUI Jateng berbeda dari website-website sebelumnya, dimana untuk kasus ini, website LPPOM MUI Jateng mengharuskan pengguna untuk memasukkan sebuah query kedalam sebuah form pencarian yang memiliki method POST untuk menampilkan tabel produk halal. Untuk LPPOM MUI Jateng sendiri penulisan kode untuk tautan yaitu:

```

1. import scrapy
2. from ..items import LppomjatengItem
3.
4. class crawlkalsel(scrapy.Spider):
5.     name = 'lppomjateng'
6.
7.     def start_requests(self):
8.         return [scrapy.Request( "http://www.halaljateng
9.             .or.id/index.php/user",
10.                 callback=self.parse_item)
11. ]
12.
13.     def parse_item(self, response):
14.         return scrapy.FormRequest.from_response(
15.             response,
16.             formdata={'query' : ''},
17.             callback=self.parse

```

Kode 5. 10 Penulisan Tautan untuk Kode Scraper

Pada **Kode 5. 10** terdapat kode yang berbeda dari website-website sebelumnya, yaitu penempatan tautan berada dalam class `start_request` pada baris ke-7. Lalu, `callback` ke method `parse_item` pada baris ke-9 bertujuan untuk membuat sebuah simulasi klik dari form pencarian yang terdapat dalam website, agar dapat menampilkan tabel produk halal.

ii. Membuat *selector*

```

1. def parse(self, response):
2.     for row in response.css('tr'):
3.         item = LppomjatengItem()
4.
5.         nama_produk    = row.xpath("td[2]//text()").extract_first()
6.         no_sertifikat  = row.xpath("td[4]//text()").extract_first()
7.         valid_date     = row.xpath("td[5]//text()").extract_first()
8.         nama_perusahaan= row.xpath("td[3]//text()").extract_first()
9.
10.        item['nama_produk'] = nama_produk
11.        item['no_sertifikat'] = no_sertifikat
12.        item['valid_date'] = valid_date
13.        item['nama_perusahaan'] = nama_perusahaan
14.
15.        yield item
16.    pass

```

Kode 5. 11 penulisan kode untuk setiap item yang datanya akan diambil

Kode 5. 11 diatas dilakukan untuk mendapatkan data yang dibutuhkan dengan cara menetapkan letak dari tiap data pada struktur html website tersebut. Data produk yang akan diambil disini terdapat empat item, yaitu nama produk, nama perusahaan, valid date, dan nomor sertifikat. Dalam scrapy ketika menulis program item seperti diatas maka diharuskan membuat field item sesuai jumlah yang di inisiasi pada kode scraper yang telah ditulis. Berikut penulisan field pada file item:

```

1. import scrapy
2.
3.
4. class LppomjatengItem(scrapy.Item):
5.     # define the fields for your item here like
6.     :
7.     # name = scrapy.Field()
8.     nama_produk      = scrapy.Field()
9.     no_sertifikat    = scrapy.Field()
10.    valid_date       = scrapy.Field()
11.    nama_perusahaan  = scrapy.Field()
12.    pass

```

Kode 5. 12 penulisan field item pada file item

2. Melakukan Web Scraping

Melakukan Web Scraping berarti menjalankan kode program yang telah ditulis sebelumnya. *Command* untuk menjalankan scraper yang telah dibuat adalah sebagai berikut:

```
scrapy crawl lppomjateng -o items.json
```

Kode 5. 13 Command untuk Menjalankan Scraper pada LPPOM MUI Jawa Tengah

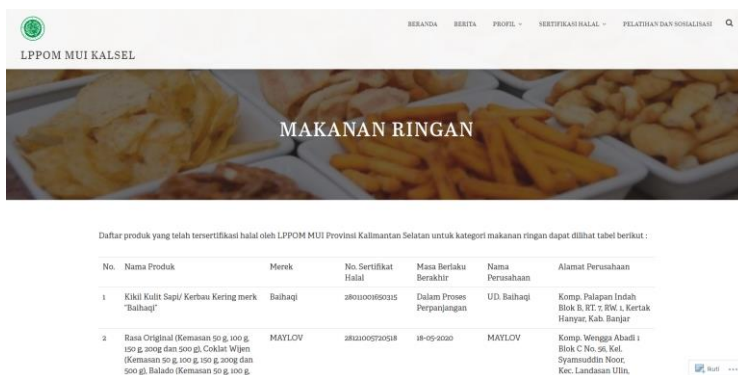
Setelah kode program untuk LPPOM MUI Jateng dijalankan terdapat 1123 data yang berhasil di scraping. Seluruh data tersebut nantinya akan dikumpulkan menjadi satu setelah seluruh kode program selesai dijalankan.

d. LPPOM MUI Kalsel

Berikut merupakan tampilan dari daftar produk halal pada website LPPOM MUI Kalsel:



Gambar 5. 5 tampilan kategori-kategori produk halal pada website LPPOM MUI Kalsel



Gambar 5. 6 tampilan tabel produk halal LPPOM MUI Kalsel

Dari Gambar 5. 5 dan Gambar 5. 6 dapat dilihat bahwa website LPPOM MUI Kalsel membagi produknya ke banyak tautan. Meskipun begitu, struktur html yang cenderung umum digunakan membuat website ini tidak menyulitkan untuk dilakukannya web scraping. Untuk itu terdapat beberapa hal yang akan dilakukan agar data dari website ini dapat diambil:

1. Menulis Kode Scraper

i. Penulisan tautan

Penulisan tautan untuk website LPPOM MUI Kassel sama halnya dengan website-website sebelumnya. Berikut penulisan kode untuk tautan dari LPPOM MUI Kassel:

```

1. import scrapy
2. from ..items import LppomkasselItem
3.
4. class crawlkassel(scrapy.Spider):
5.     name = 'lppomkassel'
6.     start_urls = [
7.         'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/bakery-aneka-kue-dan-roti/',
8.         'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/makanan-ringan/',
9.         'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/produk-olahan-ikan/',
10.        'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/produk-olahan-nabati/',
11.        'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/air-minum-dalam-kemasan-amdk/',
12.        'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/minuman-bahan-minuman-pemanis-dan-madu/',
13.        'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/aneka-sambal-dan-telur-asin/',
14.        'https://lppommui-kassel.org/daftar-perusahaan-
tersertifikat/lain-lain/']

```

Kode 5. 14 Penulisan kode untuk tautan LPPOM MUI Kassel

Karena daftar produk atau tabel dari website ini terbagi menjadi delapan tautan yang berbeda, maka delapan tautan tersebut dituliskan dengan method `start_urls` pada baris ke-6. Dengan begitu, spider nantinya akan langsung melakukan proses scraping terhadap kedelapan url tersebut.

ii. Membuat *selector*

```

1. def parse(self, response):
2.     for row in response.xpath('//table/tbody/tr'):
3.         item = LppomkalselItem()
4.
5.         nama_produk      = row.xpath("td[2]//text()").extract_first()
6.         merek            = row.xpath("td[3]//text()").extract_first()
7.         no_sertifikat    = row.xpath("td[4]//text()").extract_first()
8.         valid_date       = row.xpath("td[5]//text()").extract_first()
9.         nama_perusahaan = row.xpath("td[6]//text()").extract_first()
10.
11.
12.         item['nama_produk'] = nama_produk
13.         item['merek']      = merek
14.         item['no_sertifikat'] = no_sertifikat
15.         item['valid_date'] = valid_date
16.         item['nama_perusahaan'] = nama_perusahaan
17.
18.     yield item

```

Kode 5. 15 penulisan kode untuk setiap item yang datanya akan diambil

Kode 5. 145 menunjukkan kelima item yang akan diambil, yaitu nama produk, merek, nomor sertifikat, valid date, dan nama perusahaan. Lalu diperlukan juga kode dari field item yang sama pada file item sebagai *container* atau tempat yang menampung data item yang akan di scrape. Berikut penulisan field pada file item:

```

1. import scrapy
2.
3. class LppomkalselItem(scrapy.Item):
4.     # define the fields for your item here like:
5.     # name = scrapy.Field()
6.     nama_produk      = scrapy.Field()
7.     merek            = scrapy.Field()
8.     no_sertifikat    = scrapy.Field()
9.     valid_date       = scrapy.Field()
10.    nama_perusahaan = scrapy.Field()
11.    pass

```

Kode 5. 16 penulisan field pada file item

Penulisan field item pada **Kode 5. 16** sesuai dengan item yang ada pada file scraper.

2. Melakukan Web Scraping

Melakukan Web Scraping berarti menjalankan kode program yang telah ditulis sebelumnya. *Command* untuk menjalankan scraper yang telah dibuat adalah sebagai berikut:

```
scrapy crawl lppomkalsel -o items.json
```

Kode 5. 17 Command untuk menjalankan scraper pada LPPOM MUI Kalsel

Website LPPOM MUI Kalsel membagi daftar produk halalnya kedalam delapan tautan yang berbeda, maka, saat dijalankan scrapy melakukan proses scraping terhadap semua url yang diinginkan. Dalam proses scraping ini, spider melakukan scraping secara teracak, tidak memperhatikan urutan url yang tertera. Namun, hal ini tidak menjadi masalah dikarenakan output yang diharapkan tidaklah memerlukan sebuah urutan berdasarkan kategori, melainkan keseluruhan isi dari tabel-tabel produk itu sendiri. Setelah kode program untuk LPPOM MUI Kalsel dijalankan terdapat total 361 data yang berhasil di scraping. Seluruh data tersebut nantinya akan dikumpulkan menjadi satu setelah seluruh kode program selesai dijalankan.

e. LPPOM MUI Kaltim

Lembaga ini memiliki tampilan website sebagai berikut:



Gambar 5. 7 Tampilan Halaman Depan Website LPPOM MUI Kaltim

Dalam websitenya, lembaga ini menyediakan daftar produk halal nya dalam format file pdf, sehingga pengguna harus mengunduh file tersebut untuk menggunakan file tersebut. Berikut tampilan file pdf dari daftar produk halal LPPOM MUI Kaltim:

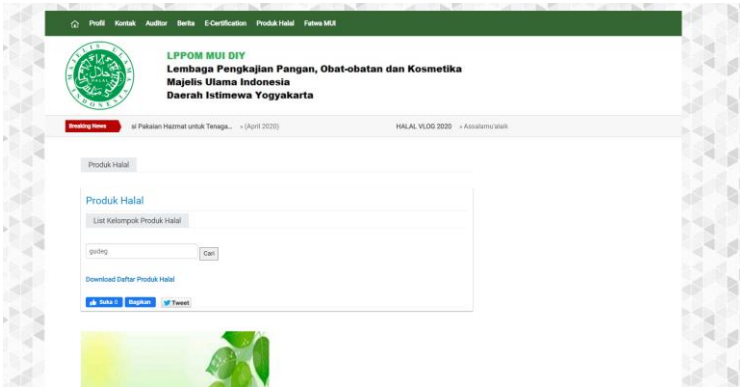
Kota Samarinda		
No.	Kelompok / Jenis Produk	Nama Perusahaan
1	Minuman Tradisional	Al Eshan Natural
2	Abon	Asslimy Gepuk Abon
3	Abon Ikan	Asslimy Gepuk Abon
4	Rumahan Potong Unggas	Ayam Sakti
5	Pengkilangan Daging dan Produk Olahahan Daging	Ayam Sakti
6	Minuman Tradisional	Bakul Jambu Mbak Bro
7	Nasi dan Lauk Pauk	Bitsani Cake
8	Bakery	Bitsani Cake
9	Snack Tradisional	Bunda Tini
10	Bakery	Cake & Bakery "Alidiah"
11	Kue Semi Basah	Citra Sari
12	Katering	CV. Andhika Katering
13	Produk Daging Olahahan	CV. Borneo Makmur Sejahtera
14	Produk Ikan Olahahan	CV. Borneo Makmur Sejahtera
15	Katering	CV. Jayatri
16	Katering	CV. Kirana Boga Caterindo
17	Bumbu Instan	CV. Ladang Indonesia
18	Air Minum Dalam Kemasan (AMOK)	CV. Mitra Sejati
19	Katering	CV. Tri Jaya
20	Restoran dan Katering	CV. Yen's Delight
21	Snack Tradisional	CV. ADRR Jaya Makmur
22	Sambal	CV. ADRR Jaya Makmur
23	Produk Ikan Olahahan	Dapoer Ikan Diana

Gambar 5. 8 Tampilan file pdf daftar produk halal LPPOM MUI Kaltim

Dalam file tersebut, dapat dilihat bahwa daftar produk yang dimiliki tidak melengkapi atribut-atribut yang diinginkan dari daftar produk halal, karena hanya memiliki nama perusahaan dan jenis produk.

f. LPPOM MUI Jogja

LPPOM MUI Yogya memiliki tampilan sebagai berikut:



Gambar 5. 9 Tampilan halaman depan website LPPOM MUI Jogja

Website lembaga ini memiliki fitur pencarian produk halal yang tidak aktif sehingga pengguna tidak dapat mencari produk halal secara langsung. Namun daftar prduk halal terdapat dalam file pdf yang dapat diunduh langsung oleh pengguna yang memiliki tampilan sebagai berikut:

DAFTAR PRODUK BERSERTIFIKAT HALAL
LEMBAGA PENGAJIAN PANGAN, OBAT, DAN KOSMETIKA (LPPOM)
MAJELIS ULAMA INDONESIA (MUI) DAERAH ISTIMEWA YOGYAKARTA
(JUNI 2020)

KELOMPOK DAGING (Termasuk Daging Olahhan, Seperti : Nugget, Bakso, DLI) [01]

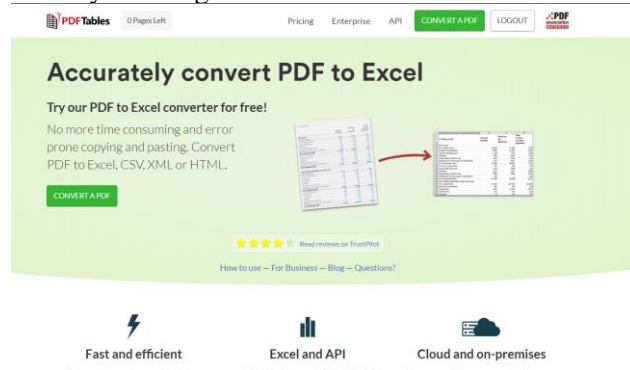
No.	Jenis Produk	Nama Produk	Nama Produsen	Alamat	Number Sertifikat	Expiry date
1.	Daging Olahhan	Daging Olahhan	PT. Ceman Adisatwa Unit Yogyakarta	Prambanan, Sleman	12010003690818	30 Agustus 2020
2.	Rambak Kulit	Rambak Kulit Saradi	Rambak Kulit Saradi	Pleret, Bantul	12010003840918	28 September 2020
3.	Rawon Daging Sapi Kaleng	Mamams	CV. Indonesia Kaya Rasa	Playen, Gunungkidul	12010003160718	31 Juli 2020
4.	Olahhan Daging	Galantim Ayam	UD. Daan Salam	Ngablak, Sleman	12010004801018	26 Oktober 2020
5.	Aneka Bakso Frozen.	Ramah Bakso	Ramah Bakso	Tirtomoko, Kasihan, Bantul	12010005901218	31 Desember 2020
6.	Nugget dan Sosis	Nugget dan Sosis.	PT. Dagsap Endura Entore	Selayu, Bantul, Yogyakarta	12010005761218	31 Desember 2020
7.	Daging Ayam	Daging Ayam "Selasih"	Perdagangan Daging Ayam "Selasih"	Ngupasan, Gondokusuman, Yogyakarta	12010003700818	30 Agustus 2020
8.	Daging Sapi	Depot Daging Sapi	UD. Panda	Kurahan, Bantul	12010004981118	21 November 2020

Gambar 5. 10 Tampilan daftar produk halal LPPOM MUI Jogja

Daftar produk halal yang dimiliki oleh LPPOM MUI Jogja cukup lengkap dan mencakup semua atribut yang diinginkan untuk digunakan lebih lanjut dalam tugas akhir ini.

1. Pengubahan file PDF ke CSV

Untuk menggunakan daftar produk halal ini lebih lanjut, file yang diperlukan haruslah bertipe JSON, namun untuk menjaga kerapian dan keutuhan dari data yang terdapat di dalam tabel, maka perlu diubah kedalam file CSV terlebih dahulu. Proses ini menggunakan *tools* berbasis web yaitu <https://pdftables.com/>. Tampilan dari website ini cukup sederhana yaitu sebagai berikut:



Gambar 5. 11 Tampilan website untuk mengubah file PDF ke CSV

Cara kerja dari website ini pun cukup sederhana, dimana pengguna cukup mengunggah file PDF yang diinginkan, lalu website ini akan menampilkan pratinjau dari tabel yang ingin diubah ke dalam file CSV. Kemudian file sudah dapat diunduh dalam format CSV. Tampilan dari pratinjau dan opsi unduhnya adalah sebagai berikut:

Daftar Produk Halal LPPOM MUI DIY terbaru.pdf How did we do? ★★★★★ DOWNLOAD AS EXCEL

Download as...
 Excel (single sheet)
 Excel (multiple sheets)
 CSV
 XML
 HTML
 Help
 Automate with our API

Page 1

DAFTAR PRODUK BERSERTIFIKAT HALAL
 LEMBAGA PENGKAJIAN PANGAN, OBAT, DAN KOSMETIKA (LPPOM)
 MAJELIS ULAMA INDONESIA (MUI) DAERAH ISTIMEWA YOGYAKARTA
 (Juni 2020)

KELOMPOK DAGING (Termasuk Daging Olahan, Seperti: Nugget, Bakso, Dli.) [01]

No.	Jenis Produk	Nama Produk	Produsen	Alamat	Nomer Sertifikat	Expire date
1.	Daging Olahan	Daging Olahan.	PT. Cionas Adisatwa Unit Yogyakarta	Prambanan, Sleman	12010003690818	30 Agustus 2020
2.	Rambak Kulit.	Rambak Kulit Suradi	Rambak Kulit Suradi	Pleret, Bantul	12010003840918	28 September 2020
3.	Rawon Daging Sapi Kaleng	Mammams	CV. Indonesia Kaya Raza	Playen, Gunungkidul	12010003100718	31 Juli 2020
4.	Olahan Daging	Galantin Ayam	UD. Daun Salam	Ngaglik, Sleman	12010004801018	26 Oktober 2020

Gambar 5. 12 Tampilan pratinjau dan opsi untuk mengunduh file dalam format yang diinginkan

Setelah didapatkan file dalam format CSV, maka data yang ada didalamnya dibersihkan dengan cara menghapus kolom yang tidak diperlukan hingga menyisakan kolom nama produk, nama produsen/perusahaan, nomor sertifikat, dan expire date. Perbedaan file CSV sebelum dan sesudah dibersihkan adalah sebagai berikut:

	A	B	C	D	E	F	G
1	No.	Jenis Produk	Nama Produk	Nama Produsen	Alamat	Nomer Sertifikat	Expire date
2	1.	Daging Olahan	Daging Olahan.	PT. Cionas Adisatwa	Prambanan, Sleman	12010003690818	30 Agustus 2020
3	2.	Rambak Kulit.	Rambak Kulit Suradi	Rambak Kulit Suradi	Pleret, Bantul	12010003840918	28 September 2020
4	3.	Rawon Daging Sap	Mammams	CV. Indonesia Kaya	Playen, Gunungkidul	12010003100718	31 Juli 2020
5	4.	Olahan Daging	Galantin Ayam	UD. Daun Salam	Ngaglik, Sleman	12010004801018	26 Oktober 2020
6	5.	Aneka Bakso Froze	Rumah Bakso	Rumah Bakso	Tirtonimolo, Kasihan, Bantul	12010005901218	31 Desember 2020
7	6.	Nugget dan Sosis	Nugget dan Sosis.	PT. Dagsap Endura	Sedayu, Bantul, Yogyakarta	12010005761218	31 Desember 2020
8	7.	Daging Ayam	Daging Ayam "Selasih"	Perdagangan Daging Ngupasan, Gondomanan, Yogyakarta		12010003700818	30 Agustus 2020

Gambar 5. 13 tabel daftar produk halal sebelum dibersihkan

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	nama_perusahaan,nama_produk,no_sertifikat,valid_date												
2	PT .Ciomas Adisatwa Unit Yogyakarta,Daging Olahhan,12010003690818,30 Agustus 2020												
3	Rambak Kulit Suradi,Rambak Kulit Suradi,12010003840918,28 September 2020												
4	CV. Indonesia Kaya Rasa,Mammams,12010003100718,31 Juli 2020												
5	UD. Daun Salam,Galantin Ayam,12010004801018,26 Oktober 2020												
6	Rumah Bakso,Rumah Bakso,12010005901218,31 Desember 2020												
7	PT. Dagsap Endura Eatore,Nuget, Sosis,12010005761218,31 Desember 2020												
8	Perdagangan Daging Ayam "Selasih",Daging Ayam "Selasih",12010003700818,30 Agustus 2020												
9	UD. Pandu,Depot Daging Sapi,12010004981118,21 November 2020												
10	Gandem Marem,Abon Gandem Marem,12010005451218,12 Desember 2020												
11	PT. Kepurun Pawana Indonesia,Olahan Daging,12010004761018,26 Oktober 2020												
12	Andrawinaloka Gudeg Bu Tjitro,Krecek Kaleng,12010000290119,28 Januari 2021												
13	Maxichic,Maxichic,12010005691218,31 Desember 2020												
14	El-As Chicken,Olahan Daging,12010005871218,31 Desember 2020												
15	Mirasa,Krecek Cap Koki, Rambak Cap Sapi,12010001240319,28 Maret 2021												
16	Eka Jaya,Eka Jaya,12010001280319,28 Maret 2021												

Gambar 5. 14 tabel daftar produk halal setelah dibersihkan

Dalam **Gambar 5. 13** dan **Gambar 5. 14** dapat dilihat bahwa kolom yang dihapus adalah kolom No., Jenis Produk, dan Alamat. Lalu kolom yang telah terbentuk dikembalikan ke dalam bentuk teks agar perubahan kedalam file JSON di tahap selanjutnya dapat dilakukan secara lebih mudah.

2. Pengubahan file CSV ke JSON

Setelah kolom dalam tabel sesuai dengan data yang diinginkan, maka file CSV diubah ke dalam file JSON agar dapat diproses lebih lanjut. Dalam penelitian tugas akhir ini untuk melakukan perubahan tipe file tersebut dilakukan dengan jupyter notebook yang memanfaatkan bahasa pemrograman python. Berikut kode program python yang ditulis untuk melakukan perubahan file tersebut:

```

1. import csv
2. import json
3.
4. file = 'Daftar Produk Halal LPPOM MUI DIY terbaru (version FIX
   ).csv'
5. json_file = 'itemsjogja.json'
6.
7. #Read CSV File
8. def read_CSV(file, json_file):
9.     csv_rows = []
10.    with open(file) as csvfile:
11.        reader = csv.DictReader(csvfile)
12.        field = reader.fieldnames
13.        for row in reader:
14.            csv_rows.extend([field[i]:row[field[i]] for i in
   range(len(field))])
15.        convert_write_json(csv_rows, json_file)
16.
17. #Convert csv data into json
18. def convert_write_json(data, json_file):
19.     with open(json_file, "w") as f:
20.         f.write(json.dumps(data, sort_keys=False, indent=4, se
   parators=(',', ': '))) #for pretty
21.
22.
23. read_CSV(file,json_file)

```

Kode 5. 18 Kode untuk mengubah data CSV ke dalam bentuk JSON

Kode 5.18 menunjukkan file CSV yang akan diubah ditetapkan diawal, lalu diikuti dengan nama hasil file JSON yang diinginkan. Kode bekerja dengan membaca dan menetapkan baris dan kolom dari tabel yang dimiliki oleh file CSV, lalu membuat sebuah file JSON dengan menggunakan baris dan kolom yang telah didapatkan dari proses membaca file CSV. File JSON yang dihasilkan akan memiliki tampilan sebagai berikut:

```

jupyter | itemsjogia.json | 10 hours ago
File Edit View Language JSON
1 [
2   {
3     "nama_perusahaan": "PT .Cinmas Adisatwa Unit Yogyakarta",
4     "nama_produk": "Daging Olah",
5     "no_sertifikat": "12010003690818",
6     "valid_date": "30 Agustus 2020"
7   },
8   {
9     "nama_perusahaan": "Rambak Kulit Suradi",
10    "nama_produk": "Rambak Kulit Suradi",
11    "no_sertifikat": "12010003840918",
12    "valid_date": "28 September 2020"
13  },
14  {
15    "nama_perusahaan": "CV. Indonesia Kaya Rasa",
16    "nama_produk": "Gummies",
17    "no_sertifikat": "12010003100718",
18    "valid_date": "31 Juli 2020"
19  },
20  {
21    "nama_perusahaan": "Ud. Daun Salam",
22    "nama_produk": "Galantin Ayam",
23    "no_sertifikat": "12010004801018",
24    "valid_date": "26 Oktober 2020"
25  },
26  ],
27 ]

```

Gambar 5. 15 Tampilan hasil file JSON

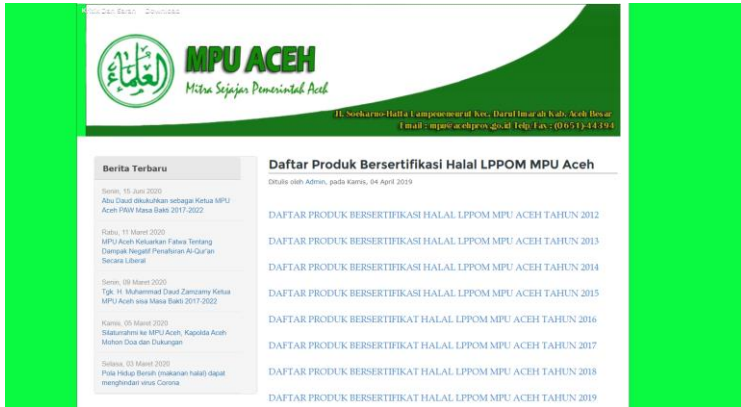
Gambar 5. 15 menjelaskan file JSON yang dihasilkan berisikan 1109 data yang sudah sesuai dengan format yang diinginkan dan dapat digunakan lebih lanjut dalam tahap selanjutnya.

g. LPPOM MUI Aceh

Tampilan website dan daftar produk halal lembaga ini adalah sebagai berikut:



Gambar 5. 16 tampilan halaman depan website MPU ACEH



Gambar 5. 17 tampilan daftar tautan untuk mengunduh daftar produk halal

Dalam **Gambar 5. 17** menunjukkan website MPU Aceh menyediakan daftar produk halalnya dalam bentuk tautan-tautan file berbentuk PDF yang dapat diunduh.

1. Pengubahan file PDF ke CSV

Proses pengubahan file PDF ke dalam CSV yang dilakukan pada lembaga ini sama dengan yang digunakan pada lembaga sebelumnya, yaitu dengan menggunakan *tools* berbasis web <https://pdftables.com/>. Tampilan dari pratinjau file PDF MPU Aceh adalah sebagai berikut:

The screenshot shows the PDFTables website interface. At the top, there are navigation links for Pricing, Enterprise, API, and a 'CONVERT API' button. Below the navigation, there is a search bar with the text 'Daftar_Sertifikat_Halal_2019_Kelompok_Industri_Pengolahan.pdf' and a 'DOWNLOAD AS EXCEL' button. The main content area displays a table with 10 columns: No, Jenis Produk, Group Code, Type Code, Nama Produk, Merek, No. Sertifikat Halal, ValidDate Mulai Berlaku Masa Berakhir, Nama Perusahaan, and Alamat Perusahaan. The table contains three rows of data.

No	Jenis Produk	Group Code	Type Code	Nama Produk	Merek	No. Sertifikat Halal	ValidDate Mulai Berlaku Masa Berakhir	Nama Perusahaan	Alamat Perusahaan
1	Madu		12 01	Uhoee 58	Uhoee 58	14120000010119	2019-01-21 2022-01-20	Uhoee 58	Jl. Singgah Mata No. 58 Desa Sukaramil Kec. Baiturrahman Kota Banda Aceh
2	Air Minum Isi Ulang		12 01	ARRA	ARRA	14120000020119	2019-01-21 2022-01-20	ARRA Water	Jl. SA Johansyah Lt. Mutiara No. 5 Gp. Neusu Aceh Kec. Baiturrahman Kota Banda Aceh
3	Cincau		19 01	Cincau Hitam	Cincau Hitam	14190000010119	2019-01-21 2022-01-20	UD. Oils Cincau	Jl. Teledan I Lt. Gunung Steniko Gp. Ganuf

Gambar 5. 18 Tampilan pratinjau file PDF daftar produk halal sebelum diunduh dalam bentuk CSV

Setelah didapatkan file dalam format CSV, maka data yang ada didalamnya dibersihkan dengan cara menghapus kolom yang tidak diperlukan hingga menyisakan kolom nama produk, nama produsen/perusahaan, merek, nomor sertifikat, dan expire date. Berikut tampilan file CSV daftar produk halal MPU Aceh yang sudah dibersihkan:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	merek,nama_perusahaan,nama_produk,no_sertifikat,valid_date												
2	Unoe 58,Unoe 58,Madu,14120000010119,2022-01-20												
3	ARRA,ARRA Water,Air Minum Isi Ulang,14120000020119,2022-01-20												
4	Cincau Hitam,UD. Olis Cincau,Cincau,14190000010119,2022-01-20												
5	UKA Crispy,Usaha Keluarga Alami,Stik Ketala,14100000010119,2022-01-20												
6	Kak Neng,Mita Raseuki,Peyek Kacang, Kue Bawang,14100000020119,2022-01-20												
7	Kuades,CV. Tirta Nagan Lestari,AMDK,14120000030119,2022-01-20												
8	Hijrah,Hijrah,Aneka Kue,14200000010119,2022-01-20												
9	Rita Cake & Cookies,Rita Cake & Cookies,Aneka Kue,14200000020119,2022-01-20												
10	Natural Food,Natural Aceh,Kerupuk, Nugget Tiram,14100000030119,2022-01-20												
11	Garindo Coffee,CV. Garindo,Bubuk Kopi Arabika, Bubuk Kopi Robusta,14120000040219,2022-02-10												
12	Garindo Coffee,CV. Garindo,Biji Kopi Arabika, Biji Kopi Robusta,14190000020219,2022-02-10												
13	Tjeudah,Tjeudah,Keripik Pisang, Keripik Singkong,14100000040219,2022-02-10												
14	Keripik Balado Rizki,Keripik Balado Rizki,Keripik Singkong Balado,14100000050419,2022-04-01												
15	Rengginang Kak Ros,Rengginang Kak Ros,Regginang,14100000060419,2022-04-01												
16	Rengginang Tamija,Rengginang Tamija,Regginang,14100000070419,2022-04-01												
17	Regginang Ulee Kareng,Regginang Ulee Kareng,Regginang,14100000080419,2022-04-01												

Gambar 5. 19 Daftar produk halal MPU Aceh setelah dibersihkan

Gambar 5.19 menunjukkan isi dari daftar produk halal yang sudah dibersihkan dan berisikan atribut yang sesuai dengan data yang diinginkan.

2. Pengubahan file CSV ke JSON

File CSV yang sudah berisikan data yang sesuai diubah ke dalam bentuk JSON dengan menggunakan *tools* yang sama dengan lembaga sebelumnya, yaitu menggunakan jupyter notebook notebook yang memanfaatkan bahasa pemrograman python. Berikut kode program python yang ditulis untuk melakukan perubahan file tersebut:

```

1. import csv
2. import json
3.
4. file = 'Daftar_Sertifikat_Halal_2019_Kelompok_Industri_Peng
   olahan_fix.csv'
5. json_file = 'itemsaceh.json'
6.
7. #Read CSV File
8. def read_CSV(file, json_file):
9.     csv_rows = []
10.    with open(file) as csvfile:
11.        reader = csv.DictReader(csvfile)
12.        field = reader.fieldnames
13.        for row in reader:
14.            csv_rows.extend([field[i]:row[field[i]] for i
   in range(len(field))])
15.    convert_write_json(csv_rows, json_file)
16.
17. #Convert csv data into json
18. def convert_write_json(data, json_file):
19.    with open(json_file, "w") as f:
20.        f.write(json.dumps(data, sort_keys=False, indent=4,
   separators=(',', ' : '))) #for pretty
21.
22.
23.
24. read_CSV(file,json_file)

```

Kode 5. 19 Kode Python untuk Mengubah File CSV menjadi JSON

Kode 5.19 menunjukkan file CSV yang akan diubah ditetapkan diawal, lalu diikuti dengan nama hasil file JSON yang diinginkan. Kode bekerja dengan membaca dan menetapkan baris dan kolom dari tabel yang dimiliki oleh file CSV, lalu membuat sebuah file JSON dengan menggunakan baris dan kolom yang telah didapatkan dari proses membaca file CSV. File JSON yang dihasilkan akan memiliki tampilan sebagai berikut:



```

1  [
2  {
3      "merek": "Unoe 50",
4      "nama_perusahaan": "Unoe 50",
5      "nama_produk": "Mudu",
6      "no_sertifikat": "I4120000010119",
7      "valid_date": "2022-01-20"
8  },
9  {
10     "merek": "ARBA",
11     "nama_perusahaan": "ARBA Water",
12     "nama_produk": "Air Minum Isi Ulang",
13     "no_sertifikat": "I4120000020119",
14     "valid_date": "2022-01-20"
15  },
16  {
17     "merek": "Cincou Hitam",
18     "nama_perusahaan": "UD, Olis Cincou",
19     "nama_produk": "Cincou",
20     "no_sertifikat": "I4100000010119",
21     "valid_date": "2022-01-20"
22  },
23  {
24     "merek": "UKA Crispy",
25     "nama_perusahaan": "Usaha Keluarga Alami",
26     "nama_produk": "Stik Katala",
27     "no_sertifikat": "I4100000010119",
28     "valid_date": "2022-01-20"
29  },
30 ]

```

Gambar 5. 20 Tampilan hasil file JSON

Gambar 5. 1920 menunjukkan tampilan file JSON yang dihasilkan yang berisi 165 data yang dapat digunakan untuk tahap selanjutnya dalam pengerjaan tugas akhir.

h. KlikIndomaret

1. Menulis Kode *Scraper*

i. Penulisan tautan

Pada bagian ini cukup menuliskan tautan yang akan digunakan untuk pengambilan data lalu dituliskan pada bagian awal kode scraper, apabila terdapat tautan yang berbeda-beda untuk beberapa produk maka dapat langsung ditambahkan setelah penulisan tautan pertama dimana harus dipisahkan dengan tanda koma. Apabila terdapat sangat banyak tautan seperti `next_page` maka dalam Scrapy dapat memanfaatkan fitur pagination pada tautan yang di ambil datanya. Berikut penulisan tautan pada kode scraper KlikIndomaret:

```

1. import scrapy
2. from ..items import IndomaretItem
3. from urllib.parse import urljoin
4.
5. class crawlindomaret(scrapy.Spider):
6.     name = 'indomaret'
7.     page_number = 2
8.     start_urls = [
9.         'https://www.klikindomaret.com/category/cemi
lan--
biskuit?categories=&sortcol=PROMO&page=1&pagesi
ze=50&attributes=&productbrandid=&startprice=&e
ndprice='
10.     ]
11.     base_url = 'https://www.klikindomaret.com/'

```

Kode 5. 20 penulisan tautan klikindomaret

Pada Error! Reference source not found.20 tulisan pada objek start_urls merupakan penulisan tautan website yang akan di lakukan pengambilan data.

i. Penulisan *selector*

```

1. def parse(self, response):
2.     indomaret = response.xpath("//*[contains(@class, 'item')]/
a/@href").extract()
3.     for i in indomaret:
4.         url = urljoin(response.url, i)
5.         yield scrapy.Request(url, callback=self.parse_item)
6.
7.     def parse_item(self, response):
8.         for info in response.css('div.produk produk-detail'):
9.             item = IndomaretItem()
10.
11.             nama_produk = info.css('h3.desktop-
xs::text').extract_first(),
12.             komposisi = info.css('span.spec_desc spec_id_KOMPO
SISI').extract_first(),
13.             daily_value = info.css('span.spec_desc spec_id_TAKAR
AN PER SERVING').extract_first(),
14.
15.             item['nama_produk'] = nama_produk
16.             item['komposisi'] = komposisi
17.             item['daily_value'] = daily_value
18.
19.             yield item

```

Kode 5. 2181 pembuatan selector dan item

Kode 5. 21821 diatas menjelaskan cara iterasi untuk melakukan

pengambilan data yang telah di pilah-pilah sesuai dengan posisinya pada tag html. Serta dibuatkan item agar data yang di dapat hasilnya terstruktur dengan baik.

```

1. import scrapy
2.
3. class IndomaretItem(scrapy.Item):
4.     # define the fields for your item here like
5.     :
6.     # name = scrapy.Field()
7.     nama_produk = scrapy.Field()
8.     komposisi = scrapy.Field()
9.     daily_value = scrapy.Field()
10.    pass

```

Kode 5. 192 penulisan item field pada file item

Kode 5. 192 merupakan data file item yang ditulis untuk mendefinisikan item field. Penulisan item field sesuai dengan kode crawl yang di buat pada file scraper/crawl.

ii. Tahap *pagination*

Pada kasus website dari klikindomaret data pada tabel ternyata tidak ditampilkan langsung dalam satu halaman, melainkan terbagi ke dalam 9 halaman yang dimana tiap halamannya hanya menampilkan 50 data. Sehingga pada tugas akhir ini dilakukan perulangan untuk tautan klikindomaret karena setelah dilihat tautannya hanya menambah angka di setiap pagenanya. Jadi:

```

1. next_page = 'https://www.klikindomaret.com/category/cemilan--
   biskuit?categories=&sortcol=PROMO&page=' + str(crawlindomaret
   .page_number) + '&pagesize=50&attributes=&productbrandid=&sta
   rtprice=&endprice='
2.     if crawlindomaret.page_number < 3:
3.         crawlindomaret.page_number += 1
4.     yield response.follow(next_page, callback = self.parse)

```

Kode 5. 203 penulisan kode untuk halaman berikutnya

Pada **Kode 5. 203** page_number sebelumnya telah di tulis pada **Kode 5. 21** sehingga, langsung dapat digunakan. Selain itu disini hanya dilakukan perulangan lalu di lakukan callback agar tautan utamanya muncul lalu ditambah hingga page ke 2.

3. Melakukan Web Scraping

Melakukan Web Scraping berarti menjalankan kode program yang telah ditulis sebelumnya. *Command* untuk menjalankan scraper yang telah dibuat adalah sebagai berikut:

```
scrapy crawl indomaret -o items.json
```

Kode 5. 214 Command untuk menjalankan scraper pada KlikIndomaret

Dari **Kode 5. 214**, terdapat perintah `-o` yang berarti menetapkan format dari *output* yang dihasilkan oleh scraper. Pada tugas akhir ini kode program dijalankan secara terpisah karena struktur, elemen html untuk setiap website berbeda-beda. Setelah kode program untuk klikindomaret dijalankan terdapat 100 data yang berhasil di scraping. Seluruh data tersebut nantinya akan dikumpulkan menjadi satu setelah seluruh kode program selesai dijalankan.

5.1.2 Menggabungkan data menjadi satu

Setelah seluruh kode web scraper dijalankan selanjutnya seluruh hasil dari implementasi kode web scraper merupakan file json. File json tersebut berisi seluruh data produk yang ada pada website LPPOM MUI Regional dan klikindomaret. Setelah menggabungkan seluruh data dari tujuh website yang telah dijelaskan diatas maka, terdapat 4.245 data yang berhasil dikumpulkan dan data tersebut selanjutnya akan memasuki tahap pembersihan.

5.1.3 Membersihkan data

Pembersihan data penelitian ini dilakukan untuk membersihkan dan memisahkan nama produk yang terkumpul dalam satu baris dan menyeragamkan format *valid date*. Data berformat JSON dibaca sebagai sebuah dataframe menggunakan *jupyter notebook* yang memanfaatkan bahasa pemrograman *python*. Berikut tahap pembersihan data yang dilakukan:

a. Pemisahan nama produk sebaris

Pada data produk hasil scraping, banyak nama produk yang berasal dari satu merek atau perusahaan yang disimpan dalam satu baris. Hal ini mengakibatkan jumlah data produk yang telah discraping tidak menggambarkan jumlah data produk sebenarnya dari semua website daftar produk halal yang telah dilakukan scrape. Berikut kode program *python* yang ditulis untuk pemisahan nama produk berdasarkan koma:

```

1. namaproduk = []
2. index = []
3.
4. for i in items_all.index:
5.     namabaru = items_all.nama_produk[i].split(",")
6.     for j in namabaru:
7.         namaproduk.append(j.lstrip(' ').rstrip('.'))
8.         index.append(i)
9.     namabaru = []
10.
11. df = items_all.reset_index()
12. df2 = df.merge(df2,how='left',
13.                on='index').drop(columns=['index'])

```

Kode 5. 225 Kode untuk memisahkan nama produk

Dari **Kode 5. 225** dapat dilihat bahwa untuk memisahkan, diperlukan sebuah kolom baru pada baris pertama untuk menampung nama produk yang sudah terpisah, dan sebuah kolom index pada baris ke-2 yang berperan dalam menetapkan urutan dari nama produk yang sudah terpisah. Setelah itu kolom “nama_produk” di split dengan parameter “,” pada baris ke-5. Dalam proses ini juga dapat sekaligus membersihkan karakter pengganggu yang mengawali data pada “nama_produk”. Berikut hasil implementasi dari **Kode 5. 225**:

1. Data sebelum dipisahkan menggunakan python:

	merek	nama_perusahaan	nama_produk	no_sertifikat	valid_date
0	Unoe 58	Unoe 58	Madu	14120000010119	2022-01-20
1	ARRA	ARRA Water	Air Minum Isi Ulang	14120000020119	2022-01-20
2	Cincau Hitam	UD. Olis Cincau	Cincau	14190000010119	2022-01-20
3	UKA Crispy	Usaha Keluarga Alami	Stik Ketala	14100000010119	2022-01-20
4	Kak Neng	Mita Raseuki	Peyek Kacang, Kue Bawang	14100000020119	2022-01-20
...
4140	Dofell	Dofell	Roti dan Kue (Bakery)	04 20 003110 0320	03 Mar 2022
4141	Aden	Aden	Pemanis	04 23 003111 0320	03 Mar 2022
4142	Najah Alkhair	CV. Najah Alkhair	Katering	04 33 002404 1017	03 Mar 2022
4143	Moro Joyo Catering	CV. Trisno Kenang Biru	Katering	04 33 003109 0320	03 Mar 2022
4144	Katering RF	PT. Rezky Sukses Mandiri	Katering	04 33 003113 0320	03 Mar 2022

Gambar 5. 201 data sebelum nama produk dipisahkan

2. Data setelah diimplementasikan kode pemisah yang telah dijelaskan diatas.

	namaproduk	merek	nama_perusahaan	nama_produk	no_sertifikat	valid_date
10848	Roti dan Kue (Bakery)	Dofell	Dofell	Roti dan Kue (Bakery)	04 20 003110 0320	03 Mar 2022
10849	Pemanis	Aden	Aden	Pemanis	04 23 003111 0320	03 Mar 2022
10850	Katering	Najah Alkhair	CV. Najah Alkhair	Katering	04 33 002404 1017	03 Mar 2022
10851	Katering	Moro Joyo Catering	CV. Trisno Kenang Biru	Katering	04 33 003109 0320	03 Mar 2022
10852	Katering	Katering RF	PT. Rezky Sukses Mandiri	Katering	04 33 003113 0320	03 Mar 2022

Gambar 5. 211 data setelah nama produk dipisahkan

Setelah menjadi data nama produk yang terpisah, jumlah keseluruhan data meningkat drastic, dari data yang berjumlah 4.245, menjadi berjumlah 10.953. Namun, jumlah data ini belumlah valid karena masih ada data yang tidak valid karena pemisahan dengan koma juga memisahkan nama produk yang dibatasi koma untuk varian ukurannya.

b. Penghapusan nama produk tidak valid

Hal ini dilakukan agar data yang memiliki nama produk dengan hanya angka dihilangkan dan membuat data menjadi efektif. Untuk penghapusan nama produk tidak valid ini juga menggunakan jupyter notebook. Berikut kode program

yang

ditulis:

```

1. is_digit = []
2. for i in df3.namaproduk:
3.     if i == "":
4.         is_digit.append('Started with number')
5.     elif i[0].isdigit():
6.         is_digit.append('Started with number')
7.     else:
8.         is_digit.append('Not Started with number'
)

```

Kode 5. 236 data setelah nama produk dipisahkan

Dari **Kode 5. 236**, “df3” pada baris ke-4 merupakan dataframe dari data nama produk yang sudah dipisahkan pada tahap sebelumnya. Kode ini bekerja dengan menandai setiap baris dengan nama produk yang berawalan karakter spasi atau berupa angka, kemudian meletakkan nilai dari tanda tersebut kedalam sebuah kolom baru bernama “is_digit”. Setelah data memiliki tanda bahwa data tersebut valid atau tidak, maka data dengan tanda bahwa data tersebut diawali dengan angka atau tidak valid dapat dibuang. Setelah dibuang, jumlah data berkurang menjadi sejumlah 10.649 data.

c. Penyeragaman format tanggal

Penyeragaman dilakukan menggunakan *library dateparser* python. Berikut kode untuk proses penyeragaman format tanggal:


```

1. import dateparser
2.
3. date = []
4. for i in itemsall.index:
5.     date.append(dateparser.parse(
6.         str(itemsall.valid_date[i]))
7.
8. itemsall['date']=date
9.
10. valid_date = []
11. for i in itemsall2.index:
12.     valid_date.append(itemsall2.date[i].strftime(
13.         '%d-%m-%Y'))
13. itemsall2['valid_date'] = valid_date
14. itemsall2 = itemsall2.drop(columns=['date'])

```

Kode 5. 247 Kode python untuk menetapkan nilai tanggal

Dari **Kode 5. 247** menunjukkan kerja dari *library dateparser* yang dapat mengenali berbagai bentuk format tanggal dan menyeragamkannya kedalam sebuah format yang ditentukan. Berikut hasil implementasi **Kode 5. 247**:

	id_makanan	nama_produk	merek	nama_perusahaan	no_sertifikat	valid_date
0	1	Madu	Unoe 58	Unoe 58	14120000010119	2022-01-20
1	2	Air Minum Isi Ulang	ARRA	ARRA Water	14120000020119	2022-01-20
2	3	Cincau	Cincau Hitam	UD. Olis Cincau	14190000010119	2022-01-20
3	4	Stik Ketala	UKA Crispy	Usaha Keluarga Alami	14100000010119	2022-01-20
4	5	Peyek Kacang	Kak Neng	Mita Raseuki	14100000020119	2022-01-20
...
10544	10545	Roti dan Kue (Bakery)	Dofell	Dofell	04 20 003110 0320	03 Mar 2022
10545	10546	Pemanis	Aden	Aden	04 23 003111 0320	03 Mar 2022
10546	10547	Katering	Najah Alkhair	CV. Najah Alkhair	04 33 002404 1017	03 Mar 2022
10547	10548	Katering	Moro Joyo Catering	CV. Trisno Kenang Biru	04 33 003109 0320	03 Mar 2022
10548	10549	Katering	Katering RF	PT. Rezky Sukses Mandiri	04 33 003113 0320	03 Mar 2022

Gambar 5. 22 Data dengan format tanggal yang belum seragam

	id_makanan	nama_produk	merek	nama_perusahaan	no_sertifikat	valid_date
0	1	Madu	Unoe 58	Unoe 58	14120000010119	20-01-2022
1	2	Air Minum Isi Ulang	ARRA	ARRA Water	14120000020119	20-01-2022
2	3	Cincau	Cincau Hitam	UD. Olis Cincau	14190000010119	20-01-2022
3	4	Stik Ketala	UKA Crispy	Usaha Keluarga Alami	14100000010119	20-01-2022
4	5	Peyek Kacang	Kak Neng	Mita Raseuki	14100000020119	20-01-2022
...
10544	10545	Roti dan Kue (Bakery)	Dofell	Dofell	04 20 003110 0320	03-03-2022
10545	10546	Pemanis	Aden	Aden	04 23 003111 0320	03-03-2022
10546	10547	Katering	Najah Alkhair	CV. Najah Alkhair	04 33 002404 1017	03-03-2022
10547	10548	Katering	Moro Joyo Catering	CV. Trisno Kenang Biru	04 33 003109 0320	03-03-2022
10548	10549	Katering	Katering RF	PT. Rezky Sukses Mandiri	04 33 003113 0320	03-03-2022

Gambar 5. 23 Data dengan format tanggal yang sudah seragam

Dari **Kode 5. 192** dan **Kode 5. 203** menunjukkan perubahan format tanggal pada kolom `valid_date` dimana format diubah menjadi `dd-mm-yyyy`. Kemudian data yang sudah sesuai ditulis Kembali dalam bentuk JSON.

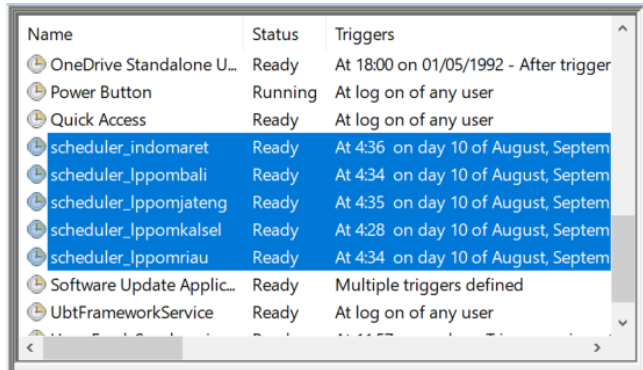
5.2 Scheduling

Scheduling dilakukan dengan menggunakan Windows Task Scheduler yang dapat menjalankan task baik secara script maupun tidak. Pada penelitian ini proses web scraping akan dilakukan secara terjadwal tiap bulan sehingga data dari daftar produk halal dapat teregenerasi tiap bulan. Berikut perintah dalam batch file dari salah satu proses web scraping:

```
@echo off
Call "C:\Users\acer\Environments\ta_env\Scripts\activate.bat"
cd "C:\Users\acer\Environments\lppomkalsel\"
call scrapy crawl lppomkalsel -o output.json
deactivate
```

Gambar 5. 24 Perintah dalam batch file untuk menjalankan web scraper

Pada **Gambar 5. 24** menunjukkan perintah di dalam batch file untuk melakukan scheduling proses web scraping dari website LPPOM MUI Kalsel. Semua batch file dijadwalkan dengan task scheduler sehingga proses web scraping dapat berjalan secara rutin secara otomatis. Daftar scheduler yang dibuat di dalam Task Scheduler tertera dalam **Gambar 5. 25**



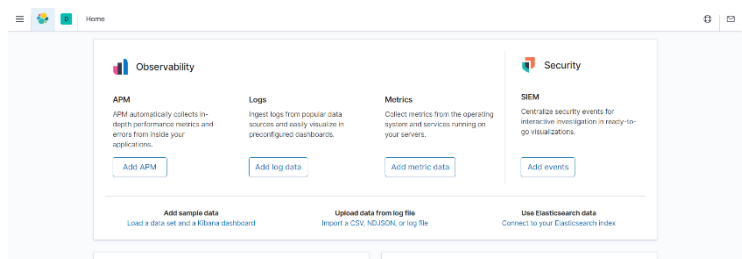
Name	Status	Triggers
OneDrive Standalone U...	Ready	At 18:00 on 01/05/1992 - After trigger
Power Button	Running	At log on of any user
Quick Access	Ready	At log on of any user
scheduler_indomaret	Ready	At 4:36 on day 10 of August, Septem
scheduler_ippombali	Ready	At 4:34 on day 10 of August, Septem
scheduler_ippomjateng	Ready	At 4:35 on day 10 of August, Septem
scheduler_ippomkalsel	Ready	At 4:28 on day 10 of August, Septem
scheduler_ippomriau	Ready	At 4:34 on day 10 of August, Septem
Software Update Applic..	Ready	Multiple triggers defined
UbtFrameworkService	Ready	At log on of any user

Gambar 5. 25 Scheduler yang sudah dibuuh pada Task Scheduler

5.3 Data Ingestion

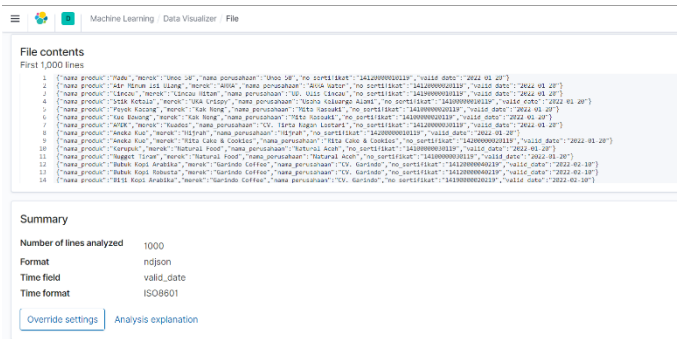
5.3.1 Membuat Index

Pada tugas akhir ini digunakan tiga buah index untuk tiga kasus pengujian. Untuk mengakses *elasticsearch*, pada penelitian ini menggunakan aplikasi bernama *Kibana*. Aplikasi ini bekerja sebagai *client* untuk melakukan pembuatan dan perubahan data pada *elasticsearch*. Berikut tampilan awal *kibana* untuk mengunggah data:

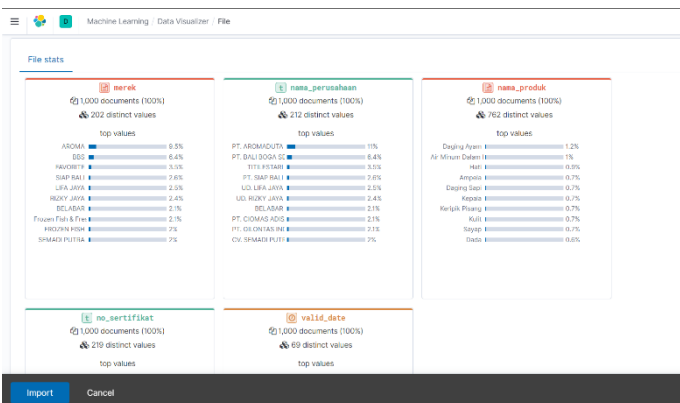


Gambar 5. 26 Tampilan awal Kibana

Pada **Gambar 5. 26** menunjukkan tampilan awal dari kibana memiliki menu “Upload data from log file”, yang akan digunakan untuk mengunggah data produk halal kedalam sebuah index *elasticsearch*. Setelah terunggah, kibana akan menunjukkan pratinjau dari *file*. Berikut tampilan pratinjau yang dihasilkan oleh kibana terhadap file yang diunggah:

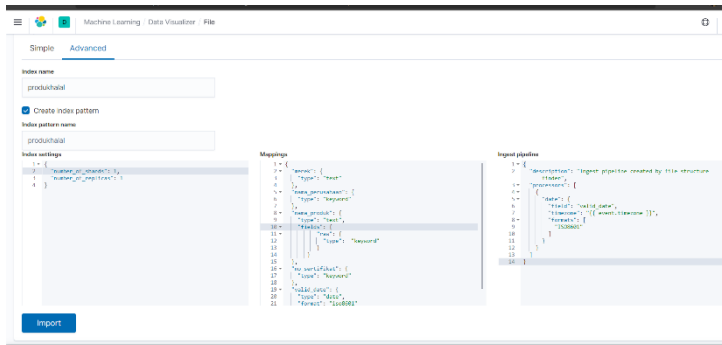


Gambar 5. 27 Tampilan pratinjau isi dari file yang diunggah



Gambar 5. 28 Tampilan statistik data dari tiap kolom file yang diunggah

Setelah data diimpor, maka diperlukan pengaturan dan *mapping* dari data yang terdapat dalam file. Hal ini diperlukan untuk menetapkan tipe data dari masing-masing field yang akan dibuat. Berikut tampilan pengaturan:



Gambar 5. 30 Tampilan pengaturan awal

Dari **Gambar 5. 30** menunjukkan *field* “index settings” yang berfungsi sebagai tempat pengaturan jumlah replika dan jumlah *shards* yang akan digunakan dan *analyzer* yang akan digunakan. Kemudian pada *field* “Mappings” menetapkan tipe data dari tiap *field* pada *index*. Sedangkan Pada *field* “Ingest Pipeline” berfungsi untuk menjalankan *processor* atau hal-hal yang akan di *generate* secara otomatis oleh elasticsearch. Setelah semua pengaturan sesuai, maka data di impor dan index dibuat yang dalam tugas akhir ini bernama “produkhalal_0analyzer”, “produkhalal_1analyzer”, dan “produkhalal_2analyzer”.

5.3.2 Mapping

Mapping menggambarkan bagaimana gambaran data yang akan disimpan didalam elasticsearch ketika membuat index. Dalam Mappings terdapat *field-field* yang akan digunakan dan tipe data dari field tersebut. Untuk menetapkan mappings, kode dijalankan pada fitur *console* yang disediakan oleh kibana untuk

melakukan kustomisasi terhadap *index* yang dibuat. Berikut tampilan kode untuk tahap *mapping*:

```
{
  "merek": {
    "type": "text",
    "fields": {
      "raw": {
        "type": "keyword"
      }
    }
  },
  "nama_perusahaan": {
    "type": "text"
  },
  "nama_produk": {
    "type": "text",
    "fields": {
      "raw": {
        "type": "keyword"
      }
    }
  },
  "no_sertifikat": {
    "type": "keyword"
  },
  "valid_date": {
    "type": "date",
    "format": "iso8601",
    "ignore_malformed": true
  },
  "daily_value": {
    "type": "text"
  },
  "komposisi": {
    "type": "text"
  }
}
```

Kode 5. 258 Kode untuk membuat mapping

Pada **Kode 5. 258** menunjukkan *mapping* dari field yang dibuat dan berisikan sebagai berikut:

1. merek

Field merek memiliki tipe multi-fields yaitu membuat sebuah field memiliki lebih dari satu tipe data. Tipe data text membuat field merek dapat dianalisa dengan dipecah kedalam bagian individu sehingga memungkinkan pencocokan parsial. Sedangkan tipe data keyword memungkinkan field merek untuk di index secara utuh. Hal ini diperlukan jika data akan dikelompokkan dalam sebuah aggregation atau melalui proses sorting.

2. nama_perusahaan

Field nama_perusahaan memiliki tipe data text untuk memungkinkan proses analisa dengan dipecah kedalam bagian individu sehingga memungkinkan pencocokan parsial.

3. nama_produk

Field nama_produk memiliki tipe multi-fields yaitu membuat sebuah field memiliki lebih dari satu tipe data. Tipe data text membuat field merek dapat dianalisa dengan dipecah kedalam bagian individu sehingga memungkinkan pencocokan parsial. Sedangkan tipe data keyword memungkinkan field merek untuk di index secara utuh. Hal ini diperlukan jika data akan dikelompokkan dalam sebuah aggregation atau melalui proses sorting

4. no_sertifikat

Field no_sertifikat memiliki tipe data *keyword* untuk menyimpan data secara utuh. *Field* ini tidak menggunakan tipe data *numeric* untuk menghindari *error* ketika menemukan nilai yang memiliki format yang berbeda.

5. `valid_date`

Field `valid_date` menggunakan tipe data `date` yang didukung dengan format `iso8601` pada `elasticsearch`. Pada field ini diberikan parameter tambahan yaitu `ignore_malformed` yang berfungsi untuk melewati data yang memiliki format berbeda.

6. `daily_value`

Field `daily_value` memiliki tipe data `text` untuk memungkinkan proses analisa dengan dipecah kedalam bagian individu sehingga memungkinkan pencocokan parsial.

7. `komposisi`

Field `komposisi` memiliki tipe data `text` untuk memungkinkan proses analisa dengan dipecah kedalam bagian individu sehingga memungkinkan pencocokan parsial.

Pada index yang menggunakan `search analyzer` yang akan dijelaskan pada bagian selanjutnya, field yang bertipe data `text` diberikan parameter tambahan yaitu “`analyzer`” dan “`search_analyzer`” yang berfungsi untuk membedakan proses analisa pada saat indexing dan pencarian.

5.3.3 *Analyzer*

Index yang dibuat dalam tugas akhir ini memiliki *analyzer* yang berbeda-beda. Berikut *analyzer* yang digunakan dalam masing-masing index:

a. *Language analyzer*

Analyzer ini bekerja dengan menganalisa tiap data pada saat proses indexing menjadi kedalam berbagai *token* atau pemisah perkata yang akan membantu proses pencarian. *Analyzer* termasuk dalam kode *settings* dalam pembuatan sebuah *index*. *Analyzer* yang digunakan adalah *Language analyzer*, yang bekerja dengan menganalisa teks berdasarkan suatu bahasa. Dengan menggunakan *analyzer* ini, teks berbahasa Indonesia yang digunakan pada penelitian ini dapat dianalisa

secara tepat. Untuk menjalankan *analyzer* ini, kode dijalankan pada fitur *console* yang disediakan oleh kibana untuk melakukan kustomisasi terhadap *index* yang dibuat. Berikut kode untuk menjalankan *language analyzer*:

```

"analysis": {
  "filter": {
    "indonesian_stop": {
      "type": "stop",
      "stopwords": "_indonesian_"
    },
    "indonesian_keywords": {
      "type": "keyword_maker",
      "keywords": ["dan"]
    },
    "indonesian_stemmer": {
      "type": "stemmer",
      "language": "indonesian"
    }
  },
  "analyzer": {
    "rebuilt_indonesian": {
      "tokenizer": "standard",
      "filter": [
        "lowercase",
        "indonesian_stop",
        "indonesian_keywords",
        "indonesian_stemmer"
      ]
    }
  }
}

```

Kode 5. 269 Kode untuk membuat language analyzer

Pada **Kode 5. 269** menunjukkan dua filter yang akan membantu kerja dari *analyzer* yang digunakan adalah *stopwords* dan *stemmer*. Filter *Indonesian_stop* menyaring kata yang tergolong dalam jenis *stopwords* dalam bahasa Indonesia. Filter “*indonesian_keyword*” memberikan kata spesifik yang tidak akan diikuti dalam proses stemming. Pada tugas akhir ini kata yang

dikeluarkan dari proses *stemming* adalah “dan”. *Filter* *indonesian_stemmer* memecah teks kedalam potongan-potongan kata berbahasa Indonesia.

b. Autocomplete analyzer

Analyzer ini bekerja dengan memanfaatkan tokenizer *edge_ngram* saat melakukan indexing. *Edge_ngram* bekerja dengan memecah teks menjadi kata-kata setiap kali bertemu salah satu dari daftar karakter yang ditentukan. Berikut query dari autocomplete analyzer:

```

"analysis": {
  "filter": {
    "autocomplete_filter": {
      "type": "edge_ngram",
      "min_gram": 1,
      "max_gram": 20
    }
  },
  "analyzer": {
    "autocomplete": {
      "type": "custom",
      "tokenizer": "standard",
      "filter": [
        "lowercase",
        "autocomplete_filter"
      ]
    }
  }
}

```

Kode 5. 30 Query untuk membuat autocomplete analyzer

Secara default *elasticsearch* menggunakan analyzer yang sama untuk proses indexing dan pencarian. Namun dengan menggunakan analyzer ini proses indexing dan pencarian dapat memiliki analyzer yang berbeda. Hal ini dilakukan dengan cara menambahkan parameter “*search_analyzer*” pada mapping field yang bertipe *text*. Value dari parameter ini diisi dengan “*standard*”. Ini

membuat saat proses indexing elasticsearch menggunakan autocomplete analyzer, lalu menggunakan standard analyzer pada saat melakukan pencarian.

5.4 Pengujian Pencarian

Proses pengujian pencarian dilakukan terhadap tiga buah index yang memiliki analyzer yang berbeda dengan query yang berbeda. Query ini berdasarkan pada statistik jumlah *value* dan kata yang terdapat pada field “nama_produk”. Berikut lima belas *value* dan kata terbanyak yang terdapat dalam field nama_produk:

Tabel 5.1 Lima belas kata terbanyak dalam field nama_produk

Kata	Jumlah
Ayam	435 docs
Roti	355 docs
Kue	351 docs
Ikan	286 docs
Daging	194 docs
Keripik	192 docs
Kerupuk	152 docs
Sambal	100 docs
Bawang	96 docs
Udang	79 docs
Tahu	76 docs
Tempe	58 docs
Tepung	44 docs
Cokelat	44 docs
Kentang	42 docs

Dari Tabel 5.1, kata Daging yang menjadi kata terbanyak ke-5 akan menjadi input yang digunakan untuk menjadi query pencarian. Sedangkan untuk kata terbanyak pada field “komposisi” adalah sebagai berikut:

Tabel 5.2 Sepuluh kata terbanyak dalam field komposisi

Kata	Jumlah
Garam	79 docs
Gula	67 docs
Tepung	62 docs
Terigu	51 docs
Nabati	50 docs
Perisa	41 docs
Susu	32 docs
Kedelai	29 docs
Pewarna	26 docs
Natrium	23 docs

Pada Tabel 5.2, kata “tepung” yang menjadi kata terbanyak ke-3 pada field “komposisi” akan digunakan sebagai query pencarian. Kata “tepung” dipilih karena field “nama_produk” sendiri memiliki 44 dokumen yang memiliki value “tepung” dan merupakan kata terbanyak ke-13, sedangkan field “komposisi” memiliki 62 dokumen yang memiliki value “tepung”, sehingga proses pencarian akan menguji value dari field manakah yang lebih dahulu ditampilkan. Pencarian dilakukan pada fitur *console* yang dimiliki oleh kibana, sedangkan *query* yang digunakan adalah *query DSL* yang bertipe JSON. Berikut *query* yang digunakan untuk melakukan pengujian pencarian pada field yang bertipe text dan keyword:

5.4.1 Full Text Query

Full text query memungkinkan untuk mencari field text yang dianalisis. Query yang dipakai dari full text query adalah Match Query yang mengembalikan dokumen yang cocok dengan teks, angka, tanggal atau nilai boolean yang disediakan. Teks yang disediakan dianalisis sebelum dicocokkan. Ada dua input query yang dimasukkan kedalam match query.

a. Input query “daging ren”

Penggunaan input query ini berdasar pada value “daging” yang menjadi salah satu value terbanyak dari field nama_produk dengan jumlah 204 dokumen. Untuk input query “ren” merupakan potongan dari kata “rendang” yang menjadi value dari field nama_produk dengan jumlah 23 dokumen. Sedangkan dokumen yang mengandung kedua kata tersebut pada index berjumlah 2 dokumen. Kombinasi dua value diatas menjadi query yang akan dicari dengan harapan diproses oleh analyzer untuk mendapatkan hasil pencarian yang menampilkan dua dokumen dengan value mengandung kata “daging” dan “rendang”. Berikut query untuk menjalankan match query:

```
{
  "query": {
    "multi_match": {
      "query": "daging_ren",
      "fields": [ "nama_produk",
                 "nama_perusahaan",
                 "komposisi", "daily_value" ]
    }
  }
}
```

Kode 5. 31 Query untuk melakukan pencarian dengan input “daging ren”

Kode 5. 31 menunjukan input yang diberikan pada query adalah “daging ren”. Query dijalankan pada semua field yang memiliki tipe data text.

b. Input query “tepung”

Penggunaan input query ini bertujuan untuk menemukan dokumen pada index yang memiliki value “tepung” baik pada field “nama_produk”, maupun “komposisi”. Field “nama_produk” sendiri memiliki 44 dokumen yang memiliki value “tepung”, sedangkan field “komposisi” memiliki

62 dokumen yang memiliki value “tepung”. Berikut query untuk menemukan value “tepung”:

```

{
  "query": {
    "multi_match": {
      "query": "tepung",
      "fields": [
        "nama_produk", "komposisi"
      ]
    }
  }
}

```

Kode 5. 32 Query untuk melakukan pencarian dengan input query "tepung"

Kode 5. 32 menunjukkan input yang diberikan pada query adalah “tepung”. Query dijalankan pada field “nama_produk” dan “komposisi” yang memiliki tipe data text.

Dua input query tersebut akan dijalankan pada tiga buah index yang memiliki analyzer berbeda.

1. Index tanpa analyzer

Index ini dibuat tanpa menetapkan analyzer yang apa yang akan digunakan didalamnya.

2. Index dengan Language Analyzer

Index ini dibuat dengan menggunakan language Analyzer yang sudah dijelaskan pada sub bab Analyzer.

3. Index dengan Autocomplete Analyzer dan Search Analyzer

Index ini dibuat dengan menggunakan dua buah analyzer yang bekerja pada secara terpisah saat melakukan indexing dan pencarian.

5.4.2 Term-level Query

Term-level query digunakan untuk menemukan dokumen berdasarkan nilai yang tepat dalam data terstruktur. Tidak

seperti pada full text query, term-level query tidak menganalisis istilah pencarian. Term-level query mencocokkan dengan istilah persis yang disimpan dalam field. Term-level query yang digunakan adalah fuzzy query. *Fuzzy query* mengembalikan dokumen yang berisi istilah yang mirip dengan istilah pencarian. Input query yang dimasukkan untuk query ini adalah “dsginh”, yaitu bentuk penulisan yang salah dari value “daging”. Dengan input query ini diharapkan pencarian dapat menghasilkan dokumen dengan nama_produk “daging”. Berikut tampilan query dari *fuzzy query*:

```
{
  "query": {
    "fuzzy": {
      "nama_produk": {
        "value": "dsginh"
      }
    }
  }
}
```

Kode 5. 273 Query untuk menampilkan hasil pencarian menggunakan Fuzzy query

Pada **Kode 5. 273** pengujian dilakukan dengan melakukan pencarian pada field “nama_produk” yang memiliki *typo* dari data yang berisi “daging” menjadi “dsginh”. Fuzzy query dilakukan pada tiga index berikut.

- a. Index tanpa analyzer
Index ini dibuat tanpa menetapkan analyzer yang apa yang akan digunakan didalamnya.
- b. Index dengan Language Analyzer
Index ini dibuat dengan menggunakan language Analyzer yang sudah dijelaskan pada sub bab Analyzer.
- c. Index dengan Autocomplete Analyzer dan Search Analyzer

Index Index ini dibuat dengan menggunakan dua buah analyzer yang bekerja pada secara terpisah saat melakukan indexing dan pencarian.

5.5 Integrasi Data

Penggabungan data yang sudah didapat kedalam Halal Nutrition Food dilakukan setelah penelitian.

BAB VI HASIL DAN PEMBAHASAN

Pada bab ini berisi penjelasan dari hasil serta pembahasan yang telah dihasilkan dari bab implementasi yaitu pengujian pencarian pada elasticsearch.

6.1 Hasil Query Pencarian

Berikut merupakan analisis hasil query dari pengujian yang dilakukan terhadap tiga index dengan analyzer yang berbeda.

6.1.1 Full Text Query

a. Input query “daging ren”

1. Index tanpa analyzer

Dengan melakukan pencarian dengan input query “daging ren” pada index yang tidak memiliki pengaturan analyzer, elasticsearch melakukan pencarian dengan menggunakan analyzer default atau standard. Hasil pencarian tertera pada **Tabel 6. 1**

Tabel 6. 1 Hasil pencarian pada index tanpa analyzer

nama_produk	merek	nama_perusahaan	_score
"Ayam Utuh (Whole)"	"SARI DAGING"	"UD. SARI DAGING"	6.5323553
"Fillet Dada"	"SARI DAGING"	"UD. SARI DAGING"	6.5323553
"Fillet Paha"	"SARI DAGING"	"UD. SARI DAGING"	6.5323553

Hasil yang tertera pada **Tabel 6. 1** merupakan 3 dokumen pertama yang dihasilkan dari proses pencarian. Dari tabel tersebut menunjukkan 3 data teratas memiliki nilai kolom score yang sama. Hal ini disebabkan kesamaan value pada kolom “merek” dan “nama_perusahaan”. Pada pencarian ini elasticsearch menemukan relevansi tinggi antara input “daging ren”

dengan value “SARI DAGING” dan “UD. SARI DAGING” pada kolom “merek” dan “nama_perusahaan”. Selain itu, tidak ditemukan value yang mengandung istilah yang sama dengan input “daging ren” pada kolom “nama_produk”.

2. Index dengan Language Analyzer

Dengan melakukan pencarian dengan input query “daging ren” pada index dengan language analyzer, elasticsearch menggunakan language analyzer berbasis bahasa Indonesia. Hasil pencarian tertera pada **Tabel 6. 2**

Tabel 6. 2 Hasil query pada index dengan language analyzer

nama_produk	merek	nama_perusahaan	_score
"Ayam Utuh (Whole)"	"SARI DAGING	"UD. SARI DAGING"	6.5323553
"Fillet Dada"	"SARI DAGING	"UD. SARI DAGING"	6.5323553
"Fillet Paha"	"SARI DAGING	"UD. SARI DAGING"	6.5323553

Hasil yang tertera pada **Tabel 6. 2** merupakan 3 dokumen pertama yang dihasilkan dari proses pencarian. Dari tabel tersebut menunjukkan 3 data teratas memiliki nilai kolom score yang sama. Hal ini dapat disebabkan kesamaan value pada kolom “merek” dan “nama_perusahaan”. Pada pencarian ini elasticsearch menemukan relevansi tinggi antara input “daging ren” dengan value “SARI DAGING” dan “UD. SARI DAGING” pada kolom “merek” dan “nama_perusahaan”. Selain itu, tidak ditemukan value yang mengandung istilah yang sama dengan input “daging ren” pada kolom “nama_produk”.

3. Index dengan Autocomplete Analyzer dan Search analyzer

Dengan melakukan pencarian dengan input query “daging ren” pada index yang memiliki autocomplete analyzer pada saat indexing dan search analyzer pada saat pencarian, elasticsearch dapat menghasilkan pencarian sebagaimana tertera pada **Tabel 6. 3**

Tabel 6. 3 Hasil pencarian pada index dengan autocomplete analyzer dan search analyzer

nama_produk	merek	nama_perusahaan	_score
"Rendang Daging Ayam (Chicken Rendang)"	"PRONA S"	"PT. CANNING INDONESIAN PRODUCTS"	14.637571
"Rendang Daging"	"Lhee Sagoe Food"	"Lhee Sagoe Food"	14.546517
"Regginang"	"Rengginang Tamija"	"Rengginang Tamija"	10.558743

Hasil yang tertera pada **Tabel 6. 3** merupakan 3 dokumen pertama yang dihasilkan dari proses pencarian. Nilai pada kolom “_score” pada pencarian ini lebih tinggi dengan jarak yang signifikan bila dibandingkan dengan dua index sebelumnya. Pada hasil pencarian ini, value pada kolom nama_produk mengandung input “daging ren”, namun dengan urutan yang tidak menentu. Sedangkan pada kolom “merek” dan “nama_perusahaan” tidak ditemukan kesesuaian antara input “daging ren” dengan value yang ada didalam kedua kolom tersebut.

b. Input query “tepung”

1. Index tanpa analyzer

Dengan melakukan pencarian dengan input query “tepung” pada index yang tidak memiliki pengaturan analyzer, elasticsearch melakukan pencarian dengan menggunakan analyzer default atau standard. Hasil pencarian tertera pada **Tabel 6. 4**

Tabel 6. 4 Hasil Pencarian pada Index tanpa Analyzer

nama_produk	mer ek	nama_perusahaa n	kom posi si	_score
"Tepung"	0	"Progress Jogja"	0	7.4525557
"Tepung"	0	"Kelompok Wanita Tani Sejahtera"	0	7.4525557
"Tepung"	0	"KWT. Rosella"	0	7.4525557

Hasil yang tertera pada **Tabel 6. 4** merupakan 3 dokumen pertama yang dihasilkan dari proses pencarian. Dari tabel tersebut menunjukkan 3 data teratas memiliki nilai kolom score yang sama. Hal ini disebabkan kesamaan value pada kolom "Tepung". Pada pencarian ini elasticsearch menemukan relevansi tinggi antara input "tepung" dengan value "Tepung" pada kolom nama_produk. Namun, tidak ditemukan value yang mengandung istilah yang sama dengan input "tepung" pada kolom "komposisi" dari 3 dokumen pertama hasil pencarian.

2. Index dengan Language Analyzer

Dengan melakukan pencarian dengan input query "tepung" pada index dengan language analyzer, elasticsearch menggunakan language analyzer berbasis bahasa Indonesia. Hasil pencarian tertera pada **Tabel 6. 5**

Tabel 6. 5 Hasil query pada index dengan language analyzer

nama_produk	mer ek	nama_perusahaa n	kom posi si	_score
"Tepung"	0	"Progress Jogja"	0	7.4525557
"Tepung"	0	"Kelompok Wanita Tani Sejahtera"	0	7.4525557
"Tepung"	0	"KWT. Rosella"	0	7.4525557

Hasil yang tertera pada **Tabel 6. 5** merupakan 3 dokumen pertama yang dihasilkan dari proses pencarian. Dari tabel tersebut menunjukkan 3 data teratas memiliki nilai kolom score yang sama. Hal ini disebabkan kesamaan value pada kolom “Tepung”. Pada pencarian ini elasticsearch menemukan relevansi tinggi antara input “tepung” dengan value “Tepung” pada kolom nama_produk. Namun, tidak ditemukan value yang mengandung istilah yang sama dengan input “tepung” pada kolom “komposisi” dari 3 dokumen pertama hasil pencarian. Hasil ini serupa dengan hasil pencarian pada index tanpa analyzer.

3. Index dengan Autocomplete Analyzer dan Search analyzer

Dengan melakukan pencarian dengan input query “tepung” pada index yang memiliki autocomplete analyzer pada saat indexing dan search analyzer pada saat pencarian, elasticsearch dapat menghasilkan pencarian sebagaimana tertera pada **Tabel 6. 6**

Tabel 6. 6 Hasil pencarian pada index dengan autocomplete analyzer dan search analyzer

nama_produk	merk	nama_perusahaan	komposisi	_score
"Tepung"	0	"Progress Jogja"	0	8.862365
"Tepung"	0	"Kelompok Wanita Tani Sejahtera"	0	8.862365
"Tepung"	0	"KWT. Rosella"	0	8.862365

Hasil yang tertera pada **Tabel 6. 6** merupakan tiga dokumen pertama yang dihasilkan dari proses pencarian. Hasil pencarian memiliki value pada kolom “nama_produk” yang sama dengan input pencarian. Sedangkan untuk tiga hasil teratas, kolom “komposisi” tidak memiliki value. Hasil pencarian ini sama dengan

hasil pencarian pada dua index sebelumnya, tetapi terdapat perbedaan pada kolom “_score” dimana value pada pencarian ini lebih besar dibanding dua index sebelumnya, walaupun value hasil pencarian pada kolom lain sama .

6.1.2 Term-level Query

a. Index tanpa analyzer

Dengan melakukan pencarian dengan input query “dsginh” pada index yang tidak memiliki pengaturan analyzer, elasticsearch melakukan pencarian dengan menggunakan analyzer default atau standard. Hasil pencarian tertera pada **Tabel 6. 7**

Tabel 6. 7 Hasil pencarian pada index tanpa analyzer

nama_produk	merek	nama_perusahaan	_score
"Daging"	0	"RPH Kalirejo Dinas Pertanian dan Pangan Kabupaten kebumen "	3.585105
"Daging")	"ZEN"	"ZEN "	3.585105
"Sate Daging"	"Minang Tanjung"	"Minang Tanjung"	2.9974358

Hasil yang tertera pada **Tabel 6. 7** merupakan tiga dokumen pertama yang dihasilkan dari proses pencarian. Kolom “_score” pada hasil pertama dan kedua memiliki nilai yang sama. Hal ini dapat disebabkan oleh value pada kolom “nama_produk” yang mengeluarkan hasil pencarian berupa “daging” dimana hasil tersebut merupakan hasil yang diharapkan dari input query “dsginh”. Sedangkan pada kolom “merek” dan “nama_perusahaan” tidak ada yang mengandung istilah dari input.

b. Index dengan Language Analyzer

Dengan melakukan pencarian dengan input query “dsginh” pada index dengan language analyzer,

elasticsearch menggunakan language analyzer berbasis bahasa Indonesia. Hasil pencarian tertera pada **Tabel 6.8**

Tabel 6.8 Hasil pencarian pada index dengan language analyzer

nama_produk	merek	nama_perusahaan	_score
"Daging"	0	"RPH Kalirejo Dinas Pertanian dan Pangan Kabupaten kebumen "	3.585105
"Daging)"	"ZEN"	"ZEN "	3.585105
"Sate Daging"	"Minang Tanjung"	"Minang Tanjung"	2.9974358

Hasil yang tertera pada **Tabel 6.8** merupakan tiga dokumen pertama yang dihasilkan dari proses pencarian. Kolom “_score” pada hasil pertama dan kedua memiliki nilai yang sama. Hal ini dapat disebabkan oleh value pada kolom “nama_produk” yang mengeluarkan hasil pencarian berupa “daging” dimana hasil tersebut merupakan hasil yang diharapkan dari input query “dsginh”. Sedangkan pada kolom “merek” dan “nama_perusahaan” tidak ada yang mengandung istilah yang mirip dengan input.

- c. Index dengan Autocomplete Analyzer dan Search analyzer

Dengan melakukan pencarian dengan input query “dsginh” pada index yang memiliki autocomplete analyzer pada saat indexing dan search analyzer pada saat pencarian, elasticsearch dapat menghasilkan pencarian sebagaimana tertera pada **Tabel 6.9**

Tabel 6.9 Hasil query pada index dengan autocomplete analyzer dan search analyzer

nama_produk	merek	nama_perusahaan	_score
-------------	-------	-----------------	--------

"Daging dan Produk Daging Olahan"	"Jawi Indah"	"Jawi Indah"	8.476187
"Daging dan Produk Daging Olahan"	"De Bakso"	"De Bakso "	8.476187
"Daging dan Produk Daging Olahan"	"Monda"	"Mondag"	8.476187

Hasil pencarian yang ditunjukkan oleh **Tabel 6. 9** menunjukkan 3 hasil teratas output pencarian berdasarkan nilai kolom “_score” tertinggi. Untuk hasil pencarian teratas pada kolom “nama_produk” menghasilkan value sama untuk beberapa baris kebawah disebabkan data pada index dengan value yang sama. Hal ini menunjukkan bahwa elasticsearch menemukan relevansi tinggi antara input “dsginh” dengan value “Daging dan Produk Daging Olahan”. Dengan begitu kolom “_score” yang dihasilkan pun akan serupa untuk hasil pencarian dengan value yang sama.

Untuk tabel hasil pencarian yang lebih lengkap dapat diakses pada tautan berikut dengan nama file hasil query pencarian.xlsx

<https://github.com/rifqyprtm/Tugas-akhir>

6.2 Pembahasan

6.2.1 Full text query

Dari hasil yang diperoleh saat melakukan query pencarian, menunjukkan pencarian full text query dengan input “daging ren” dapat menemukan relevansi yang lebih tinggi pada index dengan autocomplete_analyzer dan search analyzer. Keseluruhan bagian input juga ditemukan dalam pencarian menggunakan index dengan autocomplete analyzer dan search analyzer. Sedangkan pada dua index lainnya tidak semua bagian

input terkandung dalam hasil pencarian. Berikut tabel perbandingan input dengan hasil pencarian pada proses pencarian menggunakan 3 index tersebut:

Tabel 6. 10 Perbandingan input dengan hasil pencarian

Input	Index	Field		
		nama_produk	Hasil pertama	Hasil Kedua
"daging ren"	Index tanpa analyzer	nama_perusahaan	"UD. SARI DAGING"	"UD. SARI DAGING"
		merek	"SARI DAGING"	"SARI DAGING"
		merek	"SARI DAGING"	"SARI DAGING"
		merek	"Ayam Uruh (Whole)"	"Fillet Dada"
		merek	"UD. SARI DAGING"	"UD. SARI DAGING"
		merek	"UD. SARI DAGING"	"UD. SARI DAGING"
	Index dengan Language Analyzer	nama_perusahaan	"UD. SARI DAGING"	"UD. SARI DAGING"
		merek	"SARI DAGING"	"SARI DAGING"
		merek	"SARI DAGING"	"SARI DAGING"
		merek	"Ayam Uruh (Whole)"	"Fillet Dada"
		merek	"UD. SARI DAGING"	"UD. SARI DAGING"
		merek	"UD. SARI DAGING"	"UD. SARI DAGING"
Index dengan autocomplete analyzer dan search analyzer	nama_perusahaan	PT. INDONESIAN CANNING PRODUCTS	Lhee Sago Food	Kenginang Tamija
	merek	PRONAS	Lhee Sago Food	Kenginang Tamija
	merek	Rendang Ayam (Chicken Rendang)	Rendang Daging	Rengginang
	merek	Rendang Ayam	Rendang Daging	Rengginang
	merek	PRONAS	Lhee Sago Food	Kenginang Tamija
	merek	PRONAS	Lhee Sago Food	Kenginang Tamija

Pada **Tabel 6. 10** menunjukkan keberadaan bagian-bagian kata input pada hasil pencarian. Index dengan autocomplete analyzer dan search analyzer dapat menampilkan hasil pencarian dengan dua bagian yang terdapat pada input terkandung dalam value hasil pencarian. Sedangkan pada dua index lainnya hasil pencarian hanya mampu mencocokkan satu bagian dari input ketika menampilkan hasil pencarian. Hasil ini menunjukkan bahwa index yang mampu menghasilkan dokumen yang sesuai dengan harapan hanyalah index dengan autocomplete analyzer dan search analyzer. Hal ini berkaitan dengan tokenizer yang digunakan pada analyzer ini yaitu edge_ngram yang memecah value saat indexing menjadi bagian per karakter. Hal ini memungkinkan bagian potongan kata seperti “ren” pada input “daging ren” dicocokkan dengan potongan kata “ren” yang ada pada index. Kemudian, untuk pencarian dengan input query “tepung”, hasil teratas yang dikeluarkan saat melakukan pencarian dapat menghasilkan dokumen yang sesuai dengan input. Seluruh hasil pencarian memiliki value “tepung” pada field “nama_produk”. Berikut tabel perbandingan input dengan hasil pencarian pada proses pencarian menggunakan 3 index tersebut:

Tabel 6. 11 Perbandingan hasil pencarian menggunakan input query "tepung"

Input			Index			Field		
"tepung"			Index dengan Language Analyzer			nama_produk		
						nama_perusahaan		
			Index dengan autocomplete analyzer dan search analyzer			komposisi		
			nama_produk			nama_perusahaan		
			komposisi			komposisi		

Hasil pertama	Hasil Kedua	Hasil Ketiga
"Tepung"	"Tepung"	"Tepung"
"Progress Jogja"	"Kelompok Wanita Tani Sejahtera"	" KWT. Rosella"
0	0	0
"Tepung"	"Tepung"	"Tepung"
"Progress Jogja"	"Kelompok Wanita Tani Sejahtera"	" KWT. Rosella"
0	0	0
"Tepung"	"Tepung"	"Tepung"
"Progress Jogja"	"Kelompok Wanita Tani Sejahtera"	" KWT. Rosella"
0	0	0

Tabel 6. 11 menunjukkan bahwa pada pencarian menggunakan input query “tepung”, seluruh index dapat menghasilkan hasil pencarian yang sama. Hasil ini juga menunjukkan bahwa pencarian dokumen dilakukan pada field “nama_produk” terlebih dahulu. Barulah hasil pencarian field “komposisi” yang juga memiliki value “tepung” di tampilkan setelah pencarian pada field “nama_produk” selesai.

6.2.2 Term-level query

Pada pencarian menggunakan term-level query juga menemukan relevansi yang lebih tinggi pada index dengan autocomplete analyzer dan search analyzer. Hal ini ditunjukkan dengan hasil pencarian dengan input “dsginh” yang diharapkan, yaitu “daging”, ditemukan dalam semua hasil pencarian menggunakan index manapun. Berikut tabel perbandingan input dengan hasil pencarian pada proses pencarian menggunakan 3 index tersebut:

Tabel 6. 12 Perbandingan input dengan hasil pencarian

Input	Index	Field	Hasil pertama	Hasil Kedua	Hasil Ketiga
"dsginh"	Index dengan Language Analyzer	nama_perusahaan	"Daging"	"Daging j"	"Sate Daging"
		nama_produk	"RPH Kalirejo Dinas Pertanian dan Pangan Kabupaten Kebumen "	"ZEN"	"Minang Tanjung"
		merk	0	"ZEN"	"Minang Tanjung"
	Index dengan autocomplete analyzer dan search analyzer	nama_perusahaan	"Daging dan Produk Daging Olahan"	"Daging dan Produk Daging Olahan"	"Daging dan Produk Daging Olahan"
		nama_produk	"Jawi Indah"	"De Bakso"	"Monda"
		merk	"Jawi Indah"	"De Bakso"	"Monda"
	nama_perusahaan	"Jawi Indah"	"De Bakso"	"Monda"	

Tabel 6. 12 menunjukkan hasil pencarian menggunakan input "dsginh" dapat menampilkan hasil yang diharapkan yaitu "daging" pada field "nama_produk". Perbedaan yang terdapat

dari pencarian menggunakan ketiga index berada pada hasil pencarian dari index tanpa analyzer dan index dengan language analyzer yang mencocokkan input dengan value yang benar-benar sama pada index terlebih dahulu. Sedangkan untuk index dengan autocomplete analyzer dan search analyzer mencocokkan input dengan value pertama yang mengandung string input tersebut pada index. Hal ini disebabkan pada index dengan autocomplete analyzer menggunakan tokenizer yang memecah value pada index menjadi token tersendiri dengan proses pemecahan dilakukan per karakter.

Halaman ini sengaja dikosongkan.

BAB VII KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan yang didapat dari seluruh proses pengerjaan tugas akhir dan saran untuk penelitian kedepannya untuk dapat dikembangkan dari tugas akhir ini.

7.1. Kesimpulan

Kesimpulan yang telah didapatkan dari hasil penelitian tugas akhir ini adalah:

1. Total data yang diperoleh dari hasil *web scraping* dari berbagai website LPPOM MUI Daerah ditambah dengan website klikindomaret yaitu sebanyak 10.649 data produk.
2. Penggunaan *autocomplete analyzer* dan *search analyzer* pada *index* Elasticsearch dengan dokumen berisi data hasil *scraping* bekerja paling baik dalam menemukan relevansi antara kueri pencarian yang memiliki potongan kata dengan dokumen dalam *index*. Hal ini disebabkan oleh proses analisis teks yang dilakukan dengan memecah kata dalam indeks menjadi bagian-bagian per karakter membuat kata dapat dicocokkan sesuai dengan bagian-bagian per karakter kata yang diinput saat proses pencarian.
3. *Index* tanpa *analyzer* dan *index* yang hanya menggunakan *Language analyzer* dapat menghasilkan pencarian yang lebih tepat dalam menemukan kecocokkan antara kueri pencarian yang memiliki kesalahan penulisan dengan dokumen dalam *index*. Hal ini disebabkan proses analisis teks yang dilakukan terhadap kata yang memiliki kesalahan penulisan dicocokkan secara utuh terhadap kata yang ada dalam indeks, sehingga bentuk kata yang memiliki susunan dan jumlah huruf dalam kata paling mirip akan dihasilkan sebagai hasil pencarian yang paling relevan.

7.2. Saran

Beberapa saran yang dapat diberikan untuk penelitian selanjutnya antara lain:

1. Penelitian ini dapat dikembangkan dengan melakukan perbandingan terhadap performa sistem pencarian lain seperti Apache Lucene dan Algolia.
2. Melakukan integrasi sistem pencarian dengan database Halal Nutrition Food

DAFTAR PUSTAKA

- [1] A. Indrawan, "Inilah 10 Negara dengan Populasi Muslim Terbesar di Dunia," 2015. [Online]. Available: REPUBLICA.CO.ID.
- [2] L. MUI, "Tentang LPPOM MUI," 2019. [Online]. Available: <https://www.halalmui.org/mui14/main/page/tentang-lppom-mui>.
- [3] Yunita, "Melawan Hoax," *Kominfo*, 16 01 2017. [Online]. Available: https://kominfo.go.id/index.php/content/detail/8904/melawan-hoax/0/sorotan_media. [Accessed 10 November 2019].
- [4] N. A. Rakhmawati, J. Fatawi, A. C. Najib and A. A. Firmansyah, "Linked open data for halal food products," *Journal of King Saud University – Computer and Information Sciences*, 2019.
- [5] B. Fajar Yusuf, "Integrasi Data dan Visualisasi Graf pada Aplikasi Halal Nutrition Food," *Sistem Informasi ITS*, Surabaya, 2018.
- [6] Alkautsar, "Integrasi Data Lembaga Sertifikasi Halal di Dunia dengan Data Halal Nutrition Food," *Sistem Informasi ITS*, Surabaya, 2019.
- [7] Z. Shi, M. Shi and W. Lin, "The Implementation of Crawling News Page Based," *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence*, pp. 348-351, 2016.
- [8] D. Martinsson, "Optimization of a Scheduler for a Web," *Chalmers University of Technology*, Gothenburg, 2019.
- [9] A. P. Atmaja and S. V. Yulianto, "Pemanfaatan Elasticsearch untuk Temu Kembali Informasi Tugas

- Akhir," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 04, pp. 160-167, 2018.
- [10] F. Lestari N, D. Diniaty, Mawardi and Syamsurizal, "PENGUKURAN KINERJA MANAJEMEN RANTAI PASOK PADA SERTIFIKASI PRODUK HALAL MENGGUNAKAN SCOR MODEL (Studi Kasus LPPOM MUI Riau)," UIN SULTAN SYARIF KASIM RIAU, Pekanbaru, 2018.
- [11] Indomaret, "Cemilan-Biskuit," [Online]. Available: <https://www.klikindomaret.com/category/cemilan--biskuit>. [Accessed 1 December 2019].
- [12] A. Cholakian, "All About Analyzers," 22 January 2014. [Online]. Available: <https://www.elastic.co/blog/found-text-analysis-part-1#the-components-of-an-analyzer>. [Accessed 24 November 2019].
- [13] L. C. Gunther, E. Colangelo, H.-H. Wiendahl and C. Bauer, "Data Quality Assessment for Improved Decision-making: A Methodology for Small and Medium-sized Enterprises," *Procedia Manufacturing*, pp. 583-591, 2019.
- [14] D. Panta, "WEB CRAWLING AND SCRAPING - developing a sale-based website," TURKU UNIVERSITY OF APPLIED SCIENCES, 2015.
- [15] "Scrapy 1.6 documentation — Scrapy 1.6.0 documentation.," [Online]. Available: <https://docs.scrapy.org/en/latest/>. [Accessed 10 November 2019].

BIODATA PENULIS



Penulis lahir di Bogor pada tanggal 14 September 1998. Penulis merupakan anak kedua dari 4 bersaudara. Penulis telah menempuh pendidikan formal, yaitu: SD Islam Al-azhar 20 Cibubur, SMP Negeri 49 Jakarta, dan SMA Negeri 8 Jakarta.

Pada tahun 2016, penulis melanjutkan pendidikan ke jenjang S1 melalui jalur PKM di Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut

Teknologi Sepuluh Nopember (ITS) Surabaya. Selama menjadi mahasiswa, penulis aktif mengikuti beberapa kegiatan kemahasiswaan, antara lain sebagai pengurus HMS pada periode 2017/2018 dan 2018/2019 di Departemen Internal Affair (IA) serta menjadi Staff Explore ITS pada ITS EXPO tahun 2017. Selain itu, penulis juga aktif di bidang minat bakat di unit kegiatan mahasiswa Flag Football.

Pada tahun ke-4 perkuliahan, penulis memiliki ketertarikan dalam mempelajari pengolahan data, *data science*, *data engineering* dan mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui email: rifqybarliansyah@gmail.com