



UNIVERSITAS DIPONEGORO

**SISTEM *CRAWLING* DATA INSTRUMEN AKREDITASI
BERBASIS SELENIUM DAN PANDAS**

TUGAS AKHIR

LAILA LATHIFAH

21060116130112

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK ELEKTRO
PROGRAM STUDI SARJANA**

**SEMARANG
DESEMBER 2020**



UNIVERSITAS DIPONEGORO

**SISTEM *CRAWLING* DATA INSTRUMEN AKREDITASI
BERBASIS SELENIUM DAN PANDAS**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

LAILA LATHIFAH

21060116130112

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK ELEKTRO
PROGRAM STUDI SARJANA**

**SEMARANG
DESEMBER 2020**

HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun yang dirujuk
telah saya nyatakan dengan benar.**

NAMA : Laila Lathifah

NIM : 21060116130112

Tanda Tangan :

Tanggal : 22 Desember 2020

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

NAMA : LAILA LATHIFAH
NIM : 21060116130112
Departemen/Program Studi : TEKNIK ELEKTRO / SARJANA (S1)
Judul Skripsi : SISTEM *CRAWLING* DATA INSTRUMEN
AKREDITASI BERBASIS SELENIUM DAN
PANDAS

Telah berhasil dipertahankan di hadapan Tim Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Sarjana, Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro.

TIM PENGUJI

Pembimbing I : Eko Handoyo, S.T., M.T. (.....)
Pembimbing II : Yosua Alvin Adi Soetrisno, ST., M.Eng. (.....)
Penguji I : (.....)
Penguji II : (.....)

Semarang, 16 Desember 2020
Ketua Departemen Teknik Elektro,

Dr. Wahyudi, S.T., M.T.
NIP. 196906121994031001

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Diponegoro, saya yang bertanda tangan di bawah ini :

Nama : LAILA LATHIFAH
NIM : 210601161301112
Program Studi : SARJANA (S1)
Departemen : TEKNIK ELEKTRO
Fakultas : TEKNIK
Jenis Karya : TUGAS AKHIR

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul :

SISTEM *CRAWLING* DATA INSTRUMEN AKREDITASI BERBASIS
SELENIUM DAN PANDAS

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang
Pada Tanggal : 22 Desember 2020

Yang menyatakan,

(LAILA LATHIFAH)
21060116130112

ABSTRAK

Perkembangan teknologi informasi telah sampai pada masa dimana hampir setiap aktivitas transaksi dapat dilakukan secara daring tanpa bertemu dengan pihak yang bersangkutan. Sama halnya dengan akreditasi kampus yang evaluasinya dapat dilakukan secara daring melalui web SAPTO (Sistem Akreditasi Perguruan Tinggi Online) yang dikembangkan oleh pihak BAN-PT (Badan Akreditasi Nasional Perguruan Tinggi). Pada laporan Tugas Akhir ini akan membahas mengenai pembangunan sistem pengumpulan data dari pangkalan database berbasis web menggunakan teknik crawling dan proses filtering data yang dapat mendukung proses akreditasi secara daring. Sistem crawling data didukung oleh tools Selenium dan sistem filtering data menggunakan library Pandas Dataframe. Crawling data dilakukan untuk 4 laman web berbeda, yakni laman web Eduk yang berisi data diri dosen Universitas Diponegoro, laman web Sip3mu yang berisi data penelitian dosen Universitas Diponegoro, laman web Prestasi yang berisi data perlombaan mahasiswa Universitas Diponegoro, dan laman web Forlap yang berisi data program studi serta jumlah mahasiswa Universitas Diponegoro. Sistem crawling data menyesuaikan dengan inspect element dan interface-nya sehingga menghasilkan 9 berkas dengan total sebesar 4756,7 Kb. Sistem filtering data menyesuaikan dengan keperluan analisis data lebih lanjut, tetapi kinerjanya kurang stabil saat mengelola data, dimana semakin banyak data maka semakin besar pula kecepatan eksekusi dan penggunaan memorinya.

Kata kunci : *Crawling Data, Python, Selenium, Pandas, Dataframe*

ABSTRACT

The development of information technology has reached a time when almost every transaction activity can be done online without meeting with the party concerned. Similarly, campus accreditation evaluation can be done online through the SAPTO web developed by BAN-PT. In this Final Task report will discuss the construction of a database collection system from a web-based database using crawling techniques and data cleaning processes that can support the accreditation process online. The data crawling system is supported by Selenium tools and data filtering system using Pandas Dataframe library. Crawling data is done for 4 different websites, namely Eduk's web page containing data of Diponegoro University lecturers, Sip3mu website containing research data of Diponegoro University lecturers, Prestasi website containing data on the computation of Diponegoro University students, and Forlap web pages containing data program study and the number of Diponegoro University students. The system crawling data adjusts to inspect elements and their interfaces to produce 9 files which have total 4756.7 Kb. The system filtering data adapts to the needs of further data analysts, but its performance is less stable when managing data, where the more data, the greater the speed of execution and memory usage.

Keywords: *Data Crawling, Python, Selenium, Pandas, Dataframe*

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT atas rahmat, hidayah dan karunia-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dan penyusunan laporan ini. Tugas Akhir dengan judul “Sistem *Crawling* Data Instrumen Akreditasi Berbasis Selenium dan Pandas” ini diajukan untuk memenuhi syarat akhir menyelesaikan pendidikan Program Studi Sarjana pada Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.

Adapun penyusunan dan penyelesaian laporan Tugas Akhir ini tidak lepas dari bantuan dan dukungan semua pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku Ketua Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.
2. Bapak Yuli Christyono, S.T., M.T. selaku Ketua Program Studi Sarjana Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro
3. Bapak Munawar Agus Riyadi, S.T., M.T., Ph.D. selaku Sekretaris Program Studi Sarjana Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro.
4. Bapak Eko Handoyo, S.T., M.T. selaku Dosen Pembimbing I yang membimbing saya dalam pembuatan Tugas Akhir ini.
5. Bapak Yosua Alvin Adi Soetrisno, S.T., M.Eng. selaku Dosen Pembimbing II yang membimbing saya dalam pembuatan Tugas Akhir ini.
6. M. Arfan, ST., MT. selaku Dosen Wali, yang telah membimbing saya dalam hal perkuliahan.
7. Segenap Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro.
8. Bapak Edy Suryanto, S.Pd dan Ibu Wihayati, S.Pd selaku orang tua penulis yang senantiasa mengiringi perjalanan penulis dengan doa, cinta dan kasih serta memberikan dukungan moril dan materiil kepada penulis

9. Mas M. Syarif, S.T., M.T., selaku pembimbing Kerja Praktik saat di PT. Bumi Manunggal Sinergi yang telah membantu penulis dalam menyelesaikan mata kuliah tersebut dengan sangat baik.
10. Nurlaila Fitri Febriyanti selaku partner TA penulis yang selalu memotivasi, menerima keluhan kesah, bertukar pikiran dengan penulis.
11. Sidiq Budi Perkasa selaku partner dalam kehidupan sehari-hari yang selalu menjadi *support system* selama pengerjaan Tugas Akhir.
12. Palupi, Nabiilah, Dina, Zahirah, Diyah, Riri, Nilam, Haidar, Azmi, Syena selaku sahabat terbaik penulis di setiap suka duka perjalanan penulis
13. Najib, Willi, Haikal, Annisa, Akmal, Mimim, dan Mughaz atas bantuan ekstra kepada penulis
14. Teman-teman Gugus Radikal yang telah menemani kehidupan perkuliahan penulis.
15. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dengan ikhlas baik secara moril maupun materi

Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini tidak luput dari kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diperlukan oleh penulis demi kebaikan dan kesempurnaan penyusunan laporan di masa yang akan datang. Semoga laporan Tugas Akhir ini dapat memberikan manfaat dan menambah pengetahuan bagi kita semua.

Semarang, 16 Desember 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan.....	2
1.3 Batasan Masalah.....	2
1.4 Sistematika Penulisan.....	2
BAB II LANDASAN TEORI	4
2.1 SAPTO	4
2.2 <i>Crawling</i>	5
2.3 Selenium WebDriver.....	5
2.3.1 Navigasi	7
2.3.2 Lokasi Elemen	8
2.4 Python.....	8
2.4.1 Pandas Dataframe	9
2.4.2 Pyspark Dataframe.....	10
2.5 JSON	10
2.6 HTML.....	11
BAB III PERANCANGAN SISTEM	13
3.1 Analisis Kebutuhan	13

3.1.1 Deskripsi Sistem	13
3.1.2 Kebutuhan Fungsional	14
3.1.3 Kebutuhan Non Fungsional	15
3.1.4 Kebutuhan Perangkat Keras.....	16
3.1.5 Kebutuhan Perangkat Lunak.....	16
3.2 Perancangan Sistem Web <i>Crawling</i>	17
3.2.1 Sistem <i>Crawling</i> Data di Laman Web Eduk Undip.....	18
3.2.2 Sistem <i>Crawling</i> Data di Laman Web Sip3mu Undip.....	19
3.2.3 Sistem <i>Crawling</i> Data di Laman Web Prestasi Undip.....	20
3.2.4 Sistem <i>Crawling</i> Data di Laman Web Forlap Dikti.....	21
3.3 Perancangan Sistem <i>Fitering</i> Data	22
3.3.1 Permisahan Data dalam Baris	23
3.3.2 Permisahan Data dalam Kolom	23
3.3.3 Penghapusan Data.....	24
BAB IV HASIL DAN PEMBAHASAN	25
4.1 Pengujian <i>Blackbox</i>	25
4.2 Sistem <i>Crawling</i> Data	26
4.2.1 Sistem <i>Crawling</i> Data di Laman Web Eduk Undip.....	27
4.2.2 Sistem <i>Crawling</i> Data di Laman Web Sip3mu Undip.....	31
4.2.3 Sistem <i>Crawling</i> Data di Laman Web Prestasi Undip.....	34
4.2.4 Sistem <i>Crawling</i> Data di Laman Web Forlap Dikti.....	40
4.3 Sistem <i>Filtering</i> Data	43
4.3.1 Pengujian Sistem <i>Filtering</i> Data.....	43
4.3.2 Hasil <i>Filtering</i> Data	49
BAB V PENUTUP.....	54
5.1 Kesimpulan.....	54
5.2 Saran.....	54
DAFTAR PUSTAKA	55
BIODATA	57
LAMPIRAN A	58

DAFTAR GAMBAR

Gambar 2.1 Tampilan laman web SAPTO BAN-PT	4
Gambar 2.2 Tahapan proses teknik <i>crawling</i>	5
Gambar 2.3 Selenium WebDriver	6
Gambar 2.4 Arsitektur Selenium ChromDriver	7
Gambar 3.1 Desain Sistem	13
Gambar 3.2 <i>Flowchart</i> perancangan sistem web <i>crawling</i>	17
Gambar 3.3 Alur <i>crawling</i> data di laman web Eduk Undip	18
Gambar 3.4 Alur <i>crawling</i> data di laman web Sip3mu Undip	19
Gambar 3.5 Alur <i>crawling</i> data di laman web Prestasi Undip	20
Gambar 3.6 Alur <i>crawling</i> data di laman web Forlap Dikti	21
Gambar 3.7 Alur <i>Filtering</i> data	22
Gambar 3.8 Tampilan data di laman web Sip3mu Undip	23
Gambar 3.9 Tampilan data di laman web Prestasi Undip	23
Gambar 3.10 Tampilan data di Laman Web Forlap Dikti	24
Gambar 4.1 Tampilan penyimpanan berkas program dan data	26
Gambar 4.2 Tampilan halaman web SSO untuk <i>submit</i> nama pengguna	27
Gambar 4.3 <i>Inspect element</i> halaman web SSO untuk <i>submit</i> nama pengguna ...	28
Gambar 4.4 Tampilan halaman web SSO untuk <i>submit</i> kata sandi	29
Gambar 4.5 <i>Inspect element</i> halaman web SSO untuk <i>submit</i> kata sandi	30
Gambar 4.6 Tampilan halaman web SSO untuk lanjut ke halaman web Eduk Undip	30
Gambar 4.7 <i>Inspect element</i> halaman web SSO untuk lanjut ke laman web Eduk Undip	31
Gambar 4.8 Tampilan halaman web Sip3mu Undip untuk <i>submit</i> akun admin ...	32
Gambar 4.9 <i>Inspect element</i> halaman web Sip3mu Undip untuk <i>submit</i> akun admin	32
Gambar 4.10 Tampilan halaman web Sip3mu Undip untuk mengunduh	33

Gambar 4.11 <i>Inspect element</i> halaman web Sip3mu Undip untuk mengunduh data	34
Gambar 4.12 Tampilan halaman web Prestasi Undip untuk <i>submit</i> akun admin .	35
Gambar 4.13 <i>Inspect element</i> halaman web Prestasi Undip untuk <i>submit</i> akun admin	35
Gambar 4.14 Tampilan halaman web Prestasi Undip untuk pengumpulan <i>link</i> ...	36
Gambar 4.15 <i>Inspect element</i> halaman web Prestasi Undip untuk pengumpulan <i>link</i>	37
Gambar 4.16 Tampilan halaman web Prestasi Undip untuk pengambilan data ...	38
Gambar 4.17 <i>Inspect element</i> halaman web Prestasi Undip untuk pengambilan data.....	39
Gambar 4.18 Tampilan halaman web Forlap Dikti untuk pengambilan data daftar program studi	40
Gambar 4.19 <i>Inspect element</i> halaman web Forlap Dikti untuk pengambilan data daftar program studi.....	41
Gambar 4.20 Tampilan halaman web Forlap Dikti untuk pengambilan data jumlah mahasiswa	42
Gambar 4.21 <i>Inspect element</i> halaman web Forlap Dikti untuk pengambilan data jumlah mahasiswa.....	42
Gambar 4.22 Grafik perbandingan kecepatan eksekusi pada laman web Sip3mu Undip	45
Gambar 4.23 Grafik perbandingan penggunaan memori pada laman web Sip3mu Undip	45
Gambar 4.24 Grafik perbandingan kecepatan eksekusi pada laman web Prestasi Undip	46
Gambar 4.25 Grafik perbandingan penggunaan memori pada laman web Prestasi Undip	47
Gambar 4.26 Grafik perbandingan kecepatan eksekusi pada laman web Forlap Dikti	47
Gambar 4.27 Grafik perbandingan penggunaan memori pada laman web Forlap Dikti	48

Gambar 4.28 Hasil <i>filtering</i> data di laman web Sip3mu Undip	49
Gambar 4.29 Hasil <i>filtering</i> data di laman web Prestasi Undip	50
Gambar 4.30 Hasil pertama <i>filtering</i> data di laman web Forlap Dikti	51
Gambar 4.31 Hasil kedua <i>filtering</i> data di laman web Forlap Dikti	52

DAFTAR TABEL

Tabel 3.1 Kebutuhan perangkat keras	16
Tabel 3.2 Kebutuhan perangkat lunak	16
Tabel 4.1 Hasil pengujian <i>blackbox</i>	25
Tabel 4.2 Hasil rerata pengujian Pandas <i>dataframe</i> dan Pyspark <i>dataframe</i>	44
Tabel 4.3 Perbandingan hasil data <i>filtering</i>	52

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah sampai pada masa dimana hampir setiap aktivitas transaksi dapat dilakukan secara daring tanpa bertemu dengan pihak yang bersangkutan. Sama halnya dengan akreditasi kampus yang menunjukkan kualitas, dimana kualitas pendidikan perguruan tinggi telah menjadi masalah *transcendental*. Hal ini berkenaan dengan meningkatnya kepedulian pemerintah terhadap berbagai tingkat kualitas yang dibuktikan oleh sistem pendidikan. Menanggapi masalah ini, beberapa evaluasi dan praktik akreditasi dilaksanakan untuk memastikan dan meningkatkan kualitas karir dan institusi universitas di berbagai negara Amerika Latin, dimana pendataan sudah bisa dilakukan secara daring [1].

Berdasarkan Peraturan BAN-PT nomor 5 Tahun 2019, yang telah ditetapkan pada tanggal 23 September 2019 pendataan akreditasi di Indonesia dapat dilakukan secara daring melalui situs sapro.banpt.or.id [5]. SAPTO (Sistem Akreditasi Perguruan Tinggi Online) merupakan sistem yang dikembangkan BAN-PT untuk meningkatkan efisiensi dan kualitas proses akreditasi perguruan tinggi yang diselenggarakan oleh BAN-PT. SAPTO mendukung setiap proses yang dilakukan dalam akreditasi seperti pengajuan usulan akreditasi oleh perguruan tinggi, pemeriksaan dokumen, penugasan asesor dan validasi yang dilakukan, proses AK (asesmen kecukupan) dan AL (asesmen lapangan) oleh asesor. [2]

Berdasarkan peraturan tersebut perlu adanya sistem pengumpulan data yang dapat dijalankan secara otomatis dan berkala untuk mempermudah proses pengumpulan data yang disesuaikan dengan kebutuhan analisis data selanjutnya. Data yang telah terkumpul akan di *filtering* menggunakan *dataframe* pada library Pandas. Oleh karena itu, penelitian ini akan membahas mengenai “Sistem *Crawling* Data Instrumen Akreditasi Berbasis Selenium dan Pandas”. Selenium memudahkan untuk *crawling* data karena dapat melakukan interaksi seperti yang dilakukan oleh user ketika menelusuri web seperti melakukan klik pada tombol, mengisi form,

membuka tab baru, membuka halaman web, dan lain-lain[4]. Penggunaan *dataframe* memudahkan untuk membaca sebuah berkas dan menjadikannya table, selain itu dapat mengolah suatu data dengan menggunakan operasi seperti *join*, *distinct*, *group by*, agregasi, dan teknik lainnya yang terdapat pada SQL[6].

1.2 Tujuan

Adapun tujuan utama dari penelitian tugas akhir ini adalah sebagai berikut:

1. Memenuhi salah satu syarat untuk mendapatkan gelar sarjana
2. Membuat suatu sistem pengambilan data di pangkalan database berbasis web menggunakan teknik *crawling*.
3. Melakukan *filtering* menggunakan *dataframe* untuk menghasilkan data yang disesuaikan dengan kebutuhan analisis data selanjutnya.

1.3 Pembatasan Masalah

Untuk membatasi pembahasan dalam penelitian tugas akhir ini maka diberikan pembatasan masalah sebagai berikut:

1. Pembuatan program menggunakan Python 3.8.5
2. Sistem dibangun berdasarkan teknik *crawling* menggunakan *tool* Selenium dengan perangkat lunak pendukung ChromeDriver.
3. Proses *filtering* data menggunakan Pandas *dataframe*.
4. Data yang diperoleh hanya dari pangkalan *database* berbasis web yang diizinkan oleh pihak berwenang untuk diakses peneliti.
5. Pembahasan sistem hanya sampai berkas berekstensi .json dihasilkan.

1.4 Sistematika Penulisan

Sistematika penulisan dalam laporan Tugas Akhir dengan judul “Sistem *Crawling* Data Instrumen Akreditasi Berbasis Selenium dan Pandas” ini adalah sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, tujuan Tugas Akhir, pembatasan masalah, metodologi penulisan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas tentang teori dasar mengenai SAPTO, *Crawling*, Selenium WebDriver, Python, dan JSON, HTML.

BAB III PERANCANGAN SISTEM

Bab ini berisi perancangan sistem berupa analisis kebutuhan dan desain sistem. Pada analisis kebutuhan akan mencakup dari deskripsi sistem, kebutuhan fungsional, kebutuhan non fungsional, kebutuhan perangkat keras, dan kebutuhan perangkat lunak. Pada perancangan sistem terbagi menjadi dua, yakni perancangan sistem pengambilan data untuk setiap pangkalan *database* berbasis web dan perancangan sistem penyaringan data menggunakan dataframe Pandas.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan sistem. Pada hasil terdapat pengujian *blackbox* seluruh program dan data yang dihasilkan. Sedangkan pada pembahasan terbagi menjadi dua, yakni pembahasan mengenai sistem *crawling* data dan sistem *filtering* data.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari seluruh pembahasan Tugas Akhir.

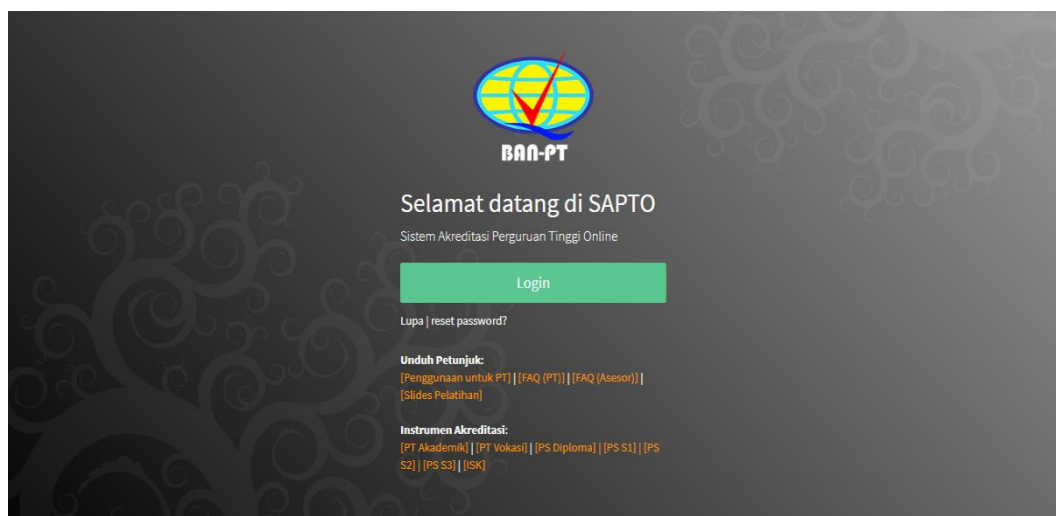
BAB II

LANDASAN TEORI

2.1 SAPTO

SAPTO adalah sistem akreditasi berbasis web yang dikembangkan BAN-PT untuk meningkatkan efisiensi dan kualitas proses akreditasi perguruan tinggi yang diselenggarakan oleh BAN-PT. SAPTO mendukung setiap proses yang dilakukan dalam akreditasi seperti pengajuan usulan akreditasi oleh perguruan tinggi, pemeriksaan dokumen, penugasan asesor dan validasi yang dilakukan, proses AK dan AL oleh asesor.[2]

Dalam sistem SAPTO Perguruan Tinggi (PT) berperan sebagai entitas yang mengajukan usulan akreditasi baik untuk Akreditasi Perguruan Tinggi (APT), maupun Akreditasi Program Studi (APS). Setiap perguruan tinggi akan diberi 1 (satu) akun menggunakan kode perguruan tinggi yang terdaftar pada Pangkalan Data Pendidikan Tinggi (PD-Dikti) . Akun tersebut digunakan untuk mengajukan akreditasi perguruan tinggi dan akreditasi program studi yang berada di lingkungan perguruan tinggi tersebut. Gambar 2.1 berikut menunjukkan halaman depan SAPTO BAN-PT.



Gambar 2.1 Tampilan laman web SAPTO BAN-PT

2.2 *Crawling*

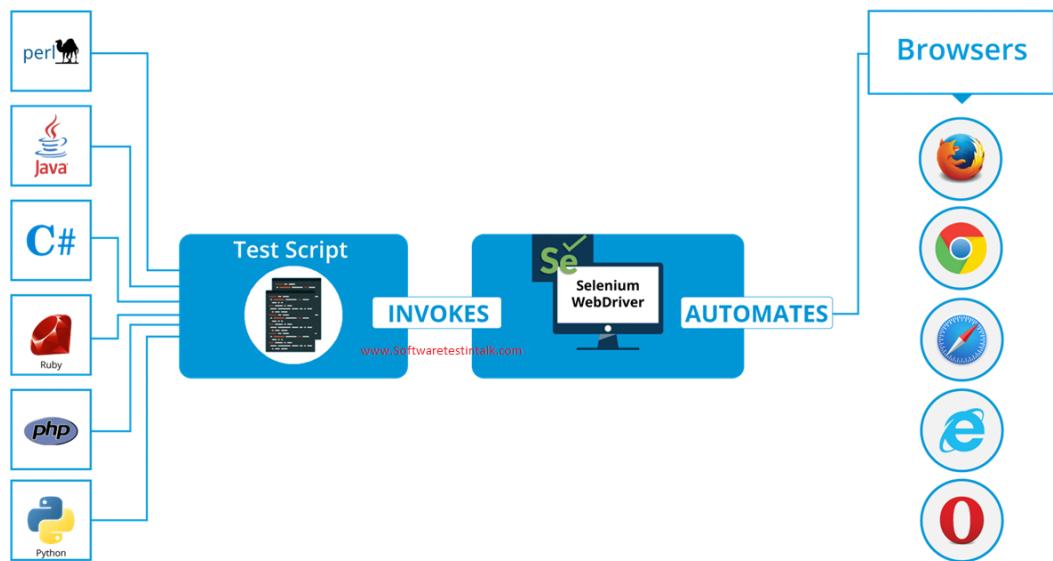
Crawling merupakan teknik mengumpulkan data pada sebuah website dengan memasukkan *Uniform Resource Locator* (URL). URL ini menjadi acuan untuk mencari semua *hyperlink* yang ada pada *website*. Kemudian dilakukan *indexing* untuk mencari kata dalam dokumen pada setiap *link* yang ada. Penerapan *crawling* dengan menggunakan *automation program* dan menggunakan *Application Programming Interface* (API) sebagai jalur komunikasi dalam mendapatkan data [7]. Gambar 2.2 berikut menunjukkan tahapan proses teknik *crawling*.



Gambar 2.2 Tahapan proses teknik *crawling* [8]

2.3 **Selenium WebDriver**

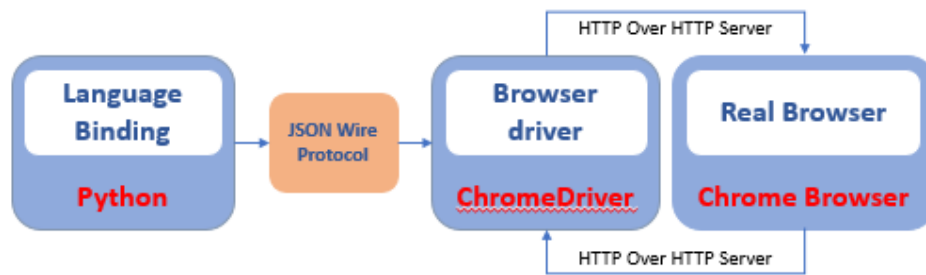
Gambar 2.3 berikut merupakan representasi prangkat pendukung penggunaan Selenium Webdriver.



Gambar 2.3 Selenium Webdriver [9]

Selenium adalah *tools auto testing* yang digunakan untuk mengotomatisasi tes aplikasi web yang dilakukan pada *browser*. Selenium akan melakukan validasi web apps pada berbagai *browser* dan *platform* [10]. Pada Gambar 2.3 menunjukkan bahwa bahasa pemrograman untuk membuat *test scripts* Selenium antara lain Java, Python, C#, Perl, JavaScript, Ruby, dan PHP. Peneliti akan menggunakan bahasa pemrograman Python. Sedangkan peramban web yang bisa digunakan untuk proses otomatisasi antara lain Chrome, Firefox, Opera, Safari dan IE. Penulis akan menggunakan peramban web Chrome. Fasilitas yang ditawarkan Selenium yang dapat menggunakan berbagai web *browser* dan bahasa pemrograman Python menjadi alasan untuk menggunakannya sebagai *tool auto testing* web dalam penelitian ini.

WebDriver adalah API dan protokol yang mendefinisikan antarmuka tanpa bahasa untuk mengontrol perilaku *browser* web. Setiap *browser* didukung oleh implementasi WebDriver tertentu, yang disebut *driver*. *Driver* adalah komponen yang bertanggung jawab untuk mendelegasikan ke *browser*, dan menangani komunikasi ke dan dari Selenium dan browser.[11] WebDriver akan menyesuaikan peramban web yang digunakan penulis, yakni ChromeDriver. Gambar 2.4 berikut menunjukkan arsitektur dari Selenium yang menggunakan ChromeDriver.



Gambar 2.4 Arsitektur Selenium ChromeDriver

Pada Gambar 2.4 menunjukkan bahwa ChromeDriver sebagai pihak ketiga untuk menjalankan *automation* web dengan membuka jendela peramban yang baru. ChromeDriver tersedia untuk Chrome di Android dan Chrome di Desktop (Mac, Linux, Windows, dan ChromeOS).

2.3.1 Navigasi

Melakukan crawling tidak hanya menuju ke suatu halaman web tertentu, tetapi juga berinteraksi dengan halaman, atau lebih khusus lagi berinteraksi dengan elemen HTML dalam halaman. Pertama-tama, kita perlu menemukan elemen tersebut. WebDriver menawarkan sejumlah cara untuk menemukan elemen. Misalnya, diberikan elemen yang didefinisikan sebagai:

```
<input type="text" name="uname" id="uname" />
```

Berikut adalah cara untuk menemukan elemen diatas:

```
element = driver.find_element_by_id("uname")
```

```
element = driver.find_element_by_name("uname")
```

```
element = driver.find_element_by_xpath("//input[@id='uname']")
```

```
element = driver.find_element_by_css_selector("input#uname")
```

Setelah menemukan elemen, berikut adalah cara memasukkan *username* :

```
element.click()
```

```
element.send_keys("some text")
```

```
element.submit()
```

2.3.2 Lokasi Elemen

Ada beberapa strategi untuk menemukan elemen di halaman. Selenium menyediakan metode berikut untuk menemukan elemen di halaman:

- `find_element_by_id`
- `find_element_by_name`
- `find_element_by_xpath`
- `find_element_by_link_text`
- `find_element_by_partial_link_text`
- `find_element_by_tag_name`
- `find_element_by_class_name`
- `find_element_by_css_selector`

Berikut metode untuk menemukan banyak elemen dan menjadikannya sebagai *list*:

- `find_elements_by_name`
- `find_elements_by_xpath`
- `find_elements_by_link_text`
- `find_elements_by_partial_link_text`
- `find_elements_by_tag_name`
- `find_elements_by_class_name`
- `find_elements_by_css_selector`

2.4 Python

Python adalah bahasa pemrograman tingkat tinggi. Dibuat oleh Guido Van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu *programmer* menulis kode yang jelas dan logis untuk proyek skala kecil dan besar.[16]

Terdapat dua versi Python yang beredar, yakni Python versi 2 dan Python versi 3. Peneliti akan menggunakan Python versi 3. Berikut fitur-fitur Python yang menjadi keunggulan darinya:

1. Berorientasi kepada objek.

2. Mudah dikembangkan dengan menciptakan modul-modul baru. Modul tersebut juga bisa dibangun dengan bahasa Python.
3. Memiliki tata bahasa yang mudah dipelajari.
4. Didukung sistem pengelolaan memori secara otomatis sehingga membutuhkan kinerja saat *coding*.
5. Python juga memiliki banyak fasilitas pendukung sehingga ketika mengoperasikannya, terhitung mudah dan cepat.

2.4.1 Pandas *Dataframe*

Pandas adalah paket dasar penting selain dari Numpy yang tersedia pada bahasa pemrograman Python. Saat bekerja dengan data tabular, seperti data yang disimpan di *spreadsheet* atau *database*, pandas adalah alat yang tepat untuk mengolah dan memproses data. Pandas dapat membantu dalam menjelajahi, membersihkan, dan memproses data [16]. Pandas dibangun di atas NumPy dan menyediakan implementasi *dataframe* yang efisien. *Dataframe* pada dasarnya adalah *array* multidimensi dengan label baris dan kolom terlampir, dan seringkali dengan tipe heterogen dan / atau data yang hilang[17].

Selain menawarkan antarmuka penyimpanan yang nyaman untuk data berlabel, Pandas mengimplementasikan sejumlah operasi data canggih yang akrab bagi pengguna kerangka kerja *database* dan program *spreadsheet*. Pandas mendukung integrasi dengan banyak format file atau sumber data secara langsung (CSV, Excel, SQL, JSON, parquet,...)[17].

Pandas menggunakan dua struktur data, salah satunya adalah *dataframe*. *Dataframe* adalah *array* dua dimensi dengan indeks baris fleksibel dan nama kolom fleksibel. Demikian pula, *dataframe* dapat dianggap sebagai kamus/*dictionary* yang spesial. Sebagaimana *dictionary* memetakan kunci ke sebuah nilai, *dataframe* memetakan nama kolom ke serangkaian data kolom.

Berikut sintaks untuk membuat Pandas *dataframe*:

```
pandas.DataFrame(data, index, columns, dtype, copy)
```

dengan keterangan:

- a. `index` merupakan label untuk baris

- b. `columns` merupakan label untuk kolom
- c. `dtype` merupakan tipe data perkolom
- d. `copy` digunakan untuk menyalin data, *default*-nya `False`

2.4.2 Pyspark *Dataframe*

Pyspark merupakan bagian dari kerangka kerja Apache Spark dengan bahasa pemrograman Python yang memungkinkan untuk memanipulasi data dalam skala besar dan bekerja dengan objek dan algoritma melalui sistem berkas terdistribusi[13]. Salah satu sistem berkas terdistribusi yang dimiliki oleh Spark adalah *dataframe*. *Dataframe* mulai muncul di Spark Release 1.3.0. Di Apache Spark, *Dataframe* adalah kumpulan baris terdistribusi di bawah kolom bernama. Secara sederhana, ini sama seperti tabel dalam database relasional atau lembar Excel dengan header Kolom dan dapat dibuat dari beragam sumber seperti: *file* data terstruktur, tabel di Hive, *database* eksternal, atau RDD (*Resilient Distributed Dataset*) yang sudah ada[12].

Berikut contoh sintaks untuk membuat Pyspark *dataframe* :

```
df = sqlContext.read.csv('PATH',header=True,inferSchema=True)
```

dengan keterangan:

- a. `sqlContext` merupakan kelas dalam Spark SQL
- b. `path` merupakan tempat penyimpanan berkas
- c. `header` digunakan untuk menjadikan baris pertama sebagai nama kolom, *default*-nya `False`
- b. `inferSchema` untuk membuat tipe data menjadi String, *default*-nya `False`

2.5 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (dibangkitkan) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa

pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan lain lain. Oleh karena sifat-sifat tersebut, JSON ideal sebagai bahasa pertukaran-data[18]. JSON terbuat dari dua struktur :

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

2.6 HTML

HTML adalah singkatan dari *Hypertext Markup Language* yang memungkinkan seorang user untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman web dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word. Dokumen HTML adalah berkas yang diakhiri dengan ekstensi .html atau .htm. Ekstensi *file* ini bisa dilihat dengan menggunakan web *browser* apapun (seperti Google Chrome, Safari, atau Mozilla Firefox). *Browser* tersebut membaca *file* HTML dan me-render kontennya sehingga *user* internet bisa melihat dan membacanya.[14]

Berikut contoh kode dari susunan atau struktur HTML:

```
<div>
  <h1>Selamat Datang!</h1>
  <h2>Jenis-jenis Bunga</h2>
  <p>Paragraph one</p>
  <p>ada with a <a href="https://bunga.com">contoh</a></p>
</div>
```

dengan keterangan :

1. Elemen teratas dan terbawah adalah *division* sederhana (<div></div>) yang bisa digunakan untuk *mark up* bagian konten yang lebih besar.
2. Susunan HTML di atas terdiri atas *heading* (<h1></h1>), *subheading* (<h2></h2>), dan dua paragraf (<p></p>),
3. Paragraf kedua meliputi sebuah *link* (<a>) dengan *attribute* href yang terdiri atas URL tujuan.

BAB III

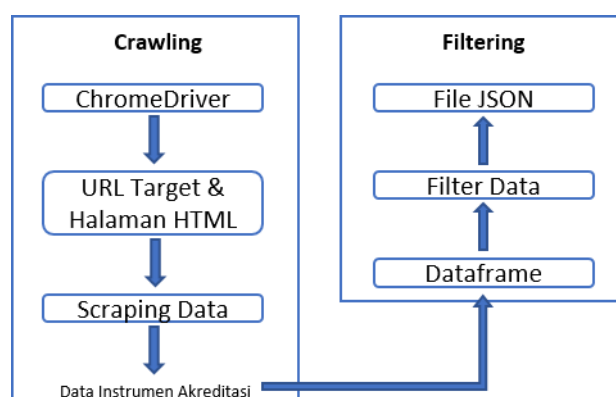
PERANCANGAN SISTEM

3.1 Analisis Kebutuhan

Analisis kebutuhan sistem ini ditujukan untuk menguraikan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna dan sesuai dengan tujuan Tugas Akhir yang berjudul Sistem *Crawling* Data Instrumen Akreditasi Berbasis Selenium dan Pandas. Dalam bab ini akan dijelaskan kebutuhan perangkat keras, kebutuhan perangkat lunak, dan kebutuhan data yang menunjukkan spesifikasi sistem yang dapat berjalan secara otomatis.

3.1.1 Deskripsi Sistem

Konsep yang dibahas adalah bagaimana aplikasi ini dapat mengambil data yang berada pada suatu laman web dan menyaring data sesuai kebutuhan secara otomatis. Selain itu, terdapat proses *login* ke dalam laman web yang bersifat rahasia dan hanya admin yang dapat mengaksesnya. Gambar 3.1 berikut akan menunjukan desain sistem *crawling* yang dibangun.



Gambar 3.1 Desain Sistem

Pengambilan data dilakukan secara otomatis menggunakan teknik *crawling* data dapat dilihat alurnya pada Gambar 3.1. Proses *crawling* data diawal dengan terbukanya ChromeDriver yang langsung mengakses URL target untuk

melakukan *login* akun admin kemudian menuju ke halaman HTML yang telah ditentukan dalam *scripts* dan melakukan *scraping* (pengambilan data). Data yang akan diambil dalam bentuk *tabel* ataupun *form* yang akan disimpan sementara pada suatu *list* atau diunduh dalam bentuk berkas berekstensi *.xls* maupun *.json*. Kemudian, data tersebut dimasukkan ke dalam *Dataframe* untuk dibersihkan sesuai dengan desain *database* yang dibutuhkan dalam melakukan proses pengolahan data. *Dataframe* yang dinilai sudah sesuai dengan kebutuhan akan disimpan dalam sebuah berkas berekstensi *.json* untuk mempermudah proses *import* ke dalam *database*. Berkas berekstensi *.json* merupakan akhir dari sistem ini, dengan kata lain berkas yang diambil di laman web dalam bentuk *.json* dianggap sebagai hasil keluaran dan tidak akan diproses dalam sistem filtering data.

Sistem *crawling* ini menggunakan *tool* Selenium dengan perangkat lunak tambahan berupa *browser driver* atau *webdriver*. Selenium dapat dijalankan menggunakan beberapa bahasa pemrograman, salah satunya adalah Python yang akan digunakan dalam pembangunan aplikasi web *crawling* ini. *Webdriver* pendukung Selenium yang dipakai adalah Chrome untuk mempermudah proses pengambilan data dengan bantuan Chrome Extension tertentu.

3.1.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan gambaran mengenai fungsi-fungsi yang dapat dilakukan oleh sistem ini. Kebutuhan fungsional sistem meliputi:

1. Mengakses halaman HTML sesuai dengan URL yang dicantumkan dalam *scripts*.
2. Mengambil data pada suatu tabel ataupun *form* untuk disimpan sementara dalam bentuk *list* atau berkas unduhan berekstensi *.xls*.
3. Menyaring data yang ada pada penyimpanan sementara menggunakan *dataframe* supaya tidak mengubah data unduhan dari halaman HTML.
4. Menyimpan hasil akhir *dataframe* ke dalam berkas berekstensi *.json*.

3.1.3 Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan sistem meliputi kinerja, kelengkapan operasi pada fungsi-fungsi yang ada, serta kesesuaian dengan lingkungan penggunaannya. Kebutuhan non-fungsional ini melingkupi beberapa kebutuhan yang mendukung kebutuhan fungsional, rumusan kebutuhan non-fungsional meliputi:

1. Kebutuhan Operasional

- Kecepatan dapat berjalan dengan baik pada sistem operasi Ubuntu dengan RAM minimal 8Gb
- Web *Crawling* membutuhkan internet yang stabil untuk menjaga keutuhan data.
- Sistem hanya dapat diakses dan digunakan oleh petugas pengelola akreditasi.
- Sistem ini dibangun menggunakan *tool* Selenium didukung ChromeDriver dan *library* Pandas oleh Python 3.

2. Performa Sistem

Sistem yang dibangun merupakan aplikasi yang berjalan pada laptop. Terdapat beberapa keterbatasan yang ditemui pada laptop meskipun sistem operasi yang digunakan adalah Ubuntu. Oleh karena itu perlu diperhatikan guna menjadi acuan dalam pengembangan sistem, diantaranya:

- Penggunaan laptop yang tidak bisa menyala secara terus menerus selama 24 jam sehari.
- *System* yang dirancang untuk web *crawling* belum bisa mendeteksi *update* data secara berkala.

Dari keterbatasan pada computer *server* tersebut, maka diusulkan beberapa alternatif untuk menunjang performa sistem dengan keterbatasan yang ada, diantaranya:

- Menggunakan *computer server* yang tersedia di institusi dan aktif selama 24 jam dalam sehari.
- Merancang *system* untuk melakukan pengambilan data setiap 24 jam sekali.

3.1.4 Kebutuhan Perangkat Keras

Dalam pembangunan sistem ini, dibutuhkan beberapa spesifikasi perangkat keras. Spesifikasi perangkat keras tersebut dapat dimasukkan ke dalam kebutuhan perangkat keras dalam analisis kebutuhan. Karena melibatkan pengambilan dan penyaringan data, perangkat keras yang dibutuhkan dalam membuat aplikasi ini adalah sebuah komputer dengan spesifikasi minimal yang ditunjukkan pada Tabel 3.1 berikut.

Tabel 3.1 Kebutuhan perangkat keras

Spesifikasi	Keterangan
<i>Processor</i>	Intel(R) Core(TM) i5-2520M
RAM	8192 MB
<i>Harddisk</i>	31 GB
Laptop	Dell Latitude E6320 Core i5

3.1.5 Kebutuhan Perangkat Lunak

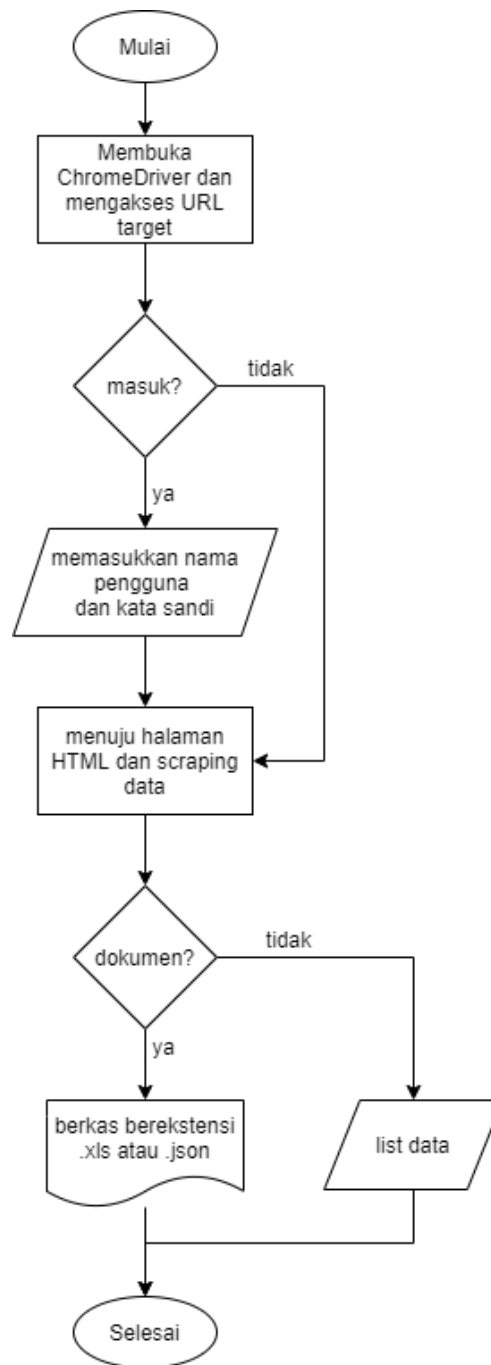
Dalam pembangunan sistem ini, dibutuhkan beberapa spesifikasi perangkat lunak. Spesifikasi perangkat lunak tersebut dapat dimasukkan ke dalam kebutuhan perangkat lunak dalam analisis kebutuhan. Perangkat lunak yang dibutuhkan baik untuk merancang sistem, membangun sistem maupun menjalankan sistem adalah seperti yang ditunjukkan pada Tabel 3.2 berikut.

Tabel 3. 2 Kebutuhan perangkat lunak

Spesifikasi	Keterangan
Sistem Operasi	Ubuntu 18.06
<i>Text Editor</i>	Notepad++
<i>Tool Otomatisasi Web</i>	Selenium 3.0
<i>WebDriver & Browser</i>	Chrome WebDriver & Chrome Browser (versi 84.0.4147)
Bahasa Pemrograman	Python 3
<i>Library</i>	Pandas

3.2 Perancangan Sistem Web *Crawling*

Sistem Web *Crawling* ini bergantung pada laman web yang akan diambil datanya. Tetapi, proses secara garis besar akan digambarkan menggunakan *flowchart* yang dapat dilihat pada Gambar 3.2 berikut ini.

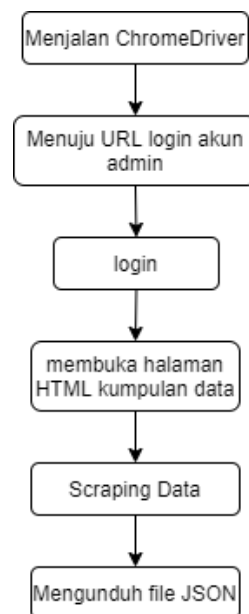


Gambar 3.2 Flowchart perancangan sistem web *crawling*

Pada Gambar 3.2 dapat dilihat bahwa program akan langsung membuka ChromeDriver dan mengakses URL target yang tertera pada *scripts*. Kemudian memasuki proses *login* supaya dapat mengakses halaman HTML yang telah ditentukan. Namun, dari empat laman web yang akan diakses untuk memperoleh data, satu diantaranya tidak perlu melalui proses *login*. Ketika tidak memerlukan proses *login*, sistem akan langsung menuju halaman HTML yang telah ditentukan dalam *scripts* kemudian mendeteksi data berdasarkan elemen tertentu. Setelah data berhasil diakses dan didapatkan, data akan langsung disimpan ke dalam *list* atau berkas dokumen berekstensi *.xls* atau *.json*.

3.2.1 Sistem *Crawling* Data di Laman Web Eduk Undip

Alur sistem *crawling* data di laman web Eduk Undip dapat dilihat pada Gambar 3.3 berikut.



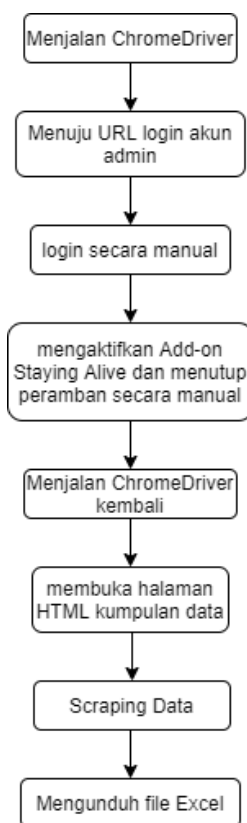
Gambar 3.3 Alur *crawling* data di laman web Eduk Undip

Pada Gambar 3.3 dapat dilihat bahwa proses diawali dengan menjalankan Chromedriver yang membuka peramban Chrome secara otomatis dan langsung mengakses URL target untuk melakukan *login* dimana admin tidak perlu mengetikkan *username* dan *password* karena akan otomatis terisi. Setelah *login*

berhasil, peramban Chrome akan langsung membuka halaman HTML data yang ditentukan dalam *scripts* dan melakukan *scraping* data. Data yang diperoleh akan secara otomatis diunduh. Dokumen JSON yang terunduh menunjukkan akhir dari proses *crawling* data di laman web Eduk Undip.

3.2.2 Sistem *Crawling* Data di Laman Web Sip3mu Undip

Alur sistem *crawling* data di laman web Sip3mu Undip dapat dilihat pada Gambar 3.4 berikut.



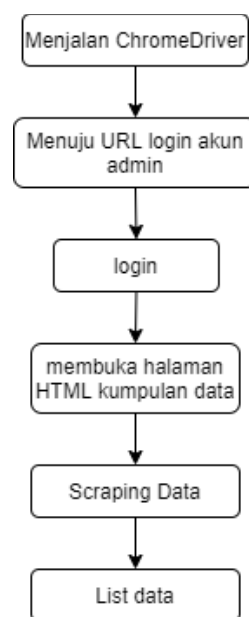
Gambar 3.4 Alur *crawling* data di laman web Sip3mu Undip

Pada Gambar 3.4 dapat dilihat bahwa proses *crawling* diawali dengan menjalankan Chromedriver yang membuka peramban Chrome secara otomatis dan langsung mengakses URL target untuk *login* yang dilakukan secara manual karena adanya *captcha*. Kemudian, mengaktifkan *add-on* Staying Alive untuk menjalankan *session* dan menutup peramban Chrome secara manual. Proses kedua diawali dengan terbukanya peramban Chrome secara otomatis dan langsung

menuju halaman HTML yang tertera dalam *scripts* untuk melakukan proses *scraping* data. Data yang berhasil didapat akan secara otomatis terunduh dalam bentuk berkas Excel berekstensi .xls.

3.2.3 Sistem *Crawling* Data di Laman Web Prestasi Undip

Alur sistem *crawling* data di laman web Prestasi Undip dapat dilihat pada Gambar 3.5 berikut.

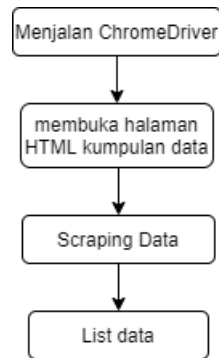


Gambar 3.5 Alur *crawling* data di laman web Prestasi Undip

Pada Gambar 3.5 dapat dilihat bahwa proses *crawling* diawali dengan menjalankan Chromedriver yang membuka peramban Chrome secara otomatis dan langsung mengakses URL target untuk *login*. Setelah login berhasil peramban akan langsung membuka halaman HTML yang berisi data dan melakukan *scraping* data. Terakhir, data akan disimpan sementara di dalam *list*.

3.2.4 Sistem *Crawling* Data di Laman Web Forlap Dikti

Alur sistem *crawling* data di laman web Forlap Dikti dapat dilihat pada Gambar 3.6 berikut.

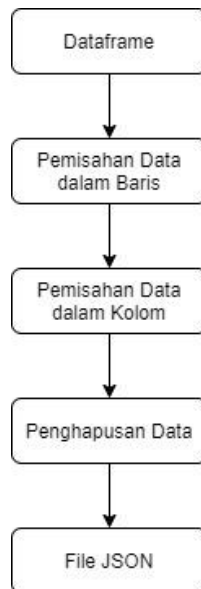


Gambar 3.6 Alur *crawling* data di laman web Forlap Dikti

Pada Gambar 3.6 dapat dilihat bahwa proses diawali dengan terbukanya Chromedriver dan tidak perlu untuk *login* sehingga *system* akan langsung membuka halaman HTML data yang diperlukan untuk dijalankan satu per satu saat pengambilan data. Data akan secara otomatis dimasukkan ke dalam *list* berdasarkan judul kolom.

3.3 Perancangan Sistem *Filtering* Data

Alur sistem *filtering* data di setiap laman web Forlap Dikti dapat dilihat pada Gambar 3.7 berikut.



Gambar 3.7 Alur *filtering* data

Gambar 3.7 menggambarkan proses penyaringan data di tiga dari empat laman web yang harus menggunakan *dataframe* untuk menghindari perubahan pada data orisinil. Pembahasan perancangan penyaringan data tidak dipisah per laman web karena secara keseluruhan akan mengalami proses yang sama, yakni pemisahan data dalam baris jika diperlukan, pemisahan data dalam kolom jika diperlukan, dan penghapusan data jika diperlukan. Setelah selesai penyaringan, data akan disimpan dalam berkas berekstensi .json untuk mempermudah proses *import* ke *database*.

3.3.1 Pemisahan Data dalam Baris

Contoh data yang harus dipisahkan dalam baris dapat dilihat pada Gambar 3.8 berikut ini.

Data Penelitian Fakultas Teknik Tahun 2020				
Download Excel 2020				
Show 10 entries		Search:		
No	Penelitian	Peneliti	Status	Aksi
1	Palm Oil Cracking to Renewable Liquid Fuels over Plasma-Assisted Catalytic Continuous Reactor to Supply National Energy Demand Jenis : Dasar (TRL 1-3) Bidang : ILMU KETEKNIKAN INDUSTRI Berkas : 6593_penelitian(1).pdf	1. Prof. Dr. Istadi, S.T., M.T. 2. Dr. Luqman Buchori, ST, MT 3. Prof. Ir. Didi Dwi Anggoro, M.Eng, Ph.D	Data Lengkap : Ya Validasi Fakultas : Tidak Tgl Validasi : 30-05-2018 Validasi LPPM : Tidak Tgl Validasi : 30-05-2018	
2	Development of Low Fouling Ultrafiltration Membranes by Reactive Phase Separation Jenis : Dasar (TRL 1-3) Bidang : Lainnya Berkas : 1005_0029057502_penelitian.pdf	1. Prof. Dr.rer.nat. Heru Susanto, ST, MM, MT	Data Lengkap : Ya Validasi Fakultas : Tidak Tgl Validasi : 27-02-2020 Validasi LPPM : Tidak Tgl Validasi : 27-02-2020	

Gambar 3.8 Tampilan data di laman web Sip3mu Undip

Gambar 3.8 menunjukkan data yang akan diperoleh pada bagian peneliti akan menjadi satu baris sedangkan data yang dibutuhkan dalam satu baris hanya boleh diisi satu peneliti. Supaya data sesuai kebutuhan maka akan ada pemisahan data perbaris menggunakan *lamda expression*.

3.3.2 Pemisahan Data dalam Kolom

Contoh data yang harus dipisahkan dalam baris dapat dilihat pada Gambar 3.9 berikut ini.

Gambar 3.9 Tampilan data di laman web Prestasi Undip

Pada Gambar 3.9 menunjukkan data waktu yang lengkap mulai dari tanggal, bulan, dan tahun sedangkan data yang dibutuhkan hanyalah tahunnya saja. Oleh karena itu, diperlukan pemisahan data dalam kolom menggunakan *method* `.str.split()`.

3.3.3 Penghapusan Data

No.	Kode	Nama Program Studi	Status	Jenjang	Data Pelaporan Tahun 2018/2019			Data Pelaporan Tahun 2019/2020		
					Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa	Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa
1	63001	Administrasi Publik	Aktif	S3	6	121	1:20.2	6	129	1:21.5
2	60001	Ekonomi	Aktif	S3	6	266	1:44.3	6	296	1:49.3
3	74001	Hukum	Aktif	S3	18	158	1:8.8	18	177	1:9.8
4	23001	Ilmu Arsitektur Dan Perkotaan	Aktif	S3	5	55	1:11	5	62	1:12.4

Gambar 3.10 Tampilan data di laman web Forlap Dikti

Pada Gambar 3.10 menunjukkan data program studi dan jenjang yang lengkap sampai S3 sedangkan data yang dibutuhkan hanyalah jenjang S1 dalam Fakultas Teknik. Maka, akan ada penghapusan beberapa data berdasarkan kondisi selain jenjang S1 dan program setudi dilingkup Fakultas Teknik.

BAB IV

HASIL DAN PEMBAHASAN

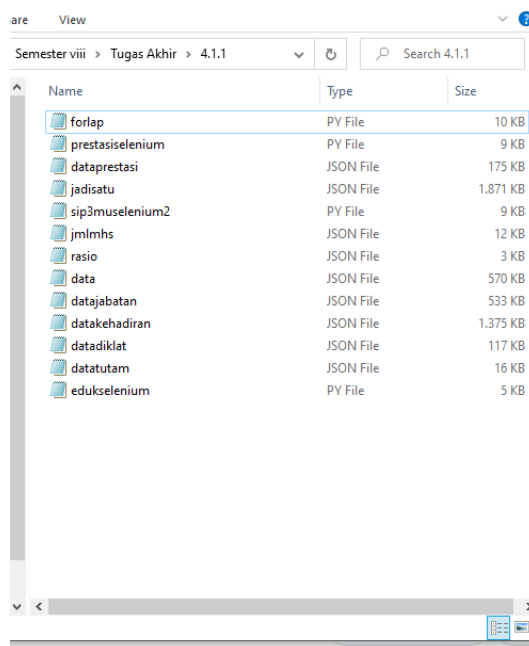
4.1 Pengujian *Blackbox*

Pengujian *blackbox* ini bertujuan untuk mengetahui perolehan data akhir yang siap untuk dimasukkan ke dalam *database*. Pengujian dinyatakan sukses ketika data dalam bentuk JSON berhasil ditambahkan dalam suatu folder bersamaan dengan berkas program berekstensi .py. Hasil pengujian *blackbox* direpresentasikan dalam Tabel 4.1 berikut.

Tabel 4.1 Hasil pengujian *blackbox*

Laman Web	Detail Pengujian	Jenis Pengujian	Data (Kb)
Eduk Undip Undip	<i>Data Crawling & Filtering</i>	<i>Blackbox</i>	2663,7
Sip3mu Undip LPPM Undip	<i>Data Crawling & Filtering</i>	<i>Blackbox</i>	1900
Prestasi Undip	<i>Data Crawling & Filtering</i>	<i>Blackbox</i>	178,5
Forlap Dikti	<i>Data Crawling & Filtering</i>	<i>Blackbox</i>	14,5

Dapat dilihat pada Tabel 4.1 bahwa seluruh program berhasil dijalankan dengan adanya besaran berkas yang diperoleh. Data paling besar berasal dari laman web Eduk Undip, yakni sebesar 2663,7 Kb. Data paling kecil berasal dari laman web Forlap Dikti, yakni sebesar 14,5 Kb. Data yang diperoleh dari laman web Sip3mu Undip sebesar 1900 Kb dan data yang diperoleh dari laman web Prestasi Undip sebesar 178,5 Kb. Besaran data secara rinci dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Tampilan penyimpanan berkas program dan data

Pada Gambar 4.1 dapat dilihat bahwa terdapat 4 macam berkas berekstensi .py dan 9 berkas berekstensi .json yang siap dimasukkan ke dalam basis data. Program dengan nama berkas edukselenium.py menghasilkan 5 berkas dengan nama datatutam.json, datakehadiran.json, datajabatan.json, datadiklat.json, dan data.json. Program dengan nama berkas sip3muselenium2.py menghasilkan 1 berkas dengan nama jadisatu.json. Program dengan nama berkas prestasiselenium.py menghasilkan 1 berkas dengan nama Prestasi Undip.json. Program dengan nama berkas forlap.py menghasilkan 2 berkas dengan nama rasio.json dan jmlmhs.json. Berkas berekstensi .json tersebut merupakan keluaran dari penelitian ini yang siap dimasukkan ke dalam *database*.

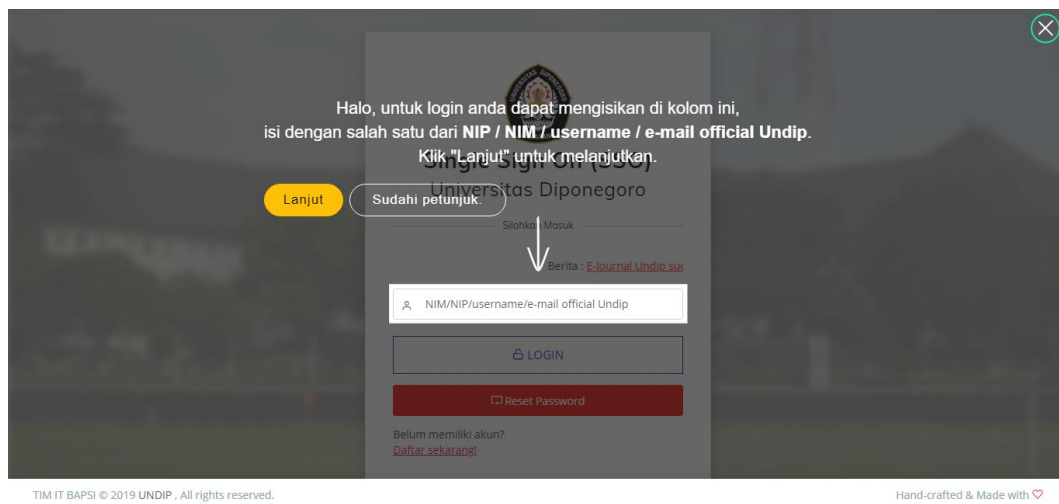
4.2 Sistem Crawling Data

Sistem *crawling* data pada Tugas Akhir ini berbasis Selenium yang bertujuan untuk mempermudah proses *login* ke dalam pangkalan *database* yang hanya bisa diakses oleh pihak-pihak tertentu, dimana proses tersebut dilakukan secara otomatis. Berikut pembahasan mengenai proses *crawling* data disetiap laman web.

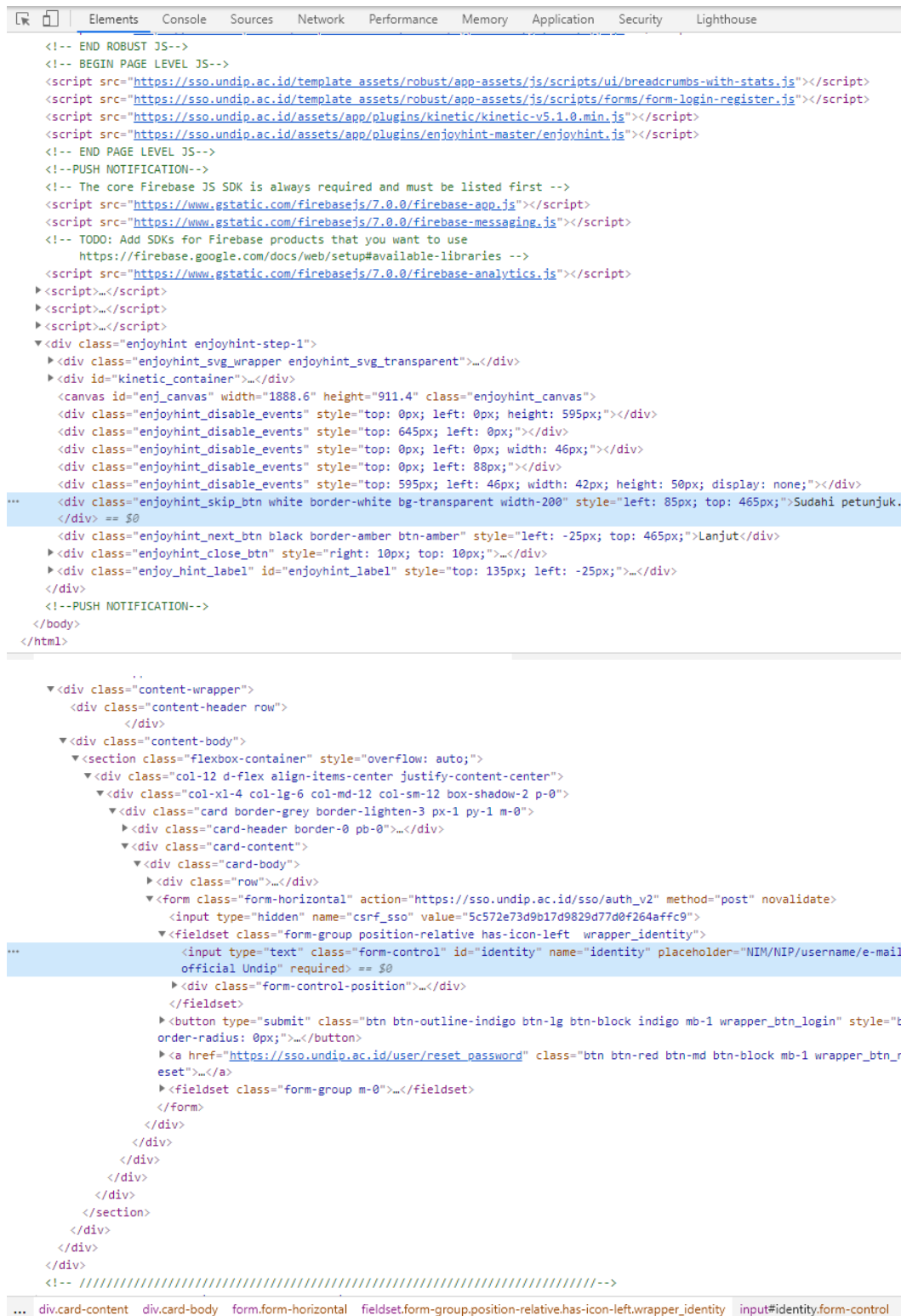
4.2.1 Sistem *Crawling* Data di Laman Web Eduk Undip

Laman Web Eduk Undip menyimpan data-data mengenai dosen ataupun karyawan yang bekerja di Universitas Diponegoro. Data dapat diakses dengan masuk menggunakan akun SSO (*Single Sign On*). Hanya akun para dosen dan karyawan yang dapat mengakses fasilitas laman web Eduk Undip.

Berikut proses *crawling* data di laman web SSO untuk *submit* nama pengguna yang diperjelas dengan *interface* pada Gambar 4.2 dan *inspect element* pada Gambar 4.3.



Gambar 4.2 Tampilan halaman web SSO untuk *submit* nama pengguna



```

<!-- END ROBUST JS-->
<!-- BEGIN PAGE LEVEL JS-->
<script src="https://sso.undip.ac.id/template/assets/robust/app-assets/js/scripts/ui/breadcrumbs-with-stats.js"></script>
<script src="https://sso.undip.ac.id/template/assets/robust/app-assets/js/scripts/forms/form-login-register.js"></script>
<script src="https://sso.undip.ac.id/assets/app/plugins/kinetic/kinetic-v5.1.0.min.js"></script>
<script src="https://sso.undip.ac.id/assets/app/plugins/enjoyhint-master/enjoyhint.js"></script>
<!-- END PAGE LEVEL JS-->
<!-- PUSH NOTIFICATION-->
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.0.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.0.0/firebase-messaging.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.0.0/firebase-analytics.js"></script>
<script>...</script>
<script>...</script>
<script>...</script>
<div class="enjoyhint enjoyhint-step-1">
  <div class="enjoyhint_svg_wrapper enjoyhint_svg_transparent">...</div>
  <div id="kinetic_container">...</div>
  <canvas id="enj_canvas" width="1888.6" height="911.4" class="enjoyhint_canvas">
    <div class="enjoyhint_disable_events" style="top: 0px; left: 0px; height: 595px;"></div>
    <div class="enjoyhint_disable_events" style="top: 645px; left: 0px;"></div>
    <div class="enjoyhint_disable_events" style="top: 0px; left: 0px; width: 46px;"></div>
    <div class="enjoyhint_disable_events" style="top: 0px; left: 88px;"></div>
    <div class="enjoyhint_disable_events" style="top: 595px; left: 46px; width: 42px; height: 50px; display: none;"></div>
  <div class="enjoyhint_skip_btn white border-white bg-transparent width-200" style="left: 85px; top: 465px;">Sudahi petunjuk.
  </div>
  <div class="enjoyhint_next_btn black border-amber btn-amber" style="left: -25px; top: 465px;">Lanjut</div>
  <div class="enjoyhint_close_btn" style="right: 10px; top: 10px;">...</div>
  <div class="enjoy_hint_label" id="enjoyhint_label" style="top: 135px; left: -25px;">...</div>
</div>
<!-- PUSH NOTIFICATION-->
</body>
</html>

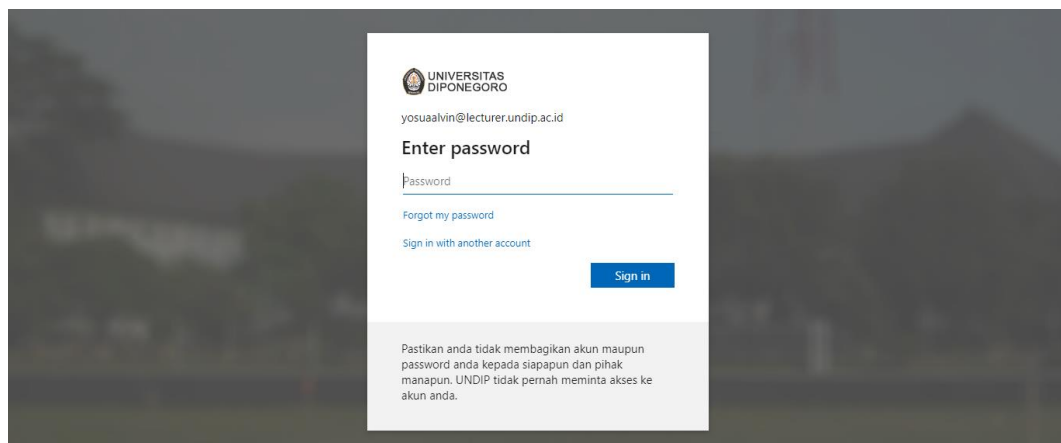
...
<div class="content-wrapper">
  <div class="content-header row">
    </div>
  <div class="content-body">
    <section class="flexbox-container" style="overflow: auto;">
      <div class="col-12 d-flex align-items-center justify-content-center">
        <div class="col-xl-4 col-lg-6 col-md-12 col-sm-12 box-shadow-2 p-0">
          <div class="card border-grey border-lighten-3 px-1 py-1 m-0">
            <div class="card-header border-0 pb-0">...</div>
            <div class="card-content">
              <div class="card-body">
                <div class="row">...</div>
                <form class="form-horizontal" action="https://sso.undip.ac.id/sso/auth_v2" method="post" novalidate>
                  <input type="hidden" name="csrf_sso" value="5c572e73d9b17d9829d77d0f264affc9">
                  <fieldset class="form-group position-relative has-icon-left wrapper_identity">
                    <input type="text" class="form-control" id="identity" name="identity" placeholder="NIM/NIP/username/e-mail
                    official Undip required" == $0
                    <div class="form-control-position">...</div>
                  </fieldset>
                  <button type="submit" class="btn btn-outline-indigo btn-lg btn-block indigo mb-1 wrapper_btn_login" style="b
                    order-radius: 0px;">...</button>
                  <a href="https://sso.undip.ac.id/user/reset_password" class="btn btn-red btn-md btn-block mb-1 wrapper_btn_r
                    eset">...</a>
                  <fieldset class="form-group m-0">...</fieldset>
                </form>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>
  </div>
</div>
<!-- ////////////////////////////////////////////////////-->
... div.card-content div.card-body form.form-horizontal fieldset.form-group.position-relative.has-icon-left.wrapper_identity input#identity.form-control

```

Gambar 4.3 *Inspect element* halaman web SSO untuk *submit* nama pengguna

Saat peramban berjalan pertama kali akan menampilkan hasil request URL yang dapat dilihat pada Gambar 4.2, dimana tombol ‘Sudahi Petunjuk’ harus ditekan terlebih dahulu sebelum memasukkan nama pengguna. Tombol ‘Sudahi Petunjuk’ ditekan berdasarkan elemen dari `class="enjoyhint_skip_btn"` yang dapat dilihat pada Gambar 4.3. Kemudian, memasukkan nama pengguna berdasarkan elemen dengan `id="identity"` yang dapat dilihat pada Gambar 4.3, lalu mengirim tulisan nama penggunaan menggunakan `method .send_keys('username')` dan `.submit()` sebagai tombol *enter*.

Berikut proses *crawling* data di laman web SSO untuk *submit* kata sandi yang diperjelas dengan *interface* pada Gambar 4.4 dan *inspect element* pada Gambar 4.5.



Gambar 4.4 Tampilan halaman web SSO untuk *submit* kata sandi



```

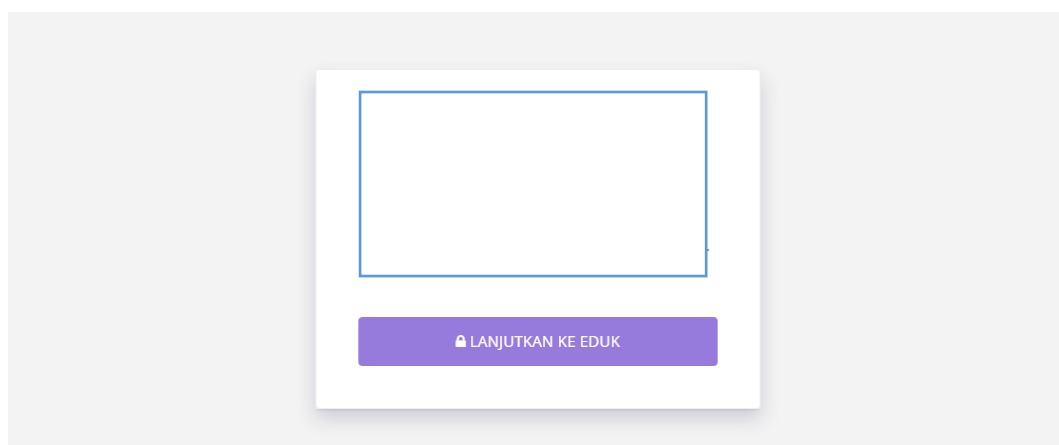
<div role="alert" aria-live="assertive">...</div>
  <div class="placeholderContainer" data-bind="component: { name: 'placeholder-textbox-field',
    publicMethods: passwordTextbox.placeholderTextboxMethods,
    params: {
      serverData: svr,
      hintText: str['CT_PWD_STR_PwdTB_Label'] },
    event: {
      updateFocus: passwordTextbox.textbox_onUpdateFocus } }">
    <!-- ko withProperties: { '$placeholderText': placeholderText } -->
    <!-- ko template: { nodes: $componentTemplateNodes, data: $parent } -->
    ...
    <input name="passwd" type="password" id="i0118" autocomplete="off" class="form-control input ext-input
    text-box ext-text-box" aria-required="true" data-bind="
      textInput: passwordTextbox.value,
      ariaDescribedBy: [
        'loginHeader',
        showCredViewBrandingDesc ? 'credViewBrandingDesc' : '',
        unsafe_pageDescription ? 'passwordDesc' : ''].join(' '),
      hasFocusEx: passwordTextbox.focused() && !showPassword(),
      placeholder: $placeholderText,
      ariaLabel: unsafe_passwordAriaLabel,
      moveOffScreen: showPassword,
      externalCss: {
        'input': true,
        'text-box': true,
        'has-error': passwordTextbox.error }" aria-describedby="loginHeader "
      placeholder="Password" aria-label="Enter the password for yosuaalvin@lecturer.undip.ac.id" tabindex=
      "0"> == $0
    <!-- ko if: svr.fUsePasswordPeek && showPassword() -->

```

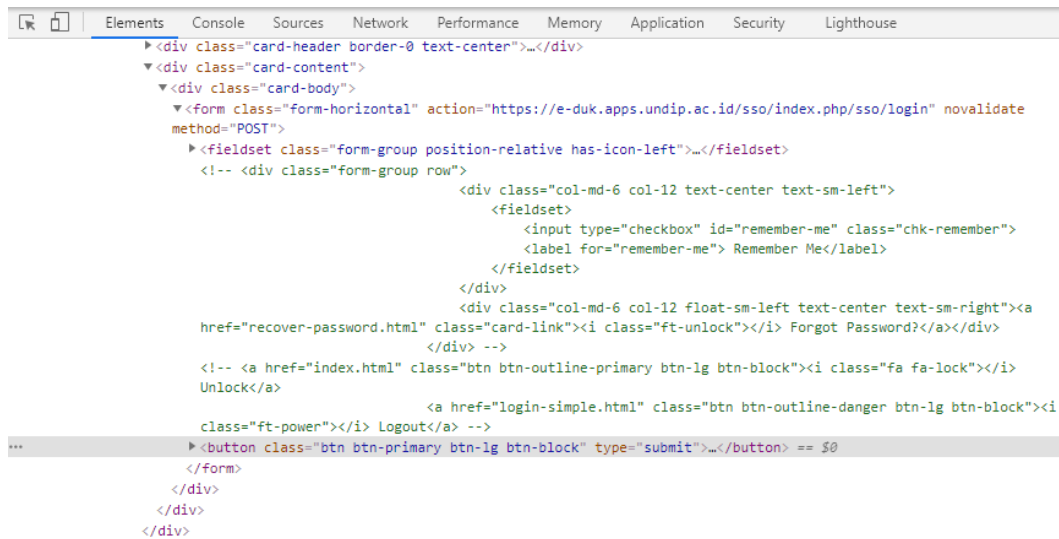
Gambar 4.5 *Inspect element* halaman web SSO untuk *submit* kata sandi

Setelah *submit* nama pengguna peramban akan lanjut ke halaman *submit* kata sandi seperti pada Gambar 4.4. Kemudian, program akan memasukkan sandi berdasarkan elemen dengan nama="passwd" dalam tag `<input>` yang dapat dilihat pada Gambar 4.5, lalu mengirim tulisan kata sandi menggunakan *method* `.send_keys('password')` dan `.submit()` sebagai tombol *enter*.

Berikut proses *crawling* data di laman web SSO untuk lanjut ke laman web Eduk Undip yang diperjelas dengan *interface* pada Gambar 4.6 dan *inspect element* pada Gambar 4.7



Gambar 4.6 Tampilan halaman web SSO untuk lanjut ke halaman web Eduk Undip



Gambar 4.7 *Inspect element* halaman web SSO untuk lanjut ke laman web Eduk Undip

Setelah *submit* kata sandi peramban akan lanjut ke halaman Eduk Undip yang memerlukan verifikasi seperti pada Gambar 4.6. Sebagai verifikasi program akan menekan tombol ‘LANJUTKAN KE EDUK UNDIP’ berdasarkan elemen dengan `class="btn"` dalam tag `<button` yang dapat dilihat pada Gambar 4.7, lalu menggunakan `method .submit()` sebagai tombol *enter*. Kemudian, peramban akan langsung menuju laman pangkalan basis data yang berbentuk JSON dan melakukan pengunduhan data.

4.2.2 Sistem *Crawling* Data di Laman Web Sip3mu Undip

Laman Web Sip3mu Undip menyimpan data-data mengenai penelitian, publikasi, dan pengabdian masyarakat para dosen di Universitas Diponegoro. Data dapat diakses dengan masuk menggunakan akun seorang admin di tingkat Fakultas. Ada dua program terpisah yang akan dijalankan pada laman web Sip3mu Undip, yakni program untuk *submit* akun admin dan program pengunduhan data.

Berikut proses *crawling* data pada program pertama di laman web Sip3mu Undip untuk *submit* akun admin yang diperjelas dengan *interface* pada Gambar 4.8 dan *inspect element* pada Gambar 4.9.

Gambar 4.8 Tampilan halaman web Sip3mu Undip untuk *submit* akun admin

Gambar 4.9 *Inspect element* halaman web Sip3mu Undip untuk *submit* akun admin

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 4.8, dimana *input* nama pengguna dan kata sandi menjadi satu halaman serta ada tambahan *input captcha*. Proses pemasukan *captcha* belum bisa dilakukan secara otomatis, berdasarkan Gambar 4.9 dapat dilihat pada

elemen dengan `name="captcha"` dalam tag `<input` menjelaskan bahwa kumpulan *captcha* memiliki halaman web nya tersendiri dan kata-kata akan muncul secara acak. Oleh karena itu, *submit* akun admin dilakukan secara manual. Kemudian, mengaktifkan *add-on* Staying Alive yang berguna untuk menjaga *session* untuk tetap aktif saat menjalankan program selanjutnya, lalu menutup peramban.

Berikut proses *crawling* data pada program kedua di laman web Sip3mu Undip untuk mengunduh data yang diperjelas dengan *interface* pada Gambar 4.10 dan *inspect element* pada Gambar 4.11.

Sistem Informasi Penelitian, Publikasi, dan Pengabdian Kepada Masyarakat Universitas Diponegoro

opt_165

HALAMAN UTAMA > Tambah Data Baru Excel 2020 > Semua Data > Semua Departemen > Lihat

SDM PT > Data Penelitian Tahun 2013-2016 Tidak Wajib Mengunggah Bukti Dukung

PENELITIAN

PENGABDIAN

LUARAN

KERJASAMA

UNIT BISNIS

ROYALTI

Data Penelitian Fakultas Teknik Tahun 2020

Download Excel 2020

Show 10 entries Search:

No	Penelitian	Peneliti	Status	Aksi
1	Palm Oil Cracking to Renewable Liquid Fuels over Plasma-Assisted Catalytic Continuous Reactor to Supply National Energy Demand Jenis : Dasar (TRL 1-3) Bidang : ILMU KETEKNIKAN INDUSTRI Berkas : 6593_penelitian(1).pdf	1. Prof. Dr. Istadi, S.T., M.T. 2. Dr. Luqman Buchori, ST, MT 3. Prof. Ir. Didi Dwi Anggoro, M.Eng, Ph.D	Data Lengkap : Ya Validasi Fakultas : Tidak Tgl Validasi : 30-05-2018 Validasi LPPM : Tidak Tgl Validasi : 30-05-2018	...
2	Development of Low Fouling Ultrafiltration Membranes by Reactive Phase Separation Jenis : Dasar (TRL 1-3) Bidang : Lainnya Berkas : 1005_0029057502_penelitian.pdf	1. Prof. Dr. rer.nat. Heru Susanto, ST, MM, MT	Data Lengkap : Ya Validasi Fakultas : Tidak Tgl Validasi : 27-02-2020 Validasi LPPM : Tidak Tgl Validasi : 27-02-2020	...
Inovasi Teknologi Ekstruder Panas Dua Tahap Untuk				

Gambar 4.10 Tampilan halaman web Sip3mu Undip untuk mengunduh



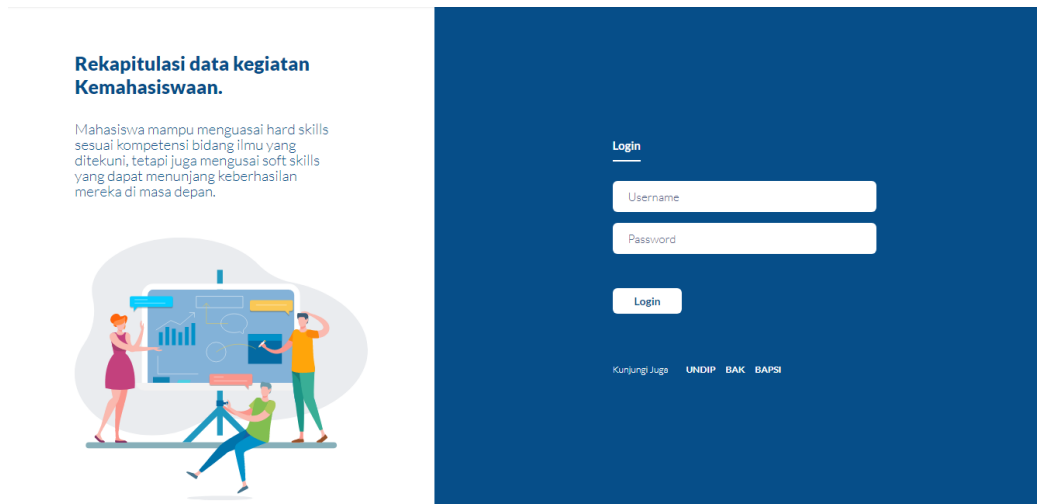
Gambar 4.11 *Inspect element* halaman web Sip3mu Undip untuk mengunduh data

Setelah menutup peramban dilanjutkan dengan menjalankan program kedua dan saat peramban berjalan akan langsung menuju halaman pengunduhan data seperti pada Gambar 4.10. Pengunduhan akan dilakukan berdasarkan tahun penelitian. Seperti pada Gambar 4.11 dapat dilihat bahwa elemen nama="tahun" di dalam tag `<select` ada 8 jenis tahun yang terdata. Artinya, akan ada 8 berkas yang akan terunduh dalam bentuk HTML. Proses pengunduhan dilakukan dengan memasukkan elemen nama="tahun" ke dalam perulangan, kemudian akan ditekan tombol 'Lihat' menggunakan `method .click()` dengan elemen nama="submit" di dalam tag `<input`, selanjutnya akan diklik tulisan 'Download Excel +tahun+' menggunakan `method .click()` dengan elemen `class="btn"` di dalam tag `<button` dan berkas akan otomatis terunduh.

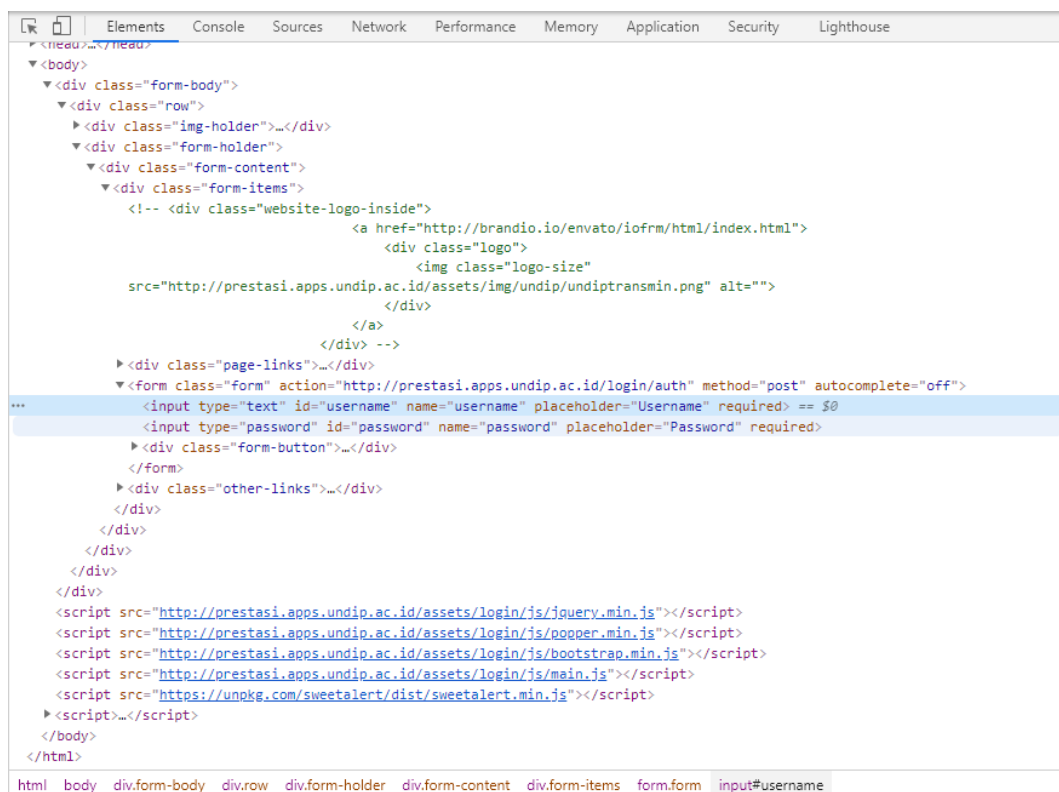
4.2.3 Sistem Crawling Data di Laman Web Prestasi Undip

Laman web Prestasi Undip menyimpan data-data mengenai kegiatan lomba para mahasiswa di Universitas Diponegoro. Data dapat diakses dengan masuk menggunakan akun seorang admin di tingkat fakultas. Pengambilan data terbagi menjadi 2 proses, yakni pengumpulan *link* dan pengambilan data berdasarkan *link* tersebut.

Berikut proses *crawling* data pada di laman web Prestasi Undip untuk *submit* akun admin yang diperjelas dengan *interface* pada Gambar 4.12 dan *inspect element* pada Gambar 4.13.



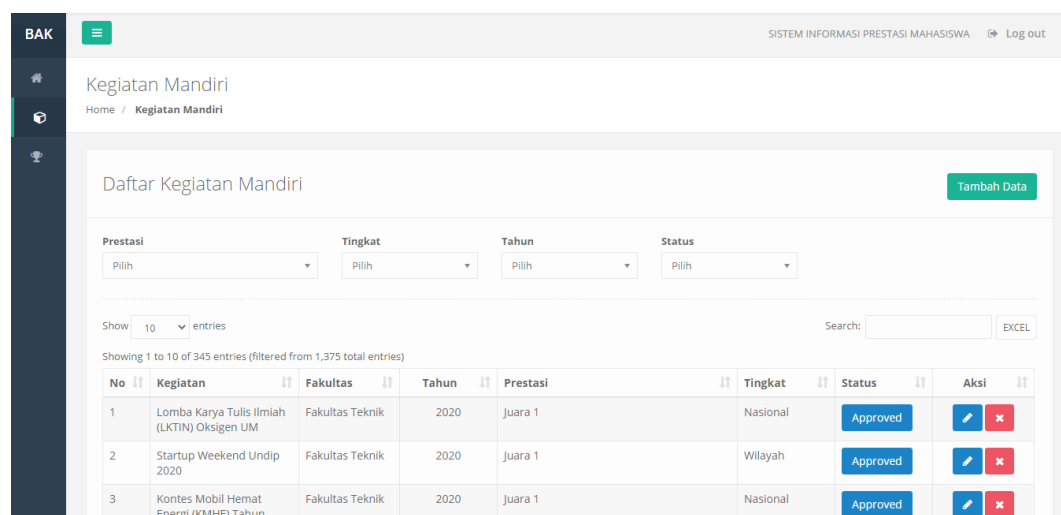
Gambar 4.12 Tampilan halaman web Prestasi Undip untuk *submit* akun admin



Gambar 4.13 *Inspect element* halaman web Prestasi Undip untuk *submit* akun admin

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 4.12, dimana *submit* nama pengguna dan kata sandi menjadi satu halaman. Proses *submit* nama pengguna menggunakan *method* `.send_keys('teks')` dan `.submit()` sebagai *enter* dengan elemen `id="username"` di dalam *tag* `<input` yang dapat dilihat pada Gambar 4.13. Proses *submit* kata sandi menggunakan *method* `.send_keys()` dan `.submit()` sebagai *enter* dengan elemen `id="password"` di dalam *tag* `<input` yang dapat dilihat pada Gambar 4.13.

Berikut penjelasan proses *crawling* data di laman web Prestasi Undip untuk pengumpulan link yang diperjelas dengan *interface* pada Gambar 4.14 dan *inspect element* pada Gambar 4.15.



The screenshot shows a web application interface for 'Prestasi Undip'. The header includes 'BAK' and 'SISTEM INFORMASI PRESTASI MAHASISWA' with a 'Log out' link. The main content area is titled 'Kegiatan Mandiri' and 'Daftar Kegiatan Mandiri'. It features a table with columns: No, Kegiatan, Fakultas, Tahun, Prestasi, Tingkat, Status, and Aksi. The table lists three activities, all with a status of 'Approved'.

No	Kegiatan	Fakultas	Tahun	Prestasi	Tingkat	Status	Aksi
1	Lomba Karya Tulis Ilmiah (LKTI) Oksigen UM	Fakultas Teknik	2020	Juara 1	Nasional	Approved	Edit Delete
2	Startup Weekend Undip 2020	Fakultas Teknik	2020	Juara 1	Wilayah	Approved	Edit Delete
3	Kontes Mobil Hemat Energi (KMHE) Tahun	Fakultas Teknik	2020	Juara 1	Nasional	Approved	Edit Delete

Gambar 4.14 Tampilan halaman web Prestasi Undip Undip untuk pengumpulan *link*



```

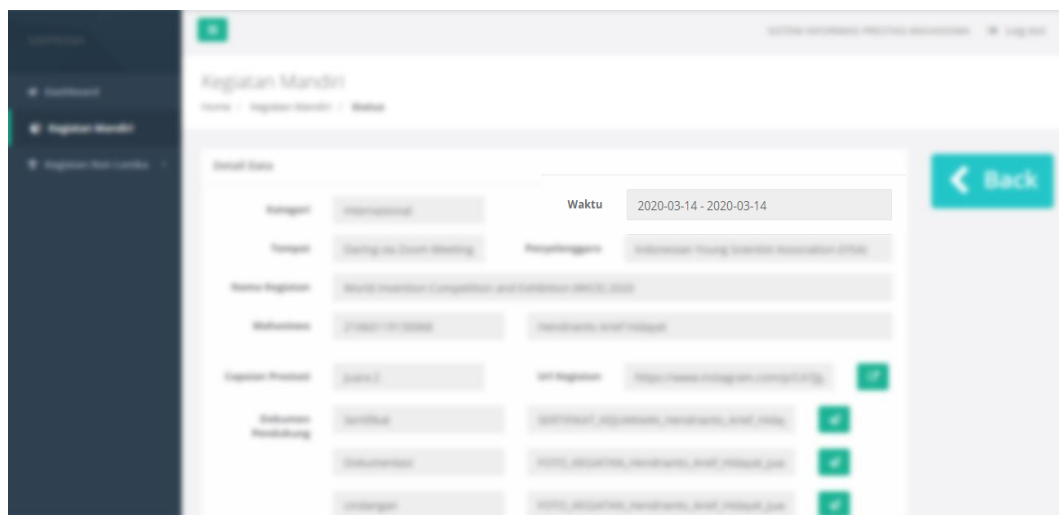
<div class="form-group col-md-2">
  <label>Status</label>
  <select name="a" class="form-control chosen" id="stats" data-placeholder="Pilih" tabindex="-1" style="display: none;"></select>
  <div class="chosen-container chosen-container-single" style="width: 100%;" title id="stats_chosen">
    <a class="chosen-single" tabindex="-1">
      <span>Approved</span>
      <div>...</div>
    </a>
    <div class="chosen-drop">...</div>
  </div>
  ::after
</div>
</form>
<div class="hr-line-dashed"></div>
<div class="table-responsive">
  <div id="DataTables_Table_0_wrapper" class="dataTables_wrapper form-inline dt-bootstrap no-footer">
    <div class="html5buttons">...</div>
    <div class="dataTables_length" id="DataTables_Table_0_length">
      <label>
        "Show "
        <select name="DataTables_Table_0_length" aria-controls="DataTables_Table_0" class="form-control input-m">
          <option value="10">10</option>
          <option value="25">25</option>
          <option value="50">50</option>
          <option value="100">100</option>
        </select>
        " entries"
      </label>
    </div>
    <div id="DataTables_Table_0_filter" class="dataTables_filter">...</div>
    <div class="dataTables_info" id="DataTables_Table_0_info" role="status" aria-live="polite">Showing 1 to 10 of 340 entries (filtered from 1,375 total entries)</div>
    <table class="table table-striped table-bordered table-hover tablex dataTable no-footer" style="font-size: 14px; width: 1184px;" id="DataTables_Table_0" aria-describedby="DataTables_Table_0_info" role="grid">
      <thead>...</thead>
      <tbody>
        <tr role="row" class="odd">
          <td>1</td>
          <td>Lomba Karya Tulis Ilmiah (LKTIN) Oksigen UM</td>
          <td>Fakultas Teknik</td>
          <td class="text-center">2020</td>
          <td>Juara 1</td>
          <td>Nasional</td>
          <td>
            <a href="http://prestasi.apps.undip.ac.id/mandiri/status/1812">
              <span class="btn btn-success">Approved</span>
            </a>
          </td>
        </tr>
        <tr role="row" class="even">...</tr>
        <tr role="row" class="odd">...</tr>
        <tr role="row" class="even">...</tr>
        <tr role="row" class="odd">...</tr>
        <tr role="row" class="even">...</tr>
        <tr role="row" class="odd">...</tr>
        <tr role="row" class="even">...</tr>
        <tr role="row" class="odd">...</tr>
        <tr role="row" class="even">...</tr>
      </tbody>
    </table>
    <div class="dataTables_paginate paging_simple_numbers" id="DataTables_Table_0_paginate">
      <ul class="pagination">
        <li class="paginate_button previous" id="DataTables_Table_0_previous">...</li>
        <li class="paginate_button ">...</li>
        <li class="paginate_button ">...</li>
        <li class="paginate_button ">...</li>
        <li class="paginate_button active">...</li>
        <li class="paginate_button next disabled" id="DataTables_Table_0_next">
          <a href="#" aria-controls="DataTables_Table_0" data-dt-idx="5" tabindex="0">Next</a>
        </li>
      </ul>
    </div>
  </div>
</div>

```

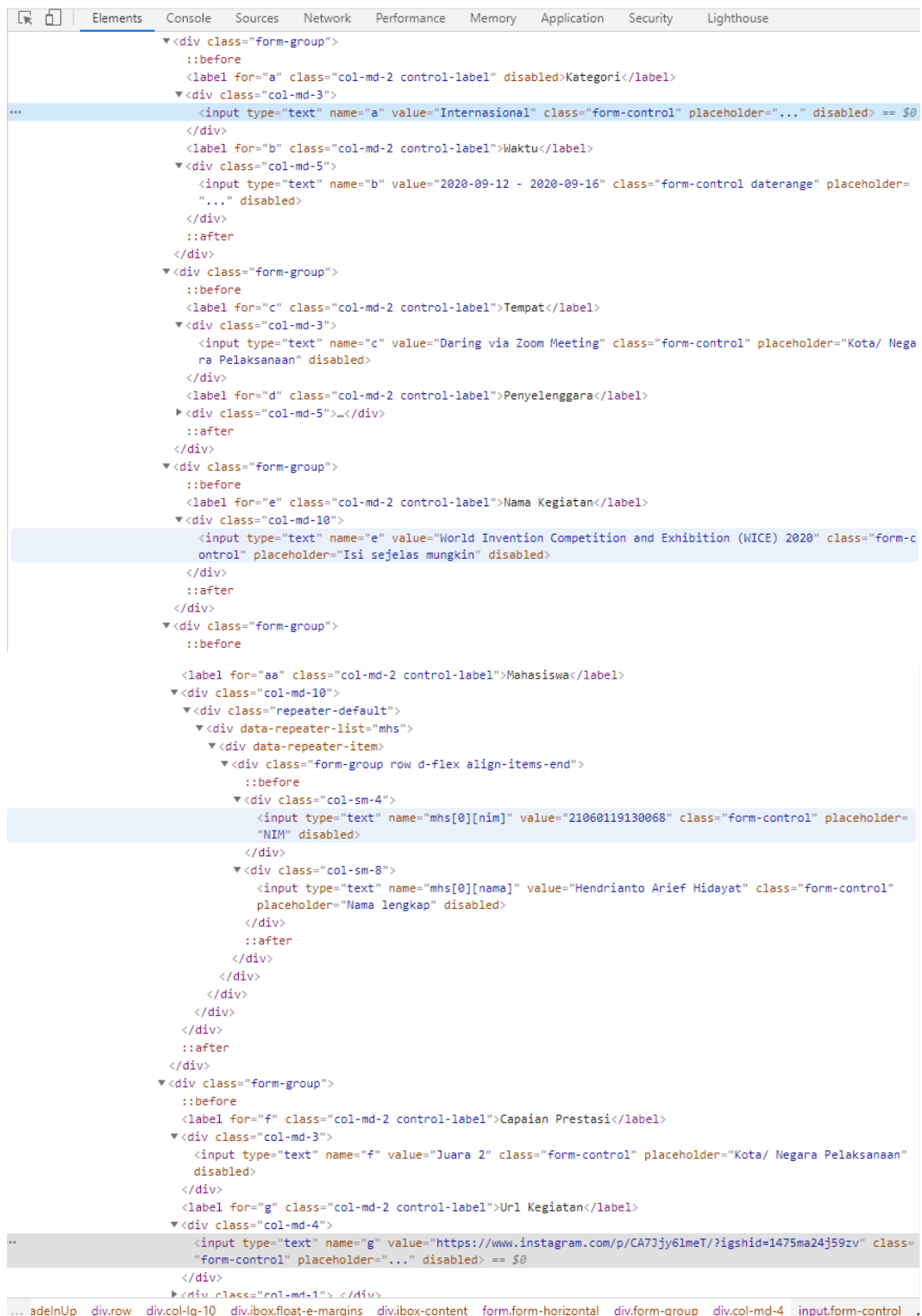
Gambar 4.15 *Inspect element* halaman web Prestasi Undip Undip untuk pengumpulan link

Setelah *submit* akun admin, peramban akan berjalan menuju halaman pangkalan basis data perlombaan yang diikuti para mahasiswa untuk pengumpulan *link* seperti pada Gambar 4.14. Pengumpulan *link* akan dilakukan berdasarkan status data. Seperti pada Gambar 4.15 dapat dilihat bahwa elemen `class="chosen-single"` di dalam *tag* `<a` memiliki teks '*Approved*', pemilihan dilakukan menggunakan *method* `.click()`. Selanjutnya, data akan ditampilkan sebanyak 100 baris per halaman web menggunakan *method* `.click()` dengan memilih atribut `value="100"` dari elemen `name="DataTables_Table_0_length"` di dalam *tag* `<select` untuk mempercepat pengumpulan *link* dan untuk berpindah ke halaman selanjutnya menggunakan *method* `.click()` pada elemen `id="DataTables_Table_0_next"` dalam *tag* `<li` yang dapat dilihat pada Gambar 4.15. Kemudian, *link* akan dikumpulkan ke dalam *list* berdasarkan elemen `id="DataTables_Table_0"` di dalam *tag* `<table` dan memasukkan *tag* `<td>` ke-7 di dalam perulangan untuk mengambil atribut `href` yang dapat dilihat pada Gambar 4.15.

Berikut proses *crawling* data di laman web Prestasi Undip untuk pengambilan data yang diperjelas dengan *interface* pada Gambar 4.16 dan *inspect element* pada Gambar 4.17.



Gambar 4.16 Tampilan halaman web Prestasi Undip Undip untuk pengambilan data



Gambar 4.17 *Inspect element* halaman web Prestasi Undip untuk pengambilan data

Setelah pengumpulan *link* selesai, peramban secara otomatis akan membuka *link* tersebut satu per satu seperti pada Gambar 4.16. Proses pengambilan

data berupa teks dilakukan per detail data menggunakan sintaks `find_elements_by_XPath` dibedakan berdasarkan *tag* `<td>` yang dapat dilihat pada Gambar 4.17 dan dimasukkan ke dalam perulangan. Data yang berhasil diambil akan ditampilkan dalam *list* dan ada 3 list yang digunakan untuk menampung data sesuai dengan jumlah kolom. Artinya, ada 3 perulangan yang akan membaca data perkolom.

4.2.4 Sistem *Crawling* Data di Laman Web Forlap Dikti

Laman web Prestasi Undip menyimpan data-data mengenai kegiatan lomba para mahasiswa di Universitas Diponegoro. Data dapat diakses dengan masuk menggunakan akun seorang admin di tingkat Fakultas. Pengambilan data menghasilkan 2 berkas berbeda, yakni berkas data daftar program studi dan berkas data daftar jumlah mahasiswa.

Berikut proses *crawling* data di laman web Forlap untuk pengambilan data daftar program studi yang diperjelas dengan *interface* pada Gambar 4.18 dan *inspect element* pada Gambar 4.19.

Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa	Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa
1.751	49.425	1 : 28.2	1.751	56.125	1 : 32.1

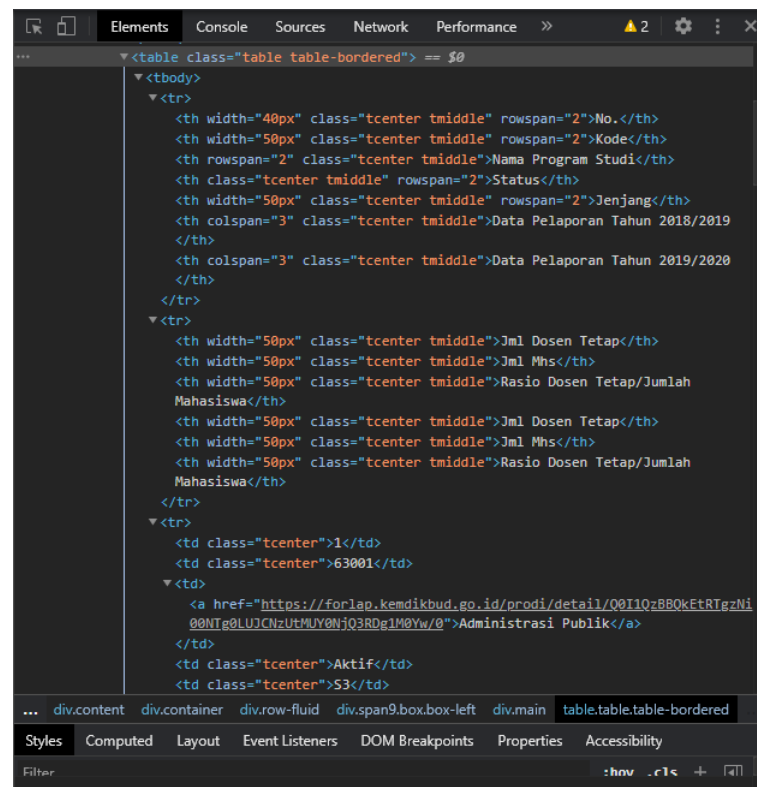
Daftar Program Studi

table.table-bordered	710 × 8254
----------------------	------------

as mahasiswa pada semester ganjil tahun ajaran tersebut. Jika tidak sesuai, laporannya melalui aplikasi PDDikti Feeder

No.	Kode	Nama Program Studi	Status	Jenjang	Data Pelaporan Tahun 2018/2019			Data Pelaporan Tahun 2019/2020		
					Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa	Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa
1	63001	Administrasi Publik	Aktif	S3	6	121	1:20.2	6	129	1:21.5
2	60001	Ekonomi	Aktif	S3	6	266	1:44.3	6	296	1:49.3
3	74001	Hukum	Aktif	S3	18	158	1:8.8	18	177	1:9.8
4	23001	Ilmu Arsitektur Dan Perkotaan	Aktif	S3	5	55	1:11	5	62	1:12.4
5	11001	Ilmu Kedokteran dan Kesehatan	Aktif	S3	5	76	1:15.2	5	104	1:20.8

Gambar 4.18 Tampilan halaman web Forlap Dikti untuk pengambilan data daftar program studi



Gambar 4.19 *Inspect element* halaman web Forlap Dikti untuk pengambilan data daftar program studi

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 4.18 dimana data daftar program studi langsung bisa diakses tanpa harus melakukan *submit* akun. Proses pengambilan data berupa teks dilakukan per kolom menggunakan sintaks `find_element_by_XPath` dibedakan berdasarkan *tag* `<td>` yang dapat dilihat pada Gambar 4.19 dan dimasukkan ke dalam perulangan. Data yang berhasil diambil akan ditampung dalam *list* dan ada 10 list yang digunakan untuk menampung data sesuai dengan jumlah kolom. Artinya, ada 10 perulangan yang akan membaca data perkolom.

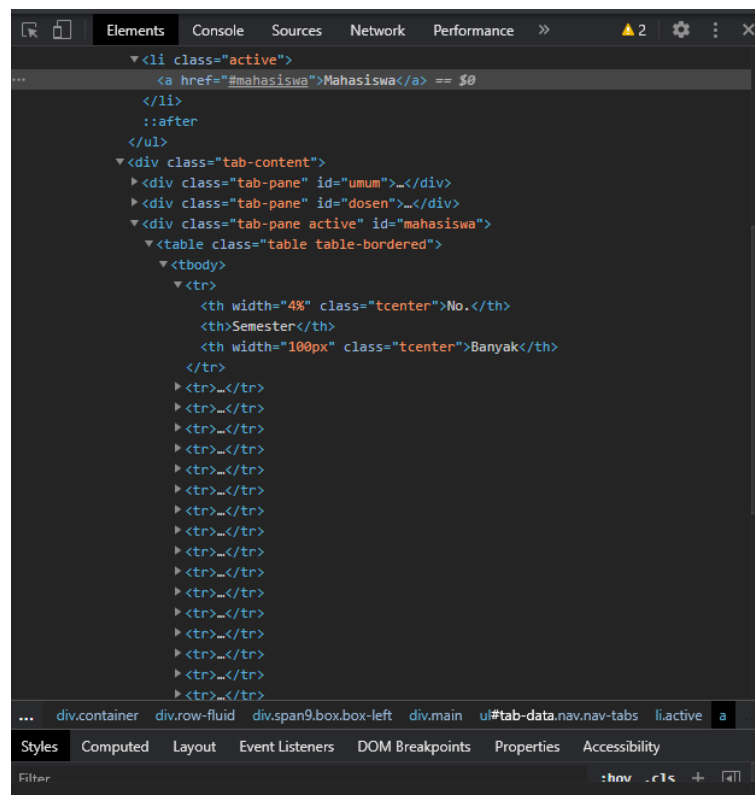
Berikut proses *crawling* data di laman web Forlap Dikti untuk pengambilan data daftar jumlah mahasiswa yang diperjelas dengan *interface* pada Gambar 4.20 dan *inspect element* pada Gambar 4.21.

Profil Program Studi [Kembali ke Hasil Pencarian](#)

Umum Dosen Mahasiswa

No.	Semester	Banyak
1	Genap 2019	120
2	Ganjil 2019	129
3	Genap 2018	110
4	Ganjil 2018	121
5	Genap 2017	112
6	Ganjil 2017	142
7	Genap 2016	121
8	Ganjil 2016	117
9	Genap 2015	94
10	Ganjil 2015	102
11	Genap 2014	87
12	Ganjil 2014	90
13	Genap 2013	88

Gambar 4.20 Tampilan halaman web Forlap Dikti untuk pengambilan data jumlah mahasiswa



Gambar 4.21 *Inspect element* halaman web Forlap Dikti untuk pengambilan data jumlah mahasiswa

Setelah pengambilan data daftar program studi selesai, peramban akan langsung mengumpulkan *link* untuk pengambilan jumlah mahasiswa per program studi. Pengumpulan *link* akan dilakukan dengan memasukkan *tag* `<td` ke-3 yang dapat dilihat pada Gambar 4.19 ke dalam perulangan. *Tag* tersebut memiliki atribut `href` dan nilai dari `href` tersebut akan ditampung dalam *list*. Kemudian, peramban secara otomatis akan membuka *link* tersebut satu per satu seperti pada Gambar 4.20. Proses pengambilan data berupa teks tidak jauh berbeda dengan data daftar program studi sebelumnya, yakni dilakukan per kolom menggunakan elemen dengan `find_elements_by_XPath` dibedakan berdasarkan *tag* `<td` yang dapat dilihat pada Gambar 4.21 dan dimasukkan ke dalam perulangan. Data yang berhasil diambil akan ditampung dalam *list* dan ada 3 *list* yang digunakan untuk menampung data sesuai dengan jumlah kolom. Artinya, ada 3 perulangan yang akan membaca data perkolom.

4.3 Sistem *Filtering* Data

Sistem *filtering* data pada Tugas Akhir ini menggunakan struktur data *dataframe* yang dimiliki oleh librari Pandas. Pembahasan mengenai sistem *filtering* data ini dibagi menjadi 2, yaitu pengujian sistem dan hasil data dari *filtering* data.

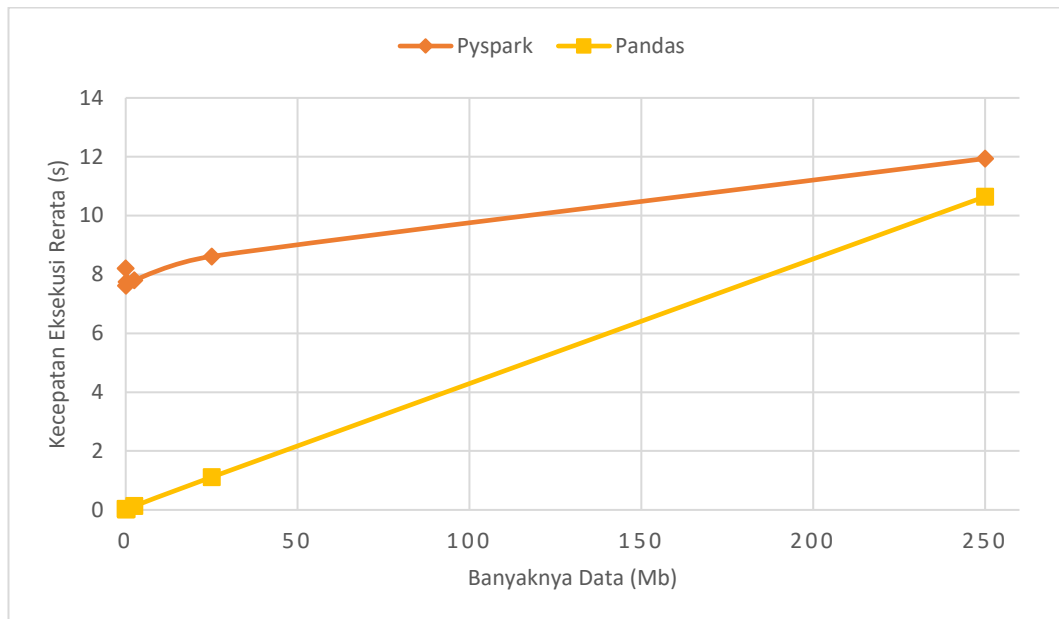
4.3.1 Pengujian Sistem *Filtering* Data

Pengujian dilakukan dengan membandingkan kinerja kecepatan eksekusi dan penggunaan memori oleh Pandas *dataframe* dan Pyspark *dataframe*. Pengujian dilakukan berdasarkan 3 kondisi untuk setiap banyaknya data, yakni saat 1 aplikasi dijalankan, saat 2 aplikasi dijalankan, dan saat 3 aplikasi dijalankan. Berikut hasil rerata pengujian Pandas *dataframe* dan Pyspark *dataframe* yang disajikan dalam Tabel 4.2.

Tabel 4.2 Hasil rerata pengujian Pandas *dataframe* dan Pyspark *dataframe*

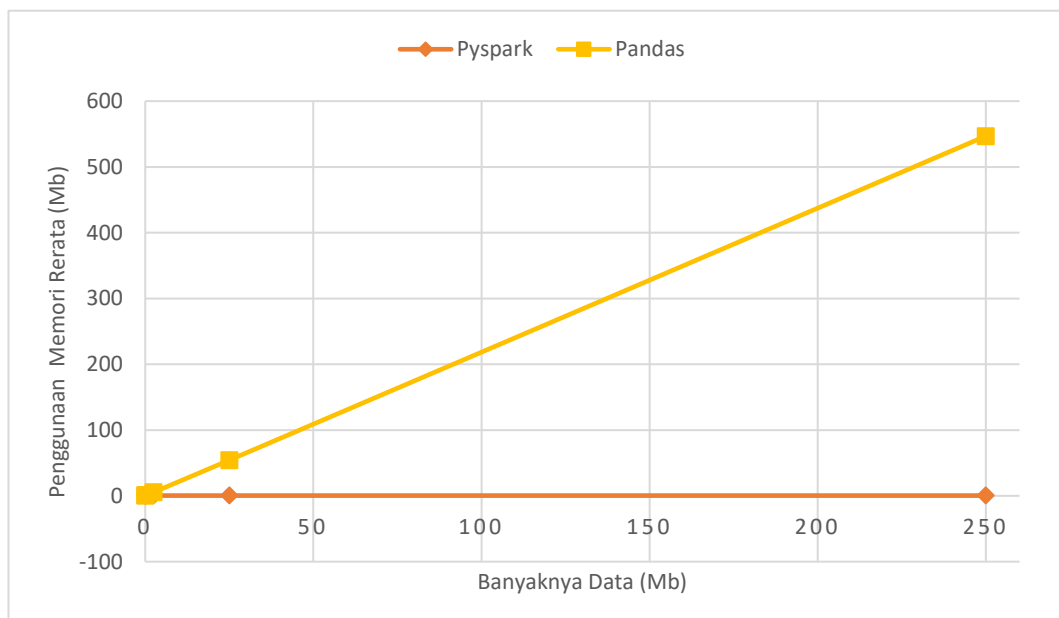
Laman Web	Banyaknya Data (Mb)	Kecepatan Eksekusi (s)		Penggunaan Memori (Mb)	
		Pyspark	Pandas	Pyspark	Pandas
Sip3mu Undip	0,003	8,206324418	0,039816	0,425955	0,339076
	0,025	7,613231817	0,018068	0,425956	0,364903
	0,250	7,748084625	0,026259	0,412361	0,676763
	2,499	7,800643285	0,128549	0,425956	3,663401
	24,996	8,612986644	1,113693	0,425959	33,56355
	249,961	11,93711193	10,64237	0,42596	267,8953
Prestasi Undip	0,004	6,699864229	0,02321	0,412361	0,347042
	0,031	6,726234674	0,024081	0,412362	0,415321
	0,312	6,869834661	0,031958	0,447895	0,883193
	3,129	7,336883624	0,187398	0,412362	5,579908
	31,291	8,035114368	1,740815	0,412365	54,26836
	312,913	10,40634084	17,35626	0,412366	546,7106
Forlap Dikti	0,002	7,400766611	0,024983	0,447902	0,367378
	0,011	7,155849059	0,057595	0,447895	0,381933
	0,110	7,296039184	0,047367	0,447902	0,469856
	1,008	7,526277622	0,110048	0,447895	1,102192
	10,075	8,08164978	0,359805	0,447897	9,434527
	100,739	10,32327882	1,921089	0,44789	85,67912

Dari data yang ada pada Tabel 4.2 dapat dibuat grafik seperti pada Gambar 4.22, Gambar 4.23, Gambar 4.24, Gambar 4.25, Gambar 4.26 , dan Gambar 27 berikut.



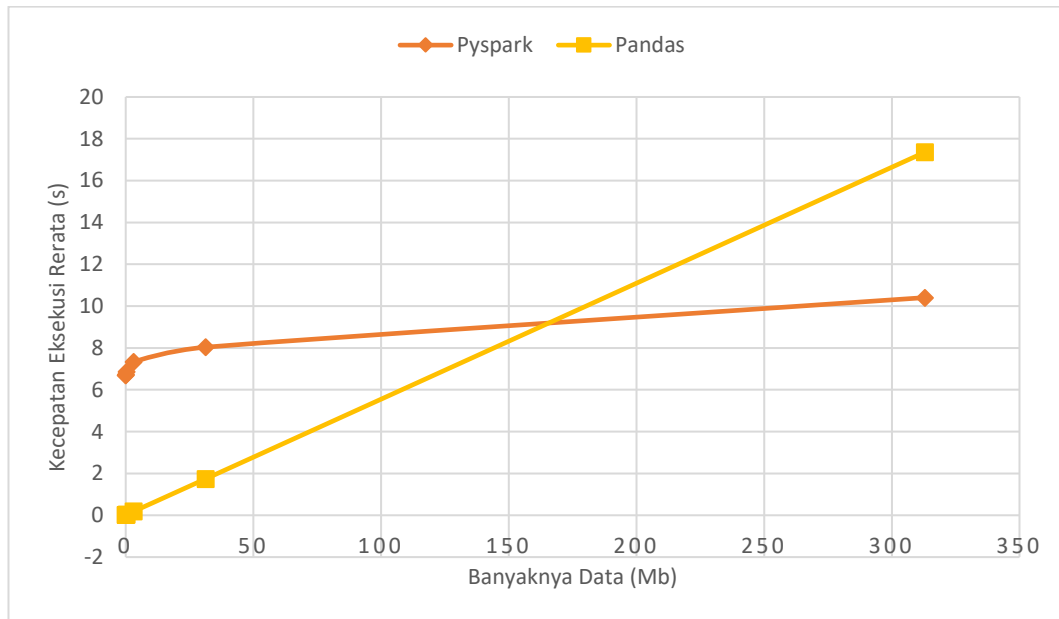
Gambar 4.22 Grafik perbandingan kecepatan eksekusi pada laman web Sip3mu Undip

Pada Gambar 4.22 terlihat bahwa kecepatan eksekusi Pandas *dataframe* lebih cepat dibandingkan dengan Pyspark *dataframe*. Tetapi, kenaikan kecepatan pada Pyspark *dataframe* berkisar 2 detik sedangkan kenaikan kecepatan pada Pandas *dataframe* mencapai 4 detik.



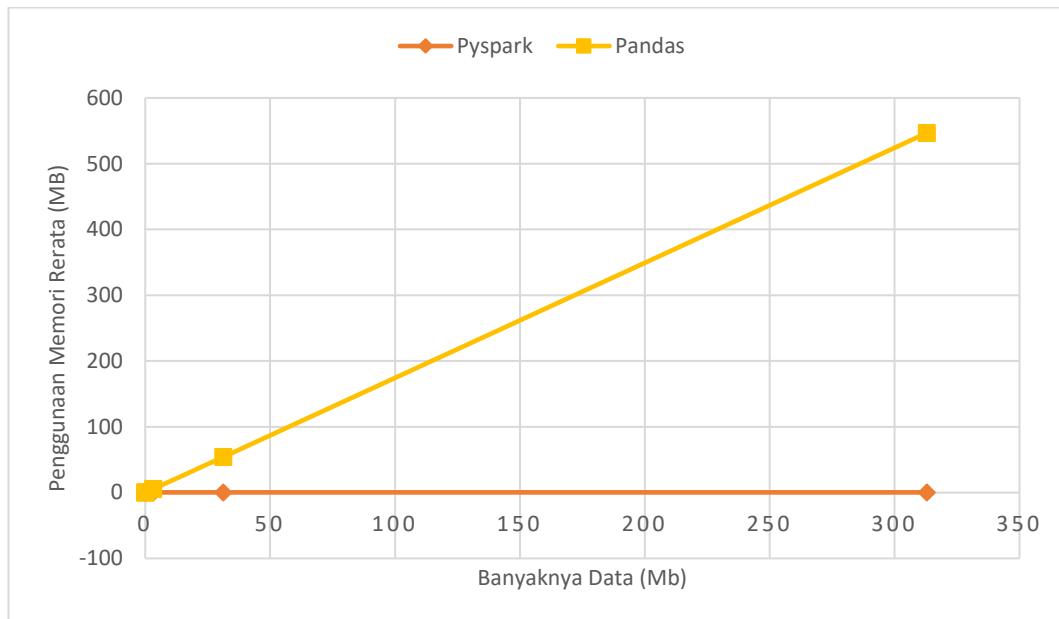
Gambar 4.23 Grafik perbandingan penggunaan memori pada laman web Sip3mu Undip

Pada Gambar 4.23 terlihat bahwa penggunaan memori pada Pyspark *dataframe* cenderung stabil dan bersekala sangat kecil dibandingkan dengan Pandas *dataframe*. Pada Pandas *Dataframe* besarnya penggunaan memori bergerak linear terhadap besarnya data yang di kelola.



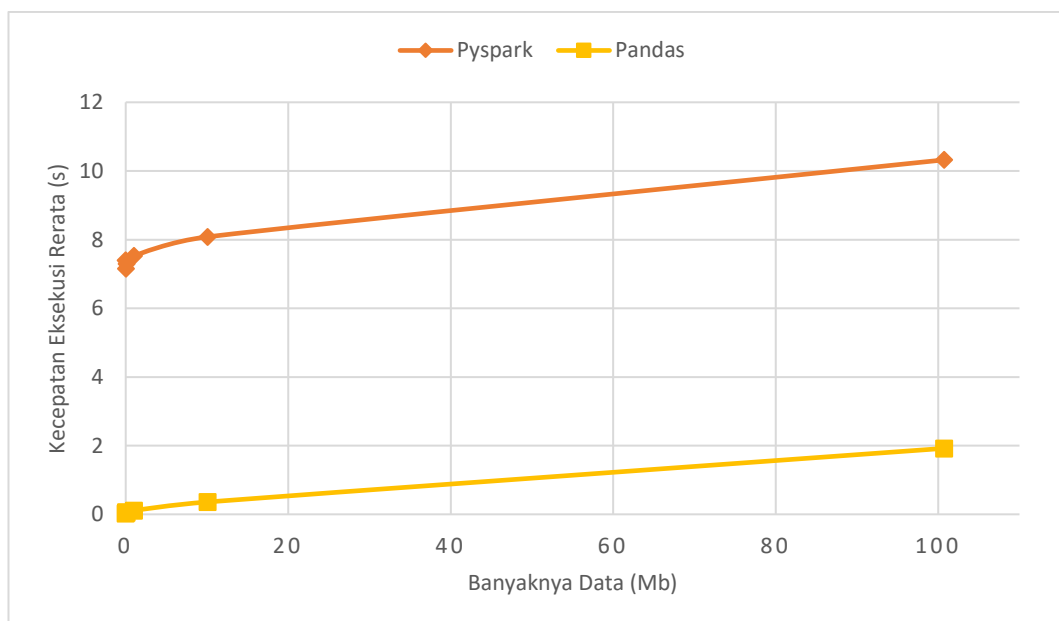
Gambar 4.24 Grafik perbandingan kecepatan eksekusi pada laman web Prestasi Undip

Pada Gambar 4.24 terlihat bahwa kecepatan eksekusi Pandas *dataframe* lebih cepat dibandingkan dengan Pyspark *dataframe* saat besaran data berada di 175 Mb. Tetapi, kenaikan kecepatan pada Pyspark *dataframe* berkisar 4 detik sedangkan kenaikan kecepatan pada Pandas *dataframe* mencapai 18 detik.



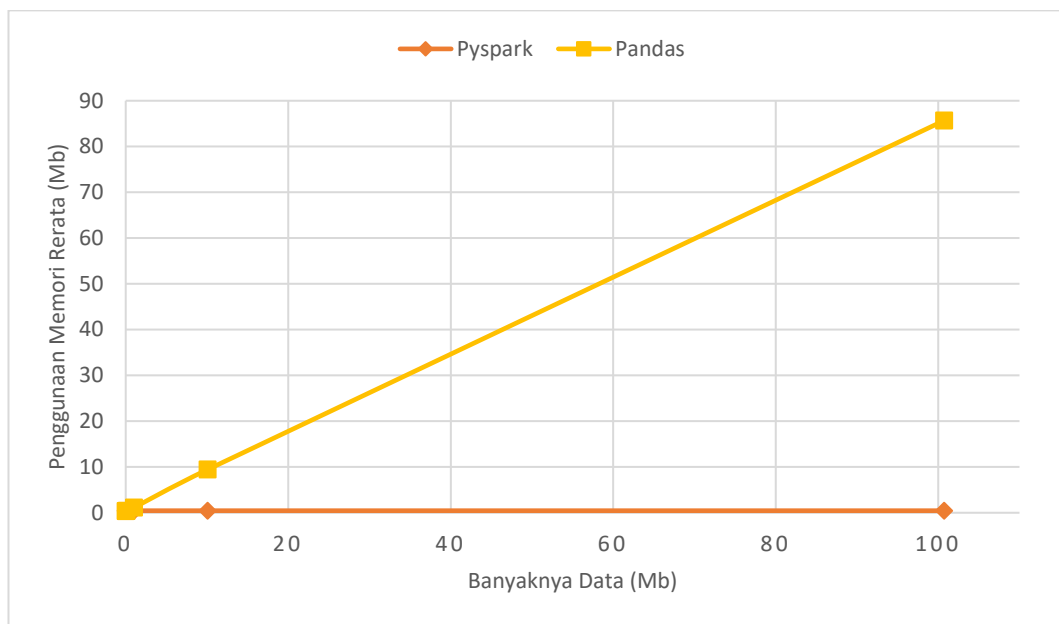
Gambar 4.25 Grafik perbandingan penggunaan memori pada laman web Prestasi Undip

Pada Gambar 4.25 terlihat bahwa penggunaan memori pada Pyspark *dataframe* cenderung stabil dan bersekala sangat kecil dibandingkan dengan Pandas *dataframe*. Pada Pandas *dataframe* besarnya penggunaan memori bergerak linear terhadap besarnya data yang di kelola.



Gambar 4.26 Grafik perbandingan kecepatan eksekusi pada laman web Forlap Dikti

Pada Gambar 4.26 terlihat bahwa kecepatan eksekusi Pandas *dataframe* lebih cepat dibandingkan dengan Pyspark *dataframe*. Kenaikan kecepatan eksekusi cenderung sama pada Pyspark *dataframe* dan Pandas *dataframe*.



Gambar 4.27 Grafik perbandingan penggunaan memori pada laman web Forlap Dikti

Pada Gambar 4.27 terlihat bahwa penggunaan memori pada Pyspark *dataframe* cenderung stabil dan bersekala sangat kecil dibandingkan dengan Pandas *dataframe*. Pada Pandas *dataframe* besarnya penggunaan memori bergerak linear terhadap besarnya data yang di kelola.

4.3.2 Hasil *Filtering* Data

Gambar 4.28 berikut merupakan hasil *filtering* data di laman web Sip3mu Undip.

Out[41]:

	Judul	Peneliti	Sumberdana	Detail_Sumberdana	Bidang_Penelitian	Jenis_Penelitian	Tahun_Pendanaan	Jumlah
0	Model Revitalisasi Pasar Tradisional Indonesia...	[1] [0011087806] Ferry Hermawan, S.T, M.T, P...	Dalam Negeri	Internal Universitas	TEKNIK SIPIL DAN PERENCANAAN TATA RUANG	Terapan (TRL 4-6)	2019	Rp 50.000.000,00

Out[36]: (2089, 8)

Out[44]:

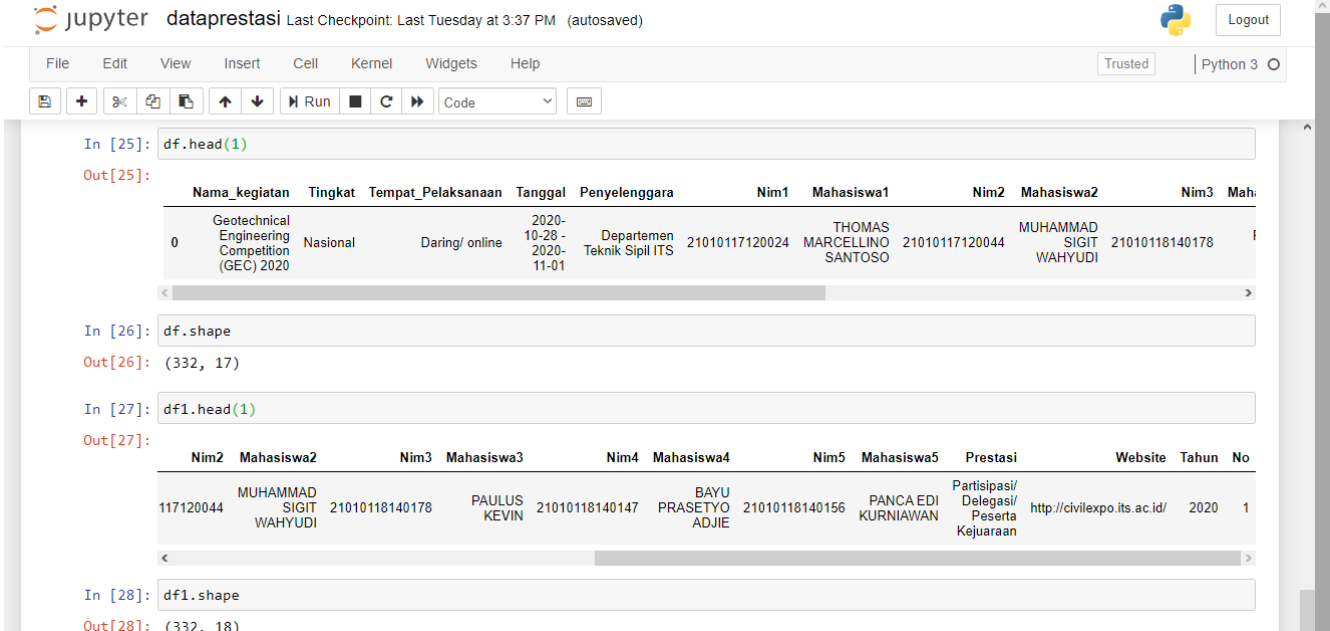
index	Judul	Sumberdana	Detail_Sumberdana	Bidang_Penelitian	Jenis_Penelitian	Tahun_Pendanaan	Jumlah	Peneliti	Nipbaru	Departemen
2	Model Revitalisasi Pasar Tradisional Indonesia...	Dalam Negeri	Internal Universitas	TEKNIK SIPIL DAN PERENCANAAN TATA RUANG	Terapan (TRL 4-6)	2019	Rp 50.000.000,00	Ferry Hermawan, S.T, M.T, Ph.D.	0011087806	Departemen Teknik Sipil Undip

Out[43]: (4042, 11)

Gambar 4.28 Hasil *filtering* data di laman web Sip3mu Undip

Pada Gambar 4.28 dapat dilihat bahwa inisial `df` merupakan data awal dan inisial `df1` merupakan data akhir yang siap disimpan dalam bentuk JSON. Data awal memiliki baris berjumlah 2089 dan kolom berjumlah 8, setelah proses data *filtering* menjadi 4042 dan kolom berjumlah 11. Penambahan baris terjadi karena data pada kolom 'Peneliti' harus dipisahkan untuk peneliti 1, peneliti 2, dan seterusnya. Penambahan kolom terjadi karena pada kolom 'Peneliti' harus dipisahkan antara nama, NIP, dan departemen.

Gambar 4.29 berikut merupakan hasil *filtering* data di laman web Prestasi Undip.



In [25]: `df.head(1)`

Out[25]:

	Nama_kegiatan	Tingkat	Tempat_Pelaksanaan	Tanggal	Penyelenggara	Nim1	Mahasiswa1	Nim2	Mahasiswa2	Nim3	Mahasiswa3
0	Geotechnical Engineering Competition (GEC) 2020	Nasional	Daring/ online	2020-10-28 - 2020-11-01	Departemen Teknik Sipil ITS	21010117120024	THOMAS MARCELLINO SANTOSO	21010117120044	MUHAMMAD SIGIT WAHYUDI	21010118140178	

In [26]: `df.shape`

Out[26]: (332, 17)

In [27]: `df1.head(1)`

Out[27]:

	Nim2	Mahasiswa2	Nim3	Mahasiswa3	Nim4	Mahasiswa4	Nim5	Mahasiswa5	Prestasi	Website	Tahun	No
0	21010117120044	MUHAMMAD SIGIT WAHYUDI	21010118140178	PAULUS KEVIN	21010118140147	PRASETYO ADJIE	21010118140156	PANCA EDI KURNIAWAN	Partisipasi/ Delegasi/ Peserta Kejuaraan	http://civilexpo.its.ac.id/	2020	1

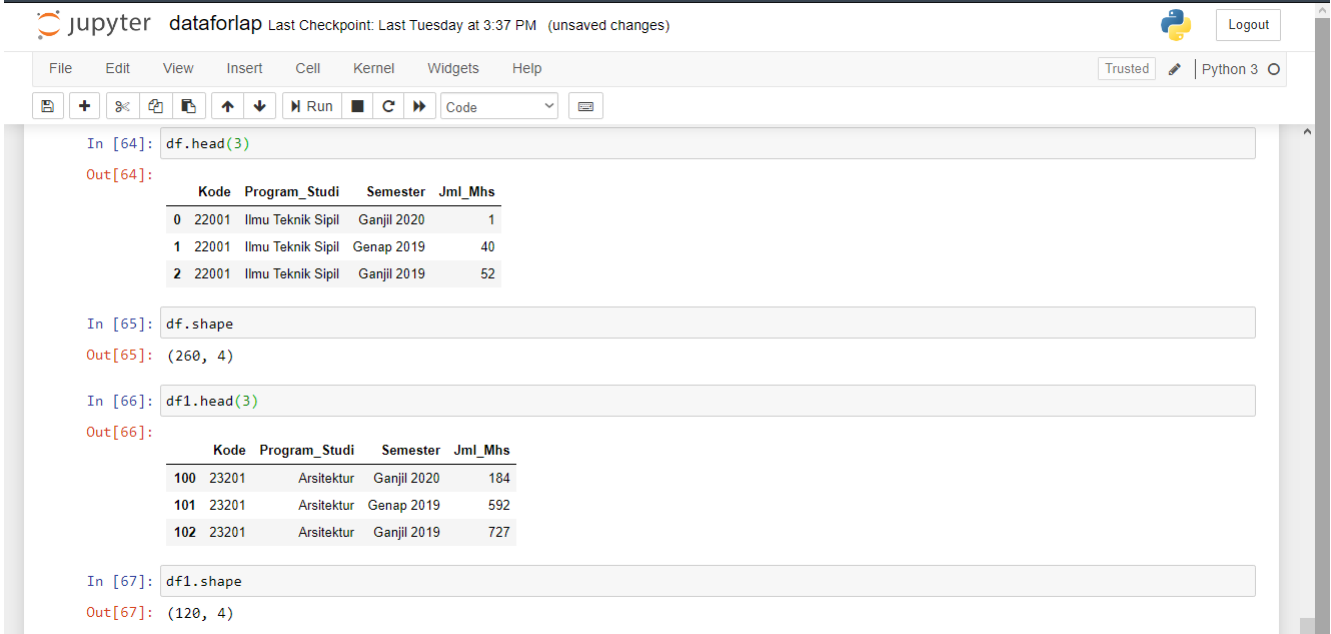
In [28]: `df1.shape`

Out[28]: (332, 18)

Gambar 4.29 Hasil *filtering* data di laman web Prestasi Undip

Pada Gambar 4.29 dapat dilihat bahwa inisial `df` merupakan data awal dan inisial `df1` merupakan data akhir yang siap disimpan dalam bentuk JSON. Data awal memiliki baris berjumlah 332 dan kolom berjumlah 17, setelah proses data *filtering* menjadi 332 dan kolom berjumlah 18. Penambahan baris terjadi karena data pada kolom 'Tanggal' harus dipisahkan untuk diperoleh tahun terselenggaranya kegiatan tersebut.

Gambar 4.30 berikut merupakan hasil *filtering* data pertama di laman web Forlap Dikti.



```

In [64]: df.head(3)
Out[64]:
   Kode  Program_Studi  Semester  Jml_Mhs
0  22001  Ilmu Teknik Sipil  Ganjil 2020      1
1  22001  Ilmu Teknik Sipil  Genap 2019     40
2  22001  Ilmu Teknik Sipil  Ganjil 2019     52

In [65]: df.shape
Out[65]: (260, 4)

In [66]: df1.head(3)
Out[66]:
   Kode  Program_Studi  Semester  Jml_Mhs
100  23201  Arsitektur  Ganjil 2020     184
101  23201  Arsitektur  Genap 2019     592
102  23201  Arsitektur  Ganjil 2019     727

In [67]: df1.shape
Out[67]: (120, 4)

```

Gambar 4.30 Hasil pertama *filtering* data di laman web Forlap Dikti

Pada Gambar 4.30 dapat dilihat bahwa inisial `df` merupakan data awal dan inisial `df1` merupakan data akhir yang siap disimpan dalam bentuk JSON. Data awal memiliki baris berjumlah 260 dan kolom berjumlah 4, setelah proses data *filtering* menjadi 120 dan kolom berjumlah 4. Pengurangan baris terjadi karena data yang akan digunakan hanya meliputi Fakultas Teknik.

Gambar 4.31 berikut merupakan hasil *filtering* data kedua di laman web Forlap.

The screenshot shows a Jupyter Notebook with the following code and output:

```
In [29]: df.head(3)
```

```
Out[29]:
```

	Kode	Program_Studi	Status	Jenjang	Jml_Dsn_TAlalu	Jml_Mhs_TAlalu	Rasio_TAlalu	Jml_Dsn_TAini	Jml_Mhs_TAini	Rasio_TAini
0	63001	Administrasi Publik	Aktif	S3	6	121	1 : 20.2	6	129	1 : 21.5
1	60001	Ekonomi	Aktif	S3	6	266	1 : 44.3	6	296	1 : 49.3
2	74001	Hukum	Aktif	S3	18	158	1 : 8.8	18	177	1 : 9.8

```
In [26]: df.shape
```

```
Out[26]: (167, 10)
```

```
In [30]: df1.head(3)
```

```
Out[30]:
```

	Kode	Program_Studi	Status	Jenjang	Jml_Dsn_TAlalu	Jml_Mhs_TAlalu	Rasio_TAlalu	Jml_Dsn_TAini	Jml_Mhs_TAini	Rasio_TAini
84	23201	Arsitektur	Aktif	S1	17	586	1 : 34.5	17	727	1 : 42.8
109	35201	Perencanaan Wilayah Dan Kota	Aktif	S1	33	679	1 : 20.6	33	779	1 : 23.6
117	20201	Teknik Elektro	Aktif	S1	27	676	1 : 25	27	762	1 : 28.2

```
In [28]: df1.shape
```

```
Out[28]: (12, 10)
```

Gambar 4.31 Hasil kedua *filtering* data di laman web Forlap Dikti

Pada Gambar 4.31 dapat dilihat bahwa inisial `df` merupakan data awal dan inisial `df1` merupakan data akhir yang siap disimpan dalam bentuk JSON. Data awal memiliki baris berjumlah 167 dan kolom berjumlah 10, setelah proses data *filtering* menjadi 12 dan kolom berjumlah 10. Pengurangan baris terjadi karena data yang akan digunakan hanya meliputi Fakultas Teknik.

Berikut hasil proses *filtering* yang dirangkum dalam Tabel 4.3.

Tabel 4.3 Perbandingan hasil data *filtering*

Laman Web	Data Awal		Data Akhir		Split	Split	Hapus	Jumlah
	Baris	Kolom	Baris	Kolom	Baris	Kolom	Cell	Proses
Sip3mu Undip	2089	8	4042	10	1	2	2	5
Prestasi Undip	332	17	332	18	0	2	1	3
Forlap Dikti	260	4	120	4	0	0	2	2
	167	10	12	10	0	0	1	1

Dapat dilihat pada Tabel 4.3 bahwa proses data *filtering* di laman web Sip3mu Undip melakukan instruksi inti sebanyak 5 kali, di laman web Prestasi Undip

melakukan instruksi inti sebanyak 3 kali, di laman web Sip3mu Undip melakukan instruksi inti sebanyak 3 kali dan menghasilkan 2 berkas. Proses data *filtering* di laman web Sip3mu Undip mengharuskan dilakukannya 1 kali *split* baris, 2 kali *split* kolom, dan 2 kali penghapusan baris. Proses data *filtering* di laman web Prestasi Undip mengharuskan dilakukannya 2 kali *split* kolom dan 1 kali penghapusan kolom. Proses data *filtering* di laman web Forlap mengharuskan dilakukannya 2 kali penghapusan kolom pada berkas pertama, dan 1 kali penghapusan kolom pada berkas kedua.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang didapat dari hasil dan pembahasan system yang dibangun adalah sebagai berikut:

1. Sistem pengambilan data berbasis Selenium dan Pandas yang dibangun untuk laman web Eduk, Sip3mu, Prestasi, dan Forlap berhasil dilakukan dan menghasilkan 9 berkas dengan total sebesar 4756,7 Kb.
2. Sistem *crawling* data untuk setiap laman web berbeda, menyesuaikan dengan *inspect element* dan *interface*-nya, pada laman web Eduk Undip pengambilan data dilakukan dengan mengunduh berkas berformat JSON, pada laman web Prestasi Undip dan Foplap Ristekdikti pengambilan data dilakukan dengan memasukkan data dalam *list*, dan pada laman web Sip3mu Undip pengambilan data dilakukan dengan mengunduh berkas berformat Excel.
3. Sistem *filtering* data berbasis Pandas *dataframe* cocok untuk data bersekala kecil, tetapi kenaikan kecepatan eksekusi dan penggunaan memori terjadi secara signifikan seiring bertambahnya jumlah data sehingga tidak cocok digunakan untuk pengolahan data bersekala besar.

5.2 Saran

1. Mengembangkan sistem untuk melakukan pengambilan data di laman web yang tersedia untuk umum seperti Scopus guna melengkapi data yang sesuai dengan instrumen akreditasi SAPTO BAN-PT.
2. Menggunakan Pyspark *dataframe* untuk sistem *filtering* data untuk mengelola data bersekala besar.

DAFTAR PUSTAKA

- [1] Panduan Tugas Akhir Teknik Elektro Undip.
- [2] Panduan Penggunaan SAPTO Versi 01 Untuk Pengguna Perguruan Tinggi oleh BAN-PT Tahun 2017.
- [3] O. Leonardo, and H. Maria, "Analytical Data Mart for the Monitoring of University Accreditation Indicators", IEEE 2019
- [4] L. Michael, N. Henry, dan R. Silvia, "Perbandingan Performa Tools Web Scraping pada Website dengan Data Statis dan Dinamis", Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra.
- [5] Peraturan BAN-PT nomor 5 Tahun 2019
- [6] Iin Mutmainah. 2019. Mengenal Pandas Dan *Dataframe*. <https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40> (diakses Oktober 2020)
- [7] Dede Brahma. 2020. Perbedaan Antara *Crawling* dan *Scraping*. <https://medium.com/@dede.brahma2/perbedaan-antara-crawling-dan-scraping-98e64e0c6439> (diakses tanggal 19 Oktober 2020)
- [8] Powerscraping.com. *Web Scraping VS Web Crawling*. <https://prowebscraping.com/web-scraping-vs-web-crawling/> (diakses Desember 2020)
- [9] Coldsript. 2019. *I will create Selenium Webdriver script for data mining*. <https://www.fiverr.com/coldsript/create-selenium-webdriver-script-for-data-mining> (diakses Desember 2020)
- [10] Beon Intermedia. 2020. Apa itu Selenium? *Tools Auto Testing Web Apps Terbaik*. <https://www.jagoanhosting.com/blog/apa-itu-selenium/> (diakses Desember 2020)
- [11] Selenium.dev Getting Started with Webdriver. https://www.selenium.dev/documentation/en/getting_started_with_webdriver/ (diakses Oktober 2020)
- [12] Spark.apache.org. *SQL Programming Guide*. <https://spark.apache.org/docs/1.6.1/sql-programming-guide.html> (diakses November 2020)

- [13] Ichi.pro. 2020. Contoh Menggunakan Apache Spark dengan PySpark Menggunakan Python. <https://ichi.pro/id/contoh-menggunakan-apache-spark-dengan-pyspark-menggunakan-python-267611095265298>. (diakses Desember 2020)
- [14] Ariata C. 2020. Apap itu HTML? Fungsi dan Cara Kerja. <https://www.hostinger.co.id/tutorial/apa-itu-html/> (diakses tanggal 13 Desember 2020)
- [15] M. Vivensius, S. Herry, dan B. Arif, “Rancang Bangun Aplikasi Web Scraping untuk Korpus Paralel Indonesia - Inggris dengan Metode HTML DOM”, Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol. 5, No. 1, Januari 2017.
- [16] McKinney, Wes. “*pandas: powerful Python data analysis toolkit (Release 1.1.0)*”. Pandas Development Team. 2020
- [17] VanderPlas, Jake. “*Python Data Science Handbook: Essential Tools for Working with Data*”. 1005 Gravenstein Highway North, Sebastopol, CA 95472 : O’Reilly Media, Inc. 2017.
- [18] json.org. 2017. Pengenalan JSON. <https://www.json.org/json-id.html> (diakses Oktober 2020)

BIODATA



Nama : Laila Lathifah
NIM : 21060116130112
Konsentrasi : Teknologi Informasi
Tempat/ Tgl. Lahir : Bukitkemuning, 16 Juni 1997
Alamat Sekarang : Jl. Baskoro Raya 43B Tembalang,
Semarang
No. Telpon / HP : 082180011377
Alamat e-mail : lathifahlailaa@gmail.com
Nama orang tua : Edy Suryanto
Alamat orang tua : Lampung
IP Kumulatif : 3,47

Pengalaman dan Prestasi yang pernah diraih:

1. Kerja Praktek PT Bumi Manunggal Sinergi Banyumas
2. Asisten Praktikum Algoritma dan Pemrograman 2018
3. Bendahara Bidang RKM HME 2019
4. Bendahara Biro PHILAR Angkatan XVI
5. Litbang FST Tahun 2019

Semarang, 22 Desember 2020

Laila Lathifah

NIM. 21060116130112

LAMPIRAN A
MAKALAH TUGAS AKHIR

SISTEM CRAWLING DATA INSTRUMEN AKREDITASI BERBASIS SELENIUM DAN PANDAS

Laila Lathifah^{*)}, Eko Handoyo, dan Yosua Alvin Adi Soetrisno

Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: gdismnis@students.undip.ac.id

Abstrak

Perkembangan teknologi informasi telah sampai pada masa dimana hampir setiap aktivitas transaksi dapat dilakukan secara daring tanpa bertemu dengan pihak yang bersangkutan. Sama halnya dengan akreditasi kampus yang evaluasinya dapat dilakukan secara daring melalui web SAPTO (Sistem Akreditasi Perguruan Tinggi Online) yang dikembangkan oleh pihak BAN-PT (Badan Akreditasi Nasional Perguruan Tinggi). Pada laporan Tugas Akhir ini akan membahas mengenai pembangunan sistem pengumpulan data dari pangkalan database berbasis web menggunakan teknik crawling dan proses filtering data yang dapat mendukung proses akreditasi secara daring. Sistem crawling data didukung oleh tools Selenium dan sistem filtering data menggunakan library Pandas dataframe. Crawling data dilakukan untuk 4 laman web berbeda, yaitu laman web Eduk yang berisi data diri dosen Universitas Diponegoro, laman web Sip3mu yang berisi data penelitian dosen Universitas Diponegoro, laman web Prestasi yang berisi data perlombaan mahasiswa Universitas Diponegoro, dan laman web Forlap yang berisi data program studi serta jumlah mahasiswa Universitas Diponegoro. Sistem crawling data menggunakan tool Selenium menyesuaikan dengan interface setiap laman web sehingga menghasilkan berkas yang siap dimasukkan ke database atau di filtering. Sistem filtering data menggunakan Pandas dataframe menyesuaikan dengan keperluan analisis data lebih lanjut, tetapi kinerjanya kurang stabil saat mengelola data, dimana semakin banyak data maka semakin besar pula kecepatan eksekusi dan penggunaan memorinya.

Kata kunci : Crawling Data, Python, Selenium, Pandas, Dataframe

Abstract

The development of information technology has reached a time when almost every transaction activity can be done online without meeting with the party concerned. Similarly, campus accreditation evaluation can be done online through the SAPTO web developed by BAN-PT. In this Final Task report will discuss the construction of a database collection system from a web-based database using crawling techniques and data filtering processes that can support the accreditation process online. The data crawling system is supported by Selenium tools and data filtering system using Pandas Dataframe library. Crawling data is done for 4 different websites, namely Eduk's web page containing data of Diponegoro University lecturers, Sip3mu website containing research data of Diponegoro University lecturers, Prestasi website containing data on the computation of Diponegoro University students, and Forlap web pages containing data program study and the number of Diponegoro University students. The system crawling data that using tool Selenium adjusts to their interfaces in website to produce files that ready to importing to the database or to filtering. The system filtering data that using Pandas dataframe adapts to the needs of further data analysts, but its performance is less stable when managing data, where the more data, the greater the speed of execution and memory usage.

Keywords: Data Crawling, Python, Selenium, Pandas, Dataframe

1. Pendahuluan

Perkembangan teknologi informasi telah sampai pada masa dimana hampir setiap aktivitas transaksi dapat dilakukan secara daring tanpa bertemu dengan pihak yang bersangkutan. Sama halnya dengan akreditasi kampus yang menunjukkan kualitas, dimana kualitas pendidikan

perguruan tinggi telah menjadi masalah transendental. Hal ini berkenaan dengan meningkatnya kepedulian pemerintah terhadap berbagai tingkat kualitas yang dibuktikan oleh sistem pendidikan. Menanggapi masalah ini, beberapa evaluasi dan praktik akreditasi dilaksanakan untuk memastikan dan meningkatkan kualitas karir dan

institusi universitas di berbagai negara Amerika Latin, dimana pendataan sudah bisa dilakukan secara daring [3].

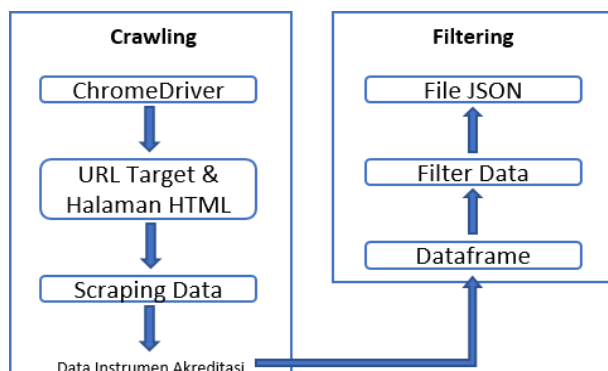
Berdasarkan Peraturan BAN-PT nomor 5 Tahun 2019, yang telah ditetapkan pada tanggal 23 September 2019 pendataan akreditasi di Indonesia dapat dilakukan secara daring melalui situs sapro.banpt.or.id [5]. SAPTO (Sistem Akreditasi Perguruan Tinggi Online) merupakan sistem yang dikembangkan BAN-PT untuk meningkatkan efisiensi dan kualitas proses akreditasi perguruan tinggi yang diselenggarakan oleh BAN-PT. SAPTO mendukung setiap proses yang dilakukan dalam akreditasi seperti pengajuan usulan akreditasi oleh perguruan tinggi, pemeriksaan dokumen, penugasan asesor dan validasi yang dilakukan, proses AK (asesmen kecukupan) dan AL (asesmen lapangan) oleh asesor. [2]

Berdasarkan peraturan tersebut perlu adanya sistem pengumpulan data yang dapat dijalankan secara otomatis dan berkala untuk mempermudah proses pengumpulan data yang disesuaikan dengan kebutuhan analisis data selanjutnya. Data yang telah terkumpul akan di *filtering* menggunakan *dataframe* pada library Pandas. Oleh karena itu, penelitian ini akan membahas mengenai “Sistem *Crawling* Data Instrumen Akreditasi Berbasis Selenium dan Pandas”. Selenium memudahkan untuk *crawling* data karena dapat melakukan interaksi seperti yang dilakukan oleh user ketika menelusuri web seperti melakukan klik pada tombol, mengisi form, membuka tab baru, membuka halaman web, dan lain-lain[4]. Penggunaan *dataframe* memudahkan untuk membaca sebuah berkas dan menjadikannya tabel, selain itu dapat mengolah suatu data dengan menggunakan operasi seperti *join*, *distinct*, *group by*, agregasi, dan teknik lainnya yang terdapat pada SQL[6].

2. Metode

2.1. Deskripsi Sistem

Desain sistem observasi data pengunjung landmark yang dilakukan dapat dilihat pada Gambar 1.



Gambar 1 Desain Sistem

Pengambilan data secara otomatis yang disebut dengan *crawling* data dapat dilihat alurnya pada Gambar 1. Proses *crawling* data diawal dengan terbukanya ChromeDriver yang langsung mengakses URL target untuk melakukan *login* akun admin kemudian menuju ke halaman HTML yang telah ditentukan dalam *scripts* dan melakukan *scraping* (pengambilan data). Data yang akan diambil dalam bentuk *tabel* ataupun *form* yang akan disimpan sementara pada suatu *list* atau diunduh dalam bentuk *berkas* berektensi .xls maupun .json. Kemudian, data tersebut dimasukkan ke dalam Dataframe untuk dibersihkan sesuai dengan desain *database* yang dibutuhkan dalam melakukan proses pengolahan data. Dataframe yang dinilai sudah sesuai dengan kebutuhan akan disimpan dalam sebuah *berkas* berektensi JSON untuk mempermudah proses *import* ke dalam *database*. Penentuan Dataframe yang sesuai dengan kebutuhan *database* merujuk pada instrumen akreditasi di laman SAPTO BAN-PT.

Web *crawling* ini menggunakan *tool* Selenium dengan perangkat lunak tambahan berupa *browser driver* atau *webdriver*. Selenium dapat dijalankan menggunakan beberapa bahasa pemrograman, salah satunya adalah Python yang akan digunakan dalam pembangunan aplikasi web *crawling* ini. *Webdriver* pendukung Selenium yang dipakai adalah ChromeDriver untuk mempermudah proses pengambilan data dengan bantuan Chrome Extension tertentu.

2.2. Analisis Kebutuhan

2.2.1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan gambaran mengenai fungsi-fungsi yang dapat dilakukan oleh sistem ini.

Kebutuhan fungsional sistem meliputi:

- 1) Mengakses halaman HTML sesuai dengan URL yang dicantumkan dalam *scripts*.
- 2) Mengambil data pada suatu *tabel* ataupun *form* untuk disimpan sementara dalam bentuk *list* atau *berkas* unduhan berektensi .xls.
- 3) Menyaring data yang ada pada penyimpanan sementara menggunakan Dataframe supaya tidak mengubah data unduhan dari halaman HTML.
- 4) Menyimpan hasil akhir Dataframe ke dalam *berkas* berektensi .json

2.2.2. Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan sistem meliputi kinerja, kelengkapan operasi pada fungsi-fungsi yang ada, serta kesesuaian dengan lingkungan penggunaannya. Kebutuhan non-fungsional ini melingkupi beberapa kebutuhan yang mendukung kebutuhan fungsional, rumusan kebutuhan non-fungsional meliputi:

- 1) Kebutuhan Operasional
 - Kecepatan dapat berjalan dengan baik pada sistem operasi Ubuntu dengan RAM minimal 4Gb dan pada sistem operasi Windows dengan RAM minimal 8Gb
 - Sistem hanya dapat diakses dan digunakan oleh petugas pengelola akreditasi.
 - Sistem ini dibangun menggunakan bahasa pemrograman Python 3 dan *library* Pandas.
- 2) Performa Sistem

Sistem yang dibangun merupakan aplikasi yang berjalan pada laptop. Terdapat beberapa keterbatasan yang ditemui pada laptop. Oleh karena itu, hal berikut perlu diperhatikan guna menjadi acuan dalam pengembangan sistem, diantaranya:

 - Penggunaan laptop yang tidak bisa menyala secara terus menerus selama 24 jam sehari.
 - *System* yang dirancang untuk web *crawling* belum bisa mendeteksi *update* data secara berkala.

Dari keterbatasan pada laptop tersebut, maka diusulkan beberapa alternatif sebagai berikut:

 - Menggunakan *computer server* yang tersedia di institusi dan aktif selama 24 jam dalam sehari.
 - Merancang *system* untuk melakukan pengambilan data setiap 24 jam sekali.

2.2.3. Kebutuhan Perangkat Keras

Dalam pembangunan sistem ini, dibutuhkan beberapa spesifikasi perangkat keras. Spesifikasi perangkat keras tersebut dapat dimasukkan ke dalam kebutuhan perangkat keras dalam analisis kebutuhan. Karena melibatkan pengambilan dan penyaringan data, perangkat keras yang dibutuhkan dalam membuat aplikasi ini adalah sebuah komputer dengan spesifikasi minimal yang ditunjukkan pada Tabel 1 berikut.

Tabel 1 Kebutuhan perangkat keras

Spesifikasi	Keterangan
<i>Processor</i>	Intel(R) Core(TM) i5-2520M
RAM	8192 MB
<i>Harddisk</i>	31 GB
Laptop	Dell Latitude E6320 Core i5

2.2.4. Kebutuhan Perangkat Lunak

Dalam pembangunan sistem ini, dibutuhkan beberapa spesifikasi perangkat lunak. Spesifikasi perangkat lunak tersebut dapat dimasukkan ke dalam kebutuhan perangkat lunak dalam analisis kebutuhan. Perangkat lunak yang dibutuhkan baik untuk merancang sistem, membangun

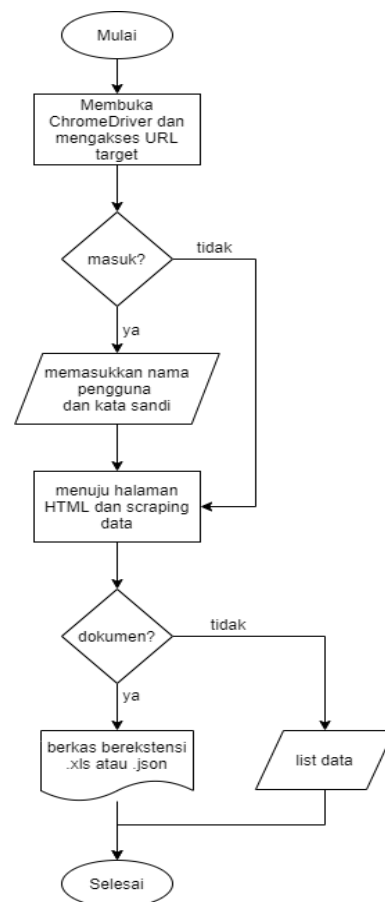
sistem maupun menjalankan sistem adalah seperti yang ditunjukkan pada Tabel 2 berikut.

Tabel 2 Kebutuhan perangkat lunak

Spesifikasi	Keterangan
Sistem Operasi	Ubuntu 18.06
<i>Text Editor</i>	Notepad++
<i>Tool</i> Otomatisasi Web	Selenium 3.0
<i>WebDriver</i>	Chrome WebDriver
<i>Browser</i>	Chrome Browser
Bahasa Pemrograman	Python 3
<i>Library</i>	Pandas

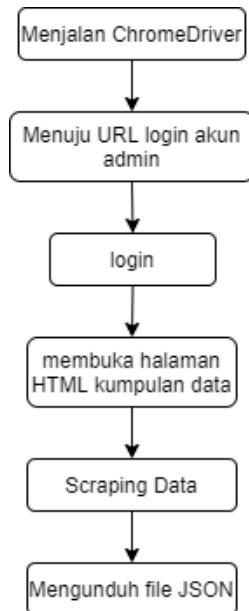
2.3. Perancangan Sistem Web *Crawling*

Sistem Web *Crawling* ini bergantung pada laman web yang akan diambil datanya. Tetapi, proses secara garis besar akan digambarkan menggunakan *flowchart* yang dapat dilihat pada Gambar 2 berikut ini.



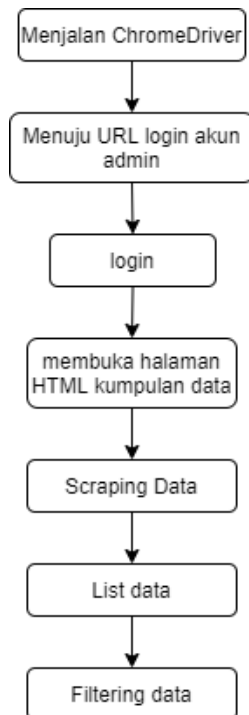
Gambar 2 Flowchart perancangan sistem web *crawling*

2.3.1. Diagram Alir Sistem *Crawling* Data di Laman Web Eduk Undip



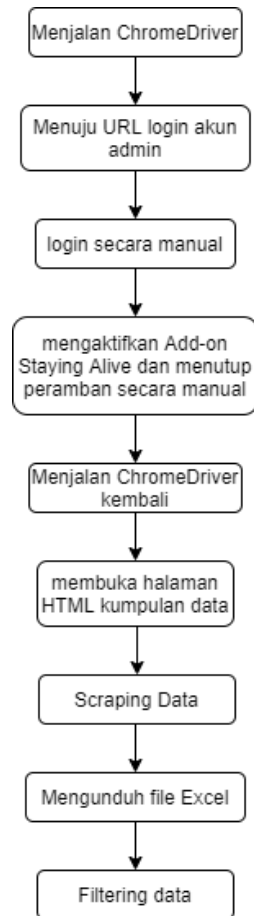
Gambar 3 Diagram alir sistem *crawling* data di laman web Eduk Undip

2.3.2. Diagram Alir Sistem *Crawling* Data di Laman Web Prestasi Undip



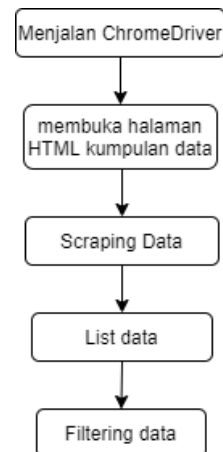
Gambar 4 Diagram alir sistem *crawling* data di laman web Prestasi Undip

2.3.3. Diagram Alir Sistem *Crawling* Data di Laman Web Sip3mu Undip



Gambar 5 Diagram alir sistem *crawling* data di laman web Sip3mu Undip

2.3.4. Diagram Alir Sistem *Crawling* Data di Laman Web Forlap Dikti

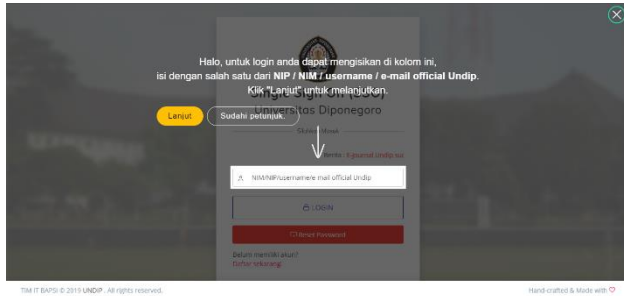


Gambar 6 Diagram Alir Analisis *Geodataframe* Pengunjung Pada Peta Wilayah Semarang

3. Hasil dan Pembahasan

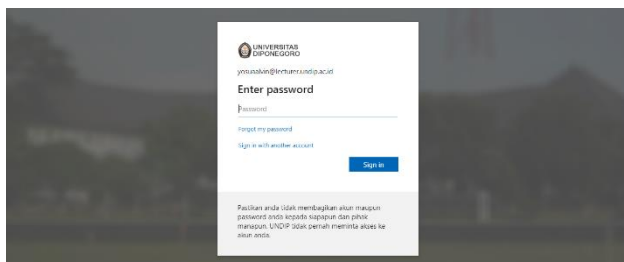
3.1. Implementasi Sistem *Crawling* Data

3.1.1. Implementasi Sistem *Crawling* Data di Laman Web Prestasi Undip



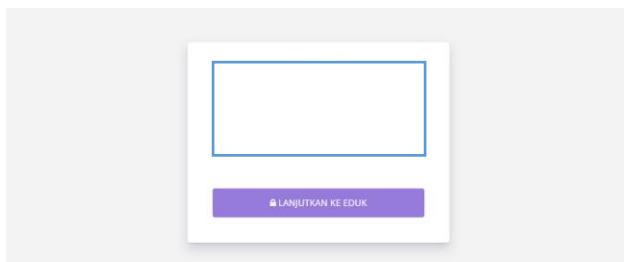
Gambar 7 Tampilan *submit* nama pengguna di laman web SSO (Single Sign On)

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 7, dimana tombol 'Sudah Petunjuk' harus ditekan terlebih dahulu sebelum memasukkan nama pengguna. Kemudian, memasukkan nama menggunakan *method* `.send_keys('sometext')` dan `.submit()` sebagai tombol *enter*.



Gambar 8 Tampilan *submit* nama pengguna di laman web SSO

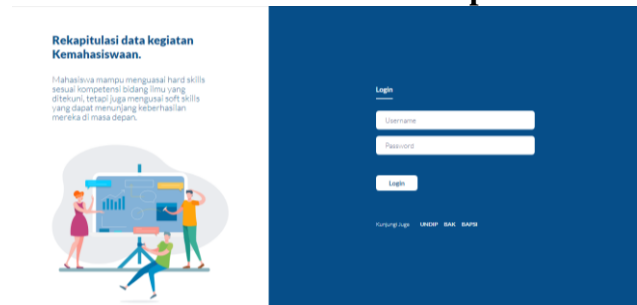
Setelah *submit* nama pengguna peramban akan lanjut ke halaman *submit* kata sandi seperti pada Gambar 8. Kemudian, program akan memasukkan sandi menggunakan *method* `.send_keys('sometext')` dan `.submit()` sebagai tombol *enter*.



Gambar 9 Tampilan verifikasi menuju laman web Eduk Undip

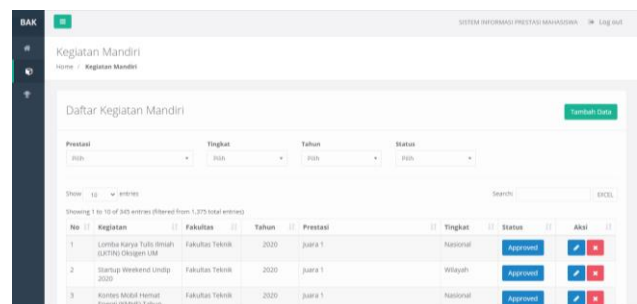
Setelah *submit* kata sandi peramban akan lanjut ke halaman *eduk* yang memerlukan verifikasi seperti pada Gambar 9. Sebagai verifikasi program akan menekan tombol 'LANJUTKAN KE EDUK' menggunakan *method* `.click()` sebagai tombol *enter*. Kemudian, peramban akan langsung menuju laman pangkalan basis data yang berbentuk JSON dan melakukan pengunduhan data.

3.1.2. Implementasi Sistem *Crawling* Data di Laman Web Prestasi Undip



Gambar 10 Tampilan *submit* akun admin di laman web Prestasi Undip

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 10, dimana *submit* nama pengguna dan kata sandi menjadi satu halaman. Proses *submit* nama pengguna menggunakan *method* `.send_keys('sometext')` dan `.submit()` sebagai tombol *enter*. Proses *submit* kata sandi menggunakan *method* `.send_keys('sometext')` dan `.submit()` sebagai tombol *enter*.



Gambar 11 Tampilan kumpulan data di laman web Prestasi Undip

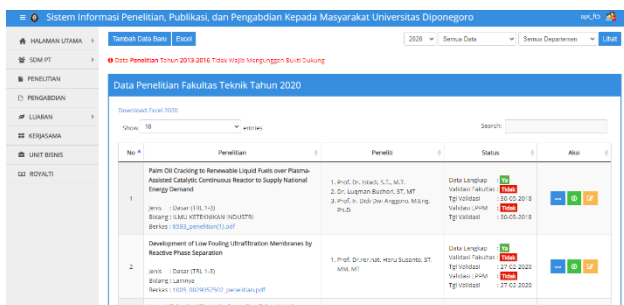
Setelah *submit* akun admin, peramban akan berjalan menuju halaman pangkalan basis data perlombaan yang diikuti para mahasiswa untuk pengumpulan *link* yang terdapat pada tombol 'Approved' seperti pada Gambar 11. Setelah pengumpulan *link* selesai, peramban secara otomatis akan membuka *link* tersebut satu per satu dan melakukan proses pengambilan data berupa teks yang akan ditampung dalam *list*. Selanjutnya, data akan di *filtering* dalam *dataframe*.

3.1.3. Implementasi Sistem *Crawling* Data di Laman Web Sip3mu Undip



Gambar 12 Tampilan submit akun admin di laman web Sip3mu Undip

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 12, dimana *submit* nama pengguna dan kata sandi menjadi satu halaman serta ada tambahan *submit captcha*. Proses pemasukan *captcha* belum bisa dilakukan secara otomatis karena kata akan muncul secara acak sehingga perlunya pemrograman lebih lanjut untuk menangani hal ini. Oleh karena itu, *submit* akun admin dilakukan secara manual. Kemudian, mengaktifkan Add-on Staying Alive yang berguna untuk menjaga session untuk tetap aktif saat menjalankan program selanjutnya, lalu menutup peramban.



Gambar 13 Tampilan kumpulan data di laman web Sip3mu Undip

Setelah menutup peramban dilanjutkan dengan menjalankan program kedua dan saat peramban berjalan akan langsung menuju halaman kumpulan data seperti pada Gambar 13. Pengumpulan data akan dilakukan dengan mengunduh berkas Excel berdasarkan tahun penelitian. Proses pengunduhan dilakukan dengan pemilihan tahun kemudian menekan tombol 'Lihat' menggunakan *method .click()* selanjutnya akan ditekan tombol 'Excel' menggunakan *method .click()* dan berkas akan otomatis terunduh. Selanjutnya, data akan di *filtering* dalam *dataframe*.

3.1.4. Implementasi Sistem *Crawling* Data di Laman Web Forlap Dikti

Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa	Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa
1.751	49.425	1 : 28.2	1.751	56.125	1 : 32.1

Daftar Program Studi

710 x 8254 as mahasiswa pada semester ganjil tahun ajaran tersebut. Jika tidak sesuai, iporannya melalui aplikasi PDDikti Feeder

No.	Kode	Nama Program Studi	Status	Jenjang	Data Pelaporan Tahun 2018/2019			Data Pelaporan Tahun 2019/2020		
					Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa	Jml Dosen Tetap	Jml Mhs	Rasio Dosen Tetap/Jumlah Mahasiswa
1	63001	Administrasi Publik	Aktif	S3	6	121	1:20.2	6	129	1:21.5
2	60001	Ekonomi	Aktif	S3	6	266	1:44.3	6	296	1:49.3
3	74001	Hukum	Aktif	S3	18	158	1:8.8	18	177	1:9.8
4	23001	Ilmu Arsitektur Dan Perkotaan	Aktif	S3	5	55	1:11	5	62	1:12.4
5	11001	Ilmu Kedokteran dan Kesehatan	Aktif	S3	5	76	1:15.2	5	104	1:20.8

Gambar 14 Tampilan kumpulan data program studi di laman web Forlap Ristekdikti

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 14 dimana data daftar program studi langsung bisa diakses tanpa harus melakukan *submit* akun. Proses pengambilan data berupa teks dilakukan per kolom dan dimasukkan ke dalam perulangan. Data yang berhasil diambil akan ditampung dalam *list* dan ada 10 list yang digunakan untuk menampung data sesuai dengan jumlah kolom. Artinya, ada 10 perulangan yang akan membaca data perkolom. Selanjutnya, data akan di *filtering* dalam *dataframe*.

Profil Program Studi

Umum	Dosen	Mahasiswa
No.	Semester	Banyak
1	Genap 2019	120
2	Ganjil 2019	129
3	Genap 2018	110
4	Ganjil 2018	121
5	Genap 2017	112
6	Ganjil 2017	142
7	Genap 2016	121
8	Ganjil 2016	117
9	Genap 2015	94
10	Ganjil 2015	102
11	Genap 2014	87
12	Ganjil 2014	90
13	Genap 2013	88

Gambar 15 Tampilan kumpulan data jumlah mahasiswa di laman web Forlap Ristekdikti

Setelah pengambilan data daftar program studi selesai, peramban akan langsung mengumpulkan *link* untuk yang terdapat pada daftar nama program studi dan nilai *link* tersebut akan ditampilkan dalam list. Kemudian, peramban secara otomatis akan membuka *link* tersebut satu per satu seperti pada gambar 15. Proses pengambilan data berupa teks tidak jauh berbeda dengan data daftar program studi sebelumnya, yaitu dilakukan per kolom dan data yang berhasil diambil akan ditampilkan dalam *list* dan ada 3 list yang digunakan untuk menampilkan data sesuai dengan jumlah kolom. Artinya, ada 3 perulangan yang akan membaca data perkolom. Selanjutnya, data akan di *filtering* dalam *dataframe*.

3.2. Pengujian Proses *Filtering* Data

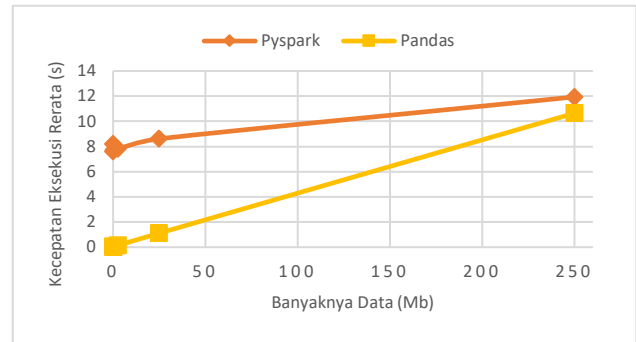
Pengujian dilakukan dengan membandingkan kinerja kecepatan eksekusi dan penggunaan memori oleh Pandas *dataframe* dan Pyspark *dataframe*. Pengujian dilakukan berdasarkan 3 kondisi untuk setiap banyaknya data, yaitu saat 1 aplikasi dijalankan, saat 2 aplikasi dijalankan, dan saat 3 aplikasi dijalankan. Berikut hasil rerata pengujian Pandas *dataframe* dan Pyspark *dataframe* yang disajikan dalam bentuk tabel dan grafik.

3.2.1. Pengujian Kecepatan Eksekusi Proses *Filtering* Data

Tabel 3 Hasil pengujian kecepatan eksekusi proses *filtering* data

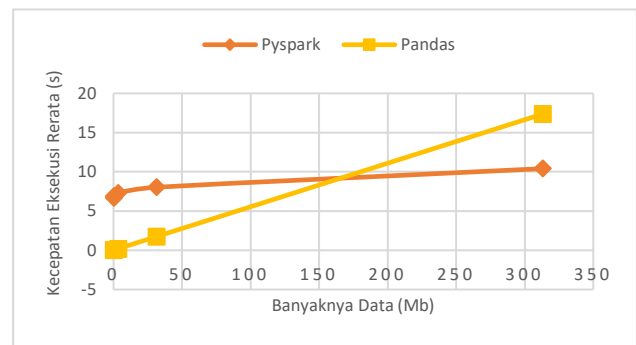
Laman Web	Banyaknya Data (Mb)	Kecepatan Eksekusi (s)	
		Pyspark	Pandas
Sip3mu	0,003	8,206324418	0,039816
	0,025	7,613231817	0,018068
	0,250	7,748084625	0,026259
	2,499	7,800643285	0,128549
	24,996	8,612986644	1,113693
Undip	249,961	11,93711193	10,64237
Prestasi	0,004	6,699864229	0,02321
	0,031	6,726234674	0,024081
	0,312	6,869834661	0,031958
	3,129	7,336883624	0,187398
	31,291	8,035114368	1,740815
Undip	312,913	10,40634084	17,35626
Forlap	0,002	7,400766611	0,024983
	0,011	7,155849059	0,057595
	0,110	7,296039184	0,047367
	1,008	7,526277622	0,110048
	10,075	8,08164978	0,359805
Dikti	100,739	10,32327882	1,921089

Dari data yang ada pada Tabel 3 dapat dibuat grafik seperti pada Gambar 16 Gambar 17 dan Gambar 18 berikut.



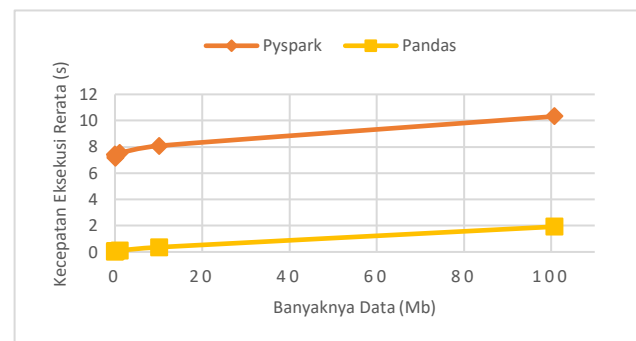
Gambar 16 Grafik perbandingan kecepatan eksekusi pada laman web Sip3mu

Pada Gambar 16 terlihat bahwa penggunaan Pandas *dataframe* lebih cepat dibandingkan Pyspark *dataframe*. Tetapi, kenaikan kecepatan pada Pandas *dataframe* cukup signifikan.



Gambar 17 Grafik perbandingan kecepatan eksekusi pada laman web Prestasi

Pada Gambar 17 terlihat bahwa kecepatan eksekusi Pandas *dataframe* lebih cepat dibandingkan dengan Pyspark *dataframe* saat besaran data berada di 175 Mb. Tetapi, kenaikan kecepatan pada Pandas *dataframe* mencapai 18 detik.



Gambar 18 Grafik perbandingan kecepatan eksekusi pada laman web Forlap

Pada Gambar 4.25 terlihat bahwa kecepatan eksekusi Pandas *dataframe* lebih cepat dibandingkan dengan

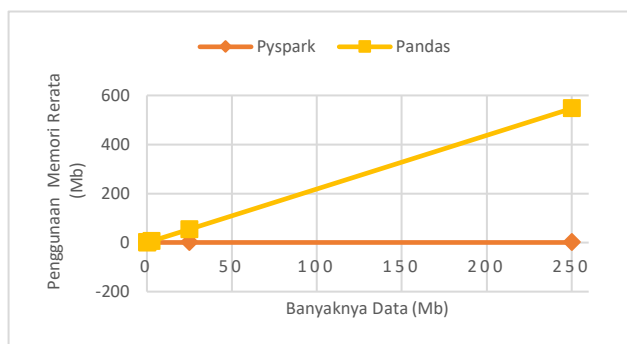
Pyspark *dataframe*. Kenaikan kecepatan eksekusi cenderung sama pada Pyspark *dataframe* dan Pandas *dataframe*.

3.2.2. Pengujian Penggunaan Memori Proses Filtering Data

Tabel 4 Hasil pengujian penggunaan memori proses *filtering* data

Laman Web	Banyaknya Data (Mb)	Penggunaan Memori (Mb)	
		Pyspark	Pandas
Sip3mu	0,003	0,425955	0,339076
	0,025	0,425956	0,364903
	0,250	0,412361	0,676763
	LPPM	0,425956	3,663401
	Undip	0,425959	33,56355
Prestasi Undip	24,996	0,425959	33,56355
	249,961	0,42596	267,8953
	0,004	0,412361	0,347042
	0,031	0,412362	0,415321
	0,312	0,447895	0,883193
Forlap Dikti	3,129	0,412362	5,579908
	31,291	0,412365	54,26836
	312,913	0,412366	546,7106
	0,002	0,447902	0,367378
	0,011	0,447895	0,381933
Forlap Dikti	0,110	0,447902	0,469856
	1,008	0,447895	1,102192
	10,075	0,447897	9,434527
	100,739	0,44789	85,67912

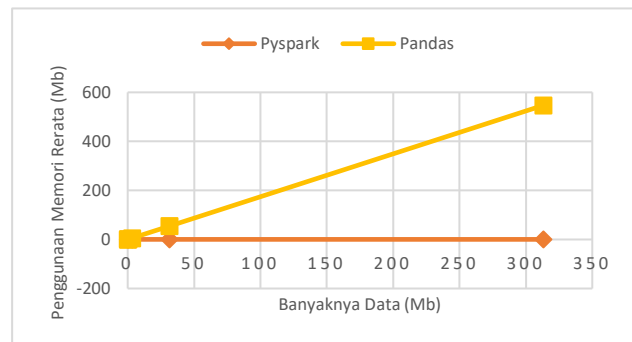
Dari data yang ada pada Tabel 4 dapat dibuat grafik seperti pada Gambar 19, Gambar 20, dan Gambar 21 berikut.



Gambar 19 Grafik perbandingan penggunaan memori pada laman web Sip3mu

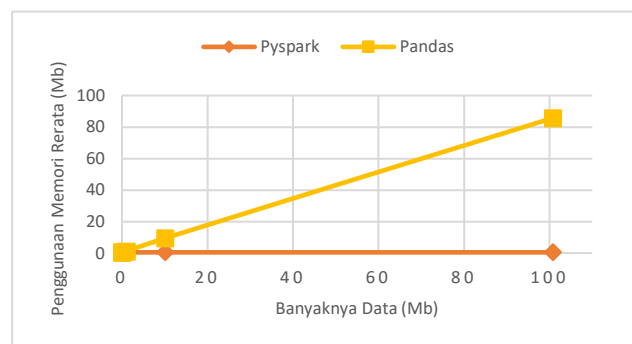
Pada Gambar 19 terlihat bahwa penggunaan memori pada Pyspark *dataframe* cenderung stabil dan bersekala sangat kecil dibandingkan dengan Pandas *dataframe*. Pada Pandas

dataframe besarnya penggunaan memori bergerak linear terhadap besarnya data yang di kelola.



Gambar 20 Grafik perbandingan penggunaan memori pada laman web Prestasi

Pada Gambar 19 terlihat bahwa penggunaan memori pada Pyspark *dataframe* cenderung stabil dan bersekala sangat kecil dibandingkan dengan Pandas *dataframe*. Pada Pandas *dataframe* besarnya penggunaan memori bergerak linear terhadap besarnya data yang di kelola.



Gambar 21 perbandingan penggunaan memori pada laman web Forlap

Pada Gambar 21 terlihat bahwa penggunaan memori pada Pyspark *dataframe* cenderung stabil dan bersekala sangat kecil dibandingkan dengan Pandas *dataframe*. Pada Pandas *dataframe* besarnya penggunaan memori bergerak linear terhadap besarnya data yang di kelola.

4. Kesimpulan

Kesimpulan yang didapat dari pembahasan implementasi sistem *crawling* data untuk setiap laman web bahwa proses pengambilan data berbeda-beda, menyesuaikan dengan tampilan *interface*-nya, pada laman web Eduk Undip pengambilan data dilakukan dengan mengunduh berkas berformat JSON dan tidak perlu dilakukan *filtering* data, pada laman web Prestasi Undip dan Foplap Ristekdikti pengambilan data dilakukan dengan memasukkan data dalam *list*, dan pada laman web Sip3mu Undip pengambilan data dilakukan dengan mengunduh berkas berformat Excel.

Kesimpulan yang didapat dari hasil pengujian proses *filtering* data didapatkan bahwa penggunaan Pandas *dataframe* cocok untuk data bersekala kecil, tetapi harus menyesuaikan ruang penyimpanan, sementara kenaikan kecepatan eksekusi dan penggunaan memori terjadi secara signifikan seiring bertambahnya jumlah data sehingga tidak cocok digunakan untuk program dengan data bersekala besar.

Referensi

- [1] Panduan Tugas Akhir Teknik Elektro Undip.
- [2] Panduan Penggunaan SAPTO Versi 01 Untuk Pengguna Perguruan Tinggi oleh BAN-PT Tahun 2017.
- [3] O. Leonardo, and H. Maria, "Analytical Data Mart for the Monitoring of University Accreditation Indicators", IEEE 2019.
- [4] L. Michael, N. Henry, dan R. Silvia, "Perbandingan Performa Tools Web Scraping pada Website dengan Data Statis dan Dinamis", Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra.
- [5] Peraturan BAN-PT nomor 5 Tahun 2019.
- [6] <https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40> (diakses Oktober 2020).
- [7] <https://medium.com/@dede.brahma2/perbedaan-antara-crawling-dan-scraping-98e64e0c6439> (diakses tanggal 19 Oktober 2020).
- [8] <https://proweb scraping.com/web-scraping-vs-web-crawling/> (diakses Desember 2020).
- [9] <https://www.fiverr.com/coldscript/create-selenium-webdriver-script-for-data-mining> (diakses Desember 2020).
- [10] <https://www.jagoanhosting.com/blog/apa-itu-selenium/> (diakses Desember 2020).
- [11] https://www.selenium.dev/documentation/en/getting_started_with_webdriver/ (diakses Oktober 2020).
- [12] <https://spark.apache.org/docs/1.6.1/sql-programming-guide.html> (diakses November 2020).
- [13] <https://ichi.pro/id/contoh-menggunakan-apache-spark-dengan-pyspark-menggunakan-python-267611095265298>. (diakses Desember 2020).
- [14] <https://www.hostinger.co.id/tutorial/apa-itu-html> (diakses tanggal 13 Desember 2020).
- [15] M. Vivensius, S. Herry, dan B. Arif, "Rancang Bangun Aplikasi Web Scraping untuk Korpus Paralel Indonesia - Inggris dengan Metode HTML DOM", Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol. 5, No. 1, Januari 2017.

Biodata



Laila Lathifah lahir di Bukitkemuning pada tanggal 16 Juni 1997. Telah menempuh pendidikan mulai dari TK Muslimin Bukitkemuning 1 tahun, melanjutkan ke SDN 1 bukitkemuning selama 6 tahun, kemudian melanjutkan ke SMPN 1 Bukitkemuning selama 3 tahun, SMAN Bukitkemuning selama 3 tahun. Saat ini penulis sedang melanjutkan pendidikan di Departemen S1 Teknik Elektro Universitas Diponegoro angkatan 2016 mengambil konsentrasi Teknologi Informasi.

Saya menyatakan bahwa segala informasi yang tersedia di makalah ini adalah benar, merupakan hasil karya sendiri, bebas dari plagiat, dan semua karya orang lain telah dikutip dengan benar.

Laila Lathifah
NIM. 21060116130112

Pengesahan

Telah disetujui untuk diajukan pada Sidang Tugas Akhir

Semarang, 22 Desember 2020

Pembimbing 1

Pembimbing 2

Eko Handoyo, S.T., M.T.
NIP. 197506082005011001

Yosua Alvin Adi Soetrisno, S.T., M.Eng.
NIP. H.7.199010132018071001