SISTEM *CRAWLING* DATA INSTRUMEN AKREDITASI BERBASIS SELENIUM DAN PANDAS

Laila Lathifah[*)], Eko Handoyo, dan Yosua Alvin Adi Soetrisno

Departemen Teknik Elektro, Fakultas Teknik, 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 hal nya 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, yaituk 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 yang kinerjanya kurang stabil saat mengelola data, dimana semakin banyak data maka semakin besar pula kecepatan eksekusi dan pengunaan 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 computiton of Diponegoro University students, and Forlap web pages containing data program study and the number of Diponegoro University students. Thesystem 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. Thesystem fitering data using Pandas dataframe that performance is less stabel 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 hal nya 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 (Badan Akreditasi Nasional Perguruan Tinggi) nomor 5 Tahun 2019, yang telah ditetapkan pada tanggal 23 September 2019 pendataan akreditasi di Indonesia dapat dilakukan

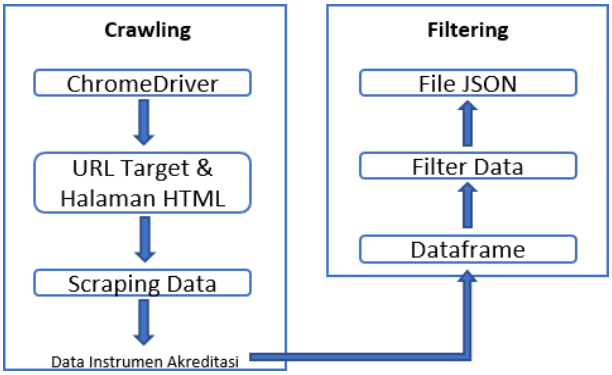
secara daring melalui situs sapto.banpt.or.id [2]. SAPTO (Sistem Akreditasi Perguruan Tinggi Online) merupakan sistem yang dikembangkan BAN-PT untuk meningkatkan efisiensi dan kualitas proses akreditasi perguruan tinggi yang diselenggarakan oleh BAN-PT. SAPTO mendukung setiap proses yang dilakukan dalam 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. [3]

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 librari 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[5], selain itu dapat mengolah suatu data dengan menggunakan operasi seperti split baris, split kolom, hapus data, dan lain-lainnya.

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 diawali dengan aktifnya 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)[6]. 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* berekstensi 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[7]. *Webdriver* pendukung Selenium yang dipakai adalah ChromeDriver untuk mempermudah proses pengambilan data dengan bantuan Chrome Extention tertentu[9].

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* berekstensi .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 penggunanya. 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[10].

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
Processor	Intel(R) Core(TM) i5-2520M
RAM	8192 MB
Harddisk	31 GB
Laptop	Dell Latitude E6320 Core i5

2.2.4. Kebutuhan Perangkat Lunak

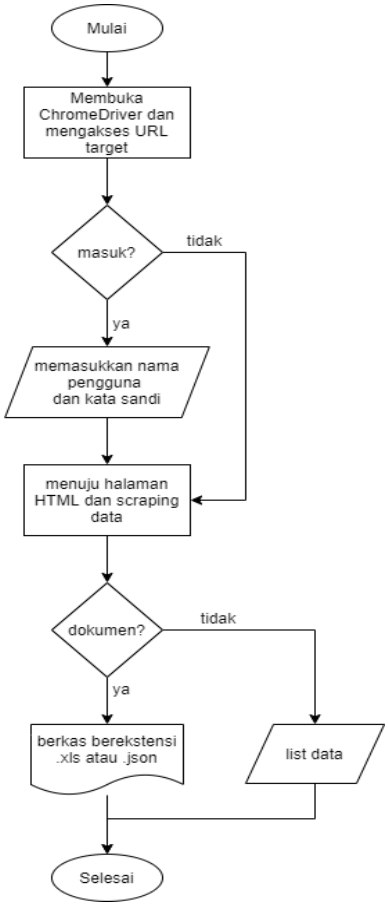
Dalam pembangunan sistem ini, dibutuhkan beberapa spesifikasi perangkat lunak. Spesifikasi perangkat lunak tersebut dapat dimasukkan ke dalam kebutuhan perangkat lunak dalam analisis kebutuhan. Perangkat lunak yang dibutuhkan baik untuk merancang sistem, membangun sistem maupun menjalankan sistem adalah seperti yang ditunjukkan pada Tabel 2 berikut.

Tabel 2. Kebutuhan perangkat lunak [11]

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

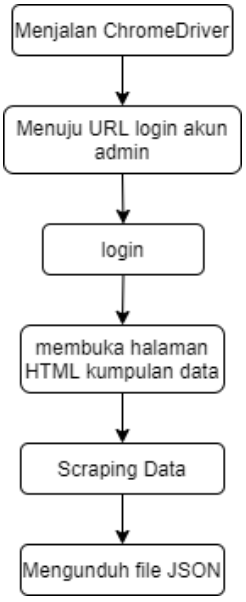
2.3. Perancangan Sistem Web *Crawling*

Sistem Web *Crawling* ini bergantung pada laman web yang akan diambil datanya[8]. Tetapi, proses secara garis besar akan digambarkan menggunakan *flowchart* yang dapat dilihat pada Gambar 2 berikut ini.



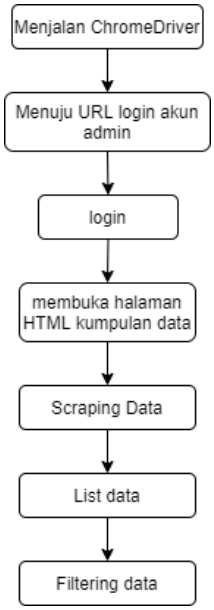
Gambar 2. *Flowchart* perancangan sistem web *crawling* [15]

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



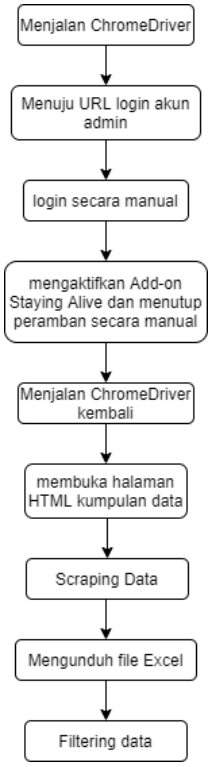
Gambar 3. Diagram alir sistem *crawling* data di laman web Eduk Undip [14]

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



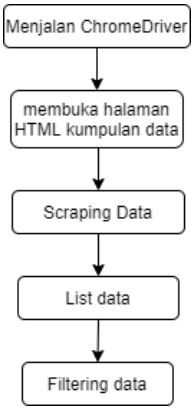
Gambar 4. Diagram alir sistem *crawling* data di laman web Prestasi Undip [14]

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 [14]

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

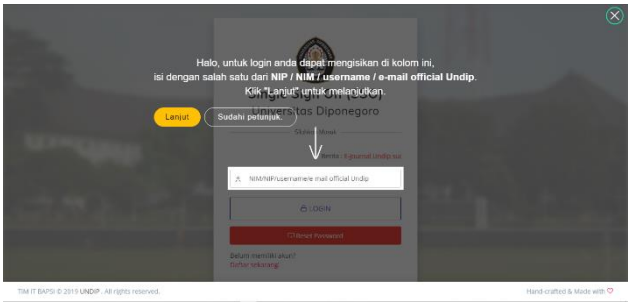


Gambar 6. Diagram alir sistem *crawling* data di laman web Forlap Dikti [14]

3. Hasil dan Pembahasan

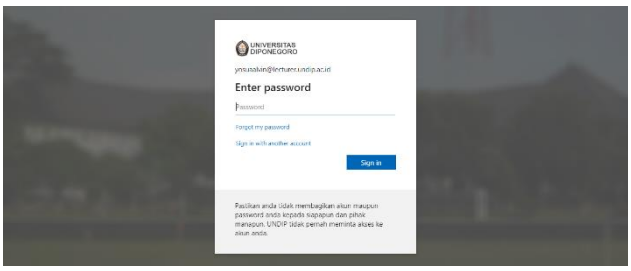
3.1. Implementasi Sistem *Crawling* Data

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



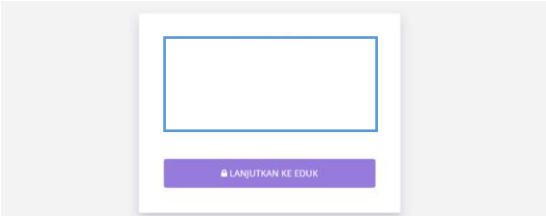
Gambar 7. Tampilan *submit* nama pengguna di laman web SSO (*Single Sign On*)

Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 7, dimana tombol 'Sudahi 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

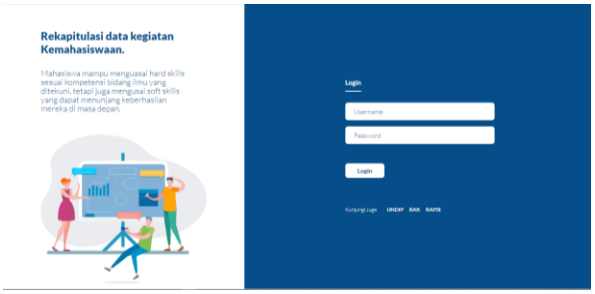
Setalah *submit* nama pengguna peramban akan lanjut ke halaman submit kata sandi seperti pada Gambar 8 Kemudian, program akan memasukkan sandi menggunakan *method* `.send_keys('sometext')` dan `.submit()` sebagai tombol *enter*.



Gambar 9. Tampilan verifikasi menuju laman web Eduk Undip

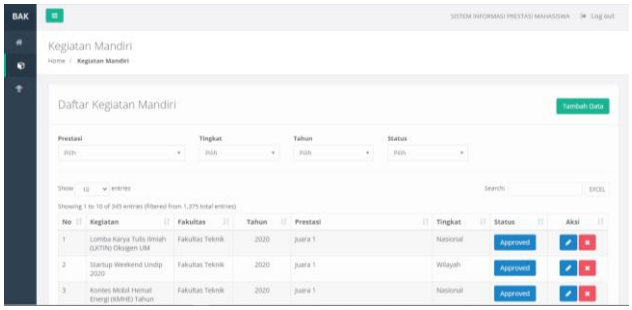
Setalah *submit* kata sandi peramban akan lanjut ke halaman eduk yang memerlukan verifikasi seperti pada Gambar 9 Sebagai verifikasi program akan menekan tombol 'LANJUTKAN KE EDUK' menggunakan *method* `.click()` sebagai tombol *enter*. Kemudian, peramban akan langsung menuju laman pangkalan basis data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



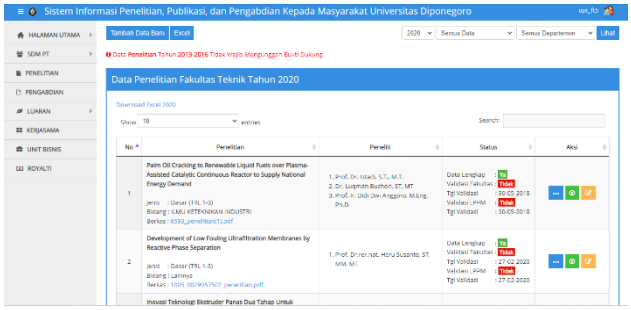
Gambar 11. Tapilan kumpulan data di laman web Prestasi Undip

Setalah *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. Setalah 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

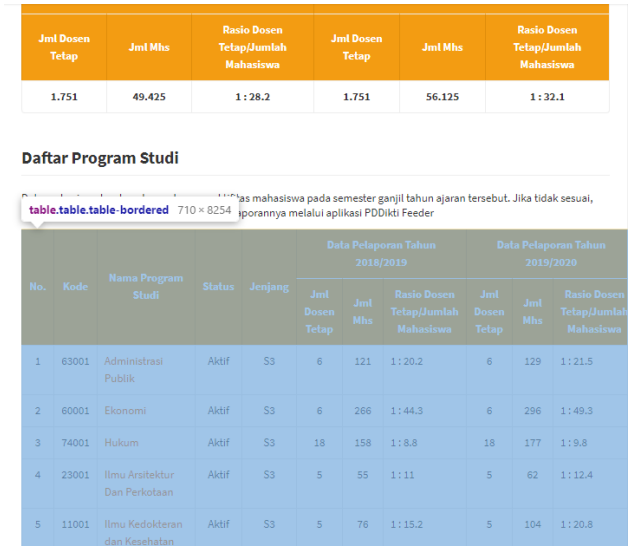
Saat peramban berjalan pertama kali akan menampilkan hasil *request* URL yang dapat dilihat pada Gambar 12, dimana *submit* nama pengguna dan kata sandi menjadi satu halaman serta ada tambahan *submit captcha*. Proses pemasukan *captcha* belum bisa dilakukan secara otomatis karena kata akan muncul secara 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

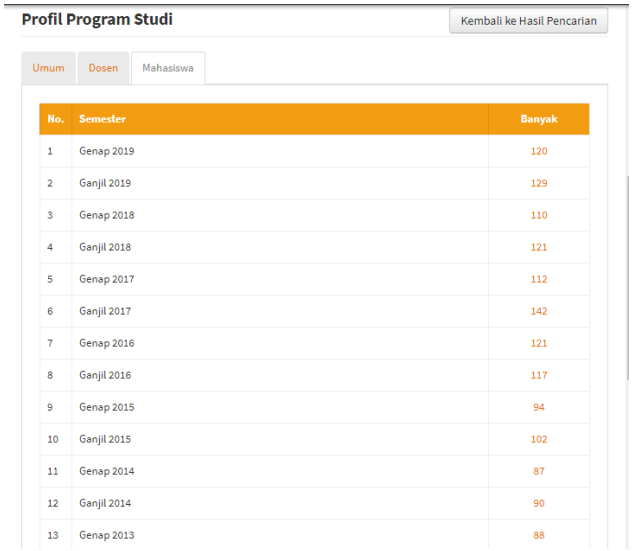
Setalah menutup peramban dilanjutkan dengan menjalan program kedua dan saat peramban berjalan akan langsung menuju halaman kumpulan data seperti pada Gambar 13. Pengumpulan data akan dilakukan dengan mengunduh berkas Excel berdasarkan tahun penelitian. Proses pengunduhan dilakukan dengan pemilihan tahun 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



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*.



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

Setalah pengambilan data daftar program studi selesai, peramban akan langsung mengumpulkan *link* untuk yang terdapat pada daftar nama program studi dan nilai link tersebut akan ditampung 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 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. Selanjutnya, data akan di *filtering* dalam *dataframe*.

3.2. Pengujian Proses *Filtering* Data

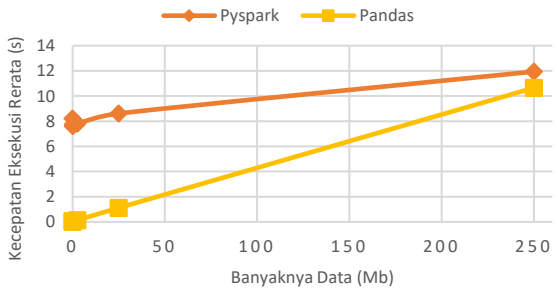
Pengujian dilakukan dengan membandingkan kinerja kecepatan eksekusi dan penggunaan memori oleh Pandas *dataframe* dan Pyspark *dataframe*. Pengujian dilakukan berdasarkan 3 kondisi untuk setiap banyaknya data, yaitu saat 1 apliksasi dijalankan, saat 2 aplikasi dijalankan, dan saat 3 aplikasi dijalankan.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

3.2.1. Pengujian Kecepatan Ekesekusi Proses *Filtering* Data

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

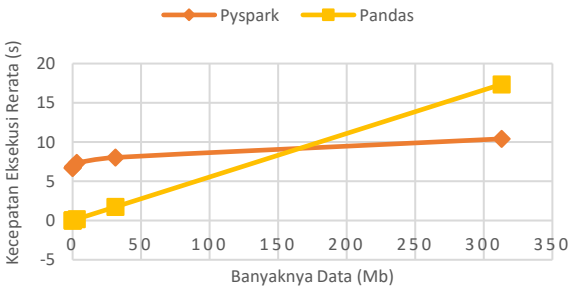
Tabel 3. Hasil pengujian kecepatan eksekusi proses *filtering* data [12]

Laman Web	Banyaknya Data (Mb)	Kecepatan Eksekusi (s)	
		Pyspark	Pandas
Sip3mu LPPM Undip	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
	249,961	11,93711193	10,64237
Prestasi Undip	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
	312,913	10,40634084	17,35626
Forlap Dikti	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
	100,739	10,32327882	1,921089



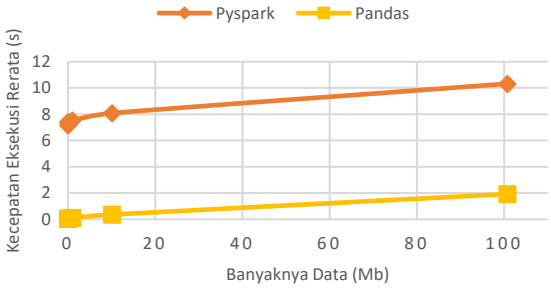
Gambar 16. Grafik perbandingan kecepatan eksekusi pada laman web Sip3mu

Pada Gambar 16 terlihat bahwa penggunaan Pandas *dataframe* lebih cepat dibandingkan Pyspark *dataframe*. Teapi, kenaikan kecepetan 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*.

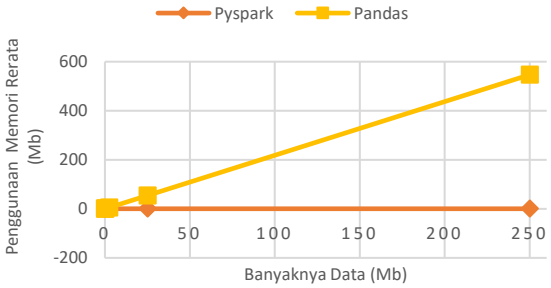
3.2.2. Pengujian Penggunaan Memori Proses *Filtering* Data

Tabel 4. Hasil pengujian penggunaan memori proses *filtering* data [13]

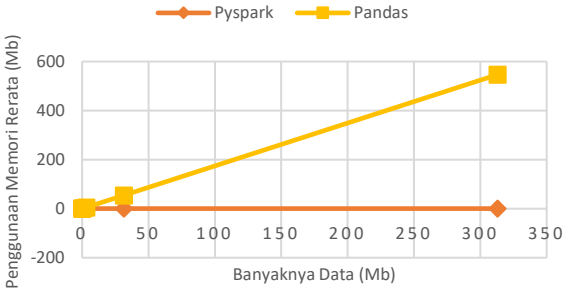
Laman Web	Banyaknya Data (Mb)	Penggunaan Memori (Mb)	
		Pyspark	Pandas
Sip3mu LPPM Undip	0,003	0,425955	0,339076
	0,025	0,425956	0,364903
	0,250	0,412361	0,676763
	2,499	0,425956	3,663401
	24,996	0,425959	33,56355
	249,961	0,42596	267,8953
Prestasi Undip	0,004	0,412361	0,347042
	0,031	0,412362	0,415321
	0,312	0,447895	0,883193
	3,129	0,412362	5,579908
	31,291	0,412365	54,26836
	312,913	0,412366	546,7106
Forlap Dikti	0,002	0,447902	0,367378
	0,011	0,447895	0,381933
	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.

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.

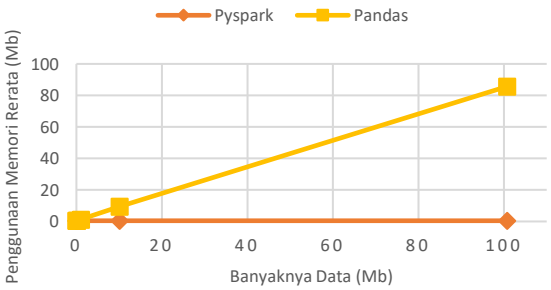


Gambar 19. Grafik perbandingan penggunaan memori pada laman web Sip3mu



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]. O. Leonardo, and H. Maria, “Analytical Data Mart for the Monitoring of University Accreditation Indicators”, IEEE 2019.
[2]. Peraturan BAN-PT nomor 5 Tahun 2019.
[3]. Panduan Penggunaan SAPTO Versi 01 Untuk Pengguna Perguruan Tinggi oleh BAN-PT Tahun 2017.
[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]. https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40 (diakses Oktober 2020).
[6]. https://medium.com/@dede.brahma2/perbedaan-antara-crawling-dan-scraping-98e64e0c6439 (diakses tanggal 19 Oktober 2020).
[7]. VanderPlas, Jake. “Python Data Science Handbook: Essential Tools for Working with Data”. 1005 Gravenstein Highway North, Sebastopol, CA 95472 : O’Reilly Media, Inc. 2017.
[8]. https://prowebscraping.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.