

Terbit online pada laman : <http://teknosi.fti.unand.ac.id/>

# Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Studi Kasus

## Implementasi *Web Scraping* Dalam Pengembangan Aplikasi Super Untuk Menunjang Kegiatan Tugas Akhir

Faridah Hanifah<sup>a</sup>, Alam Rahmatulloh<sup>b, \*</sup>, Rahmi Nur Shofa<sup>c</sup><sup>a,b,c</sup> Program Studi Informatika, Fakultas Teknik, Universitas Siliwangi, Jl. Siliwangi No.24, Kahuripan, Tasikmalaya, 46115, Indonesia

### INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 28 Maret 2025

Revisi Akhir: 18 Desember 2025

Diterbitkan Online: 14 Januari 2026

### KATA KUNCI

*Extreme Programming,*  
 SUPER,  
 SINTA,  
 Tugas Akhir,  
*Web Scraping*

### KORESPONDENSI

E-mail: [alam@unsil.ac.id](mailto:alam@unsil.ac.id)

### A B S T R A C T

Jurusan Informatika Universitas Siliwangi, telah menyediakan sebuah sistem informasi yang bernama SUPER yang mendukung proses tugas akhir, namun fokus aplikasi ini terbatas pada tahapan pemilihan dosen pembimbing. Penelitian ini bertujuan untuk memberikan solusi dalam meningkatkan efisiensi dan efektivitas proses kegiatan tugas akhir di Jurusan Informatika Universitas Siliwangi, dengan mengembangkan aplikasi yang terintegrasi dengan fitur-fitur yang diperlukan. Berdasarkan hasil jawaban kuesioner dari pengguna SUPER bahwa terdapat salah satu fitur yang perlu ditambahkan pada pengembangan aplikasi SUPER yaitu pencarian referensi jurnal dimana data yang diperoleh berasal dari *website* SINTA. Mendapatkan data dari SINTA sebelumnya memungkinkan melalui penggunaan API (*Application Programming Interface*). Namun untuk saat ini SINTA tidak menyediakan API resmi sedangkan API yang sebelumnya sudah tidak dapat diakses. Maka dari itu, solusi yang dapat digunakan untuk memperoleh data yaitu dengan melakukan *web scraping*. Proses *scraping* dilakukan menggunakan program Python, namun perintah untuk melakukan *scraping* data dijalankan melalui aplikasi SUPER. Hasil implementasi *web scraping* menunjukkan bahwa total 884 artikel berhasil diambil, yang kemudian diintegrasikan ke dalam menu referensi jurnal pada aplikasi SUPER. Pengujian dilakukan melalui *Alpha Beta Testing*. *Alpha Testing* dilakukan menggunakan *Black Box* dan didapatkan hasil bahwa semua parameter pada program *scraping* dan menu pada aplikasi yang dikembangkan sudah berjalan dengan baik dan sudah sesuai harapan. Pengujian Beta dilakukan melalui penyebaran kuesioner kepada 84 responden, dengan hasil rata-rata skor SUS sebesar 71,1, menunjukkan bahwa aplikasi mendapat grade C+ dalam kategori baik.

## 1. PENDAHULUAN

Referensi jurnal menjadi salah satu komponen penting dalam proses penelitian yang dapat memberikan dasar teoritis yang kuat serta mendukung argumentasi yang diajukan dalam tugas akhir [1]. Namun, bagi sebagian mahasiswa, mencari referensi jurnal yang relevan dengan topik penelitian dapat menjadi tugas yang rumit dan memakan waktu. Hal ini dikarenakan jumlah jurnal yang sangat banyak dan informasi yang tersebar di berbagai platform, sehingga membutuhkan waktu dan upaya ekstra untuk mengidentifikasi sumber-sumber yang sesuai [2].

Jurusan Informatika Universitas Siliwangi, menyediakan sebuah sistem informasi bernama SUPER, yang mendukung proses tugas akhir. Namun, fokus aplikasi ini terbatas pada tahapan pemilihan dosen pembimbing. Hal ini menyebabkan kebutuhan akan pengembangan aplikasi yang lebih luas dan terintegrasi, yang mampu mencakup seluruh tahapan dalam proses tugas akhir.

Data yang diperoleh dari hasil wawancara menggunakan kuesioner terhadap pengguna SUPER dari 62 responden mahasiswa angkatan 2018 dan 2019 yang mengontrak tugas akhir dalam kurun waktu 1 Januari sampai 5 Agustus 2023, beberapa responden menyatakan perlu adanya fitur yang menampilkan jurnal-jurnal dosen dari berbagai sumber agar memudahkan mahasiswa untuk memastikan relevansi usulan penelitian yang

diajukan dengan dosen pembimbing yang dipilih. Selain itu, responden juga menyatakan perlunya *e-library* untuk membantu mahasiswa tingkat akhir dalam mencari referensi terkait topik tugas akhir. Maka dari itu, salah satu fitur yang harus ditambahkan pada pengembangan aplikasi SUPER yaitu pencarian referensi jurnal yang dapat digunakan sebagai acuan dalam penelitian dan sebagai pertimbangan bagi mahasiswa dalam menentukan calon dosen pembimbing.

Sumber data jurnal yang diperoleh untuk ditambahkan pada aplikasi harus berasal dari platform terpercaya. Salah satu platform terpercaya yang dapat menjadi sumber referensi jurnal adalah SINTA (*Science and Technology Index*). SINTA merupakan sebuah platform yang dikelola oleh Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. Platform ini menyediakan akses ke berbagai jurnal ilmiah yang telah terindeks secara resmi, sehingga kredibilitasnya dapat dipertanggungjawabkan [3]. Mendapatkan data dari SINTA sebelumnya memungkinkan melalui penggunaan API (*Application Programming Interface*). Namun untuk saat ini SINTA tidak menyediakan API resmi sedangkan API yang sebelumnya sudah tidak dapat diakses. Maka dari itu, alternatif umum yang dapat digunakan untuk memperoleh data yaitu dengan melakukan *web scraping*. *Web scraping* merupakan teknik ekstraksi data dari situs web secara otomatis [4].

Terdapat beberapa penelitian terkait dengan topik penelitian yang sama dengan penelitian sebelumnya diantaranya mengenai pengembangan aplikasi pengelolaan tugas akhir untuk Jurusan Teknik Informatika Politeknik Negeri Ketapang. Aplikasi ini berbasis *website* dan memungkinkan administrator mengelola data master dengan efektif. Mahasiswa dapat dengan mudah mengajukan judul tugas akhir secara online, dan dosen dapat memberikan persetujuan pada judul yang diajukan. Meskipun demikian, terdapat kekurangan pada penelitian ini, seperti kurangnya integrasi dengan sistem informasi yang sudah ada dan keterbatasan fungsionalitas pada proses pengajuan judul saja [5]. Penelitian lainnya mengembangkan aplikasi serupa untuk Program Studi Manajemen Informatika AMIK Mahaputra Riau, berbasis android untuk kemudahan akses melalui *smartphone*. Aplikasi ini membantu mahasiswa dalam pengajuan judul dan admin dalam pembuatan jadwal sidang. Namun, kekurangannya adalah tidak adanya akses untuk dosen melihat pengajuan judul mahasiswa dan kurangnya fitur untuk mendukung pengelolaan tugas akhir secara lebih efisien [6]. Terdapat juga penelitian yang menerapkan *web scraping* pada jurnal SINTA untuk mengidentifikasi tren penelitian kesehatan di Indonesia, dengan hasil berupa data dalam format CSV yang berisi informasi tentang topik penelitian dan afiliasi penelitian [7].

Penelitian ini bertujuan untuk memberikan solusi dengan melakukan pengembangan aplikasi pada SUPER agar semua proses tugas akhir terintegrasi dalam satu aplikasi. Pengembangan tersebut melibatkan penambahan fitur-fitur yang mencakup proses kegiatan tugas akhir, seperti administrasi pengajuan sidang, penjadwalan sidang, pelaporan kemajuan penelitian, pencatatan bimbingan, dan pencarian referensi jurnal. Metode pengembangan yang dipilih pada penelitian ini adalah *Extreme Programming* karena fleksibilitasnya dalam menanggapi perubahan kebutuhan selama proses pengembangan [8]. Selain itu, pengembangan aplikasi dilakukan dengan

menggunakan teknologi PWA (*Progressive Web Application*) untuk memberikan pengalaman pengguna yang responsif dan serupa dengan aplikasi *native* di berbagai perangkat dan *browser*, tanpa memerlukan instalasi yang rumit [9].

Selain itu, dilakukan juga pengintegrasian antara SUPER dengan platform SINTA sebagai sumber referensi jurnal yang digunakan. Pengambilan data referensi jurnal dari SINTA dilakukan melalui teknik *web scraping* untuk memperoleh datanya. Data yang diambil dalam penelitian ini terbagi menjadi dua bagian, yaitu pengambilan data profil dosen dan pengambilan artikel dosen. Untuk artikel sendiri, data dibagi menjadi empat bagian, termasuk artikel dari Scopus, *Web of Science*, Google Scholar, dan Garuda. Informasi yang diperoleh mencakup berbagai atribut seperti *sinta id* dosen, foto profil dosen, judul artikel, link artikel, *index*, jurnal, penulis, tahun terbit, dikutip, akreditasi, dan doi.

Proses *scraping* dilakukan dengan menggunakan bahasa pemrograman Python dan memanfaatkan *library* seperti Selenium, BeautifulSoup, dan Flask yang dapat mengolah proses *web scraping*. Identifikasi *tag* HTML yang sesuai dilakukan untuk setiap sumber artikel. Data yang berhasil diekstraksi dari *website* SINTA disimpan dalam *format* JSON yang kemudian data tersebut dimasukkan ke dalam *database* yang akan terhubung dengan sebuah halaman khusus, yaitu halaman referensi jurnal pada aplikasi SUPER yang dikembangkan menggunakan *framework* Laravel.

## 2. METODE

Metodologi penelitian menjadi cara untuk memperoleh kebenaran dalam menggunakan penelusuran yang didasari realitas yang sedang dikaji. Metodologi penelitian berisi tahap-tahap proses penelitian. Metodologi penelitian yang diterapkan pada penelitian ini ditunjukkan pada Gambar 1.

### 2.1. Pengumpulan Data

Tahapan ini adalah tahapan untuk mengumpulkan data dan informasi yang berkaitan dengan objek permasalahan. Pengumpulan data dilakukan dengan dua cara yaitu:

#### 2.1.1. Studi Literatur

Studi literatur dilakukan untuk mencari bahan yang berkaitan dengan objek permasalahan yang berasal dari jurnal dan internet.

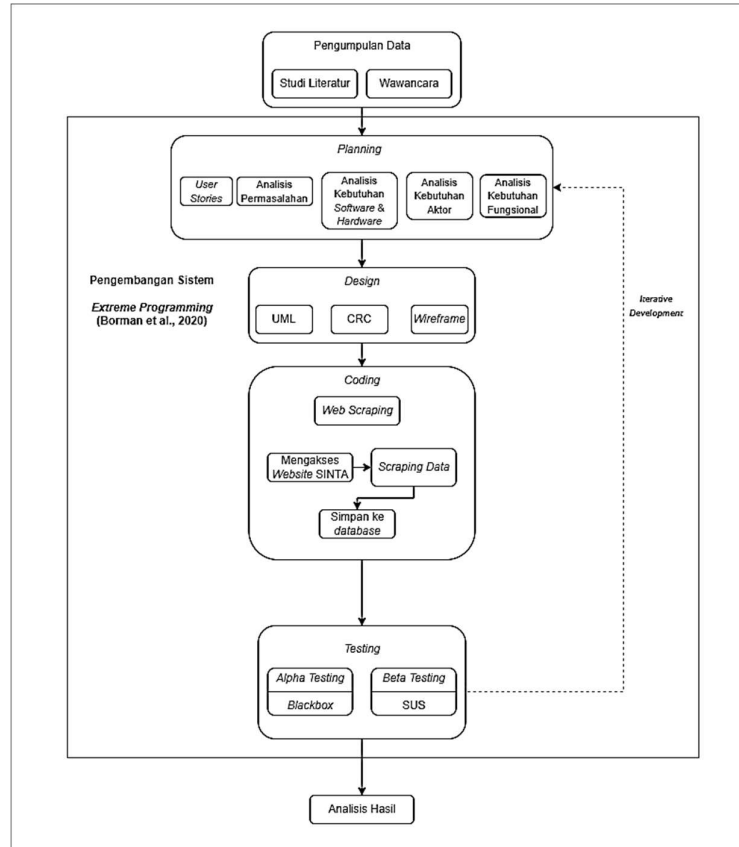
#### 2.1.2. Wawancara

Wawancara dilakukan dengan pengembang aplikasi sebelumnya untuk memperoleh informasi terkait aplikasi SUPER, sementara wawancara dengan pengguna aplikasi SUPER dilakukan untuk memahami kebutuhan dan harapan pengguna terhadap aplikasi SUPER.

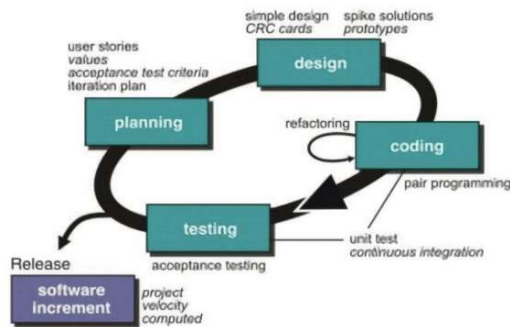
### 2.2. Pengembangan Sistem

Tahapan pengembangan sistem yaitu proses dimana sistem atau aplikasi dikembangkan berdasarkan kebutuhan dan spesifikasi yang telah ditetapkan. Metode yang digunakan dalam

pengembangan sistem yaitu model *extreme programming* dengan tahapan sebagai berikut [10]:



Gambar 1. Metode Penelitian



Gambar 2. Model *Extreme Programming* [10]

### 2.2.1. Planning

Tahapan perencanaan ini menggambarkan langkah-langkah untuk menetapkan tujuan yang diinginkan, dimulai dengan pemahaman tentang konteks bisnis aplikasi, definisi fitur-fitur aplikasi, fungsi yang diinginkan, dan kebutuhan teknologi yang diperlukan untuk pengembangan. Tahap ini juga melibatkan analisis kebutuhan yang mencakup *user stories*, diperoleh melalui kuesioner untuk mengetahui harapan dan kebutuhan pengguna, serta evaluasi setiap cerita pengguna untuk menentukan fitur yang layak dikembangkan dengan mempertimbangkan kompleksitas, urgensi, dampak, serta ketersediaan sumber daya. Selain itu, analisis juga mencakup pemetaan kondisi sistem saat ini, sistem

yang diusulkan, kebutuhan perangkat lunak dan keras, aktor pengguna yang terlibat, dan kebutuhan fungsional aplikasi berdasarkan aktor yang ditentukan.

### 2.2.2. Design

Pada tahap perancangan, fokusnya adalah pada alur proses bisnis dan antarmuka aplikasi. Perancangan melibatkan penggunaan model *Unified Modeling Language* (UML), dengan jenis UML termasuk *use case diagram* untuk menunjukkan fungsi aplikasi dan siapa yang menggunakannya, *sequence diagram* untuk alir kerja setiap fungsi atau menu, serta *class diagram* untuk struktur sistem dalam hal definisi kelas. Selain itu, pemodelan objek menggunakan CRC untuk menggambarkan interaksi kelas, tanggung jawab masing-masing kelas, dan kolaborasi antar kelas [11]. Perancangan antarmuka dilakukan menggunakan *wireframe* dalam bentuk *desktop* dan *mobile* untuk menggambarkan struktur dan tata letak antarmuka pengguna (UI) dalam pengembangan aplikasi SUPER [12].

### 2.2.3. Coding

Tahapan ini merupakan tahapan untuk implementasi analisis dan perancangan yang telah dibuat sebelumnya ke dalam coding menggunakan bahasa pemrograman PHP dengan *framework* Laravel dan database MySQL. Selain itu, dilakukan proses *web scraping* menggunakan bahasa pemrograman Python untuk membuat sebuah API dengan menggunakan *web framework*

Flask. Proses *web scraping* ini terdiri dari beberapa tahapan, dimulai dengan mengakses *website* SINTA menggunakan *library* Selenium untuk mengotomatisasi proses *login*, kemudian melakukan *scraping* data pada halaman tersebut dengan menggunakan *library* BeautifulSoup untuk ekstraksi data dalam format JSON. Setelah data berhasil diekstraksi, data disimpan ke dalam *database* SUPER. Penyimpanan data dilakukan pada tabel-tabel terpisah sesuai dengan sumber data seperti Scopus, *Web of Science*, Garuda, dan Google Scholar, untuk memudahkan proses tampilan data pada aplikasi SUPER.

### 2.2.4. Testing

Pada tahap ini dilakukan *testing* atau pengujian yang bertujuan untuk menilai apakah hasil pengembangan sudah sesuai dengan *requirement*. Pengujian dilakukan dengan menggunakan pengujian alpha dan beta. Pengujian alpha dilakukan menggunakan metode *blackbox testing* untuk menguji fungsionalitas aplikasi [13] dan pengujian beta menggunakan pendekatan SUS untuk melihat *usability* dari sebuah sistem [14].

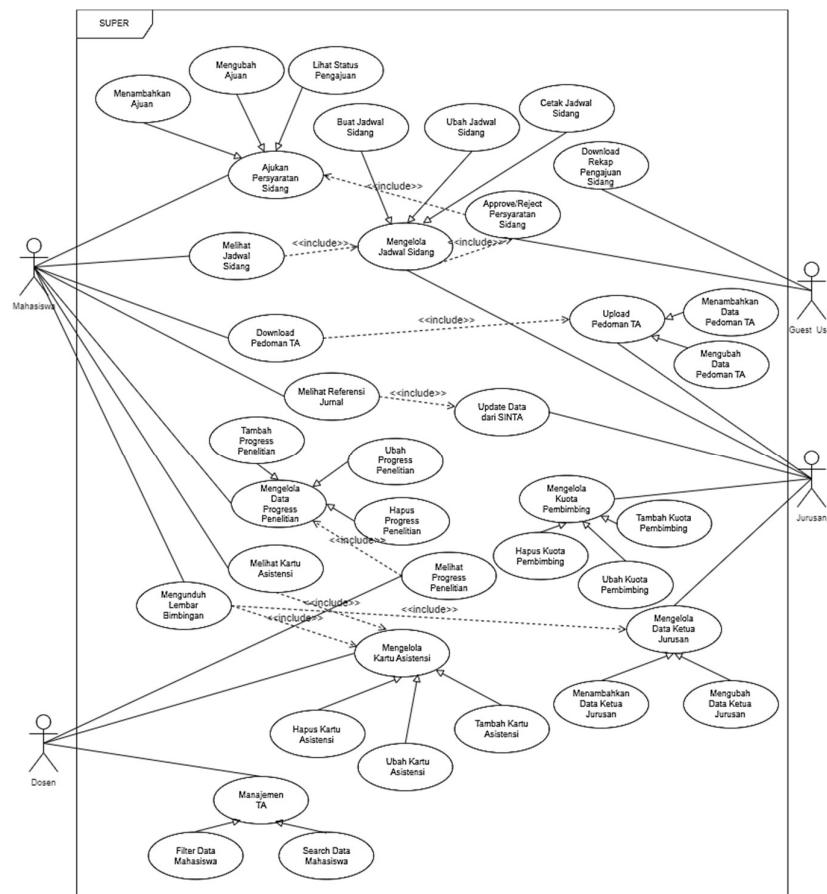
## 2.3. Analisis Hasil

Pada tahap ini dilakukan analisis kinerja sistem yang telah dikembangkan dengan sistem sebelumnya. Analisis dilakukan dengan membandingkan hasil pengujian yang telah dilakukan pada aplikasi. Tahapan ini bertujuan untuk mengetahui perbedaan dalam hal kinerja dan pengalaman pengguna saat menggunakan aplikasi.

## 3. HASIL

### 3.1. Studi Literatur

Hasil studi literatur menunjukkan bahwa penerapan aplikasi pengelolaan tugas akhir di perguruan tinggi memberikan manfaat bagi semua pihak yang terlibat, sehingga pengembangan aplikasi SUPER perlu dilakukan untuk meningkatkan fungsionalitasnya dan memastikan layanan yang lebih baik bagi pengguna. Selain itu, teknik *web scraping* yang telah banyak diterapkan dalam penelitian sebelumnya, dapat digunakan dalam aplikasi SUPER untuk memperoleh data referensi jurnal dari *website* SINTA sebagai alternatif ketika tidak tersedia API resmi.



Gambar 3. Use Case Diagram

## 3.2. Wawancara

### 3.2.1. Wawancara dengan Pengembang

Berdasarkan hasil wawancara dengan pengembang aplikasi SUPER, Alam Rahmatulloh dan Fuji Nugraha, aplikasi SUPER

saat ini menggunakan teknologi PWA (*Progressive Web Apps*) yang memungkinkan akses melalui *web browser* dan instalasi pada perangkat seluler. Teknologi yang digunakan dalam pengembangan aplikasi mencakup bahasa pemrograman PHP, *framework* Laravel 5.8, dan *database* MySQL. Pengembangan aplikasi dilakukan menggunakan metode *agile*, memungkinkan



kolaborasi tim dan adaptasi terhadap perubahan, serta peningkatan kualitas melalui iterasi pengembangan. Selain itu, terdapat perancangan sistem sebelumnya termasuk *use case*, *activity diagram*, *class diagram*, *flowchart*, dan *wireframe*, yang membantu mempermudah pengembangan selanjutnya bagi peneliti.

### 3.2.2. Wawancara dengan Pengguna

Dalam penelitian ini, tanggapan terhadap aplikasi SUPER dikumpulkan melalui wawancara menggunakan kuesioner dengan mahasiswa dan dosen dari Jurusan Informatika Universitas Siliwangi. Teknik *sampling* yang digunakan dalam pengambilan sampel adalah *Purposive Sampling*. Dimana teknik ini melakukan pengambilan sampel dari populasi yang ada berdasarkan karakteristik atau kriteria tertentu yang relevan dengan tujuan penelitian [15]. Sebanyak 62 mahasiswa diantaranya 38 mahasiswa dari angkatan 2018 dan 24 dari angkatan 2019 yang mengontrak Tugas Akhir pada tahun 2023 dalam kurun waktu 1 Januari sampai dengan 5 Agustus, dinyatakan “Belum Lulus” dan telah melaksanakan sidang UP, serta 22 dosen yang dapat dikontrak pada aplikasi SUPER, menjadi sampel penelitian. Hasil kuesioner menunjukkan beragam tanggapan, termasuk kepuasan terhadap kemudahan penggunaan dan pemilihan dosen pembimbing, tetapi juga kendala teknis seperti kebingungan dalam memahami tampilan dan kendala pada server. Pengguna juga mengemukakan harapan terkait peningkatan interaktifitas dan fitur tambahan seperti tutorial penggunaan, forum diskusi, referensi jurnal dan integrasi semua proses kegiatan tugas akhir dalam satu aplikasi.

## 4. PEMBAHASAN

### 4.1. Design

*Use Case Diagram* dibuat untuk menggambarkan fungsi-fungsi yang tersedia dalam aplikasi SUPER dan siapa yang dapat mengaksesnya. Pada Gambar 3, terdapat empat aktor yang berhubungan langsung dengan jalannya proses bisnis pada aplikasi yaitu Mahasiswa, Dosen, Jurusan, dan *Guest User*. Dapat dilihat pada diagram tersebut bahwa Jurusan bertanggung jawab untuk memperbarui referensi jurnal dalam aplikasi SUPER, dimana dalam proses ini *web scraping* diterapkan. Hasil dari *scraping* akan otomatis tersimpan dalam *database* sistem dan nantinya dapat diakses oleh Mahasiswa.

### 4.2. Coding

Setelah tahap analisis dan desain perangkat lunak selesai, langkah selanjutnya adalah mengimplementasikan hasil analisis dan perancangan ke dalam kode program. Berikut merupakan *Source Code* untuk proses *scraping website* SINTA menggunakan bahasa pemrograman *Python*. Gambar 4 merupakan modul yang digunakan untuk proses *scraping*.

```
from flask import Flask, jsonify
from selenium import webdriver
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
import time
import re
```

Gambar 4 Modul yang digunakan untuk proses *scraping*

```
# Fungsi untuk login menggunakan Selenium
def login_sinta(driver, username, password):
    login_url = 'https://sinta.kemdikbud.go.id/logins'
    driver.get(login_url)

    username_elem = driver.find_element(By.NAME, "username")
    password_elem = driver.find_element(By.NAME, "password")

    username_elem.send_keys(username)
    password_elem.send_keys(password)
    password_elem.submit()
    time.sleep(5)
```

Gambar 5 *Source Code Login* ke Website SINTA

Gambar 5 merupakan *source code* untuk proses *login* SINTA. Fungsi *login\_sinta* digunakan untuk melakukan proses *login* ke situs *web* menggunakan Selenium. Program ini membuka URL *login* dari situs SINTA menggunakan *driver.get(login\_url)*. Kemudian, program mencari elemen *username* dan *password* menggunakan *driver.find\_element()* dengan menggunakan metode pencarian *By.NAME*. Setelah menemukan elemen-elemen tersebut, program mengirimkan *username* dan *password* yang diberikan menggunakan *send\_keys()* dan menekan tombol *login* dengan *submit()*, untuk memberi waktu bagi halaman untuk merespons, program melakukan *time.sleep(5)* selama 5 detik.

```
def scrape_data_article_scopus(driver, article_urls):
    # datas.append([total_pages_scopus])

    if total_pages_scopus is None:
        continue

    for page in range(1, total_pages_scopus + 1):
        article_page_url = f"{article_url}?page={page}&view=scopus"
        driver.get(article_page_url)
        soup = BeautifulSoup(driver.page_source, 'html.parser')
        article = soup.findAll('div', 'ar-list-item mb-5')

        for art in article:
            try:
                title_div = art.find('div', 'ar-title')
                title = title_div.a.text.strip()
                link_article = title_div.a['href']
                index = art.find('a', 'ar-quartile').text.strip()
                jurnal = art.find('a', 'ar-pub').text.strip()
                link_jurnal = art.find('a', 'ar-pub')['href']
                creator_element = art.select('div.ar-meta a:nth-of-type(3)')
                creator = creator_element[0].get_text()
                year = art.find('a', 'ar-year').text.strip()
                cited = art.find('a', 'ar-cited').text.strip()

                datas.append([title, link_article, index, jurnal,
                               link_jurnal, creator, year, cited])
            except Exception as e:
                print(f"An error occurred while scraping article: {e}")

    return datas
```

Gambar 6 *Source Code Scraping Artikel Scopus*

Gambar 6 merupakan *source code* dari fungsi *scrape\_data\_article\_scopus* yang bertujuan untuk melakukan *scraping* data dari *website* SINTA yang memuat artikel Scopus dan mengembalikan data yang berhasil diambil dalam bentuk *array*. Pertama, fungsi membuat *array* kosong untuk menampung data yang akan diambil. Selanjutnya, fungsi

melakukan iterasi melalui daftar URL artikel yang diberikan. Iterasi ini dilakukan berdasarkan total halaman artikel Scopus yang didapatkan dari fungsi `get_total_pages_scopus`. Jika total halaman tidak ditemukan atau mengembalikan nilai `None`, iterasi dilanjutkan ke URL berikutnya. Namun, jika total halaman ditemukan, fungsi melakukan iterasi melalui setiap halaman artikel dan melakukan *scraping* data menggunakan BeautifulSoup. Data yang diambil meliputi judul artikel, link artikel, indeks, jurnal, link jurnal, pembuat artikel, tahun penerbitan, dan jumlah sitasi. Data-data ini kemudian ditambahkan ke dalam *array* `datas`. Jika terjadi kesalahan selama proses *scraping*, pesan kesalahan akan dicetak, tetapi proses akan terus berlanjut. Setelah semua data berhasil diambil, *array* `datas` akan dikembalikan dari fungsi.

```
@app.route('/article_scopus')
def get_article_scopus_data():
    # Initialize the WebDriver
    driver = webdriver.Firefox()

    login_sinta(driver, "USERNAME", "PASSWORD")

    scraped_data_article_scopus = scrape_data_article_scopus(driver,
                                                            article_urls)

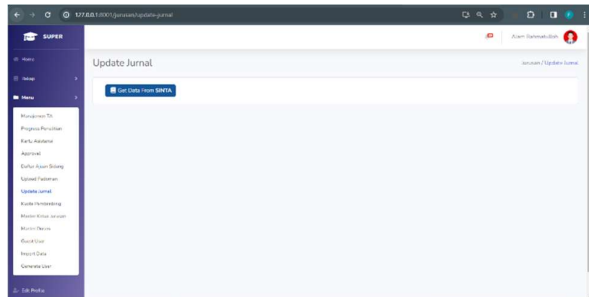
    driver.quit()

    return jsonify(scraped_data_article_scopus)
```

Gambar 7 Source Code Proses Mengambil dan Mengembalikan Data

Gambar 7 merupakan *source code* untuk proses pengambilan dan pengembalian data. Ketika endpoint atau parameter `/article_scopus` diakses, fungsi `get_article_scopus_data()` akan dieksekusi. Awalnya, WebDriver diinisialisasi menggunakan *browser Firefox*. Selanjutnya, fungsi `login_sinta()` dipanggil untuk melakukan proses *login* ke situs web SINTA dengan menggunakan kredensial yang diberikan. Setelah berhasil *login*, fungsi `scrape_data_article_scopus()` dipanggil untuk mengambil data dari situs web terkait artikel Scopus menggunakan *WebDriver* yang sama. Setelah selesai pengambilan data, WebDriver ditutup. Data yang berhasil diambil kemudian dikembalikan dalam format JSON menggunakan fungsi `jsonify()`. Dengan demikian, endpoint ini bertanggung jawab untuk mengambil dan mengembalikan data artikel dari situs Scopus melalui aplikasi *web* menggunakan Selenium dan Flask.

Berikut merupakan hasil dari pengimplementasian *web scraping* yang telah diintegrasikan dengan aplikasi SUPER, khususnya pada menu referensi jurnal yang dapat diakses oleh aktor Jurusan dan Mahasiswa.

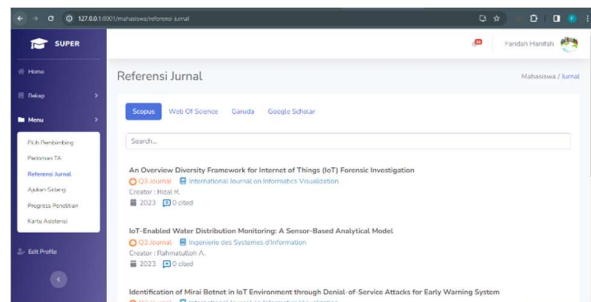


Gambar 8. Tampilan Halaman Untuk Update Referensi Jurnal

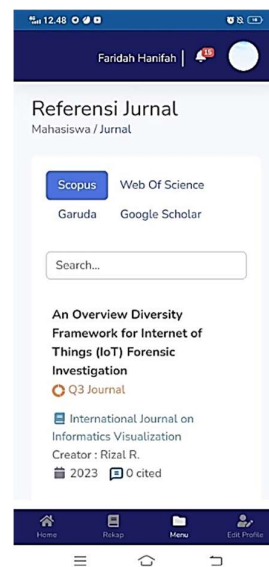
Proses *scraping* dilakukan menggunakan program Python, namun perintah untuk melakukan *scraping* data dijalankan melalui aplikasi SUPER. Seperti yang ditunjukkan pada Gambar

8, halaman *update* referensi jurnal dalam *format desktop* memungkinkan Jurusan untuk memperbarui referensi jurnal SINTA pada aplikasi SUPER dengan menekan tombol "Get Data From SINTA". *Scraping* dilakukan pada situs *web* SINTA untuk mengambil data profil dosen dan artikel dosen, termasuk dari empat sumber yaitu Scopus, *Web of Science*, Garuda, dan Google Scholar. Kode program dibagi menjadi beberapa fungsi agar memudahkan proses *scraping*, karena struktur html yg digunakan berbeda-beda untuk setiap sumbernya. Hasil *scraping* berupa data JSON yang diproses oleh Laravel dan disimpan ke dalam *database*. Dimana jumlah total *rows* pada *database* berjumlah 884 artikel yang didapatkan. Data yang telah ada di *database* kemudian ditampilkan pada menu referensi jurnal bagi aktor Mahasiswa.

Gambar 9 menampilkan halaman referensi jurnal untuk aktor Mahasiswa yang telah diimplementasikan dalam *format desktop*, sementara Gambar 10 menunjukkan tampilannya dalam *format mobile*.



Gambar 9. Tampilan Halaman Referensi Jurnal - Desktop



Gambar 10. Tampilan Halaman Referensi Jurnal – Mobile

Pada halaman referensi jurnal, mahasiswa dapat melihat referensi yang telah disediakan oleh sistem dimana referensi jurnal/artikel ini berasal dari *website* SINTA yang berperan sebagai perantara antara aplikasi SUPER dan berbagai jurnal atau artikel yang tersedia. Sebagai hasilnya, *file* jurnal atau artikel tidak dapat diunduh secara langsung melalui aplikasi SUPER. Referensi ini secara khusus menampilkan artikel penelitian yang dilakukan

oleh dosen Informatika dan Sistem Informasi Universitas Siliwangi. Untuk memastikan relevansi dengan bidang minat mahasiswa, data yang ditampilkan akan disesuaikan. Oleh karena itu, mahasiswa disarankan untuk melengkapi bidang minat pada menu *Edit Profile* agar dapat menampilkan referensi jurnal/artikel yang sesuai. Namun, jika mahasiswa memerlukan referensi yang lebih spesifik dalam bidang minatnya, mahasiswa dapat menggunakan menu pencarian untuk mencari judul artikel yang sesuai dengan penelitian yang dilakukan.

### 4.3. Testing

#### 4.3.1. Alpha Testing

*Alpha Testing* dilakukan dengan menggunakan *Black Box Testing* untuk memastikan setiap parameter pada proses *scraping* dan setiap fungsi pada aplikasi berjalan dengan benar. Pengujian ini dilakukan pada dua jenis perangkat yaitu perangkat *mobile* dan perangkat *laptop*.

Tabel 1 merupakan hasil pengujian *alpha* menggunakan *black box testing* pada setiap parameter *scraping*.

Tabel 1. Hasil *black box testing* pada setiap parameter *scraping*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menjalankan url berikut: <a href="http://127.0.0.1:5000/profile">http://127.0.0.1:5000/profile</a>	Sistem akan melakukan pengambilan data <i>profile</i> dosen dari <i>website</i> SINTA dan menampilkan hasilnya	Sesuai Harapan	Diterima
2	Menjalankan url berikut: <a href="http://127.0.0.1:5000/article_scopus">http://127.0.0.1:5000/article_scopus</a>	Sistem akan melakukan pengambilan data artikel <i>scopus</i> dari <i>website</i> SINTA dan menampilkan hasilnya	Sesuai Harapan	Diterima
3	Menjalankan url berikut: <a href="http://127.0.0.1:5000/article_wos">http://127.0.0.1:5000/article_wos</a>	Sistem akan melakukan pengambilan data artikel <i>wos</i> dari <i>website</i> SINTA dan menampilkan hasilnya	Sesuai Harapan	Diterima
4	Menjalankan url berikut: <a href="http://127.0.0.1:5000/article_garuda">http://127.0.0.1:5000/article_garuda</a>	Sistem akan melakukan pengambilan data artikel <i>garuda</i> dari <i>website</i> SINTA dan menampilkan hasilnya	Sesuai Harapan	Diterima
5	Menjalankan url berikut: <a href="http://127.0.0.1:5000/article_google_scholar">http://127.0.0.1:5000/article_google_scholar</a>	Sistem akan melakukan pengambilan data artikel <i>google scholar</i> dari <i>website</i> SINTA dan menampilkan hasilnya	Sesuai Harapan	Diterima

Hasil pengujian *alpha* menggunakan *Black Box Testing* menunjukkan bahwa semua parameter dalam proses *scraping*, serta semua halaman aplikasi yang ditujukan untuk setiap aktor, berfungsi sesuai dengan harapan.

#### 4.3.2. Beta Testing

Populasi dan sampel yang digunakan pada proses pengujian *beta testing* merupakan populasi dan sampel yang digunakan para proses *user stories*. Responden yang terlibat merupakan dosen dan mahasiswa dari Jurusan Informatika Universitas Siliwangi yang telah menggunakan aplikasi SUPER sebelumnya, dengan total sampel sebanyak 84 orang. Pengujian ini dilakukan dengan penyebaran kuesioner *System Usability Scale* (SUS) kepada responden sebelum aplikasi didistribusikan. Dari 84 responden yang mengisi kuesioner, skor SUS yang diperoleh adalah 71,1. Berdasarkan hasil ini, aplikasi SUPER dinilai dapat diterima (*Acceptable*) dengan Grade "C+" dalam kategori baik (*Good*).

### 4.4. Analisis Hasil

Berdasarkan hasil pengujian pada aplikasi SUPER yang telah dikembangkan, umpan balik dari pengguna sangat positif. Pengguna mengapresiasi integrasi semua tahapan untuk tugas akhir dalam satu sistem dengan adanya beberapa fitur baru yang dapat memudahkan pengguna khususnya yaitu fitur referensi jurnal. Meskipun demikian, beberapa saran diberikan untuk meningkatkan pengalaman pengguna, seperti peningkatan dalam tampilan antarmuka pengguna (UI), menambahkan kolom komentar pada fitur progres penelitian, *live chat*, penyesuaian kuota pembimbing dengan jumlah mahasiswa yang mengontrak Tugas Akhir, kolom catatan hasil UP untuk memastikan judul skripsi berlanjut atau tidaknya atau mengalami perubahan judul/data dan meningkatkan integrasi dengan sistem-sistem yang sudah ada di lingkungan Unsil.

## 5. KESIMPULAN

Pengembangan aplikasi SUPER telah berhasil dilakukan dengan menerapkan model pengembangan *Extreme Programming* dan memanfaatkan PHP dan Python sebagai bahasa pemrograman utama, termasuk proses *scraping* pada *website* SINTA. Setiap fitur dirancang sesuai dengan level pengguna, memungkinkan interaksi yang lebih baik antar pengguna dengan tingkat akses yang berbeda. Pengembangan ini menghasilkan tampilan *desktop* dan *mobile*, dengan menggunakan teknologi PWA untuk memberikan pengalaman pengguna yang optimal saat mengakses aplikasi melalui perangkat seluler. Pengujian aplikasi dilakukan dengan metode *alpha* dan *beta testing* serta pendekatan *black box* dan SUS. Hasil pengembangan menunjukkan umpan balik positif dari pengguna, dengan setiap fungsi berjalan dengan baik dan sudah sesuai dengan harapan. Pada menu referensi jurnal, teknik *web scraping* pada *website* SINTA berhasil mengumpulkan data profil dosen dan artikel dosen dari berbagai sumber seperti Scopus, *Web of Science*, Garuda, dan Google Scholar. Total artikel yang diperoleh dari hasil *web scraping* berjumlah 884, yang kemudian ditambahkan ke dalam *database* aplikasi SUPER untuk membantu mahasiswa menemukan dosen pembimbing yang relevan dan untuk memperkuat argumen dalam penulisan

laporan TA. Skor SUS yang diperoleh dari pengujian *beta* sebesar 71,1 menunjukkan penerimaan yang baik dalam *grade* "C+".

## DAFTAR PUSTAKA

- [1] F. Lubis *et al.*, "Analysis the Role of References in Scientific Articles: Influence on Research Credibility and Impact," *Formosa J. Sci. Technol.*, vol. 2, no. 11, pp. 3065–3074, 2023, doi: [10.55927/fjst.v2i11.6822](https://doi.org/10.55927/fjst.v2i11.6822).
- [2] I. D. A. M. Budhyani and M. D. Angendari, "Kesulitan dalam Menulis Karya Ilmiah," *Mimb. Ilmu*, vol. 26, no. 3, p. 400, 2021, doi: [10.23887/mi.v26i3.40678](https://doi.org/10.23887/mi.v26i3.40678).
- [3] A. Saputra, "Pemanfaatan Science and Technology Index (SINTA) untuk Publikasi Karya Ilmiah dan Pencarian Jurnal Nasional Terakreditasi," *Media Pustak.*, vol. 27, no. 1, pp. 56–68, 2020, [Online]. Available: <https://ejournal.perpusnas.go.id/mp/article/view/674/pdf>
- [4] S. I. Nurhafida, F. Sembiring, and W. Jatmiko, "Optimasi Query Database Webscraping Pada Jurnal Sinta," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2521–2532, 2022, doi: [10.35957/jatisi.v9i3.2806](https://doi.org/10.35957/jatisi.v9i3.2806).
- [5] E. Wahyudi and I. Pratiwi, "Sistem Informasi Pengelolaan Tugas Akhir Mahasiswa dan Jurnal Penelitian Internal Jurusan Teknik Informatika Politeknik Negeri Ketapang," *Petir*, vol. 13, no. 1, pp. 12–20, 2020, doi: [10.33322/petir.v13i1.766](https://doi.org/10.33322/petir.v13i1.766).
- [6] J. Simatupang and M. Muhammad, "Sistem Aplikasi Pengelolaan Tugas Akhir Berbasis Mobile," *It J. Res. Dev.*, vol. 3, no. 2, pp. 66–75, 2019, doi: [10.25299/itjrd.2019.vol3\(2\).2339](https://doi.org/10.25299/itjrd.2019.vol3(2).2339).
- [7] Y. Sahria, "Implementasi Teknik Web Scraping pada Jurnal SINTA Untuk Analisis Topik Penelitian Kesehatan Indonesia," *URECOL (Unversity Res. Colloquium)*, pp. 297–306, 2020, [Online]. Available: <http://repository.urecol.org/index.php/proceeding/article/view/1079>
- [8] S. D. Pohan and I. Firdaus, "Implementation of Extreme Programming Method in the Development of Pekanbaru Community Training Information System," *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, p. 20, 2022, doi: [10.22373/cj.v6i1.11851](https://doi.org/10.22373/cj.v6i1.11851).
- [9] M. F. Arsa, A. S. Abdullah, and J. Rejito, "Pengembangan Sistem Informasi Geografis Kebun Binatang Berbasis Progressive Web Application (PWA) dengan Metode Prototype (Studi Kasus Kebun Binatang Bandung)," *J. Nas. Teknol. dan Sist. Inf.*, vol. 7, no. 3, pp. 119–129, 2021, doi: [10.25077/teknosi.v7i3.2021.119-129](https://doi.org/10.25077/teknosi.v7i3.2021.119-129).
- [10] R. I. Borman, A. T. Priandika, and A. R. Edison, "Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 3, p. 272, 2020, doi: [10.26418/justin.v8i3.40273](https://doi.org/10.26418/justin.v8i3.40273).
- [11] R. A. S. and M. Salahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. 2016.
- [12] B. D. Jaya and E. P. Agustini, "Perancangan Wireframe User Interface Semester Baca," *J. Comput. Sci.*, vol. 4, no. 3, pp. 689–703, 2022, [Online]. Available: [https://scholar.google.com/citations?view\\_op=view\\_citation&hl=en&user=-b8Y2tgAAAAJ&pagesize=100&citation\\_for\\_view=-b8Y2tgAAAAJ:NaGl4SEjCO4C](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=-b8Y2tgAAAAJ&pagesize=100&citation_for_view=-b8Y2tgAAAAJ:NaGl4SEjCO4C)
- [13] Y. F. Achmad and A. Yulfitri, "Pengujian Sistem Pendukung Keputusan Menggunakan Black Box Testing Studi Kasus E-Wisudawan Di Institut Sains Dan Teknologi Al-Kamal," *J. Ilmu Komput.*, vol. 5, p. 42, 2020.
- [14] E. Kaban, K. Candra Brata, and A. Hendra Brata, "Evaluasi Usability Menggunakan Metode System Usability Scale (SUS) Dan Discovery Prototyping Pada Aplikasi PLN Mobile (Studi Kasus PT. PLN)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 10, pp. 3281–3290, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [15] A. Fauzy, *Metode Sampling*, vol. 9, no. 1. 2019. [Online]. Available: <http://jurnal.globalhealthsciencegroup.com/index.php/JPPP/article/download/83/65%0Ahttp://www.embase.com/search/results?subaction=viewrecord&from=export&id=L603546864%5Cnhttp://dx.doi.org/10.1155/2015/420723%0Ahttp://link.springer.com/10.1007/978-3-319-76>