

PEMBANGUNAN APLIKASI PENGUMPUL BERITA DARI MEDIA DARING MENGGUNAKAN *WEB FRAMEWORK* CODEIGNITER DAN FLASK

Salim Satriajati¹, Ibnu Santoso²

^{1,2} Program Studi D4 Komputasi Statistik, Politeknik Statistika STIS
Jl. Otto Iskandardinata No.64C Jakarta Timur, DKI Jakarta, Indonesia

¹ 221709994@stis.ac.id

² ibnu@stis.ac.id

Abstrak— Kemajuan teknologi informasi dan komunikasi saat ini menyebabkan akses terhadap berita menjadi lebih cepat. Pengumpulan berita dari media daring menjadi hal yang penting untuk menunjang berbagai kepentingan. Pengumpulan berita juga perlu dilakukan secara otomatis supaya efisien dan efektif. Maka, dalam penelitian ini akan dibangun sebuah aplikasi pengumpul berita untuk mendapatkan berita dari media daring. Aplikasi dibangun dengan menggunakan dua buah *web framework*, yakni Codeigniter dan Flask. Selain itu, juga digunakan *package* Scrapy yang berbasis Python sebagai alat untuk melakukan *web crawling* dan *web scraping*. *Black-box testing* digunakan untuk mengevaluasi fungsionalitas sistem. Berdasarkan hasil dari *black-box testing*, dapat disimpulkan bahwa seluruh fungsi pada aplikasi pengumpul berita dari media daring yang telah dibuat dapat berjalan sesuai harapan Peneliti.

Kata kunci— berita, *web scraping*, *web crawling*, Flask, Codeigniter, Scrapy

I. PENDAHULUAN

Saat ini, dunia sedang berada pada era dimana teknologi informasi dan komunikasi berkembang sangat pesat. Perkembangan teknologi informasi dan komunikasi ditandai dengan semakin mudahnya masyarakat dalam mengakses internet. Internet kini seakan menjadi kebutuhan pokok yang dibutuhkan masyarakat untuk menunjang kegiatannya, baik di bidang bisnis, ekonomi, sosial, pendidikan, dan bidang lainnya. Mencari sumber dan literatur pembelajaran, mengakses informasi dan menggunakan media sosial adalah beberapa dari kegiatan masyarakat yang tentunya membutuhkan koneksi internet [1].

Berkembangnya teknologi internet tersebut mengakibatkan banyak dari sendi kehidupan masyarakat yang perlu melakukan transformasi. Transformasi dilakukan dari yang sebelumnya mengandalkan bentuk konvensional kini perlu diubah menjadi bentuk digital [2]. Salah satu dari bentuk transformasi digital adalah adanya media berita daring atau *online*. Saat ini, media daring sudah semakin banyak ragamnya, seiring kebutuhan masyarakat akan informasi yang dapat diakses dengan cepat [3]. Jika sebelumnya berita didapatkan melalui televisi, radio, maupun media cetak seperti koran, majalah, tabloid dan sebagainya, kini masyarakat dapat mengakses berita hanya bermodalkan gawai dan koneksi

internet. Beragam jenis media dapat menyediakan berita terkini secara cepat, sehingga memudahkan masyarakat dalam mengakses fenomena dan kejadian yang terjadi di tengah-tengah mereka. Jenis media yang beragam tersebut berdasar pada fakta, dimana jumlah perusahaan pers yang terdaftar dalam situs Dewan Pers (dewanpers.or.id) dan bergerak pada jenis media daring adalah berjumlah 795 perusahaan pers [4].

Berita dari media daring yang melimpah ruah tersebut tentunya perlu dimanfaatkan dengan baik. Kumpulan berita tersebut dapat digunakan untuk berbagai kepentingan atau hanya sekadar sebagai dokumentasi. Kumpulan berita juga dapat dianalisis lebih lanjut, misalnya analisis sentimen dan *topic modelling* terhadap fenomena tertentu. Maka dari itu, pengumpulan berita dapat menjadi kegiatan yang penting untuk menunjang berbagai kepentingan tersebut.

Pengumpulan berita dari media daring sebenarnya dapat dilakukan secara manual, baik itu dengan melakukan *copy-paste* terhadap artikel berita, maupun menyimpannya langsung dari peramban atau *browser* ke dalam format .pdf. Namun, pengumpulan berita secara manual tentunya tidak efisien dan efektif, baik dari segi tenaga, waktu bahkan biaya.

Berdasarkan hal tersebut, maka diperlukan suatu cara pengumpulan berita yang dilakukan secara otomatis dengan memanfaatkan teknik khusus. Teknik khusus tersebut yakni *web scraping* dan *web crawling*.

Web crawling adalah suatu tindakan yang dilakukan untuk mengumpulkan informasi dari internet [5]. *Web crawling* memanfaatkan program yang disebut dengan *web crawler*. *Web crawling* mengumpulkan link URL (*Uniform Resource Locator*) dari satu atau beberapa situs web yang kemudian dikelola dalam bentuk indeks dan disimpan dalam *database*. Salah satu contoh dari kegiatan *web crawling* adalah pengumpulan link URL dari situs pencarian (*search engine*) yang mengandung berbagai situs web dan pengumpulan link URL berita dari kolom pencarian di situs berita daring yang mengandung artikel berita.

Web crawling seringkali disamakan dengan *web scraping*. Namun keduanya memiliki pengertian dan praktek yang berbeda. Jika proses *web crawling* diawali dari satu set URL dimana URL tersebut terhubung dengan URL lainnya, maka *web scraping* hanya menyarikan sebagian data yang dibutuhkan dari suatu situs web [6].

Web scraping dilakukan secara terotomatisasi dengan memanfaatkan mesin atau robot yang dikenal dengan sebutan *web scraper*. *Web scraper* akan menjelajahi dokumen HTML dalam situs web, kemudian mengekstraksi dan mentransformasinya menjadi bentuk yang lebih terstruktur dan dapat digunakan.

Dengan mengimplementasikan teknik *web crawling* dan *web scraping*, kegiatan pengumpulan berita dari media daring dapat dilakukan secara cepat dan tentunya efektif dan efisien.

Maka, berdasarkan latar belakang di atas, selanjutnya dirumuskan tujuan penelitian, yakni (1) merancang pengumpul berita untuk mendapatkan berita dari media daring dan (2) membangun aplikasi antar muka pengumpul berita dari media daring berbasis web.

II. METODOLOGI PENELITIAN

A. Ruang Lingkup Penelitian

Penelitian ini akan menghasilkan artefak berupa sebuah aplikasi yang dapat melakukan pengumpulan berita, dimana berita dikumpulkan dari media daring berdasarkan kata kunci yang diinput oleh pengguna.

Berita yang dapat dikumpulkan dibatasi dari empat situs media daring yakni Detik, Kompas, Antara dan Bisnis.com. Jumlah berita yang dapat dikumpulkan juga dibatasi, yakni seluruh berita yang muncul pada halaman pertama pencarian kata kunci pada masing-masing situs media daring.

Entitas yang diambil dari situs media daring dapat dilihat pada Tabel I di bawah ini.

TABEL I
ENTITAS YANG DIAMBIL DARI SITUS MEDIA DARING

No.	Entitas	Keterangan
1.	Judul	Judul dari berita
2.	Konten	Konten atau isi dari berita
3.	Tanggal	Tanggal berita tersebut dimuat pada media daring
4.	Link	URL berita

B. Metode Analisis

Metode analisis dalam penelitian ini digunakan sebagai dasar prosedur dalam pembuatan aplikasi, mulai dari identifikasi masalah sampai implementasi sistem hingga penarikan kesimpulan. Adapun metode analisis yang digunakan dalam penelitian ini adalah metode *Design-Oriented Research*. *Design-Oriented Research* berfokus pada evaluasi yang dilakukan pada artefak hasil perancangan [7]. Artefak harus memenuhi seperangkat kriteria demi memenuhi kepuasan pengguna dan pemangku kepentingan.

Penyelesaian penelitian ini akan melalui lima tahap sebagai berikut:

1) *Awareness of Problem*: Identifikasi masalah dilakukan dengan menemukan permasalahan dalam hal pengumpulan berita dari media daring.

2) *Suggestion*: Berdasarkan analisis permasalahan dan juga analisis kebutuhan, selanjutnya akan dirancang sistem usulan berupa aplikasi pengumpul berita. Perancangan dibuat dalam bentuk diagram berupa rancangan arsitektur sistem, *use case diagram* dan *activity diagram*. Selain itu juga dibuat rancangan antarmuka sistem.

3) *Development*: Dalam pengembangan sistem ini, digunakan beberapa alat. Bahasa pemrograman yang digunakan di sisi *backend* adalah Python. Python dimanfaatkan dalam penggunaan web framework Flask dan package Scrapy dalam pembuatan aplikasi pengumpul berita. Scrapy merupakan salah satu *library* berbasis Python yang dapat digunakan dalam *web crawling* dan juga *web scraping*. Scrapy dapat digunakan untuk berbagai tujuan, diantaranya yakni proses *data mining* hingga *monitoring* dan pengujian otomatis (*automated testing*) [8]. Sedangkan, implementasi sistem di sisi *frontend* menggunakan HTML, CSS, Javascript dengan memanfaatkan *web framework* Codeigniter yang berbasis PHP.

4) *Evaluation*: Evaluasi dilakukan untuk memastikan sistem sudah berjalan. Dalam evaluasi sistem, penelitian ini akan menggunakan *black-box testing*. Pengujian dengan *Black-box testing* berfokus pada spesifikasi atau persyaratan fungsional yang terdapat pada perangkat lunak [9]. Karena pengujiannya berfokus pada spesifikasi dan fungsionalitas sistem, *black-box testing* tidak memperhatikan struktur internal dari sistem yang dibuat. *Black-box testing* dibuat dalam bentuk tabel yang memuat deskripsi pengujian atau daftar fungsi apa saja yang diuji, hasil yang diharapkan dari fungsi yang diuji dan hasil aktual. Ketika fungsi yang dijalankan sesuai dengan hasil yang diharapkan, maka hasil aktual diisi dengan kata "Berhasil". Sebaliknya, ketika fungsi yang dijalankan tidak sesuai dengan hasil yang diharapkan, maka hasil aktual diisi dengan kata "Tidak Berhasil".

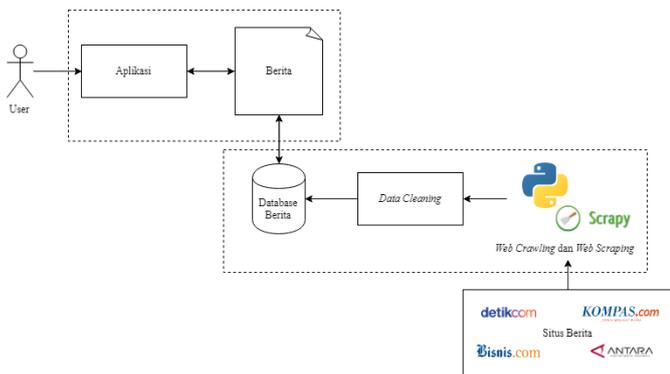
5) *Conclusion*: Dalam tahap ini, dilakukan penarikan kesimpulan apakah sistem berjalan sudah memenuhi harapan atau belum. Selain kesimpulan, juga dipaparkan saran agar

dapat dimanfaatkan dalam pengembangan sistem pada penelitian selanjutnya.

III. HASIL DAN PEMBAHASAN

A. Rancangan Arsitektur Sistem

Rancangan arsitektur sistem pada aplikasi pengumpul berita dari media daring dapat dilihat pada Gambar 1. di bawah ini.

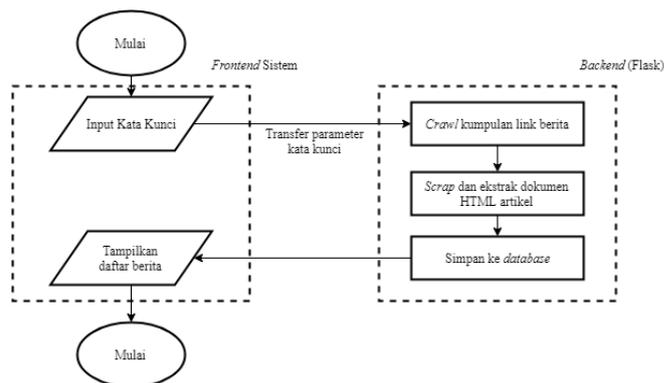


Gambar 1. Rancangan arsitektur sistem

Web framework yang digunakan dalam penelitian ini yakni Codeigniter dan Flask. CodeIgniter digunakan untuk merancang aplikasi berbasis web, sedangkan Flask digunakan untuk proses pada backend, termasuk di dalamnya terdapat modul pengumpul berita.

Berdasarkan Rancangan Arsitektur Sistem pada Gambar 1. di atas, proses pada sistem diawali oleh User. User mengakses aplikasi untuk mendapatkan berita. Untuk mendapatkan berita, User terlebih dahulu mengakses aplikasi pengumpul berita dengan melakukan input kata kunci berita. Kemudian dari kata kunci tersebut akan muncul daftar berita.

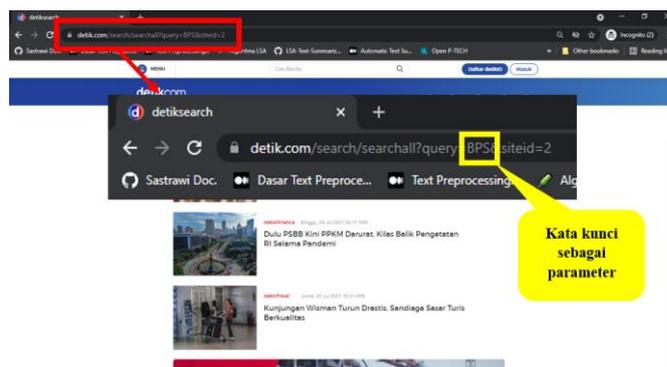
Modul pengumpul berita sendiri dibuat dengan memanfaatkan Python dan package Scrapy. Modul pengumpul berita mengumpulkan berita berdasarkan kata kunci yang diinput oleh User. Kata kunci tersebut kemudian menjadi parameter pada URL dalam proses web crawling untuk mendapatkan URL berita, dimana URL berita yang didapat adalah kumpulan berita berdasarkan kata kunci pencarian pada halaman pertama. URL berita tersebut digunakan untuk proses web scraping dengan mengambil data sesuai entitas yang telah ditetapkan sebelumnya, yakni judul berita, tanggal berita dan konten berita. Rancangan arsitektur modul pengumpul berita dapat dilihat pada Gambar 2. di bawah ini.



Gambar 2. Rancangan arsitektur modul pengumpul berita

Dalam modul pengumpul berita, terdapat web crawler dan web scraper di dalamnya. Web crawler digunakan untuk mendapatkan URL berita berdasarkan kata kunci yang diinput oleh User. Sedangkan web scraper digunakan untuk mendapatkan konten atau entitas dari dokumen HTML pada situs berita.

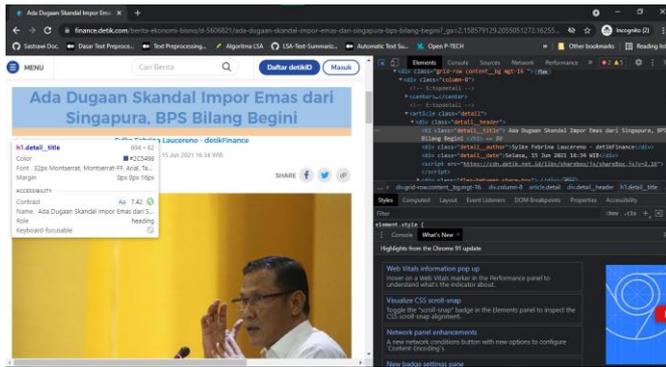
Kata kunci yang diinput oleh user, misalnya “BPS” akan menjadi parameter untuk mendapatkan artikel berita dari media daring terkait kata kunci tersebut. Contohnya pada situs berita Detik, parameter “BPS” akan tampak seperti pada Gambar 3 di bawah ini.



Gambar 3. Parameter kata kunci pada situs berita Detik

Sedangkan, dalam pembuatan web scraper, peneliti harus mengidentifikasi dan mempelajari struktur dokumen HTML pada situs berita terlebih dahulu serta menyesuaikan dengan entitas dari artikel berita yang ingin didapatkan [10]. Hal ini bertujuan agar hasil yang didapatkan dari proses web scraping dapat tepat guna. Kegiatan identifikasi ini dapat memanfaatkan fitur inspect element dalam browser.

Gambar 4. di bawah ini adalah adalah contoh identifikasi dokumen HTML pada salah satu artikel di situs berita Detik. Misalnya, entitas yang ingin didapatkan adalah judul berita, maka dapat dilihat bahwa judul berita pada artikel di bawah dilingkupi oleh tag <h1> dengan class bernama “detail__title”.



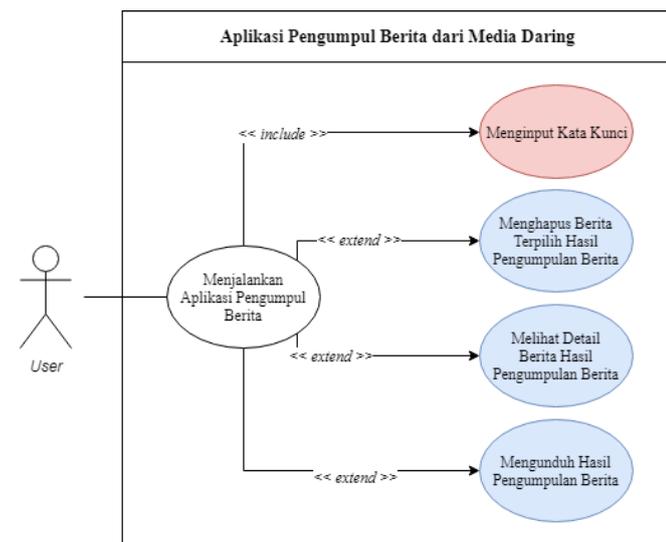
Gambar 4. Contoh identifikasi dokumen HTML artikel

Setelah dokumen HTML selesai diidentifikasi, selanjutnya buat *web scraper* untuk mendapatkan seluruh entitas yang diinginkan. *Web scraper* dibuat dalam bentuk *spyder*. Lalu, ulangi hal yang sama pada situs media daring lainnya, yakni Antara, Kompas, dan Bisnis.com. *Web scraper* yang digunakan untuk mengambil entitas pada artikel di situs berita yang satu tidak dapat digunakan pada situs berita lainnya karena struktur dokumen HTML yang tentunya berbeda antar situs berita.

Web crawler dan *web scraper* yang telah dibuat, perlu diatur agar berita yang berhasil dikumpulkan dapat disimpan ke dalam *database*. Hasil dari pengumpulan berita juga perlu dilakukan *data cleaning* untuk menghilangkan karakter yang tidak diinginkan, contohnya karakter “\n”. Dokumen berita yang telah dikumpulkan melalui *web crawler* dan *web scraper*, selanjutnya dapat disimpan dalam *database* yang kemudian dapat menghasilkan output yang dapat diakses *User* berupa tabel.

B. Rancangan Use Case

Rancangan *Use Case* pada aplikasi pengumpul berita dari media daring ini dapat dilihat pada Gambar 5. di bawah ini.



Gambar 5. Use Case Diagram

Pada rancangan *Use Case* di atas, terlihat bahwa *User* dapat melakukan aktivitas utama berupa menjalankan aplikasi pengumpul berita. Sedangkan lingkaran berwarna merah dan biru adalah detail aktivitas yang dapat dilakukan oleh *User* pada aktivitas utama.

C. Rancangan Activity Diagram

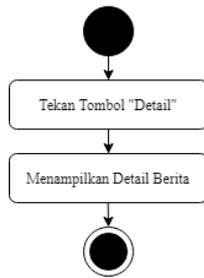
Rancangan *activity diagram* pada setiap aktivitas adalah sebagai berikut.

1) *Mengakses Aplikasi Pengumpul Berita*: Gambar 6. di bawah ini adalah *Activity Diagram* untuk aktivitas pengumpulan berita. Untuk mengakses aplikasi pengumpul berita, *User* dapat memilih menu "Pengumpul Berita". Setelah itu *User* diminta untuk memasukkan kata kunci berita yang ingin dikumpulkan kemudian tekan tombol "Scrap Berita". Hasilnya, muncul daftar berita beserta entitas lainnya yang juga dikumpulkan.



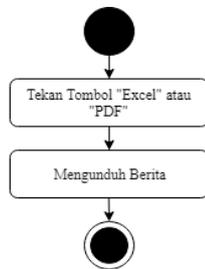
Gambar 6. Activity Diagram akses aplikasi pengumpul berita

2) *Melihat Detail Berita*: Pada daftar berita, di setiap berita yang dikumpulkan terdapat tombol "Detail". Untuk melihat detail berita berisi judul, tanggal, konten dan sumber berita pada salah satu berita yang ingin dilihat detailnya, tekan tombol tersebut. Sistem akan mengarahkan pada tampilan detail berita.



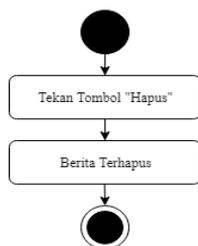
Gambar 7. Activity Diagram detail berita

3) *Mengunduh Berita*: Pada daftar berita, terdapat tombol “Excel” dan “PDF”. Tombol “Excel” berfungsi untuk mengunduh daftar berita dengan format .xlsx, sedangkan tombol “PDF” berfungsi mengunduh daftar berita dengan format .pdf. Untuk mengunduh berita, tekan salah satu tombol tersebut sesuai format yang diinginkan *User*.



Gambar 8. Activity Diagram unduh berita

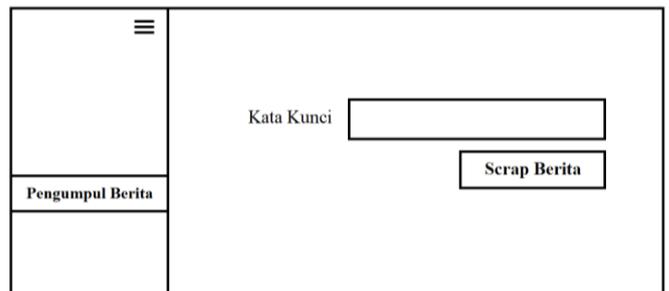
4) *Menghapus Berita*: Selain tombol “Detail”, di setiap berita yang dikumpulkan juga terdapat tombol “Hapus”. Untuk menghapus berita yang ingin dihapus, tekan tombol tersebut. Tabel daftar berita akan diperbarui dan berita yang dihapus tadi sudah tidak ada lagi di dalam daftar berita.



Gambar 9. Activity Diagram hapus berita

D. Rancangan Antarmuka

Rancangan antarmuka digunakan sebagai purwarupa atau *prototype* dalam pembangunan aplikasi pengumpul berita dari media daring. Rancangan antarmuka dibuat sesuai dengan kebutuhan sistem. Rancangan ini menjadi acuan dalam pembuatan aplikasi tersebut. Adapun detail rancangan antarmukanya adalah sebagai berikut.



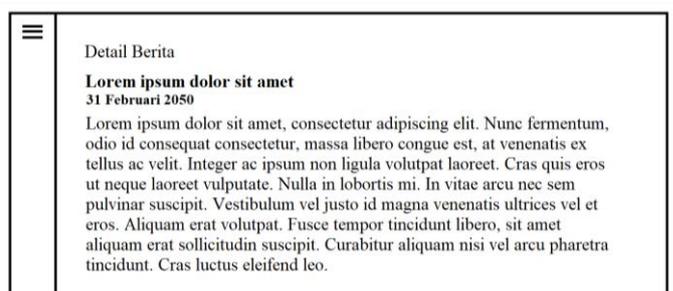
Gambar 10. Rancangan input kata kunci pada modul pengumpul berita

Pada Gambar 10. di atas, dapat dilihat tampilan awal dari aplikasi pengumpul berita. Tampilan tersebut akan muncul setelah *User* memilih menu “Pengumpul Berita” pada *sidebar*. Terdapat kolom isian kata kunci yang dapat digunakan user untuk mendapatkan berita terkait kata kunci tersebut.

No.	Judul	Tanggal	Konten Berita	Opsi
1.	Lorem ipsum dolor sit amet	31 Februari 2050	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc fermentum, odio id consequat consectetur, massa libero congue est, at venenatis ex tellus ac velit. Integer ac ipsum non ligula volutpat laoreet....	Detail

Gambar 11. Rancangan tampilan daftar berita hasil pengumpulan berita

Setelah kata kunci diinput dan *User* menekan tombol “Scrap Berita”, maka akan muncul daftar berita yang berhasil dikumpulkan dari keempat media daring seperti yang ditunjukkan Gambar 11. di atas. *User* juga dapat melihat detail berita yang rancangannya dapat dilihat pada Gambar 12. di bawah ini.



Gambar 12. Rancangan tampilan detail berita

E. Spesifikasi Perangkat

Dalam proses pelaksanaan penelitian mulai dari perancangan hingga implementasi sistem beserta penulisan dokumentasinya, Peneliti menggunakan sebuah perangkat

keras utama berupa laptop Asus VivoBook X505Z dengan spesifikasi sebagai berikut:

- Sistem Operasi Windows 10 Home Single Language
- Processor AMD Ryzen 5 2500U 2.0 GHz (up to 3.6 GHz)
- RAM 8 GB

Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

- XAMPP versi 3.2.3
- Python versi 3.7.3
- php MySQL
- Visual Studio Code versi 1.56.2
- Google Colaboratory
- CodeIgniter
- Flask
- Browser (Firefox)
- Windows Terminal

F. Pengembangan Modul Pengumpul Berita

Proses pengumpulan berita dimulai saat *User* melakukan klik tombol “Scrap Berita” setelah menginput kata kunci. Kata kunci yang diinput oleh *User* nantinya digunakan sebagai parameter pada URL dalam proses *web crawling* untuk mendapatkan URL berita.

Di sisi lain, modul pengumpul berita pada sisi *backend* dijalankan pada Flask, dimana modul tersebut memiliki *port* yang berbeda dengan aplikasi berjalan. Untuk menjalankan program pengumpulan berita, modul tersebut tentunya memerlukan parameter kata kunci. Maka dari itu, parameter kata kunci dari sistem perlu ditransfer. Transfer parameter dijalankan menggunakan *script* sebagai berikut, dimana parameter kata kunci pada *script* tersebut bernama “keyword”.

```
<script type="text/javascript">
function scraper_berita() {
var keyword = document.getElementById("inputkeyword").value;
fetch("http://127.0.0.1:5000/scraper/${keyword}", {method: 'post'})
}
</script>
```

Gambar 13. Script transfer parameter menuju *backend*

Di sisi *backend*, kata kunci yang telah diterima dari aplikasi selanjutnya digunakan sebagai parameter untuk menjalankan fungsi *web crawling* dalam mendapatkan URL berita dimana fungsi tersebut akan menjalankan *spyder*. Adapun proses *web crawling* menggunakan *script* sebagai berikut.

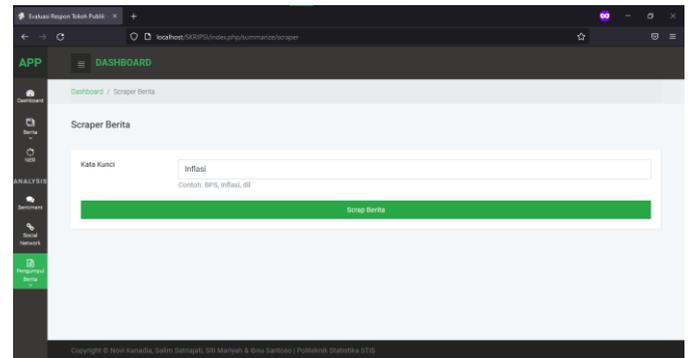
```
@app.route('/scraper/<keyword>', methods=['POST'])
def scraper(keyword):
    """Mengambil data berita berdasarkan keyword"""
    scraper_berita_detik(keyword)
    scraper_berita_antara(keyword)
    scraper_berita_bisnis(keyword)
    scraper_berita_kompas(keyword)
```

Gambar 14. Script fungsi *web crawling*

Setiap fungsi “*scraper_berita*” pada masing-masing media daring akan menjalankan *spyder*-nya masing-masing dan ketika fungsi dijalankan, maka secara otomatis, *spyder* akan melakukan proses *web scraping* untuk mendapatkan artikel berita. Artikel berita yang didapat kemudian disimpan ke dalam *database*.

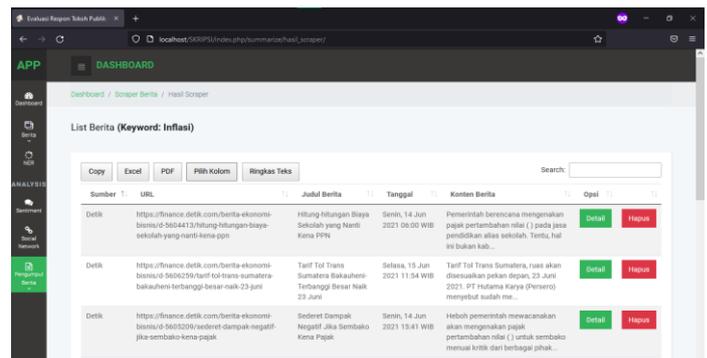
G. Pengembangan Sistem dan Impelementasi Antarmuka

Selanjutnya, berikut ini adalah tangkapan layar hasil implementasi sistem. Untuk mengakses aplikasi pengumpul berita dari media daring, pilih menu “Pengumpul Berita” yang ada pada *sidebar*.



Gambar 15. Tampilan input kata kunci pada modul pengumpul berita

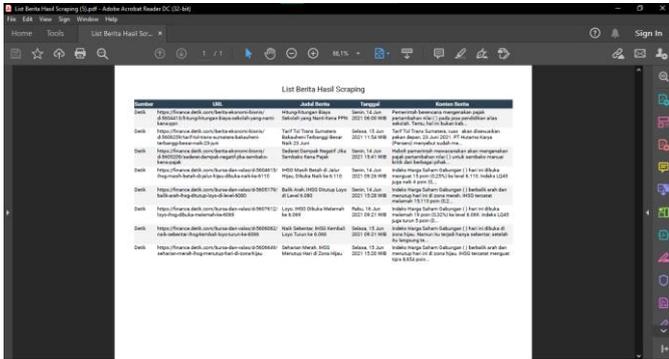
Gambar 15 di atas adalah tampilan awal aplikasi pengumpul berita dari media daring. *User* diminta untuk melakukan input kata kunci yang terkait dengan berita yang ingin dikumpulkan. Pada Gambar 15 di atas, dicontohkan berita yang ingin dikumpulkan adalah berita yang memiliki kata kunci “Inflasi”. Setelah itu tekan tombol “Scrap Berita”.



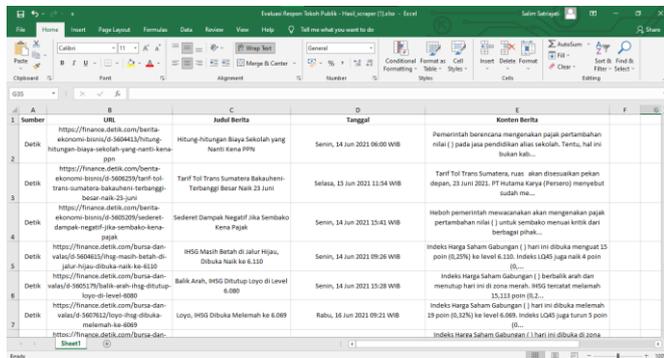
Gambar 16. Tampilan daftar berita hasil pengumpulan berita

Setelah itu, muncul daftar berita yang berhasil dikumpulkan dari keempat media daring seperti yang ditunjukkan Gambar 16 di atas. *User* dapat menghapus berita yang tidak diinginkan.

Berita hasil pengumpulan berita menggunakan modul pengumpul berita dapat diunduh dengan format .pdf dan .xlsx. Gambar 17 dan Gambar 18 berikut ini adalah contoh berita hasil unduhan.

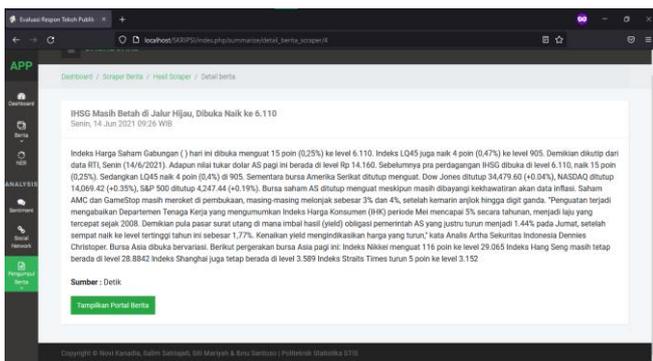


Gambar 17. Unduhan berita dalam format .pdf



Gambar 18. Unduhan berita dalam format .xlsx

User juga dapat melihat detail berita seperti yang ditunjukkan Gambar 19 di bawah ini.



Gambar 19. Tampilan detail berita

G. Evaluasi Sistem

Evaluasi sistem dalam penelitian ini menggunakan *black-box testing*. Hasil *black-box testing* dapat dilihat pada Tabel II di bawah ini.

TABEL II
HASIL EVALUASI DENGAN *BLACK-BOX TESTING*

No	Deskripsi Pengujian	Hasil yang diharapkan	Hasil Aktual
1.	Masuk ke menu "Pengumpul Berita"	Muncul isian kata kunci untuk pengumpul berita	Berhasil
2.	Menjalankan modul pengumpul berita. Klik "Scrap Berita"	Muncul daftar berita hasil pengumpul berita	Berhasil
3.	Menghapus berita terpilih	Berita yang dihapus, hilang dari daftar berita	Berhasil
4.	Mengunduh hasil pengumpulan berita dengan format .pdf	Berita berhasil diunduh dalam format .pdf	Berhasil
5.	Mengunduh hasil pengumpulan berita dengan format .xlsx	Berita berhasil diunduh dalam format .xlsx	Berhasil
6.	Melihat detail berita	Dapat melihat detail berita, berisi judul, tanggal, isi berita dan sumber	Berhasil

Berdasarkan hasil *black-box testing* pada Tabel II. di atas, dapat disimpulkan bahwa seluruh fungsi pada aplikasi pengumpul berita dari media daring yang telah dibuat dapat berjalan sesuai harapan Peneliti.

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan penelitian ini, dapat ditarik beberapa kesimpulan. Modul pengumpul berita untuk mendapatkan berita dari media daring berhasil dirancang dan dibangun dengan menggunakan *package* Scrapy yang berbasis Python sebagai alat untuk melakukan *web crawling* dan *web scraping*.

Penelitian ini berhasil menghasilkan sebuah artefak berupa aplikasi pengumpul berita dari media daring berbasis web yang berhasil diimplementasikan dengan memanfaatkan *web framework* Codeigniter dan Flask. Hasil dari *black-box testing* menunjukkan bahwa seluruh fungsi pada aplikasi pengumpul berita dari media daring yang telah dibuat dapat berjalan sesuai harapan Peneliti.

UCAPAN TERIMA KASIH

Atas penyelesaian penelitian ini, Penulis ingin mengucapkan terima kasih pada beberapa pihak. Pertama, Penulis ingin mengucapkan terima kasih kepada Ayah dan Ibu yang selalu memberikan doa dan dukungan kepada Penulis. Penulis juga berterima kasih kepada Bapak Ibnu Santoso, SST., M.T.. sebagai dosen pembimbing Penulis yang telah bersedia meluangkan waktu dan dengan sabar mengarahkan dan membimbing penyusunan penelitian ini, serta kepada Ibu Siti Maryah, SST., M.T., Kak Novi Kanadia, S.Tr.Stat., Tim Akademik Angkatan 59 Politeknik Statistika STIS dan pihak lainnya yang turut serta mendukung penelitian ini.

REFERENSI

- [1] Palinggi, S., & Limbongan, E. C. (2020). Pengaruh Internet Terhadap Industri E-Commerce dan Regulasi Perlindungan Data Pribadi Pelanggan di Indonesia. *Seminar Nasional Riset Dan Teknologi (SEMNASRISTEK)*, 4(1), 225–232.
- [2] Respati, W. (2014). Transformasi Media Massa Menuju Era Masyarakat Informasi di Indonesia. *Humaniora*, 5(1), 39-51.
- [3] Adli, M. A., & Firgia, L. (2018). Rancang Bangun Web Scraping Pada Media Online Berita Nasional. *Jurnal ENTER*, 1(1), 118–128.
- [4] Dewan Pers. Data Perusahaan Pers. Diakses pada 23 Maret 2021, dari <https://dewanpers.or.id/data/perusahaanpers>
- [5] Hong, S. H., Lee, S. K., & Yu, J. H. (2019). Automated management of green building material information using web crawling and ontology. *Automation in Construction*, 102, 230–244.
- [6] Rizaldi, T., & Putranto, H. A. (2017). Perbandingan Metode Web Scraping Menggunakan CSS Selector dan Xpath Selector. *Teknika*, 6(1), 43-46.
- [7] Verschuren, P., & Hartog, R. (2005). Evaluation in design-oriented research. *Quality and Quantity*, 39(6), 733–762.
- [8] Scrapy. Diakses pada 7 Mei 2021 dari <https://scrapy.org/>
- [9] Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), 31–36.
- [10] Satriajati, S., Panuntun, S. B., & Pramana, S. (2020). Implementasi Web Scraping dalam Pengumpulan Berita Kriminal pada Masa Pandemi COVID-19. *Seminar Nasional Official Statistics, 2020(1)*, 300-308.