

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Berikut penelitian terkait Analisis Perbandingan Metode *Web Scraping BeautifulSoup* dan *lxml* pada Aplikasi *ConvexView* Berbasis Web

Tabel 1 Penelitian Terkait

No (1)	Nama (Tahun) (3)	Keterangan (4)
1	T. Pricillia and Zulfachmi, 2021	Judul: Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD). Metode: Waterfall, Prototype, dan RAD. Software: - Hasil: Model Waterfall cocok untuk sistem generik, sedangkan Prototype dan RAD lebih sesuai untuk sistem yang menyesuaikan kebutuhan pengguna. Setiap metode memiliki kelebihan dan kekurangan masing-masing.
2	V. Lisia, A. E. Widjaja, A. R. Mitra, and C. A. Haryani, 2022	Judul: Visualisasi Data Bencana Geologi Di Indonesia Berbasis Web Metode: <i>Prototyping</i> Software: PHP, <i>Python</i> , Power BI Hasil: Sistem berbasis web memungkinkan pelaporan bencana geologi oleh masyarakat dan visualisasi data menggunakan Power BI. Namun, fitur penangkapan lokasi berbasis Mapbox belum sepenuhnya berfungsi.
3	E. W. Fridayanthie, H. Haryanto, and T. Tsabitah, 2021	Judul: Penerapan Metode <i>Prototype</i> Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web. Metode: <i>Prototype</i> . Software: PHP, MySQL Hasil: Sistem informasi penggajian berbasis web berhasil dirancang untuk PT Bina Angkasa, meningkatkan efisiensi

		pencatatan gaji karyawan dan mengurangi kesalahan data.
4	A. Yulianto, 2021	Judul: Perancangan Sistem Informasi Absensi Sekolah Menggunakan Metode Prototipe berbasis Web. Metode: <i>Prototyping</i>. Software: PHP, MySQL. Hasil: Sistem absensi berbasis web memungkinkan pencatatan kehadiran siswa secara lebih efisien dibandingkan metode manual, serta dapat diakses dalam jaringan intranet bagi sekolah tanpa internet.
5	Widhiyanti, Kathryn Atmani, Agnes Karina Prita, 2021	Judul: Penerapan Metode <i>Prototyping</i> dalam Perancangan Interface Sistem Unggah Portofolio Penerimaan Mahasiswa Baru Diploma ISI Yogyakarta. Metode: Metode <i>Prototyping</i>. Software: - Hasil: Penggunaan metode <i>Prototyping</i> berhasil menghasilkan antarmuka sistem unggah portofolio dengan tingkat usability tinggi (93–100%), meningkatkan efisiensi komunikasi antara pengguna dan pengembang, serta mempercepat proses identifikasi kebutuhan pengguna.
6	P. I Gusti Bagus Indra, 2023.	Judul: Analisis Data Produk Marketplace Menggunakan <i>Web Scraping</i> Menggunakan Metode Beautiful Soup Dan <i>LXML</i>. Metode: <i>Python</i> (Beautiful Soup, <i>LXML</i>.) Software: Beautiful Soup dan <i>LXML</i> Selenium, Pandas, Matplotlib, Streamlit, Visual Studio <i>Code</i>. Hasil: Sistem <i>Scraping</i> berhasil mengambil data produk dari Tokopedia, termasuk harga, rating, jumlah penjualan, lokasi, dan rekomendasi produk, serta

		menampilkan hasil analisis dalam bentuk tabel dan grafik.
7	K. Sharma and G. M. Borkar, 2024	Judul: Comparative Analysis of Dynamic <i>Web Scraping</i> Strategies: Evaluating Techniques for Enhanced Data Acquisition. Metode: <i>Scraping</i>. Software: <i>Python (BeautifulSoup, LXML, Selenium)</i> Hasil: Selenium lebih efisien dibandingkan <i>BeautifulSoup</i> dan <i>LXML</i> untuk <i>Scraping</i> halaman web dinamis, dengan konsumsi data 90% lebih sedikit dan waktu pemrosesan lebih cepat 70%.
8	I. Darmawan, M. Maulana, R. Gunawan, and N. Widiyasono, 2022	Judul: Evaluating <i>Web Scraping</i> Performance Using <i>XPath</i>, <i>CSS Selector</i>, Regular Expression, and <i>HTML DOM</i> With Multiprocessing Technical Applications Metode: <i>Scraping</i> berbasis <i>XPath</i>, <i>CSS Selector</i>, <i>Regex</i>, dan <i>HTML DOM</i> Software: <i>Python (Multiprocessing, BeautifulSoup, LXML, Regex)</i> Hasil: <i>XPath</i> memiliki waktu eksekusi tercepat, sedangkan <i>Regex</i> lebih hemat CPU dan memori. Penerapan multiprocessing mengurangi waktu eksekusi dan penggunaan bandwidth secara signifikan.
9	K. Dwicahyo and C. Indah Ratnasari, 2023.	Judul: Perbandingan Metode <i>Web Scraping</i> Dalam Pengambilan Data: Kajian Literatur Metode: Metode <i>Scraping (XPath Selector, CSS Selector, JSON Parsing, HTML Parsing)</i>. Software: Scrapy, Selenium, <i>BeautifulSoup, LXML</i>. Hasil: <i>XPath Selector</i> lebih efisien dalam pengambilan data dibandingkan <i>CSS</i>

		<i>Selector</i> , dan <i>JSON API</i> lebih cepat dibandingkan <i>HTML Parsing</i> dalam beberapa kasus.
10	A. Mulyani, D. Kurniadi, and I. L. Hakim, 2021	Judul: <i>Web Scraping</i> Pada Web Media Digital untuk Membangun Aplikasi Android. Metode: <i>Web Scraping</i>. Software: <i>Python, REST API, Android</i>. Hasil: <i>Web Scraping</i> memungkinkan pengumpulan data dari media digital secara otomatis dan integrasi dengan aplikasi Android tanpa database lokal.
11.	Yılmaz Dikilitaş et al. (2024)	Judul: Performance Analysis for <i>Web Scraping</i> Tools: Case Studies on <i>BeautifulSoup</i>, <i>Scrapy</i>, <i>HTMLUnit</i> and <i>Jsoup</i> Metode: Studi eksperimental dengan pengujian langsung terhadap empat <i>library Scraping</i> menggunakan dataset nyata dari situs e-commerce. Software: <i>BeautifulSoup, Scrapy, HTMLUnit, Jsoup</i>. Hasil: Penelitian menunjukkan bahwa <i>BeautifulSoup</i> dan <i>Jsoup</i> unggul dalam efisiensi memori dan CPU, cocok untuk <i>Scraping</i> sederhana. <i>Scrapy</i> menawarkan fitur lanjutan dan dukungan JavaScript dengan performa moderat, cocok untuk proyek kompleks. <i>HTMLUnit</i>, dengan kemampuan eksekusi JavaScript, memiliki konsumsi CPU tinggi tetapi berguna untuk <i>Scraping</i> konten dinamis. Pemilihan <i>library</i> harus mempertimbangkan trade-off antara kecepatan kerja, konsumsi resource, dan kebutuhan eksekusi JavaScript.
12	Erdoğan Uzun et al. (2018)	Judul: <i>Comparison of Python Libraries Used for Web Data Extraction</i> Metode: Eksperimen komparatif tiga

		metode (<i>BeautifulSoup</i>, <i>lxml</i>, <i>regex</i>) pada berbagai <i>DOM</i>ain situs web. Software: <i>Python</i>, <i>BeautifulSoup</i>, <i>lxml</i>, <i>regex</i>. Hasil: <i>Regex</i> menghasilkan waktu ekstraksi tercepat (0.071 ms) namun hanya berhasil mengekstrak data dengan akurasi 43.5%. <i>Library lxml</i> memberikan hasil terbaik untuk metode berbasis <i>DOM</i>, dengan rata-rata waktu ekstraksi 9.047 ms, lebih cepat dari <i>BeautifulSoup</i>. Meskipun <i>BeautifulSoup</i> populer, proses parsing-nya membutuhkan waktu lebih lama terutama jika tidak menggunakan <i>parser lxml</i>. Secara keseluruhan, <i>lxml</i> direkomendasikan untuk <i>Scraping</i> dengan struktur <i>DOM</i> kompleks dan efisiensi tinggi.
13	J. Chen, 2023	Judul: Application Technology of Atmospheric Dispersion Models Based on <i>FastAPI</i>+Vue. Metode: CALPUFF. Software: <i>Python</i> (<i>FastAPI</i>), <i>Vue.js</i>, <i>MATLAB</i>, <i>WebGIS</i>. Hasil: Implementasi model dispersi atmosfer berbasis web menggunakan <i>FastAPI</i> dan <i>Vue</i> meningkatkan efisiensi prediksi polusi udara. Model ini mengoptimalkan pemrosesan data GIS dan memberikan visualisasi interaktif untuk perencanaan respons darurat.
14	U. Khandelwal and A. KB, 2024	Judul: <i>FastAPI</i> vs. The Competition: A Security Feature Showdown with a Proposed Model for Enhanced Protection Metode: CALPUFF. Software: <i>Python</i> (<i>FastAPI</i>), <i>Vue.js</i>, <i>MATLAB</i>, <i>WebGIS</i>. Hasil: Implementasi model dispersi atmosfer berbasis web menggunakan

		<i>FastAPI</i> dan Vue meningkatkan efisiensi prediksi polusi udara. Model ini mengoptimalkan pemrosesan data GIS dan memberikan visualisasi interaktif untuk perencanaan respons darurat
15	A. N. Safitri and I. Harkespan, 2024.	Judul: Pengembangan Web Service Menggunakan <i>Framework FastAPI</i> untuk Meningkatkan Kemudahan Integrasi Sistem Informasi Akademik Multiaplikasi. Metode: Extreme Programming (XP). Software: <i>Python (FastAPI)</i>, JMeter. Hasil: Web service berhasil dibangun untuk sistem akademik multiaplikasi dengan performa baik (rata-rata waktu respons 6198ms saat diakses oleh 1000 user). Dokumentasi <i>API</i> yang lengkap memudahkan integrasi antar sistem dan mengurangi kesalahan dalam pengambilan data akademik.
16	R. Azhari, 2022	Judul: Web Service <i>Framework</i>: Flask dan <i>FastAPI</i> Metode: Deskriptif-komparatif Software: Flask dan <i>FastAPI (Python-based web service frameworks)</i> Hasil: Flask dirancang sebagai <i>framework</i> web general-purpose dengan fleksibilitas tinggi, sedangkan <i>FastAPI</i> dikembangkan khusus untuk pembuatan web service berbasis <i>API</i> yang mendukung konsep asynchronous. <i>FastAPI</i> dinilai lebih ringan dan efisien dalam pengembangan layanan <i>API</i> modern, tetapi memiliki keterbatasan pada struktur templating. Perbandingan diarahkan pada struktur, keamanan, asinkronitas, dan kompatibilitas <i>framework</i>.

17	X. Wu, 2024	Judul: <i>FastAPI</i> as a <i>Backend Framework</i> Metode: Evaluasi performa dan arsitektur sistem. Software: <i>FastAPI</i>, <i>Python</i>, <i>HTML/CSS/JS</i>, Uvicorn (ASGI server), <i>VSCode</i>, <i>VirtualBox</i>, token-based authentication. Hasil: Integrasi <i>FastAPI</i> berhasil menggantikan <i>backend</i> berbasis PHP dan meningkatkan performa sistem, khususnya dalam menangani kebutuhan asinkron yang tinggi serta efisiensi pengelolaan <i>API</i>. <i>FastAPI</i> juga dinilai unggul dalam kecepatan, manajemen memori, serta fitur dokumentasi otomatis. Evaluasi menunjukkan sistem mampu merender <i>frontend</i> dengan baik dan mendukung otorisasi berbasis token secara efektif. <i>Framework</i> ini dinilai cocok untuk proyek web modern yang membutuhkan skalabilitas dan performa <i>real-time</i>.
18	Suroyo et al, 2024	Judul: Pengembangan Web Admin Aplikasi Pulsa Dengan <i>ReactJS</i> Menggunakan metode Rapid Application Development. Metode: Rapid Application Development (RAD) Software: <i>ReactJS</i>, Postman, Katalon Studio, Xendit Payment Gateway, Axios (JavaScript <i>library</i>). Hasil: Aplikasi admin pulsa berhasil dikembangkan dengan waktu pengujian <i>API</i> yang efisien (GET: 6.8ms, POST: 1630ms, PUT: 27.5ms). Semua fitur diuji melalui test case dengan hasil 100% sukses. Implementasi payment gateway (Xendit) juga meningkatkan fleksibilitas metode pembayaran dalam sistem.

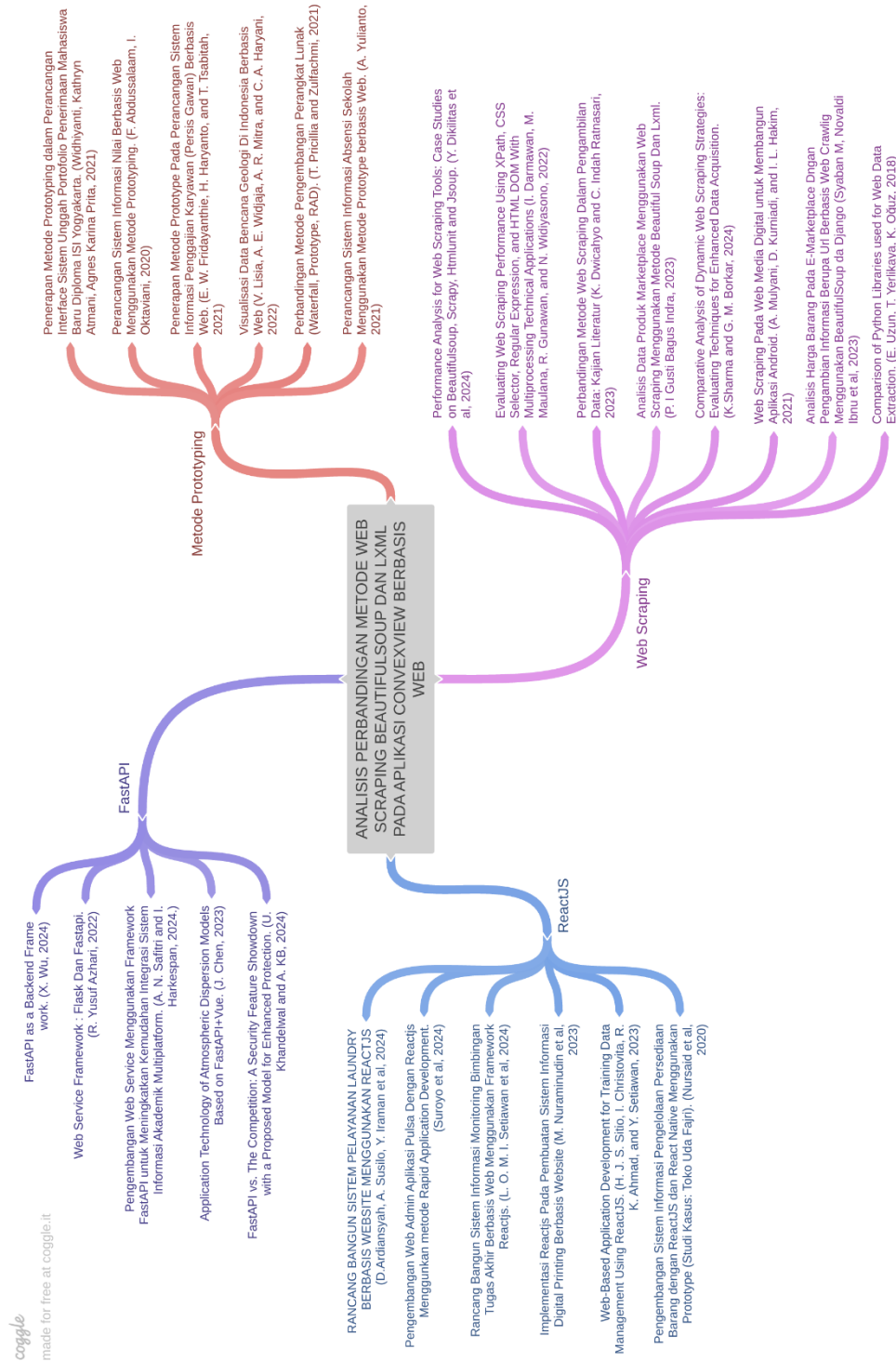
19	Ardiansyah et al. 2024	Judul: Rancang Bangun Sistem Pelayanan Laundry Berbasis Website Menggunakan <i>ReactJS</i>. Metode: Software Development Life Cycle (SDLC) model Waterfall Software: JavaScript (<i>ReactJS</i>), Visual Studio <i>Code</i>, PostgreSQL, Web Server Vercel. Hasil: Sistem website laundry berhasil dibangun dan diuji dengan hasil 17/17 test case berhasil (100%). UAT menunjukkan tingkat kepuasan pengguna sebesar 92,25% (aspek desain) dan 90,25% (aspek usability). Sistem mendukung fitur login admin, manajemen pesanan, ekspor data, dan status pesanan pelanggan secara <i>real-time</i> melalui antarmuka web yang responsif.
20	L. O. M. I. Setiawan, Gunawan, and A. Tenriawaru, 2024	Judul: Rancang Bangun Sistem Informasi Monitoring Bimbingan Tugas Akhir Berbasis Web Menggunakan <i>Framework ReactJS</i> Metode: Pengembangan sistem berbasis web. Software: <i>ReactJS</i>. Hasil: Sistem ini dikembangkan untuk membantu mahasiswa dan dosen dalam memantau progres bimbingan tugas akhir secara <i>real-time</i>. Dengan antarmuka berbasis <i>ReactJS</i>, sistem ini memungkinkan mahasiswa untuk mengunggah laporan bimbingan dan mendapatkan feedback langsung dari dosen. Selain itu, fitur notifikasi otomatis membantu mengingatkan jadwal pertemuan, sehingga meningkatkan efisiensi dalam proses bimbingan akademik.

21	M. Nuraminudin, M. M. Dewi, S. Suharsono, A. Dahlan, and L. Lukman, 2023	Judul: Implementasi <i>ReactJS</i> Pada Pembuatan Sistem Informasi Digital Printing Berbasis Website Metode: Pengembangan sistem informasi berbasis web. Software: <i>ReactJS</i>. Hasil: Sistem ini dikembangkan untuk meningkatkan efisiensi layanan digital printing dengan antarmuka yang lebih interaktif dan responsif. Dengan menggunakan <i>ReactJS</i>, sistem memungkinkan pelanggan untuk mengunggah file, memilih jenis cetakan, dan melakukan pembayaran secara online. Hal ini mempercepat proses produksi serta meningkatkan kepuasan pelanggan dengan layanan yang lebih cepat dan terorganisir.
22	H. J. S. Sitio, I. Christovita, R. K. Ahmad, and Y. Setiawan, 2023	Judul: Web-Based Application Development for Training Data Management Using <i>ReactJS</i> Metode: Pengembangan aplikasi berbasis web. Software: <i>ReactJS</i>. Hasil: Aplikasi ini dikembangkan untuk memudahkan pengelolaan data pelatihan dalam berbagai bidang, termasuk pendidikan dan pelatihan profesional. Dengan <i>ReactJS</i>, sistem dapat memberikan pengalaman pengguna yang lebih baik dalam mengakses, mengedit, dan menganalisis data pelatihan. Selain itu, fitur pencatatan riwayat data memungkinkan pelacakan perkembangan peserta pelatihan secara lebih sistematis.
23	Nursaid et al, 2020	Judul: Pengembangan Sistem Informasi Pengelolaan Persediaan Barang dengan

		<i>ReactJS</i> dan <i>React Native</i> Menggunakan Prototype (Studi Kasus: Toko Uda Fajri). Metode: <i>SDLC Prototyping</i>. Software: <i>ReactJS</i>, <i>React Native</i>, <i>PostgreSQL</i>, <i>Postman</i>. Hasil: Sistem berhasil dikembangkan dan diuji dengan hasil valid pada semua metode pengujian (100% valid dari 15 uji fungsional). Pengujian UAT menunjukkan kepuasan pengguna tinggi: 80% setuju terhadap sistem, 96% sangat mudah digunakan, 76% setuju terhadap fungsionalitas, dan 68% cukup puas terhadap antarmuka.
--	--	--

Berdasarkan penelitian terkait disimpulkan Gap penelitian ini terletak pada jenis data dan implementasinya. Penelitian sebelumnya berfokus pada *Scraping* teks *HTML* dengan evaluasi peforma saja, sedangkan penelitian ini menerapkan perbandingan metode *BeautifulSoup* dan *lxml* pada *Scraping* data citra satelit yang kemudian diintegrasikan ke dalam sistem aplikasi web *ConvexView* dalam Visualisasi informasi awan konvektif secara real time

Diagram *mindmap* dicontohkan pada Gambar 1.



Gambar 1 *Mindmap* Penelitian

B. Landasan Teori

1. Meteorologi dan Awan Konvektif.

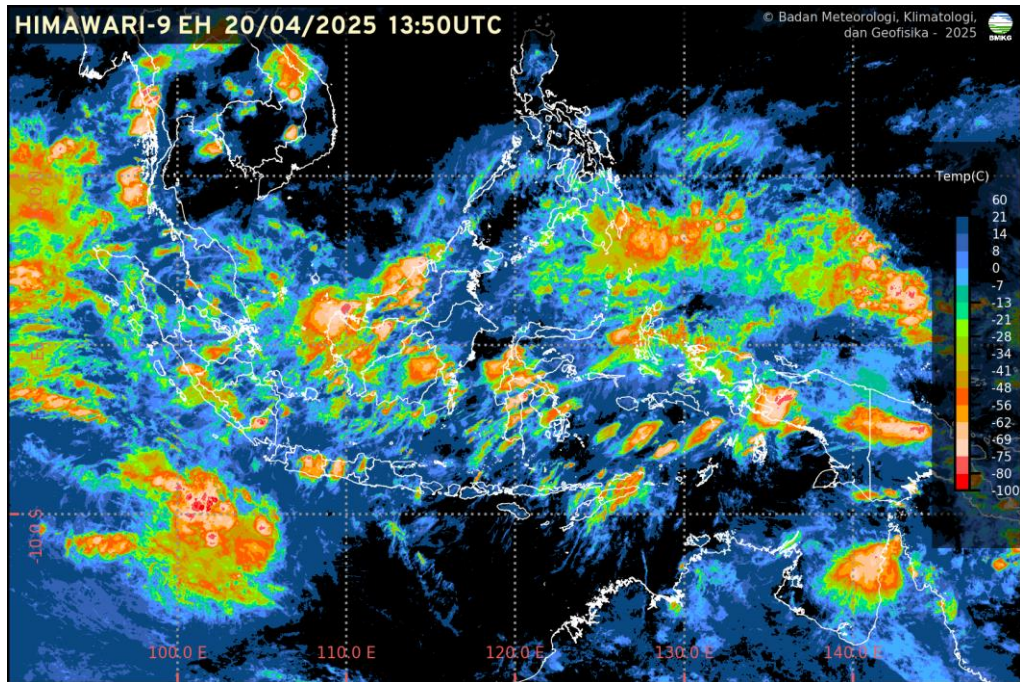
Meteorologi adalah cabang ilmu dalam mempelajari kondisi cuaca yang terjadi di atmosfer dan proses-proses fisis yang terjadi di atmosfer bumi. Salah satu aspek penting dalam meteorologi adalah pemantauan dan analisis menggunakan citra satelit, yang dapat memberikan informasi mengenai kondisi atmosfer, termasuk awan konvektif. Satelit Himawari-9 digunakan dalam berbagai penelitian untuk mengamati kejadian cuaca ekstrem, termasuk hujan lebat dan awan konvektif yang berpotensi menyebabkan bencana [12], [13].

Awan konvektif merupakan jenis awan yang terbentuk akibat proses konveksi, yaitu pergerakan udara hangat yang naik ke atmosfer dan mengalami pendinginan sehingga mencapai titik kondensasi. Proses ini sering terjadi di daerah tropis dan menyebabkan pertumbuhan awan cumulonimbus yang dapat menghasilkan hujan lebat, petir, serta angin kencang.

Analisis sebaran awan konvektif dapat dilakukan menggunakan metode *RGB-Night Microphysics* (RGB-NM) dan *Cloud Convective Overlays* (CCO), yang memungkinkan identifikasi awan konvektif berdasarkan perbedaan spektral citra satelit [14]. Penelitian terkait menunjukkan bahwa awan cumulonimbus memiliki suhu puncak awan sering kali mencapai $-77,9^{\circ}\text{C}$ hingga $-82,9^{\circ}\text{C}$, yang menandakan adanya aktivitas konvektif yang berpotensi menyebabkan hujan lebat [15].

2. Citra Himawari-9 IR Enhanced

Himawari-9 adalah satelit cuaca terbaru dari Jepang yang diluncurkan untuk menggantikan Himawari-8 dalam memantau kondisi atmosfer secara *real-time*, satelit ini dilengkapi dengan *Advanced Himawari Imager* (AHI) yang mampu menangkap citra dalam berbagai kanal spektral, termasuk Infrared (IR) Enhanced, yang sangat berguna dalam analisis cuaca ekstrem dan pemantauan awan konvektif [16].



Gambar 2 Citra Satelit Himawari-9 IR Enhanced

Citra Himawari-9 IR Enhanced merupakan salah satu hasil dari metode pemrosesan data citra yang digunakan untuk meningkatkan visibilitas struktur awan berdasarkan perbedaan suhu puncak awan. Kanal inframerah ini dapat digunakan untuk mendeteksi keberadaan awan cumulonimbus dengan suhu puncak yang sangat rendah, menandakan adanya aktivitas awan konvektif berpotensi menyebabkan hujan lebat serta angin kencang [15], [17].

Metode analisis citra Himawari-9 IR Enhanced dilakukan dengan menggunakan berbagai parameter seperti *Cloud Top Temperature* (CTT) dan *Brightness Temperature* (BT) untuk mengidentifikasi karakteristik awan konvektif. Selain itu, metode *RGB-Night Microphysics* (RGB-NM) dan *Cloud Convective Overlays* (CCO) juga digunakan untuk meningkatkan akurasi dalam mengklasifikasikan awan konvektif yang berpotensi menimbulkan cuaca ekstrem [1], [2], [14].

3. Teknik *Web Scraping*

Web Scraping adalah teknik ekstraksi data dari situs web secara otomatis dengan memanfaatkan kode program untuk mengakses dan mengambil informasi yang tersedia dalam struktur *HTML* halaman web[4]. Teknik ini banyak digunakan dalam penelitian pengembangan web dalam pengambilan informasi[5][18]. Dalam aplikasi *ConvexView*, *Web Scraping* digunakan untuk mengambil citra satelit Enhanced dari situs resmi BMKG dan memprosesnya lebih lanjut guna menampilkan awan konvektif. Dengan menerapkan teknik *Scraping* yang tepat, aplikasi dapat secara otomatis mengakses data terbaru dari

sumber terpercaya dan mengolahnya menjadi informasi yang dapat digunakan dalam penelitian meteorologi dan cuaca ekstrem[3].

Dalam implementasi *Web Scraping*, terdapat beberapa yang umum digunakan untuk mengekstrak data dari web, di antaranya *BeautifulSoup* dan *lxml*. *BeautifulSoup* merupakan *library Python* yang digunakan untuk melakukan *parsing* dokumen *HTML* dan *XML* dengan mudah, Karena *BeautifulSoup* menggunakan *parsing* berbasis *CSS Selector* sehingga sangat efektif dalam menangani data semi-terstruktur dengan mencari elemen berdasarkan tag, atribut, dan hierarki *DOM*[7], sedangkan *lxml* adalah pustaka yang lebih cepat dibandingkan *BeautifulSoup* dalam hal *parsing* dokumen *HTML* dan *XML*. *lxml* menggunakan *parsing* berbasis *XPath*, yang memungkinkan pencarian elemen dalam struktur dokumen dilakukan dengan efisiensi tinggi. Namun, menurut K. Dwicahyo and C. Indah Ratnasari penggunaan *lxml* memerlukan lebih banyak konfigurasi dibandingkan *BeautifulSoup*[4].

Beberapa penelitian terkait membandingkan performa kedua metode ini dalam proses *Scraping* halaman web dinamis. Hasilnya menunjukkan bahwa Selenium dan *lxml* lebih efisien dibandingkan *BeautifulSoup* dalam mengambil data dari situs yang memiliki banyak elemen dinamis, karena Selenium dapat mensimulasikan interaksi pengguna dengan website[7].

4. *Library Web Scraping*

Web Scraping adalah teknik ekstraksi data dari halaman web menggunakan program otomatis untuk mengambil informasi dalam format terstruktur. Teknik ini digunakan secara luas dalam berbagai bidang, salah satunya pengambilan data cuaca, dan pengolahan citra satelit[4]. Dalam implementasi *Web Scraping*, beberapa *library* yang umum digunakan dalam bahasa pemrograman *Python* adalah *BeautifulSoup* dan *lxml*. Kedua *library* ini memiliki keunggulan masing-masing dalam hal efisiensi, kemudahan penggunaan, dan kompatibilitas dengan berbagai format dokumen *HTML* dan *XML*[7].

BeautifulSoup adalah pustaka *Python* yang digunakan untuk *parsing* dan manipulasi dokumen *HTML* dan *XML*. *Library* menggunakan pendekatan *CSS Selector-based* yang sederhana dan cocok untuk *HTML* dengan struktur tidak terlalu kompleks, ini bekerja dengan membuat *tree traversal*, yang memudahkan pengguna untuk menavigasi elemen-elemen dalam dokumen *HTML*[7]. Sedangkan *lxml* adalah pustaka *Python* yang yaImgeng digunakan untuk *parsing* dokumen *HTML* dan *XML* menggunakan pendekatan *XPath-based* *parsing* yang lebih cepat dan akurat untuk struktur *HTML* yang lebih stabil *XPath* dan *tree*

traversal, yang memungkinkan pencarian elemen dalam dokumen *HTML* dengan efisiensi tinggi [4].

Adapun kelebihan dan kekurangan dari *BeautifulSoup* dan *lxml* berdasarkan penelitian terkait[6]–[8].

Tabel 2 Kelebihan dan Kekurangan *BeautifulSoup*

Kelebihan	Kekurangan
Mudah digunakan dan sintaks intuitif (cocok untuk pemula).	Lebih lambat dibandingkan <i>lxml</i> , terutama untuk <i>Scraping</i> dokumen besar
Dapat menangani <i>HTML</i> yang tidak terstruktur dengan baik.	Tidak mendukung <i>XPath</i> , yang lebih efisien dalam pencarian elemen <i>HTML</i> .
Kompatibel dengan berbagai <i>parser</i> seperti <i>HTML.parser</i> , <i>lxml</i> , dan <i>HTML5lib</i> .	Kurang optimal untuk <i>Scraping</i> halaman dinamis berbasis JavaScript

Tabel 3 Kelebihan dan kekurangan *lxml*

Kelebihan	Kekurangan
<i>Parsing</i> lebih cepat dibandingkan <i>BeautifulSoup</i> karena menggunakan <i>XPath</i> .	Sintaks lebih kompleks dibandingkan <i>BeautifulSoup</i> , memerlukan pemahaman tentang <i>XPath</i> .
Dapat menangani dokumen besar dan kompleks dengan efisiensi tinggi.	Tidak selalu kompatibel dengan <i>HTML</i> yang tidak terstruktur.
Lebih optimal untuk <i>Scraping</i> halaman dengan banyak elemen struktural	Memerlukan instalasi pustaka tambahan untuk performa optimal.

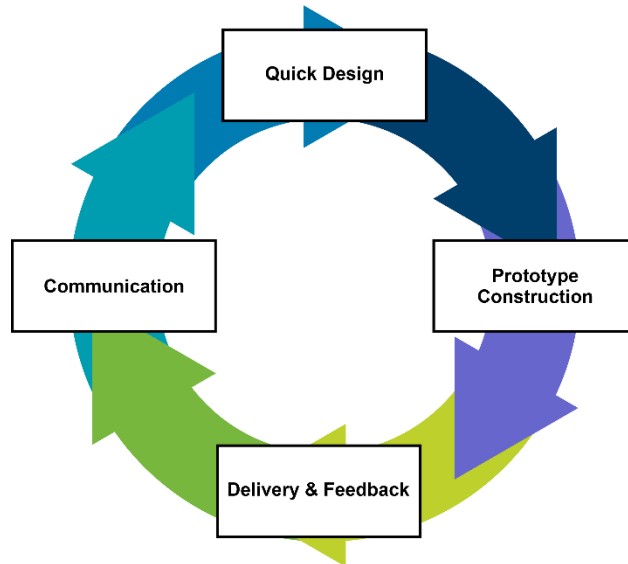
Pemilihan *BeautifulSoup* dan *lxml* didasarkan pada hasil beberapa penelitian sebelumnya yang menunjukkan bahwa keduanya memiliki kinerja yang baik dalam proses *Web Scraping*. *BeautifulSoup* menggunakan pendekatan *CSS Selector-based* yang sederhana dan cocok untuk *HTML* dengan struktur yang tidak terlalu kompleks, sedangkan *lxml* menggunakan pendekatan *XPath-based parsing* yang lebih cepat dan akurat untuk struktur *HTML* yang lebih stabil. Keduanya relevan untuk dibandingkan karena mewakili dua pendekatan parsing yang efisien dalam *Web Scraping*.

5. Bahasa Pemrograman *Python*

Python merupakan bahasa pemrograman tingkat tinggi yang bersifat interpretatif, mudah dipelajari, dan memiliki sintaks yang jelas serta ringkas. Bahasa ini dirancang untuk mendukung pengembangan aplikasi dengan cepat serta pemrosesan data secara efisien. *Python* banyak digunakan dalam berbagai bidang seperti pengembangan web, otomatisasi, analisa data, kecerdasan buatan, hingga pengolahan citra dan *Web Scraping*[7], [4].

Dalam penelitian-penelitian sebelumnya, *Python* merupakan bahasa utama yang digunakan dalam proses *Web Scraping* karena mendukung berbagai pustaka seperti *BeautifulSoup*, *lxml*, dan *Request*. Menurut K. Dwicahyo dan C. Indah Ratnasari, *Python* memiliki pustaka yang lengkap dan mendukung integrasi dengan berbagai format data yang umum digunakan dalam dunia sains dan industri[4].

6. Metode *Prototyping*



Gambar 3 Metode *Prototyping*

Metode *prototyping* merupakan salah satu model pengembangan perangkat lunak yang berfokus pada pembuatan purwarupa (*prototype*) sebagai representasi awal sistem yang akan dikembangkan. Pendekatan ini memungkinkan pengguna untuk berinteraksi langsung dengan sistem awal guna memberikan umpan balik yang kemudian digunakan untuk menyempurnakan sistem pada tahap selanjutnya.

Model pengembangan *prototyping* sesuai digunakan ketika kebutuhan pengguna belum sepenuhnya jelas di awal, karena memungkinkan terjadinya brainstorming antara pengguna dan pengembang secara berulang hingga sistem sesuai dengan harapan pengguna. Tahapan utama dalam metode ini meliputi

identifikasi kebutuhan awal, design, pembangunan prototipe, evaluasi pengguna, revisi, dan pengembangan akhir produk[21].

Kelebihan utama dari metode ini adalah kemampuannya dalam mengakomodasi perubahan kebutuhan di tengah proses pengembangan, serta meningkatkan kepuasan pengguna karena mereka dilibatkan sejak tahap awal. Sehingga memungkinkan deteksi kesalahan atau kekurangan sistem lebih awal. Dengan demikian, metode ini cocok untuk project yang membutuhkan pemahaman lebih mendalam terhadap kebutuhan pengguna dan eksplorasi desain antarmuka yang optimal[19].

7. *FastAPI*

FastAPI merupakan *framework* modern berbasis *Python* yang digunakan untuk membangun aplikasi web, khususnya web service berbasis *REST API*. Keunggulan utama dari *FastAPI* adalah performanya yang tinggi karena dibangun di atas *Starlette* dan memanfaatkan tipe data dari *Python 3.7* untuk validasi otomatis, dokumentasi interaktif, dan efisiensi pengembangan[22], [23]. Menurut Wang dkk, *framework* ini memungkinkan pengembang untuk membangun layanan web yang cepat serta efisien[24].

FastAPI mempermudah proses pengembangan dan mempercepat integrasi antar sistem melalui *endpoint* yang terdokumentasi secara otomatis. Penelitian lain menunjukkan bahwa *FastAPI* tidak hanya unggul dalam hal waktu respons dan kemudahan implementasi, tetapi juga menyediakan fitur keamanan bawaan seperti *Oauth2* dan validasi *API Key*[25]. Hal tersebut menjadikan *FastAPI* sebagai pilihan ideal dalam membangun layanan web modern yang membutuhkan performa tinggi dan keamanan yang baik.

8. *ReactJS*

ReactJS merupakan pustaka JavaScript yang dikembangkan oleh Meta (Facebook) untuk membangun antarmuka pengguna (*user interface*) yang interaktif dan efisien. *ReactJS* memiliki kemampuan untuk membuat aplikasi *single page application (SPA)* dengan komponen yang dapat digunakan ulang, serta mendukung *virtual DOM* yang memungkinkan pembaruan elemen secara cepat tanpa harus me-refresh seluruh halaman. Dalam Setiawan dkk menggunakan *ReactJS* dalam pengembangan sistem informasi monitoring bimbingan tugas akhir karena memiliki kemampuan kecepatan, kemudahan dan skalabilitas yang luas sehingga *ReactJS* sesuai untuk sistem yang membutuhkan pembaruan data secara *real-time* dan efisien[26].

Selain itu, *ReactJS* juga mendukung pengembangan aplikasi web yang efisien dalam berbagai konteks. Dalam penelitian Suroyo Dkk bahwa pengguna *ReactJS* dalam pengembangan aplikasi admin pulsa berbasis web berhasil meningkatkan efektivitas dan fleksibilitas sistem karena didasarkan pada

komponen yang interaktif, stateful, dan dapat digunakan kembali, *ReactJS* memudahkan pengembang untuk membuat aplikasi web[27]. Menurut Sitio Dkk [28] “pengembangan sistem informasi berbasis web menggunakan *ReactJS* meningkatkan efektivitas hingga 90,4% dan kepuasan pengguna sebesar 92,6%.” Penelitian-penelitian tersebut menunjukan *ReactJS* bukan hanya populer karena kemudahan penggunaannya, *restAPI* juga karena kemampuannya dalam memberikan dampak nyata pada kualitas dan efisiensi sistem berbasis web.