2. LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Publikasi Ilmiah

Sebuah publikasi yang ditulis oleh dosen akan menjadi tolak ukur kinerja dosen tersebut, kinerja dosen akan berpengaruh kepada kinerja fakultas, dan perguruan tinggi (Purwanto et al., 2021). Publikasi juga menjadi sebuah cara untuk dosen dapat mengembangkan karir nya. Kualitas dari sebuah publikasi sendiri dapat dilihat dari jumlah sitasi publikasi tersebut, semakin besar angka sitasi dari sebuah publikasi berarti publikasi tersebut semakin terlihat, dan juga semakin diakui oleh peneliti lainnya. Semakin tinggi kualitas publikasi dosen juga berarti nama dosen tersebut akan semakin diakui di dunia peneliti. Data yang menentukan kualitas dari sebuah publikasi ini juga tersedia untuk umum. Publisher dari publikasi biasanya akan mencantumkan metrics dari sebuah publikasi bersama dengan publikasi tersebut. Data yang menandakan dampak sebuah publikasi juga dapat dilihat melalui situs yang menyediakan dan mengumpulkan data publikasi, beberapa contoh layanan yang mengumpulkan data publikasi adalah Google Scholar, ScimagoJr, Web of Science, dan SINTA.

Google scholar adalah website yang mencatat publikasi yang dikeluarkan dari pihak lain, dan menyediakan layanan untuk user nya bisa mencari karya publikasi baik itu jurnal, buku ataupun bentuk publikasi lainnya. Google scholar juga menampilkan abstrak, penulis sebuah publikasi dan juga jumlah sitasi yang dimiliki oleh sebuah publikasi. Google scholar sendiri mempunyai angka sitasi yang lebih tinggi dibandingkan website lainnya yang menyediakan layanan pencarian publikasi lainnya(Martín-Martín et al., 2021). Dikarenakan kelengkapannya Google Scholar dipilih menjadi salah satu sumber *web scraping* pada aplikasi ini.

SINTA (Science and Technology Index) juga sebuah layanan yang melakukan pencatatan publikasi. SINTA sendiri juga mencatat melalui publikasi dari berbagai sumber seperti Google Scholar, dan Google Scholar menjadi salah satu sumber pencatatan SINTA. Ada beberapa meta data detail yang disediakan SINTA yang tidak ada di di Google Scholar, salah satunya adalah ISBN. Tetapi hak akses di SINTA cukup terbatas. Karena kelengkapan meta data nya SINTA dipilih menjadi salah satu sumber *web scraping* pada aplikasi ini.

Scimago adalah website lainnya yang melakukan pencatatan publikasi juga. Kelebihan website ini adalah Q ranking jurnal yang bisa disediakan oleh website ini. Q ranking atau quartile rank, rank ini terpisah menjadi Q1, Q2, Q3 dan Q4 dengan Q1 sebagai rank paling bagus, dan Q4 rank paling rendah. Jika sebuah publikasi mempunyai rank Q1 maka jurnal itu berada di 25% posisi teratas pada kategori nya. Untuk mencatat Q ranking dari sebuah publikasi maka ScimagoJr juga dipilih menjadi salah satu website sumber *web scraping* pada aplikasi ini.

2.1.2 Web Scraping

Web scraping adalah proses membaca sebuah informasi dari web, mengambil data yang diperlukan, dan dimasukkan ke sebuah database, proses ini akan dilakukan secara otomatis. Seperti melakukan browsing biasa dalam proses web scraping pertama akan mengirim request GET ke website yang akan di scrape, ketika website mengirim file HTML kembali akan di parse lalu dicari data yang perlu diambil. Pencarian data yang penting ini dapat dilakukan dengan mencari HTML tag dari data, id ataupun class.

Web scraper adalah sebuah nama untuk alat yang digunakan untuk melakukan kegiatan Web scraping. Sebuah web scraper adalah sebuah software yang menstimulasi seseorang yang membuka sebuah website untuk mengambil data (Diouf et al., 2019). Sementara Web Scraping sendiri adalah sebuah teknik pengumpulan data yang tidak terstruktur dari internet dan dikumpulkan pada sebuah file spreadsheet ataupun database (Khder, 2021).

Regex, HTML DOM, dan Xpath adalah beberapa cara untuk mengambil data dari website, regex dapat digunakan untuk mencari string yang sama atau mencari string yang sesuai kondisi yang ditentukan. Xpath dapat digunakan untuk memilih node di dokumen XML yang digunakan di HTML. HTML DOM adalah cara untuk mendapatkan dan memodifikasi file HTML (Gunawan et al., 2019).

2.1.3 Information Retrieval

Information Retrieval adalah sebuah aktivitas mengambil sebuah data yang relevan atau penting kepada sebuah topik dari kumpulan data yang banyak (Guo et al., 2020). Data yang relevan ini adalah data yang ingin didapatkan oleh *user* dan sumber data tersebut biasanya akan memberikan banyak data yang tidak relevan, seperti data iklan . Sumber data ini biasanya merupakan sebuah data yang tidak terstruktur, seperti sebuah halaman *website,*

dokumen, ataupun katalog barang. Data yang tidak terstruktur tersebut akan diolah sehingga dapat diakses dengan mudah sesuai dengan kriteria yang diberikan oleh *user*. Beberapa contoh sistem Information Retrieval adalah sebuah *search engine* yang akan mengembalikan data yang relevan berdasarkan apa yang diminta oleh *user* nya, perpustakaan digital seperti IEEE xplore (https://ieeexplore.ieee.org/Xplore/home.jsp) yang mengembalikan publikasi penelitian sesuai dengan yang diminta oleh *user* nya.

Untuk dapat menampilkan data yang relevan pada sebuah sistem Information Retrieval ada tahapan yang dinamakan indexing untuk memastikan data yang diberikan pada *user* adalah data yang relevan. Indexing sendiri adalah sebuah proses melakukan *mapping* dari dokumen dan sebuah kriteria atau ketentuan (Kaur & Gupta, 2016).

2.1.4 Regular Expression

Regex atau *regular expression* adalah cara untuk menemukan sebuah *string* atau bagian dari *string* yang sesuai dengan pola yang sudah ditentukan. Penjelasan lain untuk *regex* adalah sebuah cara untuk mendeskripsikan *string* dengan sebuah pola (Davis et al., 2019). Pada *web scraping* regex bisa digunakan untuk mencari *string* yang diinginkan berdasarkan ketentuan atau format kata yang sudah ditentukan sebelumnya. *Regex* mempunyai peraturan penulisan sendiri yang akan digunakan untuk menentukan format kata yang akan dicari. Dengan menggunakan regex *web scraping* bisa dilakukan tanpa memerlukan nama yang lengkap dari *id* ataupun *class* dari *tag* elemen di file HTML yang dilakukan akan di *scraping*. Dengan menggunakan *regex* bisa dibuat sebuah *script* untuk melakukan *web scraping* dari berbagai sumber dengan topik yang sama dengan mencari *keyword*.

2.1.5 Automated Web Browser

Automated web browser atau *headless browser* adalah sebuah alat otomasi yang bisa digunakan untuk mengoperasikan sebuah browser secara otomatis atau tanpa interaksi manusia. Sebuah *automated web browser* akan bisa berinteraksi dengan website tanpa memerlukan *GUI (Graphical User Interface), automated web browser* akan diatur dengan instruksi khusus melalui program untuk melakukan interaksi seperti *scroll*, dan menekan tombol. Sebuah *automated web browser* bisa digunakan sebagai alat pembantu untuk melakukan *web scraping* untuk menampilkan data dari sebuah website yang interaktif.

Mempunyai kemampuan untuk bisa berinteraksi dengan fitur yang disediakan oleh sebuah website seperti melakukan aksi ‘klik’ pada sebuah url, ataupun melakukan sebuah

'search' sangatlah berguna (Haddaway, 2015). Salah satu contohnya adalah melakukan 'search' pada Google untuk menemukan informasi yang kita cari. 'Klik' juga digunakan untuk mengakses halaman dan informasi yang ingin kita cari. Dengan bantuan aksi 'klik' juga bisa mengakses lebih dari satu url.

2.1.6 Application Programming Interface

Application Programming Interface (API) adalah sebuah aturan ataupun protocol yang bisa digunakan untuk 2 aplikasi bisa berkomunikasi satu sama lain. Sebuah API adalah sekumpulan prosedur, *function,* protokol dan aturan yang dibuat untuk mengatur bagaimana sebuah aplikasi bisa berinteraksi dengan aplikasi lainnya (Ghute & Raghuwanshi, 2016). Sebuah API akan mengirimkan data yang bisa dibaca dan diproses oleh berbagai macam aplikasi, contohnya adalah format data JSON (*JavaScript Object Notation*). API akan bekerja dengan format *request* dan *response* .Aplikasi yang akan menggunakan API akan mengakses API tersebut dari *endpoint URL* yang disediakan oleh API sebagai *request*, lalu setelah menerima *request* API akan memberikan Kembali data yang diminta dari *endpoint* tersebut sebagai *response.* API menggunakan HTTP dan *method* yang digunakan oleh HTTP seperti *GET* untuk membaca data, *POST* untuk menambahkan data, *PUT* untuk mengganti atau melakukan *update* pada data yang sudah ada, *DELETE* untuk menghapus data yang ada.

2.2 Tinjauan Studi

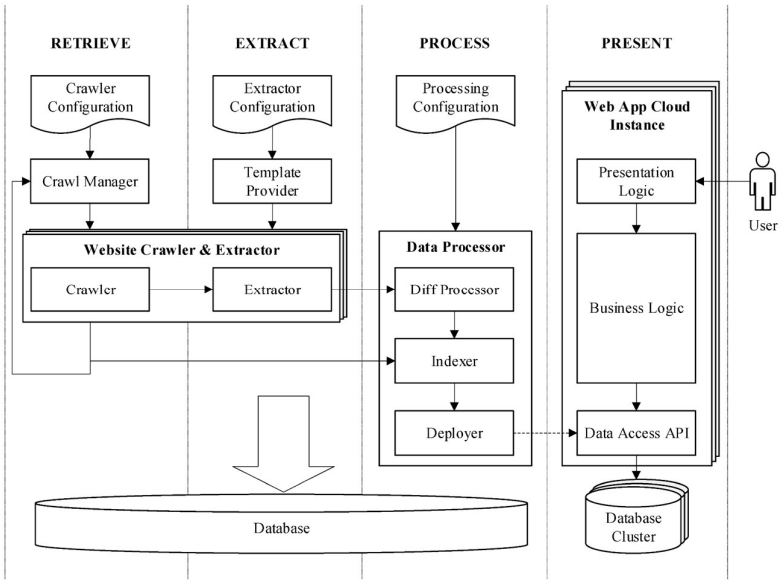
2.2.1 Social Media Web Scraping using Social Media Developers API and Regex (Dewi et al., 2019)

Masalah yang diangkat di penelitian ini adalah kesusahan mengambil data yang diperlukan dari sosial media, fitur share yang disediakan seringkali menambahkan informasi atau data yang tidak relevan. Metode yang digunakan di penelitian ini adalah menggunakan API yang disediakan oleh developer Facebook dan Twitter, hasil yang didapat lalu akan disamakan dengan preferensi user dengan menggunakan regex. Hasil dari penelitian ini adalah data yang didapat sosial media akan disaring sehingga mendapatkan data yang penting dan diinginkan oleh user dan dapat disimpan pada sebuah database. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah metode Web Scraping, dimana penelitian ini menggunakan API yang disediakan oleh developer Facebook dan Twitter skripsi ini akan menggunakan library BeautifulSoup untuk pembuatan Scraper

2.2.2 Optimization and Security in Information Retrieval, Extraction, Processing, and Presentation on a Cloud Platform (Alexandrescu, 2019)

Masalah yang diangkat pada penelitian ini adalah menemukan data yang bisa digunakan dan juga mengambil data tersebut, lalu memproses data tersebut untuk dapat digunakan pada sebuah *search engine* untuk mendapatkan informasi produk *board game*

Metode yang diusulkan adalah membuat sebuah sistem dengan arsitektur dimana akan ada sebuah *crawler* yang mengakses *link* yang disediakan di awal. Setiap *link* akan lalu di *extract* dengan pengetahuan awal dimana data yang ingin diambil. Sistem lalu akan menganalisa data yang akan ditampilkan kepada *user* dalam bentuk *web application.* Flowchart penjelasan sistem dapat dilihat di Gambar 2.1



Gambar 2.1 Flowchart alur yang diusulkan

Sumber: Information | Free Full-Text | Optimization and Security in Information Retrieval, Extraction, Processing, and Presentation on a Cloud Platform (mdpi.com)

Hasil dari penelitian adalah sebuah *web application* yang dapat diakses oleh *user* untuk mencari data yang sudah diproses setelah di *scrape* dari website sumber. Perbedaan penelitian

yang dilakukan dengan skripsi ini objek penelitian, dimana skripsi akan mencari data publikasi dosen untuk diproses dan ditampilkan kepada *user.*

2.2.3 Web Scraping Techniques to Collect Weather Data in South Sumatera (Fatmasari et al., 2018)

Masalah yang diangkat di penelitian ini kesusahan mendapatkan data cuaca yang diperlukan untuk analisa perkiraan cuaca di Sumatera Selatan. Metode yang digunakan pada penelitian ini adalah membuat sebuah scraper yang akan mengambil data dengan library BeautifulSoup, dan membuat sebuah scheduler yang dapat menjadwalkan pengambilan data secara otomatis. Data tersebut akan disimpan pada sebuah file excel sebelum diproses lebih lanjut. Hasil dari penelitian ini adalah sebuah sistem yang dapat mengambil data cuaca yang dilakukan setiap jam, data yang diambil juga sangat lengkap dan dapat digunakan untuk penelitian lebih lanjut. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah objek data yang dikumpulkan, penelitian ini mengumpulkan data cuaca dimana skripsi akan mengumpulkan data publikasi

2.2.4 Implementation of Web Scraping for Journal Data Collection on the SINTA Website (Adila, 2022)

Penelitian ini bertujuan untuk meningkatkan kualitas penelitian Indonesia.Metode yang digunakan adalah mengumpulkan dan literatur dari berbagai sumber dan pengumpulan data akan dijadwalkan dengan menggunakan cron job scheduling. Hasil dari penelitian adalah 7412 data yang berasal dari sumber SINTA dan setelah di filter berkurang menjadi 977 data. Perbedaan penelitian yang dilakukan dengan skripsi ini adalah data yang didapatkan dari scraping akan digunakan untuk keperluan melengkapi data aplikasi IKP2M secara otomatis