BAB I

PENDAHULUAN

1.1. Latar Belakang

Evolusi *World Wide Web* (WWW) menjadi scenario pertukaran data dan penggunaan internet yang sangat cepat berubah. Banyaknya pengguna yang menggunakan fasilitas internet membuat data yang ada di internet terus bertambah setiap harinya. Akibat dari pertukaran, pembagian dan penyimpanan data di internet yang begitu cepat, maka muncul masalah baru yaitu bagaimana menangani kelebihan data tersebut dan bagaimana cara menangani data yang berlebihan. *Web scraping* menjadi solusi untuk mendapatkan data terstruktur dari kumpulan data yang tersedia di web (Saurkar & Gode, 2018).

Adanya teknik *web scraping* ini dapat digunakan untuk kepentingan penelitian, analisis data, pengumpulan informasi dari berbagai media di internet secara otomatis. Contoh penggunaan *web scraping* diantaranya analisis terkait bencana alam terhadap media *online*, akuisisi dan kategorisasi informasi laman web tentang *hidroponik*, membuat klasifikasi lowongan kerja dari pencarian di internet (Priyanto & Ma'arif, 2018; Slamet et al., 2018; Sonya, 2016).

Beberapa beberapa bahasa pemrograman telah memiliki berbagai *library* yang dapat digunakan untuk melakukan *web scraping* dengan mengkombinasikanya menggunakan teknik *web scraping* yang sudah ada. *Beautifulsoup* dan *lxml* merupakan contoh *library* yang umum digunakan untuk melakukan web scraping pada pemrograman python (Uzun, Erdinc; Yerlikaya, Tarik; Kirat, 2018).

Bahasa pemrograman Python merupakan bahasa pemrograman yang populer dan baik untuk *web scraping*. Python dapat menangani banyak tugas perayapan data atau *web scraping* dengan nyaman dan tidak perlu mempelajari kode yang sangat rumit (Richard Brown, 2018). Library scraping web yang canggih dan sangat berkembang ini menjadikan Python bahasa pemrograman yang populer dan terbaik untuk web scraping. Python juga memiliki fleksibilitas, efisiensi perayapan, kemudahan coding, pemeliharaan yang mudah juga kekokohan dan skalabilitas yang baik (Prowebscraper, 2018; X-Byte Enterprise Crawling, 2019).

Perbandingan metode *web scraping* menggunakan *XPath* dan *CSS Selector*, dengan parameter: jumlah item, ukuran file, penggunaan memori dan waktu pemrosesan telah dilakukan pada penelitian sebelumnya (Rizaldi & Putranto, 2017). *Scrappy* yang merupakan *web scraping framework* pada bahasa pemrograman python digunakan pada penelitian tersebut. Hasil dari penelitian tersebut diketahui bahwa *Xpath* memiliki kelengkapan data yang lebih unggul dan pemrosesan yang lebih cepat. Namun pada ukuran data *CSS Selector* jumlah *item* dan ukuran *file* yang didapatkan lebih kecil dibandingkan dengan metode *Xpath*, sedangkan penggunaan memori dari keduanya tidak jauh berbeda (Rizaldi & Putranto, 2017).

Perbandingan metode *web scraping* menggunakan *Regex, HTML DOM* dan *XPath* dengan parameter pengukuran waktu, penggunaan memory dan penggunaan data yang dilakukan menggunakan bahasa pemrograman *Java* telah dilakukan dalam penelitian sebelumnya (Gunawan et al., 2019). Hasil dari penelitian tersebut

menunjukkan bahwa *HTML DOM* lebih unggul dalam waktu pemrosesan dan penggunaan data dari *Regex* dan *Xpath*.

Banyak data yang akan diambil dari internet menggunakan teknik *web scraping* akan mempengaruhi proses dalam melakukan *scraping*, selain waktu pemrosesan yang lama efisiensi yang terjadi akan terganggu terlebih jika *URL* yang digunakan lebih dari satu. *Multiprocessing* dapat digunakan pada kondisi seperti ini, dengan sifatnya yang paralel *multiprocessing* dapat melakukan *scraping* pada beberapa *URL* secara bersaamaan sehingga menghemat waktu dan membuat proses tersebut lebih efisien (Ahuja, 2018). *Multiprocessing* lebih tepat digunakan bila program yang dibuat merupakan program statis yang tidak memilliki *input output I/O*, hal ini karena *multiprocessing* yang paralel sedangkan *multithreading* bersifat serial (Baatout, 2018)

Penelitian sebelumnya yang dilakukan Rizaldi & Purnomo (2017) membandingkan dua teknik *web scraping CSS Selector* dan *XPath* menggunakan bahasa pemrograman *python* dengan hasil dari keduanya hanya memiliki selisih nilai yang tipis pada parameter yang diujikan. Penelitian lainnya yang dilakukan Gunawan (2019) membandingkan tiga teknik *web scraping HTML DOM, Regex* dan *XPath* menggunakan bahasa pemrograman *java* dengan sedikit parameter yang berbeda dari penelitian sebelumnya sehingga didapati hasil *HTML DOM* lebih unggul dari teknik lainnya yang digunakan dalam penelitian tersebut. Penelitian sebelumnya belum melakukan pengujian terhadap kapasitas data yang lebih banyak dan menggunakan lebih dari satu *URL* sebagai objek data yang diuji.

Diusulkannya penelitian ini berlandaskan dari usulan-usulan penelitian sebelumnya dengan menggunakan empat teknik *web scraping* yang diambil dari penelitian sebelumnya sebagai bentuk penelitian lanjutan serta menambahkan beberapa parameter yang belum diujikan sebelumnya. Sehingga teknik *web scraping* yang akan digunakan, yaitu *CSS Selector, HTML DOM, Regex* dan *XPath* dengan parameter yang digunakan dalam pengujian berupa pengukuran waktu proses, penggunaan memori, penggunaan *bandwith*, penggunaan *CPU* dan jumlah objek data yang didapat.

Dipilihnya bahasa pemrograman *python* pada penelitian ini selain bahasa pemrograman yang populer digunakan juga merupakan bahasa pemrograman yang cocok digunakan untuk *web scraping*, karena kompleksibilitas dan banyaknya dukungan *library* yang akan digunakan pada penelitian ini. Implementasi *multiprocessing* digunakan selain untuk meningkatkan performa dalam menghemat waktu, juga digunakan agar lebih efisien saat melakukan *scraping* karena pengujian dilakukan terhadap beberapa *URL* secara bersamaan, sedangkan pada prosesnya *singleprocessing* tidak dapat melakukan *scraping* pada beberapa *URL* secara bersamaan.

Perbandingan teknik *web scraping* ini bertujuan untuk mengukur performa dari setiap teknik *web scraping* yang diusulkan berdasarkan beberapa parameter tertentu yang berguna sebagai tolak ukur dalam memilih teknik *web scraping* yang akan digunakan berdasarkan performa yang dibutuhkan. Penggunaan *multiprocessing* pada *web scraping* ditujukan sebagai bentuk *paralel programing* yang tidak hanya berfungsi untuk mempercepat suatu proses, namun juga efisiensi dalam proses *web*

scraping dengan performa yang lebih baik terutama saat melakukan *scraping* pada beberapa *URL* secara bersamaan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, permasalahan yang akan dibahas adalah:

1. Bagaimana mengetahui jumlah objek data dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*?
2. Bagaimana mengetahui ukuran penggunaan *CPU* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*?
3. Bagaimana mengetahui ukuran penggunaan *Memory* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*?
4. Bagaimana mengetahui ukuran *execution time* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*?
5. Bagaimana mengetahui ukuran *bandwith usage* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*?

1.3. Batasan Masalah

Beberapa batasan masalah dalam penelitian ini adalah:

1. Bahasa pemrograman yang digunakan adalah *Python.*
2. Program yang dibuat bersifat statis dan tidak menggunakan *input output.*
3. Pengukuran objek data adalah pengukuran yang dilakukan dengan menghitung jumlah data yang terekstrak dari program yang telah dibuat.
4. Pengukuran penggunaan *CPU* & *Memory* adalah pengukuran yang dilakukan pada *process ID* dari program yang telah dibuat.
5. Pengukuran *bandwith usage* adalah pengukuran yang dilakukan pada *process ID* dari program yang telah dibuat.
6. Objek data yang menjadi uji adalah text pada sebuah website.
7. *Library* yang digunakan menyesuaikan dengan kompatibilitas metode yang digunakan, diantaranya *library lxml* untuk metode *Xpath*, *library BeautifulSoup* untuk metode *CSS Selector*, *library htmldom* untuk metode *HTML DOM* dan *library re* untuk metode *Regex*.
8. Pengujian yang dilakukan secara *offline* pada target website yang telah dibuat.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, tujuan dalam penelitian ini adalah:

1. Mengetahui jumlah objek data dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*.
2. Mengetahui ukuran penggunaan *CPU* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*.

3. Mengetahui ukuran penggunaan *Memory* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*.
4. Mengetahui ukuran penggunaan *bandwith usage* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*.
5. Mengetahui ukuran *execution time* dalam proses *web scraping* menggunakan *XPath, Regex, HTML DOM* dan *CSS Selector* dengan menerapkan *multiprocessing*.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut :

1. Pengujian pada penelitian ini diharapkan dapat menjadi tolak ukur kualitas metode *web scraping* dengan penerapan *multiprocessing*.
2. Dengan adanya penelitian ini diharapkan menjadi solusi pemilihan metode *web scraping* bagi pengguna individu ataupun kelompok dalam melakukan *web scraping*.
3. Penelitian ini diharapkan menjadi solusi untuk pemilihan metode *web scraping* yang akan digunakan.

1.6. Metodologi Penelitian

1. Analisis Permasalahan

   Memahami latar belakang permasalahan yang dihadapi dan alasan penelitian perlu dilakukan.

2. Persiapan Pengujian

Pengaturan pengujian dilakukan sebagai sarana jalannya aplikasi yang akan dilakukan perbandingan. Tahapan persiapan pengujian terdapat analisis objek data, persiapan program yang masing-masing bagiannya disiapkan secara berurutan sampai persiapan siap digunakan untuk melakukan pengukuran pada proses web scraping.

3. Pengukuran dan Pengujian

Tahap pengukuran dan pengujian mengukur jumlah objek data, penggunaan *CPU*, penggunaan memori, *execution time* dan *bandwith usage* dalam proses *web scraping* diukur dan dilakukan pencatatan.

4. Dokumentasi

Proses dokumentasi hasil penelitian dilakukan selama penelitian dengan menyusun laporan dalam bentuk skripsi.

1.7. Sistematika Penulisan

Penulisan dalam laporan tugas akhir ini menggunakan sistematika pembahasan sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan berisikan latar belakang, identifikasi masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab landasan teori berisikan teori-teori yang berkaitan dengan uraian pembahasannya yaitu *Web Scraping*, *Multiprocessing*, metode *web scraping* yang digunakan dalam penelitian ini, dll.

BAB III METODOLOGI

Bab metodologi berisikan uraian analisis mengenai cara yang dilakukan dalam pengukuran *objek data*, penggunaan *CPU*, penggunaan *Memory*, *execution time* dan penggunaan *bandwith* pada proses *web scraping* dengan menerapkan *multiprocessing*.

BAB IV HASIL DAN PEMBAHASAN

Membahas perbandingan hasil pengukuran dari proses *web scraping* dengan menerapkan *multiprocessing*.

BAB V KESIMPULAN

Bab kesimpulan akan memuat kesimpulan dan saran keseluruhan dari bab sebelumnya sebagai hasil yang diperoleh diharapkan dapat bermanfaat dalam pengembangan selanjutnya.