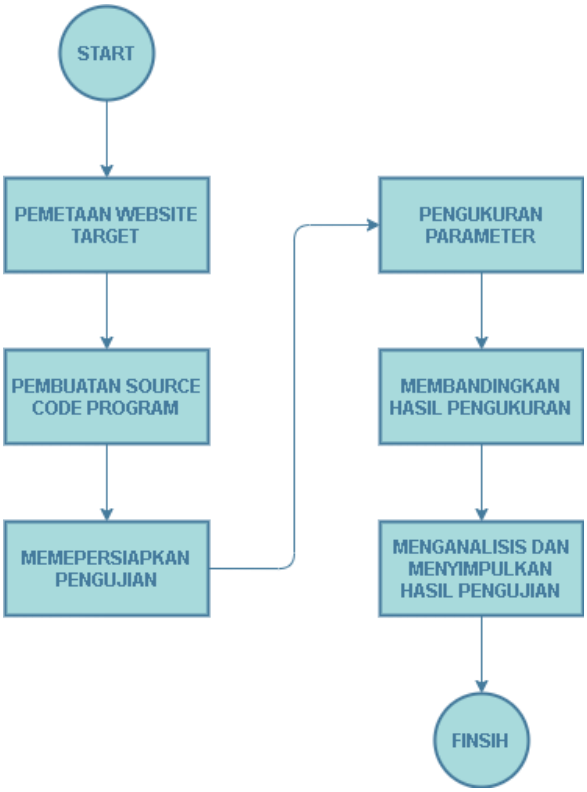
BAB III

METODOLOGI

Terdapat enam tahapan yang dilakukan dalam penelitian ini, diantaranya: pemetaan website target, pembuatan source code program, mempersiapkan pengujian, pengukuran parameter, membandingkan hasil pengukuran dan menganalisis serta menyimpulkan hasil pengujian seperti yang ditampilkan pada *Gambar III.1*.



Gambar III.1 Tahapan Penelitian

3.1. Pemetaan Website Target

Pemetaan website merupakan salah satu cara yang dilakukan untuk mengidentifikasi meta atribut yang mengandung objek data. Dilakukan pemetaan

ini agar objek data yang diambil lebih spesifik seperti yang ditunjukan pada *Gambar III.2*.

Gambar III.2 Pemetaan Website Target

Data yang akan diambil dari website target berupa data profil yang berbeda-beda. Gambar IV.2 menunjukkanbahwa setiap satu record memiliki lima buah entri data yang berbeda dengan atribut entri yang berbeda. Meta atribut yan g ada pada website tersebut menjadi sebuah jalur pemetaan yang dapat digunakan untuk melakukan scraping pada sejumlah record yang ada pada website tersebut, yaitu atribut nama, alamat, tgl, email dan ssn. Data *dummy* yang dibuat dapat diakses pada laman *https://mmaul.tech/scrapeit* atau *https://codelatte.org/maul* dan setiap laman memiliki *subdirektori* dengan setiap direktori berjumlah 5000 data *dummy* dan 5 *subdirektori* sehingga total data *dummy* yang digunakan sebanyak 25000 data.

3.2. Pembuatan Source Code Program

Pembuatan source code program terbagi menjadi dua versi untuk versi yang pertama code program dibuat tanpa menggunakan *multiprocessing*, sedangkan

versi kedua dibuat menggunakan *multiprocessing*. Hal ini dilakukan untuk meneliti perbedaan keduanya dari hasil parameter uji yang didapat. Setiap versi code terdiri dari empat buah metode *web scraping* yang berbeda, sehingga total *source code* program yang dibuat ada delapan code program.

Source code program yang dibuat tidak sepenuhnya menggunakan *library* yang sama, hal ini karena setiap metode memiliki kompatibilitas yang berbeda, sehingga *library* yang digunakan setiap metode akan disesuaikan dengan kompatibilitas masing-masing metode.

Library yang digunakan diantaranya metode *xpath web scraping* menggunakan *lxml, regex web scraping* menggunakan *re, css selector web scraping* menggunakan *beautifull soup* dan *html dom web scraping* menggunakan *htmldom.* Penerapan *multiprocessing* hanya digunakan pada versi kedua dari program yang dibuat. *Multiprocessing* akan berjalansaat memanggil fungsi dari setiap kode program, sehinga program versi pertama akan langsung menjalankan fungsi yang telah dibuat, sedangkan versi kedua fungsi pad program dijalankan melalui *multiprocessing*.

3.3. Mempersiapkan Pengujian

Proses persiapan pengujian mendefinisikan kesiapan source code program yang akan digunakan untuk melakukan *scraping* pada website target, dengan melengkapi library yang akan digunakan untuk menghitung parameter pengujian. Library yang digunakan untuk pengujian diantaranya *os, time, psutil, requests* dan *urllib.request*.

a. Menghitung Waktu Proses

```
1. import os
2. # Menghitung Waktu Proses
3. start = time.time()
4. # Bagian Kode Program
5.
6. # End Time Process
7. end = time.time()
8. # Hasil Waktu Proses
9. print("Waktu proses : {} second".format(end-
   start))
```

Library *os* digunakan untuk menghitung parameter waktu, dengan menggunakan fungsi *time.time()* yang dimiliki library tersebut. Fungsi ini digunakan pada dua sesi yaitu waktu awal dan waktu akhir dan dilakukan proses pengulangan dari waktu akhir dikurangi waktu awal maka akan didapatkan hasil penggunaan waktu yang digunakan sebuah program.

b. Menghitung Parameter Penggunaan Memori

```
1. import psutil
2. # Menghitung Penggunaan Memori Start
3. memori1 = psutil.Process(os.getpid()).memory_info
   ().rss
4. # Bagian Kode Program
5.
6. # Menghitung Penggunaan Memori End
7. memori2 = psutil.Process(os.getpid()).memory_info
   ().rss
8. # Hasil Penggunaan Memori
9. print("Penggunaan memory : {} bytes".format(memor
   i2-memori1))
```

Library *psutil* digunakan untuk menghiting penggunaan bandwith pada program. Library tersebut memiliki fungsi untuk mendapatkan informasi sistem perangkat yang sedang digunakan, dengan adanya library ini

menghitung penggunaan memori pada sebuah program ataupun sistem yang sedang berjalan menjadi lebih mudah.

c. Menghitung Penggunaan CPU

```
1. import psutil
2. # Menghitung Penggunaan CPU Awal
3. cpu0 = psutil.cpu_percent()
4. # Bagian Kode Program
5.
6. # Menghitung Penggunaan CPU Awal
7. cpu = psutil.cpu_percent()
8. # Hasil Penggunaan CPU
9. print("CPU yang digunakan : {} %".format(cpu))
```

Library *psutil* dapat digunakan juga untuk menghitung penggunaan cpu suatu program dengan nilai data yang muncul berupa persentase penggunaan cpu.

d. Menghitung Penggunaan Bandwith Network

```
1. import psutil
2. # Menghitung Bandwith Upstream dan Downstream
3. up = psutil.net_io_counters().bytes_sent
4. down = psutil.net_io_counters().bytes_recv
5. # Bagian Kode Program
6.
7. # Menghitung Bandwith Usage
8. up0 = psutil.net_io_counters().bytes_sent
9. down0 = psutil.net_io_counters().bytes_recv
10.# Hasil Bandwidth Upstream & Downstream
11.print("Penggunaan Bandwidth : Upstream = {} Do
   wnstream = {}".format(up0-up,down0-down))
```

Penggunaan bandwith network dapat dihitung menggunakan library *psutil* dengan memanfaatkan fungsi *net_io_counters()* dari penggunaan fungsi tersebut dilakukan pengurangan dengan cara membuat fungsi

penghitungan network akhir dikurangi fungsi penghitungan network awal.

e. Mempersiapkan Perangkat Pengujian

Perangkat pengujian yang disiapkan berupa perangkat keras berupa *Virtual Private Server (VPS Single Server)* dan perangkat lunak yang digunakan pada pengujian dengan spesifikasi seperti yang disajikan pada *Tabel III.1*

Table III-1 Spesifikasi Perangkat Pengujian

NO	ITEM	SPESIFIKASI
1.	*CPU*	*Intel(R) Xeon(R) CPU @ 2.30GHz*
2.	*RAM Memory*	*4 GB*
3.	*Operating System*	*Debian 4.9.228-1 (2020-07-05) x86_64*
4.	*Programing Language*	*Python 3.8.0*

3.4. Pengukuran Parameter

Pengukuran parameter dilakukan dengan menjalankan setiap program yang telah dibuat satu persatu, hal ini dilakukan agar kinerja program tidak terganggu program lain nya yang berjalan secara bersamaan. Parameter yang diukur akan lebih akurat apabila hanya satu program yang berjalan. Kelima parameter yang diukur akan muncul sebagai output dari setiap program yang telah dijalankan.

3.5. Membandingkan Hasil Pengukuran

Hasil pengujian berupa beberapa parameter output dimuat kedalam tabel dan dilakukan pengambilan nilai rata-rata dari hasil pengujian setiap parameter yang telah diuji sebanyak dua puluh kali. Pengujian dilakukan sebanyak dua puluh kali

karena pada angka pengujian tersebut hasil dari parameter pengujian sudah terlihat benar-benar stabil, sehingga parameter output dari setiap program dibandingkan untuk mendapatkan hasil akhir dari pengukuran parameter tersebut.

3.6. Menganalisis dan Menyimpulkan Hasil Pengujian

Menganalisis data pengujian antara keempat metode dengan dua penerapan yang berbeda digunakan kemudian menentukan mana yang lebih baik untuk setiap parameter yang diujikan.