

**LAPORAN KERJA PRAKTIK**

**DATA CRAWLING OTOMATIS PADA TWITTER**



Oleh :

Sarikhin

311710871

**PROGRAM STUDI TEKNIK INFROMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PELITA BANGSA**

**BEKASI**

**2020**

**LEMBAR PERSETUJUAN**

**LAPORAN PRAKTEK**

**DATA CRAWLING OTOMATIS PADA TWITTER**

Diajukan untuk memenuhi syarat kelulusan pada Program Studi Kuliah Kerja  
Praktek Universitas Pelita Bangsa

Disusun oleh :

**Sarikhin**

**311710871**

Telah diperiksa dan dilaksanakan

Pada tanggal :

Pembimbing Lapangan

Dosen Pembimbing

**Ahmad Turmudizy, S.Kom., M.Kom**  
**NIDN. 0408087907**

**Arif Siswandi, S.Kom., M.M**  
**NIDN. 0414077406**

Mengetahui,  
Ketua Program Studi Teknik Informatika

**Aswan S. Sunge, S.E, M.Kom**  
**NIDN. 0426018003**

**LEMBAR PENGESAHAN**

**LAPORAN PRAKTEK**

**DATA CRAWLING OTOMATIS PADA TWITTER**

Disusun oleh :

**Sarikhin**

**311710871**

Telah dipertahankan didepan Dewan Penguji  
pada tanggal : 08 Agustus 2020

Dosen Penguji I

Dosen Penguji II

**Sufajar Butsianto, S.Kom, M.Kom**  
**NIDN. 0424068106**

**Donny Maulana, SKom.,MMSi**  
**NIDN. 0408107502**

Mengetahui,  
Ketua Program Studi Teknik Informatika

**Aswan S. Sunge, S.E, M.Kom**  
**NIDN. 0426018003**

## KATA PENGANTAR

Puji dan Syukur penulis panjatkan kehadiran Allah Subhanallohu Wata'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penulisan laporan Kuliah Kerja Praktek ini.

Pada kesempatan ini penulis mengucapkan rasa terima kasih yang sedalam-dalamnya kepada :

1. Hamzah Muhammad Mardi Putra, S.K.M., M.M., sebagai Rektor Universitas Pelita Bangsa
2. Dr. Ir. Supriyanto, M.P., sebagai Wakil Rektor I Universitas Pelita Bangsa
3. Putri Anggun Sari, S.Pt., M.Si. Sebagai Dekan Fakultas Teknik, Universitas Pelita Bangsa
4. Aswan S. Sunge, S.E, M.Kom Sebagai Ketua Program Studi Teknik Informatika S-1
5. Bapak Ahmad Turmudizy S.Kom, M.Kom, dan Bapak Arif Siswandi, S.Kom,. M.M selaku dosen pembimbing dan senantiasa ikhlas dijadikan tempat bertanya untuk berbagi ilmu.
6. Semua Bapak/Ibu dosen, keluarga, teman, yang telah membantu penulis dalam penyusunan tugas akhir ini hingga selesai yang tidak dapat saya sebutkan satu per-satu.
7. Dan tidak lupa juga untuk Siti lailatul karimah istriku tercinta yang telah memberikan semangatnya

Penulis menyadari bahwa dalam penulisan laporan Kuliah Kerja Praktek ini jauh dari kata sempurna, untuk itu penulis sangat mengharapkan dukungan dari para pembaca untuk memberikan kritik dan saran yang membangun guna perbaikan di masa yang akan datang.

Bekasi, 12 Juli 2020

Sarikhin

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>v</b>
<b>DAFTAR TABEL.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>viii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Pembatasan Masalah .....	1
1.3. Tujuan .....	2
1.4. Manfaat .....	2
1.5. Sistematika Penulisan .....	2
<b>BAB II LANDASAN TEORI .....</b>	<b>4</b>
2.1. Definisi Judul .....	4
2.1.1. Definisi Web Crawler.....	4
2.1.2. Definisi Crawling Data .....	4
2.1.3. Twitter .....	4
2.1.4. API Twitter .....	5
2.2. Basis Data.....	6
2.2.1. MySQL.....	7
2.3. Teori Perancangan Sistem .....	8
2.3.1. Flowchart .....	8
2.3.2. UML .....	11
2.4. Bahasa Pemrograman.....	18
2.4.1. Bahasa pemrograman Python.....	18
2.4.2. Framework Flask .....	19
2.4.3. Unicorn .....	20
2.4.4. Nginx .....	20

2.4.5. HTML.....	21
2.4.6. Pengertian Bootstrap .....	21
2.4.7. Datatable .....	24
<b>BAB III PEMBAHASAN .....</b>	<b>25</b>
3.1. Flowchart Sistem Yang Berjalan.....	25
3.1.1. Crawling menggunakan <i>script</i> Pyhon dan <i>library tweepy</i> .....	25
3.1.2. Crawling menggunakan TAGS V6.....	26
3.1.3. Crawling menggunakan RapidMiner .....	27
3.2. Usulan Perancangan Sistem .....	28
3.2.1. <i>Use Case Diagram</i> .....	29
3.2.2. <i>Activity Diagram</i> .....	30
3.2.3. <i>Sequence Diagram</i> .....	33
3.2.4. <i>Class Diagram</i> .....	36
3.3. Desain Tabel.....	37
3.3.1. Tabel Tweet .....	37
3.4. Prototipe.....	37
<b>BAB IV PENUTUP .....</b>	<b>38</b>
4.1. Kesimpulan.....	38
4.2. Saran.....	38
<b>DAFTAR PUSTAKA .....</b>	<b>39</b>

## DAFTAR TABEL

Tabel : 2.1 Simbol Flow Direction .....	8
Tabel :2.2 Simbol Processing .....	9
Tabel :2.3 Simbol Masukan dan Keluaran .....	11
Table 2.4 Simbol Use Case Diagram .....	12
Table 2.5 Simbol Activity Diagram .....	14
Table 2.6 Simbol Sequence Diagram .....	16
Tabel 3.1 Keterangan Activity Diagram Pencarian Tweets .....	30
Tabel 3.2 Keterangan Activity Diagram Melihat Tweet Hasil Crawling .....	32
Tabel 3.3 Keterangan Activity Diagram Download Tweets .....	33
Tabel 3.4 Keterangan Sequence Diagram Pencarian Tweets .....	34
Tabel 3.5 Keterangan Sequence Diagram Melihat Tweet Hasil Crawling .....	35
Tabel 3.6 Keterangan Sequence Diagram Download Tweet .....	36
Tabel 3.7 Tabel Tweet .....	37

## DAFTAR GAMBAR

Gambar 2.1 Contoh Activity Diagram .....	15
Gambar 2.2 Contoh Sequence Diagram .....	17
Gambar 3.1 Flowchart Crawling menggunakan Script Python .....	25
Gambar 3.2 Flowchart Crawling menggunakan TAGS .....	26
Gambar 3.3 Flowchart Crawling menggunakan Rapidminer .....	27
Gambar 3.4 Flowchart Crawling menggunakan Web Crawling .....	28
Gambar 3.5 Use Case Diagram Sistem Yang Diusulkan .....	29
Gambar 3.6 Activity Diagram Pencarian Tweets .....	30
Gambar 3.7 Activity Diagram Melihat Tweet Hasil Crawling .....	31
Gambar 3.8 Activity Diagram Download Tweets .....	32
Gambar 3.9 Sequence Diagram Pencarian Tweets .....	33
Gambar 3.10 Sequence Diagram Melihat Tweet Hasil Crawling .....	34
Gambar 3.11 Sequence Diagram Download Tweets .....	35
Gambar 3.11 Class Diagram Aplikasi .....	36



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Analisis sentimen terhadap tokoh publik, produk dan lainnya pada media sosial merupakan alat yang ampuh untuk memahami sikap, referensi dan opini masyarakat terhadap objek tersebut. Saat ini jumlah data pada media sosial sudah terbilang sangatlah besar, akan tetapi jumlah data yang besar tersebut belum banyak digunakan untuk menjadi sesuatu informasi yang memiliki nilai guna. Twitter merupakan salah satu media sosial yang terbesar di dunia yang memiliki data yang dapat kita manfaatkan menjadi informasi yang memiliki nilai guna. Twitter menyediakan *Application Program Interface* (API) yang memungkinkan kita sebagai *programmer* dapat memperoleh data dari mereka, melalui API twitter ini kita dapat melakukan dua sistem pencarian, yaitu *by user* dan *by keyword*. Pada penelitian ini penulis hanya akan membahas mengenai pencarian *by keyword* yaitu pencarian data menggunakan penggalan kata maupun hashtag dan data yang diperoleh salah satunya yaitu *text* atau yang biasa disebut *tweet*. *Tweet* merupakan diposting yang diunggah oleh pengguna twitter berupa tulisan dan *emoticon* yang nanti dapat dimanfaatkan untuk analisis sentimen.

Hanya saja untuk mendaftarkan diri sehingga mendapatkan API twitter tidaklah mudah belum lagi cara untuk mendapatkan data dari twitter masih sangat sulit dan tidak semua dapat melakukannya. Sehingga mengakibatkan proses pengumpulan data untuk analisis sentimen menggunakan data twitter menjadi kurang efisien dari segi waktu. Dari permasalahan tersebut mendorong penulis untuk membangun aplikasi berupa web yang berfungsi untuk crawling data twitter secara otomatis sehingga dapat membantu pengumpulan informasi dari twitter.

### 1.2. Pembatasan Masalah

Berdasarkan latar belakang diatas penulis melakukan pembatasan masalah sebagai berikut :

- 1.2.1. Aplikasi yang akan penulis bangun merupakan Web yang berfungsi hanya sebagai crawler data twitter secara otomatis.
- 1.2.2. Aplikasi Web ini dapat menyimpan data hasil crawling pada database dan data dapat disimpan oleh pengguna dalam format CSV, PDF ataupun File Excel.

### **1.3. Tujuan**

Adapun tujuan dari penelitian ini adalah sebagai berikut :

- 1.3.1. Mempermudah dalam pengumpulan data dari twitter karena tidak perlu mendaftar sebagai developer Twitter.
- 1.3.2. Mempercepat proses pengumpulan data dari twitter karena dilakukan secara otomatis.
- 1.3.3. Membuat web yang dapat diimplementasikan dan dapat dipakai secara luas, tidak hanya sebatas konsep.

### **1.4. Manfaat**

Adapun manfaat dari pembuatan penelitian ini adalah sebagai berikut :

- 1.4.1. Pengguna web dapat dengan mudah melakukan web crawling tanpa mendaftar sebagai developer twitter.
- 1.4.2. Efisiensi waktu proses crawling twitter dari metode sebelumnya.
- 1.4.3. Manfaat yang didapat oleh penulis adalah menambah pengetahuan mengenai bahasa pemrograman python.
- 1.4.4. Menambah pengetahuan terkait pemakaian bootstrap pada tampilan website.

### **1.5. Sistematika Penulisan**

Sistematika penulisan laporan terdiri dari :

BAB I : Pendahuluan

Bab ini terdiri dari latar belakang, pembatasan masalah, tujuan dan manfaat, waktu dan tempat pelaksanaan dan sistematika penulisan.

BAB II : Landasan Teori

Bab ini menjelaskan tentang definisi judul, Basis Data, UML (Unified Modeling Language).

### BAB III : Hasil dan Pembahasan

Bab ini membahas sekilas tentang flowchart sistem yang berjalan dan sistem yang diusulkan.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Definisi Judul**

##### **2.1.1. Definisi Web Crawler**

Web Crawler adalah program yang menelusuri Word Wide Web dengan cara metodelis, otomatis dan teratur. Istilah lain dari web crawler adalah And, otomatis indexer, bos, web, spiders, web robot. Web Crawler adalah salah satu jenis bot atau agen perangkat lunak. Secara umum, proses crawling dimulai dengan list URL (*Unifrom Resource Locator*) yang akan dikunjungi, disebut seeds. Kemudian web crawler akan mengunjungi URL tersebut satu per satu. Setiap page URL yang dikunjungi akan diidentifikasi apakah ada hyperlink di dalamnya. Jika ada maka akan ditambahkan ke dalam list URL yang akan dikunjungi. Ini disebut crawl frontier. URL yang didapat dari crawl frontier akan dikunjungi secara rekursif dengan beberapa keijakan tertentu.

##### **2.1.2. Definisi Crawling Data**

Crawling data merupakan tahap di dalam penelitian yang bertujuan untuk mengumpulkan atau mengunduh data dari suatu database. Pengumpulan data dari laporan ini yaitu data yang diunduh dari server twitter berupa tweet dari pengguna twitter. Sehingga kita memerlukan API Twitter untuk mengakses data tersebut.

##### **2.1.3. Twitter**

Twitter adalah layanan jejaring sosial, yang memungkinkan pengguna untuk berkomunikasi satu sama lain. Pengguna dapat mengirim dan membaca pesan yang terdiri dari maksimal 140 karakter, disebut tweets. Mulai Januari 2018, Twitter telah mengubah jumlah karakter yang digunakan untuk mengirim komentar menjadi 280 karakter. Tweet dari pengguna lain yang menghadiri atau mengikuti akan muncul di halaman beranda untuk

dibaca. Pengguna dapat men-tweet atau mengirim ulang pesan yang dikirim oleh pengguna lain. Dalam pesan yang dikirim, jika nama pengguna lain aktif maka tweet tulisan tanda @ diikuti oleh nama pengguna. Pengguna dapat menggunakan tanda # (tagar) untuk menulis pesan berdasarkan topik.

Twitter hingga yang kedua sebagai situs jejaring sosial adalah yang paling banyak dikunjungi di dunia, yang sebelumnya berada di peringkat dua puluh dua. Menurut data dari Direktorat Jenderal Informasi dan Komunikasi Publik Layanan Informasi (IKP), situs jejaring sosial adalah yang paling mudah diakses oleh Facebook dan Twitter. Indonesia menempati urutan kelima dalam pengguna Twitter terbesar di dunia di bawah negara bagian AS, Brasil, Jepang, dan Inggris. Tingginya popularitas Twitter menyebabkan layanan ini telah dimanfaatkan untuk berbagai keperluan dalam berbagai aspek, misalnya sebagai sarana protes, kampanye politik, alat belajar, dan sebagai media komunikasi.

#### **2.1.4. API Twitter**

*Application Programming Interface* (API) twitter adalah suatu program atau aplikasi yang disediakan oleh twitter untuk mempermudah developer aplikasi lain dalam mengakses informasi yang ada di website twitter. Melalui API twitter program yang kita buat dapat berkomunikasi langsung ke server twitter agar dapat meminta dan menyajikan informasi yang diminta.

Pendaftaran sebagai developer aplikasi twitter untuk menggunakan API twitter harus dilakukan untuk mendapatkan consumer Key, consumer access, access token dan access token seret yang digunakan sebagai syarat otentifikasi dari aplikasi yang akan kita bangun. Tujuan dari otentifikasi adalah untuk hak akses developer dalam mengunduh data yang ada di twitter.

## 2.2. Basis Data

Basis Data adalah kumpulan data yang saling berhubungan secara logikal serta deskripsi dari data tersebut, yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Basis Data adalah sebuah penyimpanan data yang besar yang bisa digunakan oleh banyak pengguna dan departemen. Semua data terintegrasi dengan jumlah duplikasi yang minimum. Basis Data tidak lagi dipegang oleh satu departemen, tetapi dibagikan ke seluruh departemen pada perusahaan. Basis Data itu sendiri tidak hanya memegang data operasional organisasi tetapi juga penggambaran dari data tersebut (Connolly & Begg, 2010:64).

Basis data adalah kumpulan data store yang terintegrasi yang diatur dan di kontrol secara sentral. Sebuah basis data biasanya menyimpan ribuan class. Informasi yang disimpan termasuk class attribute dan relasi antar class. Basis data juga menyimpan informasi yang deksriptif seperti nama atribut, pemberian batasan suatu nilai, dan kontrol akses untuk data-data yang sensitif (Satzinger, Robert, & Stephen, 2005: 398).

Basis data juga diartikan sebagai sekumpulan file dikomputer yang saling terhubung. File file ini diatur sesuai kesamaan elemennya, sehingga data yang diinginkan dapat dicari secara mudah (Williams & Sawyer, 2007: 181).

Basis data adalah dua atau lebih simpanan data dengan elemenelemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data. Dari definisi diatas, maka dapat dikatakan bahwa basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya (Iskandar & Rangkuti, 2008: 3). Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data (Abdillah, 2012: 1).

Dapat disimpulkan basis data adalah penyimpanan data yang terstruktur, terintegrasi dan saling berkaitan dengan elemen-elemen penghubungnya dan dapat di akses dengan berbagai cara, oleh karena itu basis data juga bisa didefinisikan sebagai kumpulan yang menggambarkan sendiri dari catatan yang terintegrasi dan penggambaran dari data dikenal sebagai sistem katalog (atau kamus data atau metadata). Definisi data disini dibedakan dari program aplikasi, yang umumnya sama dengan pendekatan pengembangan modern perangkat lunak, dimana definisi internal dan eksternal dari sebuah objek dipisahkan. Salah satu keuntungan dari pendekatan tersebut adalah abstraksi data dimana kita dapat mengubah definisi internal dari sebuah objek tanpa mempengaruhi pengguna dari objek jika definisi eksternal objek tersebut tidak berubah.

### **2.2.1. MySQL**

MySQL adalah suatu Relational Database Management System (RDBMS) yang mendukung database yang terdiri dari sekumpulan relasi atau tabel Relational Database Management System (RDBMS) yang tersedia, tetapi MySQL khususnya sangat cocok untuk bekerja sama dengan PHP. Bagaimanapun, konsep dasar SQL tetap sama apa pun jenis database yang Anda gunakan. MySQL dilepaskan dengan suatu lisensi open source, dan tersedia secara cuma-cuma. MySQL bekerja pada berbagai sistem operasi, dan banyak masalah. MySQL bekerja dengan cepat dan baik dengan data yang besar. PHP menyediakan banyak fungsi untuk mendukung databaseMySQL. (Santoso, 2014). MySQL adalah multithreaded, multi-user, SQL (Structured Query Language) merupakan database server relasional yang menggunakan GNU General Public License. MySQL juga merupakan cross-platform, yang artinya dapat berjalan di Windows maupun Linux server.

## 2.3. Teori Perancangan Sistem

### 2.3.1. Flowchart

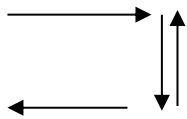

Indrajani (2015:36), “Flow chart adalah penggambaran secara grafik dari langkah -langkah dan urutan prosedur suatu program.”

Indrajani (2015:38), menjelaskan simbol - simbol dalam Flow Chart adalah sebagai berikut :


#### a. *Flow Direction Symbols*

Yaitu, simbol yang dipakai untuk menghubungkan antara simbol yang satu dengan simbol lainnya atau disebut juga *connecting line*.

**Tabel : 2.1 Simbol Flow Direction**

Gambar	Nama	Keterangan
	Garis Alir	Simbol yang digunakan untuk menunjukkan arah aliran algoritma, dari satu proses ke proses berikutnya.
	<i>On-page Connector</i>	Biasanya digunakan dalam pengulangan. Digunakan untuk menghubungkan satu proses ke proses lainnya, sama halnya seperti tanda panah. Boleh saja lebih dari satu proses yang mengarah kepadanya, namun hanya bisa menghasilkan satu keluaran. Sehingga diagram alir terlihat lebih rapi karena mengurangi


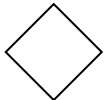


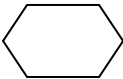

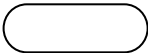

Gambar	Nama	Keterangan
		tanda panah yang lalu lalang di dalam diagram alir.
	<i>Off-page Connector</i>	Terkadang, diagram alir tidak muat dalam satu halaman saja, oleh karena itu, lambang ini berfungsi untuk menghubungkan satu proses ke proses lainnya, sama halnya seperti tanda panah, hanya saja untuk merujuk ke halaman yang berbeda

**b. Processing Symbols**

Merupakan simbol yang menunjukkan jenis operasi pengolahan dalam suatu prosedur.

**Tabel :2.2 Simbol Processing**

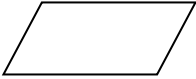


Gambar	Nama	Keterangan
	Proses / Langkah	Simbol yang digunakan untuk menunjukkan pengolahan / kegiatan yang dilakukan komputer.
	<i>Decision</i>	Simbol untuk proses / langkah di mana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan

Gambar	Nama	Keterangan
		aliran kondisi yang berbeda.
	<i>Preparation Symbol</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage.
	<i>Predefined Process</i>	Simbol yang digunakan untuk menunjukkan suatu proses yang begitu kompleks, sehingga tidak bisa dijelaskan di diagram alir ini dan merujuk pada diagram alir yang terpisah.
	<i>Terminal</i>	Simbol yang digunakan sebagai permulaan atau akhir dari suatu proses.
	Manual Input	Simbol yang digunakan untuk menunjukkan <i>input</i> data secara manual menggunakan <i>Online keyboard</i> .

**c. Input Output Symbols**

Simbol yang dipakai untuk menyatakan jenis peralatan yang digunakan sebagai media input atau output.

**Tabel :2.3 Simbol Masukan dan Keluaran**

Gambar	Nama	Keterangan
	<i>Input - Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya. Simbol ini hanya bisa untuk proses masukan tidak untuk proses keluaran.
	<i>Document</i>	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau <i>output</i> di cetak di kertas.
	<i>Disk and On-line Storage</i>	Simbol untuk menyatakan input berasal dari disk atau output di simpan ke disk.

**2.3.2. UML**

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2010: bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantic. Ketika


kita membuat model menggunakan konsep UML ada aturan – aturan yang harus diikuti. Bagaimana elemen pada model – model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan sistem yang kita buat? Dan sebagainya dapat dijawab dengan UML. Menurut Henderi (2010:5), Berikut ini adalah definisi mengenai 4 diagram UML



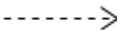




#### *a. Use Case*


Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara ‘aktor’—inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana.

Perilaku sistem adalah bagaimana sistem beraksi dan bereaksi. Perilaku ini merupakan aktifitas sistem yang bisa dilihat dari luar dan bisa diuji. Perilaku sistem ini dicapture di dalam USE CASE. USE CASE sendiri mendeskripsikan sistem, lingkungan sistem, serta hubungan antara sistem dengan lingkungannya.

**Table 2.4 Simbol Use Case Diagram**

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

Gambar	Nama	Keterangan
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Gambar	Nama	Keterangan
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).


### ***b. Activity Diagram***

Activity diagram adalah sesuatu yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya atau internal processing.

Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

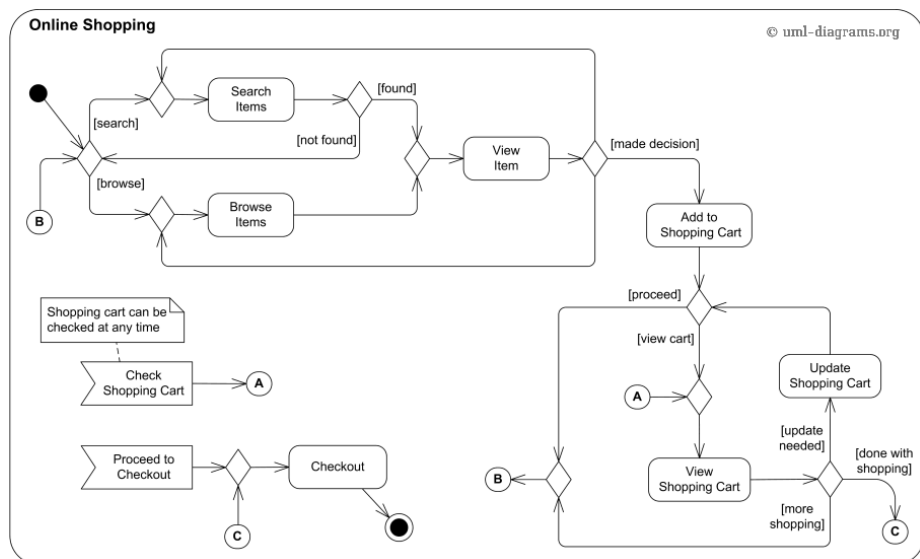
Dan berikut adalah symbol dan contoh Activity Diagram

***Table 2.5 Simbol Activity Diagram***

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.

Gambar	Nama	Keterangan
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Berikut adalah contoh Activity Diagram



**Gambar 2.1 Contoh Activity Diagram**

### c. Sequence Diagram

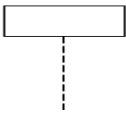

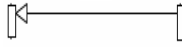
Diagram *sequence* merupakan salah satu yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu.

Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Diagram sequence menampilkan interaksi antar objek dalam dua dimensi. Dimensi vertikal adalah poros waktu, dimana waktu berjalan ke arah bawah. Sedangkan dimensi horizontal merepresentasikan objek-objek individual. Tiap objek (termasuk *actor*) tersebut mempunyai waktu aktif yang direpresentasikan dengan kolom vertical yang disebut dengan *lifeline*.

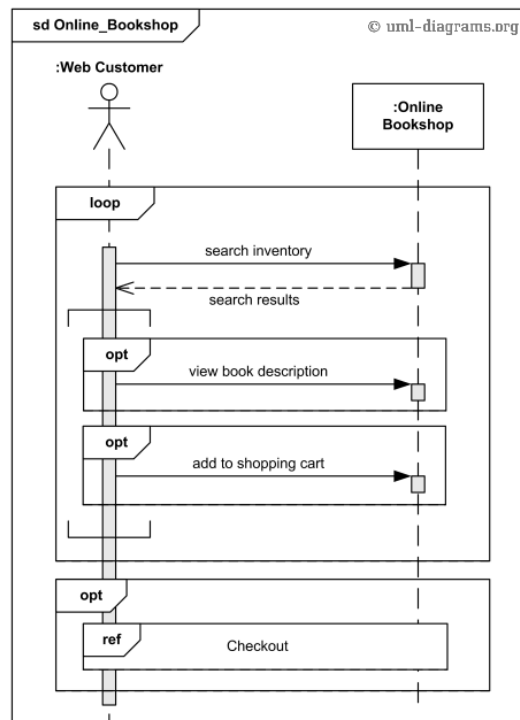
Pesan (*message*) direpresentasikan sebagai panah dari satu *lifeline* ke *lifeline* yang lain. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari *class*.

**Table 2.6 Simbol Sequence Diagram**

Gambar	Nama	Keterangan
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.



Berikut adalah contoh Diagram Sequence




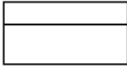

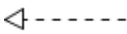


**Gambar 2.2 Contoh Sequence Diagram**

#### d. Class Diagram

Class diagram adalah model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class. Class diagram mirip ER-Diagram pada perancangan database, bedanya pada ER-diagram tdk terdapat operasi/methode tapi hanya atribut. Class terdiri dari nama kelas, atribut dan operasi/method.

**Tabel 2.7 Simbol Class Diagram**

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).

	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

## 2.4. Bahasa Pemrograman

### 2.4.1. Bahasa pemrograman Python

Python adalah sebuah bahasa pemrograman yang bisa digunakan pada beberapa platform (multiplatform), dan berifat sumber perangkat bebas terbuka (opensource), pertama kali dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Belanda. Bahasa ini dikategorikan sebagai bahasa tingkat tinggi (very-high-level language) dan merupakan bahasa berorientasi objek yang dinamis (object-oriented-dynamic language). Hal utama yang

membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program. Python memiliki aturan yang berbeda dengan bahasa lain, seperti indentasi, tipe data, tuple, dan dictionary. Python adalah bahasa pemrograman dinamis yang mendukung pemrograman berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi seperti Linux, Windows, Unix, Symbian dan masih banyak lagi. Python merupakan salah satu bahasa pemrograman favorit saat ini, karena Python menawarkan banyak fitur seperti:

1. Kepustakaan yang luas, menyediakan modul-modul untuk berbagai keperluan.
2. Mendukung pemrograman berorientasi objek.
3. Memiliki tata bahasa yang mudah dipelajari.
4. Memiliki sistem pengelolaan memori otomatis.
5. Arsitektur yang dapat dikembangkan (extensible) dan ditanam (embeddable) dalam bahasa lain, misal objek oriented Python dapat digabungkan dengan modul yang dibuat dengan C++.

#### **2.4.2. Framework Flask**

Flask adalah sebuah web framework yang ditulis dengan bahasa python dan tergolong sebagai jenis *microframework*. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu *web*. Dengan menggunakan flask dan bahasa Python, pengembang dapat membuat sebuah *web* yang restruktur dan dapat mengatur *behaviour* suatu *web* dengan lebih mudah.

Flask termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum seperti validasi form, *database*, dan sebagainya tidak terpasang *default* di Flask. Hal ini dikarenakan fungsi komponen-komponen tersebut sudah disediakan

oleh pihak ketiga dan Flask dapat menggunakan ekstensi yang membuat fitur seakan diimplementasikan oleh Flask sendiri,

Selain itu, meskipun Flask disebut sebagai *microframework*, bukan berarti Flask mempunyai kekurangan dalam hal fungsionalitas. *Microframework* di sini berarti bahwa Flask bermaksud untuk membuat *core* dari aplikasi ini sesederhana mungkin tetapi tetap dapat dengan mudah ditambahkan. Dengan begitu, fleksibilitas serta kapabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan *framework* lainnya.

#### 2.4.3. Gunicorn

Gunicorn merupakan singkatan dari “Green Unicorn” Gunicorn merupakan sebuah HTTP Server untuk Python yang berbasis *web server gateway interface* (WSGI) dan dikhususkan untuk lingkungan Unix-like.

Aslinya Gunicorn merupakan sebuah proyek yang diambil dari proyek Unicorn untuk Ruby. Gunicorn memiliki kompatibilitas tinggi dengan berbagai web framework berbasis WSGI seperti Django, Flask, Falcon, dan lainnya. Selain cepat dalam menangani *request*, Gunicorn juga cukup hemat dalam menggunakan sumber daya di mesin *server*.

#### 2.4.4. Nginx

Nginx adalah sebuah *server* HTTP dan *reverse proxy* bebas berbasis *open-source* berkemampuan tinggi, juga dapat digunakan sebagai *server proxy* IMAP/POP3. Perangkat lunak ini diciptakan oleh Igor Sysoev pada tahun 2002, dan dirilis untuk pertama kalinya secara umum pada 2004. Saat ini Nginx terkenal karena performanya yang tinggi, stabil, memiliki banyak fitur, mudah dikonfigurasi, dan menggunakan sedikit sumber daya pada *server*. Nginx adalah salah satu dari sebagian kecil perangkat lunak untuk *server* yang diciptakan untuk mengatasi masalah C10K. tidak

seperti perangkat lunak server yang umum lainnya, Nginx menggunakan arsitektur *asynkronus* yang lebih stabil. Arsitektur ini membutuhkan lebih sedikit memori, dan yang lebih penting, dapat diperkirakan.

Bahkan jika pengguna tidak mengharapkan server untuk mengatasi ribuan koneksi pada saat yang bersamaan, pengguna masih dapat diuntungkan dengan pemakaian memori yang sedikit namun berkemampuan tinggi. Nginx dapat digunakan dalam semua skala mulai dari VPS kecil hingga *cluster server* dalam jumlah besar.

#### 2.4.5. HTML

Sebuah bahasa markah untuk membuat halaman *web* dan bahasa yang digunakannya masih sangat standar seperti salah satu fungsinya untuk membuat tabel, menambahkan objek suara, video dan animasi adalah pengertian dari HTML (Hidayatullah dan kawistara, 2017:15).

Menurut Sibero (2013:19), "*Hypertext Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen sebagai bahasa untuk pertukaran dokumen *web*."

Pengertian di atas dapat disimpulkan bahwa HTML adalah sebuah dokumen yang berisikan tag, beberapa elemen dan atribut untuk menampilkan halaman pada *web browser*.

#### 2.4.6. Pengertian Bootstrap

*Bootstrap* sebuah alat bantu untuk membuat sebuah tampilan halaman website yang dapat mempercepat pekerjaan seorang pengembang website ataupun pendesain halaman website. Sesuai namanya, website yang dibuat dengan alat bantu ini memiliki tampilan halaman yang sama / mirip dengan tampilan halaman Twitter atau desainer juga dapat mengubah tampilan halaman sesuai dengan ketentuan. Tampilan web yang dibuat bootstrap akan menyesuaikan ukuran layar *browser* yang kita gunakan baik di

desktop, tablet ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun dinon-aktifkan sesuai dengan keinginan kita sendiri. Sehingga, kita bisa membuat web untuk tampilan desktop saja dan apabila dirender oleh mobile browser maka tampilan dari web yang kita buat bisa beradaptasi sesuai layar. Dengan bootstrap kita juga bisa membangun web dinamis ataupun statis. (Wahyu Widiyantoro, *Buku Panduan Bootstrap*, 2014 hal:6)

*Bootstrap* merupakan sebuah framework css yang memudahkan pengembang untuk membangun website yang menarik dan responsive. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk membangun dan memeliharanya. *Bootstrap* adalah css tetapi dibentuk dengan LESS, sebuah pre-processor yang member fleksibilitas dari css biasa. *Bootstrap* memberikan solusi dan seragam terhadap solusi yang umum, tugas *interface* yang setaip pengembang hadapi. Bootstrap dapat dikembangkan dengan tambahan lainnya karena ini cukup *flexible* terhadap pekerjaan design butuhkan. (Sopian Handianto, *Membangun Responsive Website dengan Twitter Bootstrap 2.0+PHP dan MySQL*, 2014 hal:8)

Twitter *Bootstrap* adalah sebuah alat bantu atau framework atau bisa dibilang kerangka untuk membuat sebuah tampilan halaman website yang dapat mempercepat pekerjaan seorang pengembang website ataupun pendesain halaman website karena tidak dibutuhkan *coding HTML, CSS*, maupun JavaScript terlalu banyak. Sesuai namanya, website yang dibuat dengan *bootstrap* memiliki tampilan halaman yang mirip dengan tampilan halaman Twitter. Twitter *Bootstrap* dibangun dengan teknologi HTML dan CSS yang dapat membuat layout halaman website, tabel, tombol, form, navigasi, dan komponen lainnya dalam sebuah website hanya dengan memanggil fungsi CSS (*class*) dalam berkas HTML yang telah didefinisikan. Selain itu juga terdapat komponen-komponen lainnya yang dibangun menggunakan JavaScript. (Sopian Handianto,

*membangun responsive website dengan Twitter Bootstrap 2.0+PHP dan MySQL, 2014 hal:8).*

#### **a. Sejarah Bootstrap**

Istilah *bootstrap* berasal sejak awal abad ke-19 Amerika Serikat (khususnya dalam kalimat “menarik diri atas pagar atas *bootstrap* seseorang”) sepasang sepatu bot dengan satu *bootstrap* terlihat sepatu bot tinggi mungkin memiliki tab, lingkaran atau menangan di bagian atas yang dikenal sebagai *bootstrap*, yang memungkinkan seseorang untuk menggunakan jari atau alat *booting* kait untuk membantu menariknya. *Bootstrap* sudah digunakan selama abad ke-19” sebagai contoh tugas yang mustahil. Pada tahun 1834, ketika muncul di advokat workingman. Pada tahun 1860 muncul dalam komentar filsafat omnieta *physical* yaitu upaya pemikiran yang menganalisis sendiri. *Bootstrap* sebagai metafora yang berarti yang berarti memperbaiki diri dengan upaya tanpa bantuan sendiri. Dan telah digunakan semenjak tahun 1922. *Bootstrap* istilah komputer mulai sebagai metafora pada 1950-an. Dalam komputer, menekan tombol *bootstrap* menyebabkan program tertanam untuk membaca program *bootstrap* dari unit input. komputer kemudian akan mengeksekusi program *bootstrap*, yang menyebabkan untuk membaca instruksi program yang lebih mendalam. (Wahyu Widyanoro, *Buku Panduan Bootstrap, 2014 hal: 8*).

#### **b. Pengembangan Bootstrap Dengan Perangkat Lunak**

Bootstrap juga dapat merujuk kepada pengembangan berturut-turut lebih kompleks, lingkungan pemrograman lebih cepat. Lingkungan yang paling sederhana mungkin *editor* teks yang sangat dasar (misalnya *red*) dan program *assembler*. Menggunakan alat ini, seseorang dapat menulis *editor* teks yang lebih kompleks dan *compiler*.

Sederhana untuk bahasa tingkat tinggi dan seterusnya sampai seseorang memiliki grafis IDE dan bahasa pemrograman tingkat tinggi yang sangat kuat. Secara historis, *bootstrap* juga mengacu pada teknik awal untuk pengembangan program komputer pada *hardware* baru. *Bootstrap* dalam pengembangan program dimulai pada 1950-an ketika setiap program dibangun diatas kertas dalam kode desimal atau dalam kode biner. Sedikit demi sedikit (1 0), karena tidak ada tingkat tinggi bahasa komputer, tidak ada *compiler*, *assembler* tidak ada, dan tidak ada *linker*. Sebuah program *assembler* kecil adalah tangan kode untuk komputer baru (misalnya IBM 650 ) yang di korvensi beberapa instruksi ke dalam lode biner atau desimal. *Compiler*, *linker*, *loader*, dan utilitas kemudian di kodekan dalam bahasa *assembly*, lanjut melanjut proses *bootstrap* pengembangan sistem *software* yang kompleks dengan menggunakan *software* sederhana. Istilah ini juga diperjuangkan oleh *Doug Engel bart* untuk merujuk keyakinannya bahwa organisasi bisa lebih baik berkembang dengan meningkatkan proses yang mereka gunakan untuk perbaikan ( sehingga mendapatkan efek peracikan dari waktu ke waktu ) tim SRI-nya yang mengembangkan *system hypertext* NLS meerapkan strategi ini dengan menggunakan alat yang telah dikembangkan.

#### 2.4.7. Datatable

Menurut Purbadian (2016:29) menerangkan demikian Menurut penulis *data tables* itu adalah sebuah plugin jQuery yang sangat *powerful*, kenapa demikian? Di samping mudah untuk mengimplementasikannya, *plugin jQuery data tables* sudah terintegrasi dengan *search form*, *page initation*, serta dengan jumlah data yang ditampilkan. Dengan adanya fitur-fitur itu tentu saja kita dapat menghemat waktu dalam membangun sebuah aplikasi *web*.



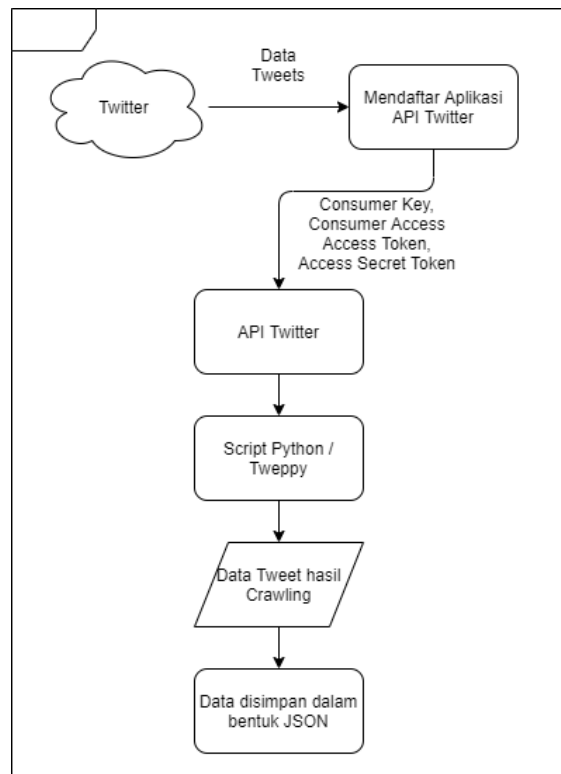
## BAB III

### PEMBAHASAN

#### 3.1. Flowchart Sistem Yang Berjalan

Sebelumnya terdapat beberapa metode crawling data dari twitter yang sering digunakan, metode tersebut antara lain :

##### 3.1.1. Crawling menggunakan *script* Pyhon dan *library* *tweepy*

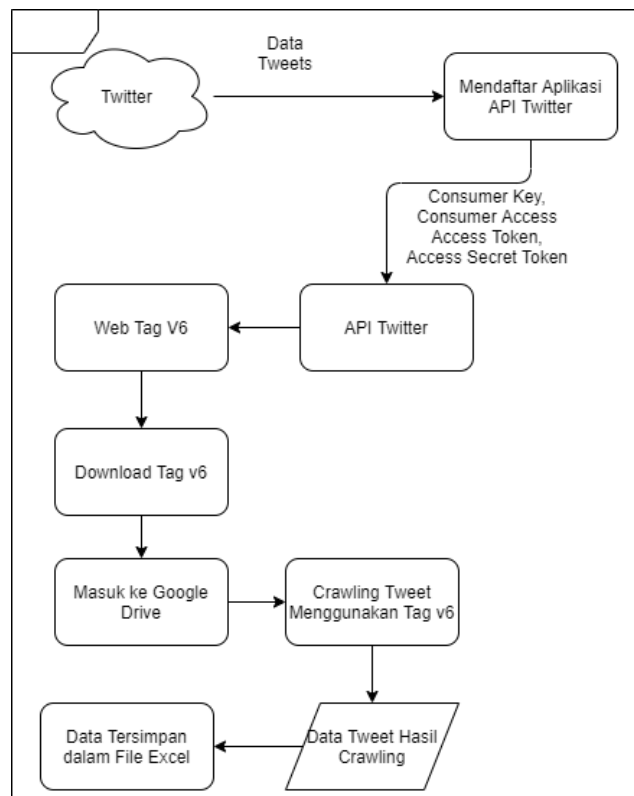


**Gambar 3.1 Flowchart Crawling menggunakan Script Python**

Crawling data twitter menggunakan *script* Python seperti *flowchart* diatas tidaklah mudah, karena harus memiliki akun *developer* twitter dan computer client harus sudah terpasang Python dan library pendukung lainnya. Belum lagi file hasil crawling yang masih berbentuk json yang masih memiliki data-data yang tidak diperlukan untuk proses sentiment analisis. Metode ini tentunya belum efektif dan efisien dari segi waktu.

### 3.1.2. Crawling menggunakan TAGS V6

TAGS v6 merupakan salah satu metode yang dapat digunakan untuk crawling data tweet dari server twitter. TAGS ini berupa file excel yang bisa didapat pada laman <https://tags.hawksey.info/get-tags/>. Tidak seperti cara sebelumnya, menggunakan TAGS V6 ini tidak perlu mengerti tentang bahas pemrograman Python. Berikut *flowchart* penggunaan TAGS sebagai media untuk mengambil data tweets.

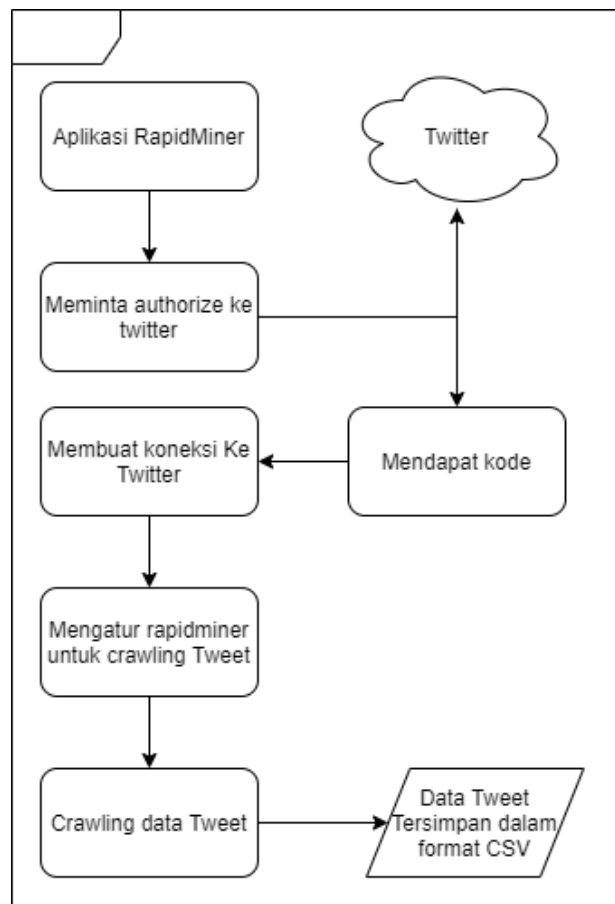


**Gambar 3.2 Flowchart Crawling menggunakan TAGS**

Bisa dilihat melalui flowchat bahwa menggunakan TAGS untuk proses Crawling data dari server twitter memiliki alur yang cukup panjang, selain harus memiliki API Twitter kita juga harus mengakses *google drive* yang nantinya sebagai media penyimpanan file excelnya sebelum di unduh ke komputer kita, dan proses crawling menggunakan TAGS ini melalui google spreadsheet.

### 3.1.3. Crawling menggunakan RapidMiner

Metode selanjutnya yang penulis ketahui untuk crawling data twitter adalah menggunakan rapidminer. Pada Rapidminer terdapat fitur *search* yang dapat digunakan untuk crawling tweet pada server twitter. Berikut flowchart proses crawling data twitter menggunakan rapidminer.

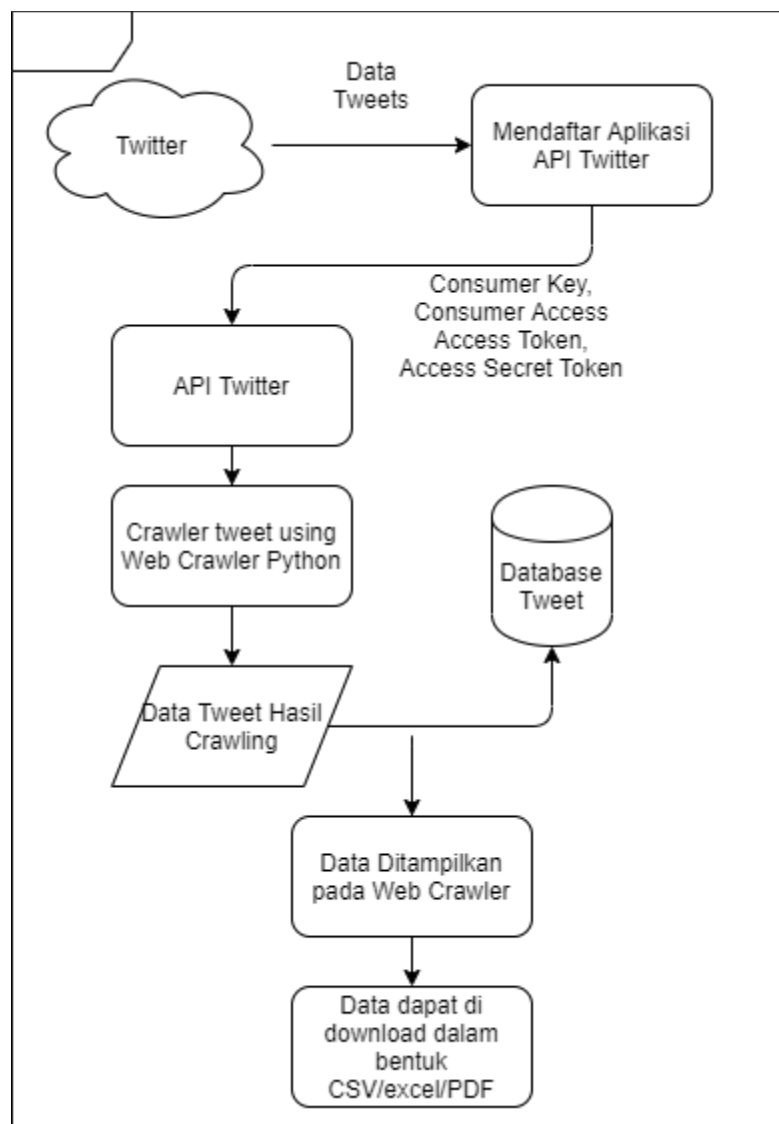


**Gambar 3.3 Flowchart Crawling menggunakan Rapidminer**

Bisa dilihat dari flowchart diatas, penggunaan rapidminer sebagai crawler data twitter juga belum efektif dari segi waktu, karena kita juga harus mendapatkan autentikasi dari twitter dan juga harus mengkonfigurasi aplikasi Rapidminer untuk melakukan crawling dan juga pastinya pada komputer kita harus pasang aplikasi *RapidMiner*.

### 3.2. Usulan Perancangan Sistem

Dari beberapa metode crawling data tweet diatas penulis mengusulkan perancangan sistem baru sebagai pemenuhan kekurangan dalam proses crawling data pada metode- metode sebelumnya. Perancangan sistem baru ini berupa webiste yang dapat digunakan untuk proses crawling data twitter secara otomatis dan data dapat diunduh ke komputer client dalam beberapa format yang dapat mempermudah tahap selanjutnya untuk proses sentimen analisis. Berikut adalah flowchat sistem yang diusulkan oleh penulis :



**Gambar 3.4 Flowchart Crawling menggunakan Web Crawling**

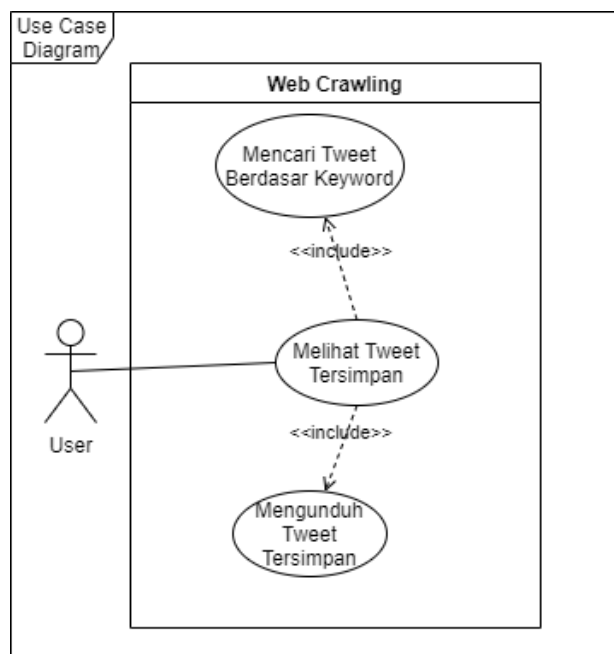
Penjelasan singkat dari flowchart sistem yang diusulkan adalah sebagai berikut :

- a. API twitter sudah diatur pada web crawler sehingga pengguna yang akan melakukan crawling tidak harus memiliki API twitter.
- b. Pengguna cukup mengisi form pencarian dan menekan tombol pencarian untuk melakukan crawling.
- c. Data tweet hasil crawling akan muncul pada halaman website.
- d. Pengguna dapat mengunduh data dari website dalam beberapa format.

Dengan diusulkannya sistem baru untuk crawling data twitter secara otomatis ini penulis mengharapkan proses crawling data twitter menjadi lebih efisien.

### 3.2.1. Use Case Diagram

*Use Case* diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antara aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan use Case diagram antaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka use case diagram untuk web crawling data tweet secara otomatis dapat di lihat pada gambar berikut :



**Gambar 3.5 Use Case Diagram Sistem Yang Diusulkan**

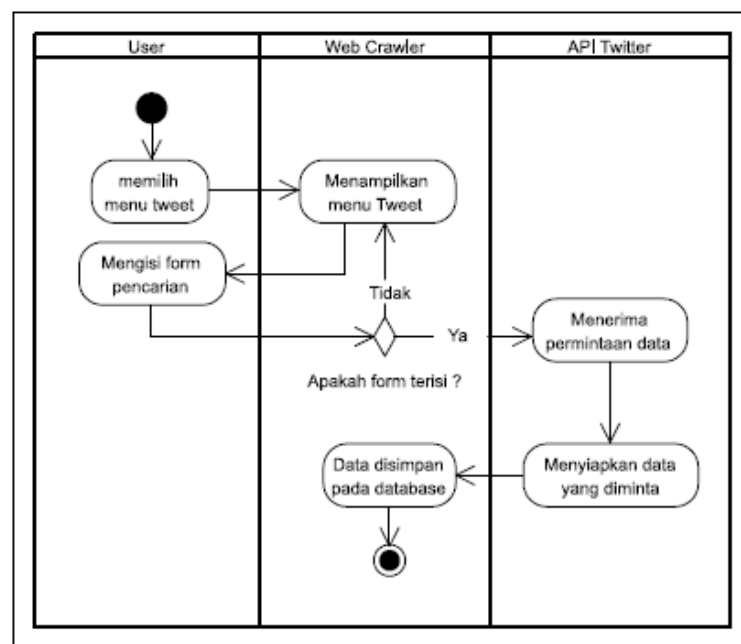
Berdasarkan gambar Use Case Diagram Web Crawling yang diusulkan terdapat :

- Satu sistem yang mencakup kegiatan crawling twitter.
- Aktor yang dapat mengakses sistem yaitu pengguna web.
- 3 (Tiga) use case yang dapat dilakukan oleh aktor yaitu : melakukan pencarian tweet, melihat data tweet hasil pencarian, mengunduh data tweet tersimpan.

### 3.2.2. Activity Diagram

Dari *Use Case* di atas didapat beberapa *Activity Diagram* yang akan penulis jelaskan sesuai dengan fungsi *Activity Diagram* yaitu untuk menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. *Activity Diagram* tersebut antara lain :

#### a. Activity Diagram Pencarian Tweet



**Gambar 3.6 Activity Diagram Pencarian Tweets**

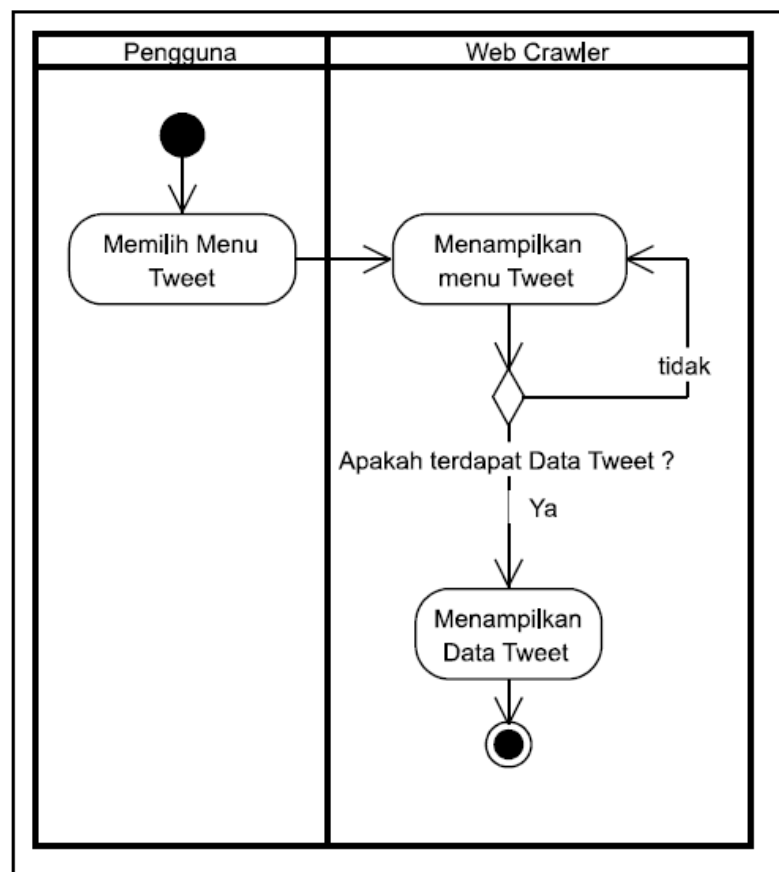
Pada *Activity Diagram* diatas dapat dijelaskan beberapa hal, antara lain :

**Tabel 3.1 Keterangan Activity Diagram Pencarian Tweets**

Nama Activity	Activity Pencarian Tweets
Aktor	Pengguna

Deskripsi Singkat	Pengguna melakukan pencarian tweets dan hasil pencarian disimpan dalam database sistem.
Pra Kondisi	Terdapat koneksi internet.
Tindakan Utama	Pengguna mengisi <i>field</i> untuk melakukan pencarian tweet dan hasil pencarian akan tersimpan pada database sistem.
Tindakan Alternatif	-
Pasca Kondisi	Data hasil pencarian tersimpan pada database.

**b. Activity Diagram Melihat Tweet Hasil Crawling**

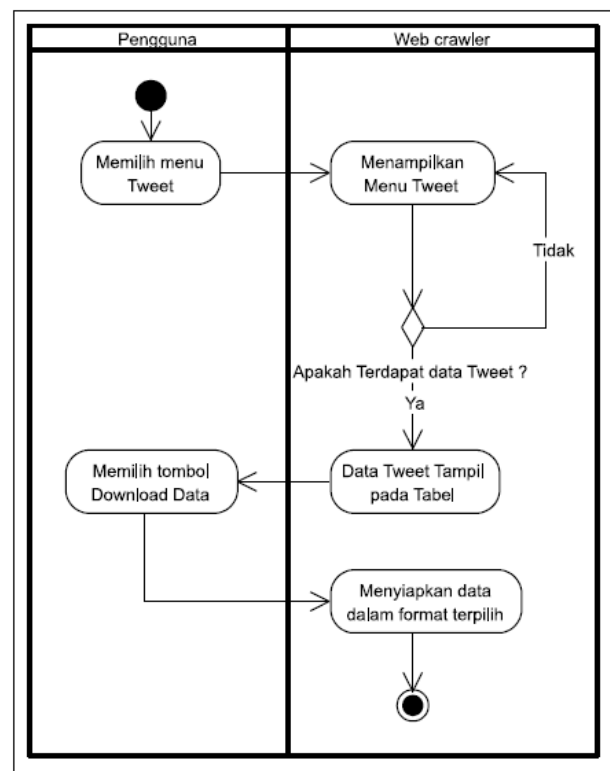


**Gambar 3.7 Activity Diagram Melihat Tweet Hasil Crawling**

**Tabel 3.2 Keterangan Activity Diagram Melihat Tweet Hasil Crawling**

Nama Activity	Activity Melihat Tweet Hasil Crawling
Aktor	Pengguna
Deskripsi Singkat	Pengguna masuk ke menu tweets dan melihat tweet yang tersimpan
Pra Kondisi	Terdapat koneksi internet
Tindakan Utama	Pengguna memilih menu tweet kemudian pengguna dapat melihat tweet yang tersimpan pada database
Tindakan Alternatif	-
Pasca Kondisi	Tweet tersimpan muncul di menu tweets

**c. Activity Diagram Download Tweet Tersimpan**



**Gambar 3.8 Activity Diagram Download Tweets**

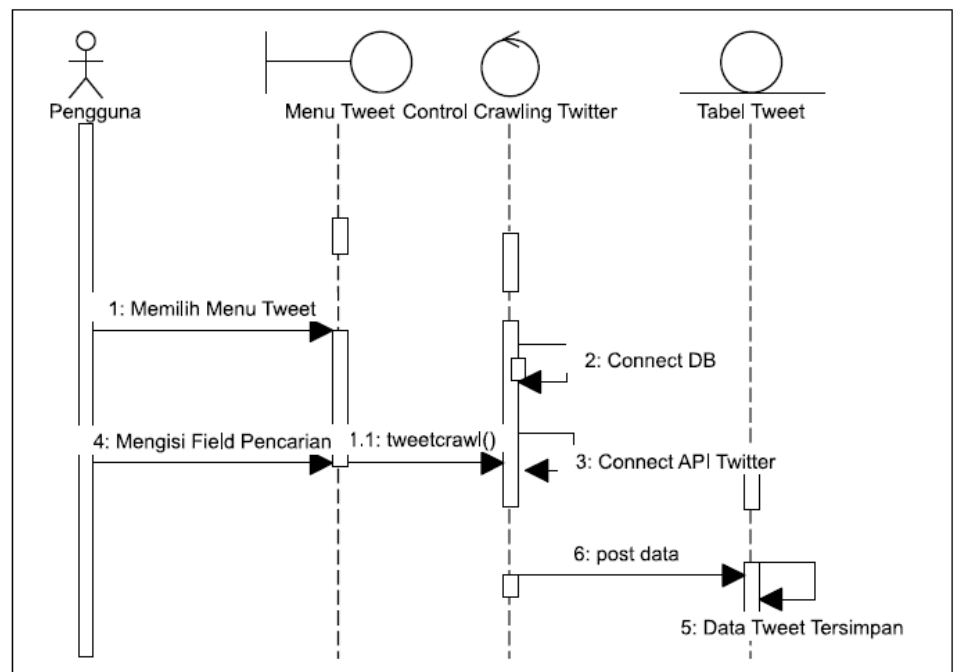


**Tabel 3.3 Keterangan Activity Diagram Download Tweets**

Nama Activity	Activity Download Tweets
Aktor	Pengguna
Deskripsi Singkat	Pengguna masuk ke menu tweets dan mendownload tweets yang ada
Pra Kondisi	Terdapat koneksi internet
Tindakan Utama	Pengguna memilih menu tweet kemudian pengguna memilih tombol download untuk menyimpan tweet yang ada
Tindakan Alternatif	-
Pasca Kondisi	Tweet berhasil di download

### 3.2.3. Sequence Diagram

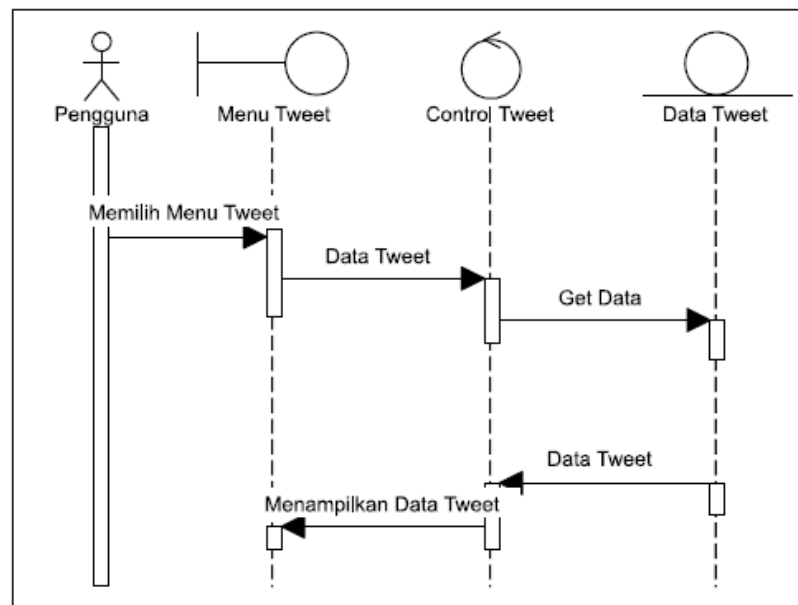
#### a. Sequence Diagram Pencarian Tweet

**Gambar 3.9 Sequence Diagram Pencarian Tweets**

**Tabel 3.4 Keterangan Sequence Diagram Pencarian Tweets**

Nama Sequence	Sequence Diagram Pencarian Tweet
Aktor	Pengguna
Deskripsi Singkat	Menjelaskan proses terjadinya pencarian tweet pada sistem
Pra Kondisi	Terdapat koneksi internet
Tindakan Utama	Pengguna memilih menu tweet kemudian mengisi form untuk pencarian tweet, di <i>background</i> aplikasi terhubung ke <i>database</i> dan meminta data ke API twitter untuk mengambil data
Tindakan Alternatif	-
Pasca Kondisi	Aplikasi terhubung dengan API dan Tweet berhasil disimpan pada <i>database</i>

**b. Sequence Diagram Melihat Tweet Hasil Crawling**

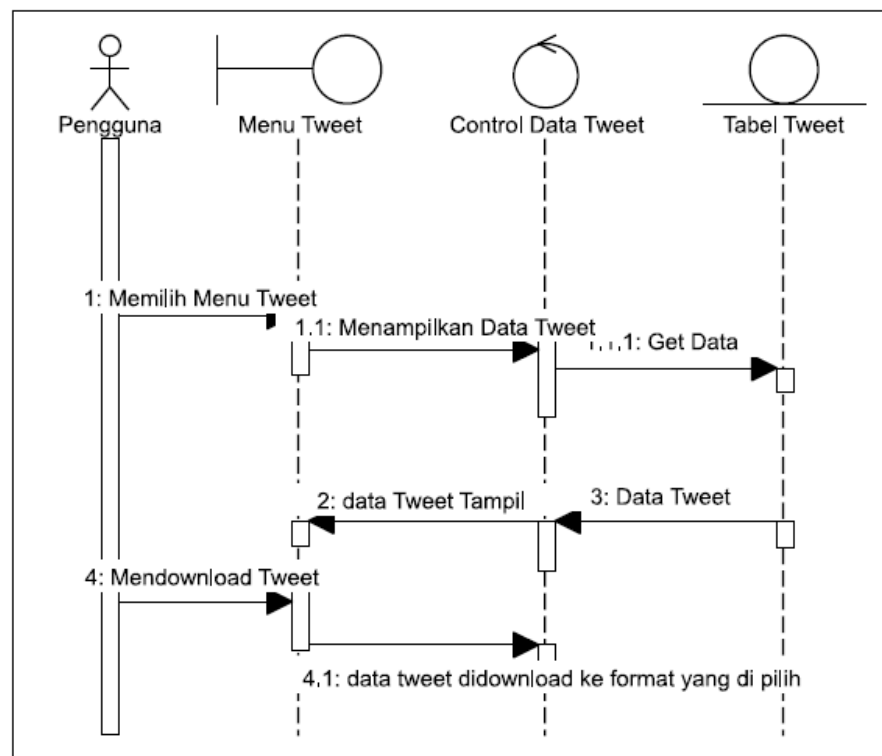


**Gambar 3.10 Sequence Diagram Melihat Tweet Hasil Crawling**

**Tabel 3.5 Keterangan Sequence Diagram Melihat Tweet Hasil Crawling**

Nama Sequence	Sequence Diagram Melihat Tweet Hasil Crawling
Aktor	Pengguna
Deskripsi Singkat	Menjelaskan proses dalam melihat tweet hasil crawling
Pra Kondisi	Terdapat koneksi internet
Tindakan Utama	Pengguna memilih menu tweet dan lihat tweet hasil crawling
Tindakan Alternatif	-
Pasca Kondisi	Tweet terimpan pada database muncul pada tabel

**c. Sequence Diagram Download Tweet**

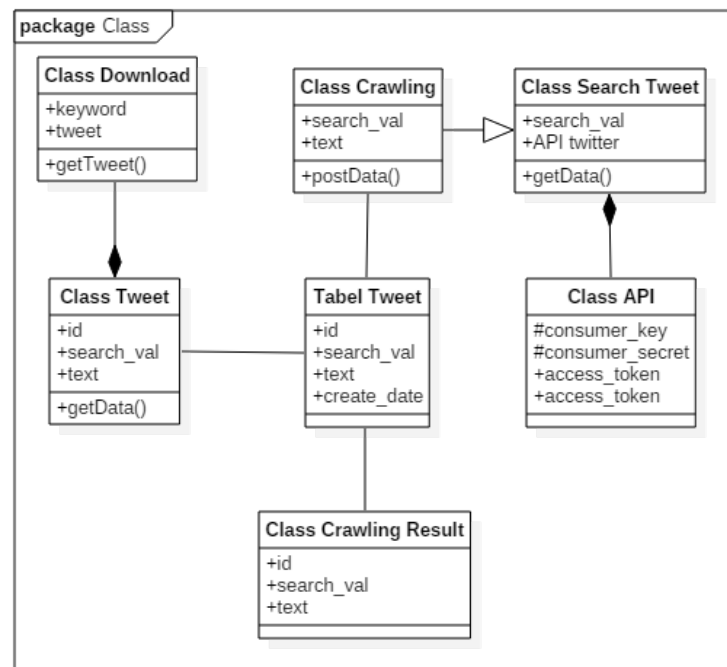


**Gambar 3.11 Sequence Diagram Download Tweets**

**Tabel 3.6 Keterangan Sequence Diagram Download Tweet**

Nama Sequence	Sequence Diagram Download Tweet
Aktor	Pengguna
Deskripsi Singkat	Menjelaskan proses dalam mendownload tweet yang terimpan pada database
Pra Kondisi	Terdapat koneksi internet
Tindakan Utama	Data tweet yang tersimpan pada database tampil pada tabel dan pengguna memilih format untuk data hasil download
Tindakan Alternatif	-
Pasca Kondisi	Tweet Berhasil disimpan sesuai format terpilih

### 3.2.4. Class Diagram

**Gambar 3.11 Class Diagram Aplikasi**

Pada class diagram aplikasi diatas dapat terlihat bahwa menu tweet dikelola oleh *classTweet* yang mengakses database dan

menu crawling tweet dikelola oleh *class crawling* yang juga terkoneksi ke *database* dan terkoneksi ke API Twitter untuk proses permintaan datannya.

### 3.3. Desain Tabel

#### 3.3.1. Tabel Tweet

Pada sistem ini terdapat satu tabel pada *database* yang digunakan untuk menyimpan semua data tweet yang dicari oleh *user*. Berikut desain tabelnya :

**Tabel 3.7 Tabel Tweet**

No	Field	Tipe	Ukuran	Key	Default
1.	id	Integer	10	Primary	Not Null
2.	search_val	text	65535	-	Not Null
3.	text	text	65535	-	Not Null
4.	create_date	Datetime	-	-	Not Null

### 3.4. Prototipe

## **BAB IV PENUTUP**

### **4.1. Kesimpulan**

Adapun kesimpulan yang dapat diambil dari penelitian crawling data otomatis dari twitter ini antara lain :

- 4.1.1. Sistem yang diusulkan dapat mempermudah dalam pengumpulan data dari twitter karena tidak perlu mendaftar sebagai developer Twitter.
- 4.1.2. Sistem yang diusulkan dapat mempercepat proses pengumpulan data dari twitter karena dilakukan secara otomatis.
- 4.1.3. Sistem yang diusulkan dapat menjadi solusi dalam kekurangan pada metode yang dilakukan untuk crawling data twitter sebelumnya.

### **4.2. Saran**

Dengan laporan berjudul Data Crawling Otomatis Pada Twitter ini, diharapkan sistem dapat diimplementasikan dan dengan adanya kekurangan yang ada semoga penelitian ini dapat dilanjutkan ke penelitian yang lebih lanjut.

## DAFTAR PUSTAKA

- [1] Purbandian, Yenda. 2016. Framework Codeigniter 3 Membangun Aplikasi Penggajian Untuk Panduan Skripsi. Cirebon: CV. ASFA Solution, Software Development, IT & Publishing.
- [2] Winda Widya Ariestya, Ida Astuti, I Made Wiryana, “*Preprocessing For Crawler Of Short Message Social Media*”, Universitas Gunadarma, Jakarta, Indonesia.
- [3] Agus Halim, Rudy Dwi Nyoto, Novi Safriadi, “*Perancangan Aplikasi Web Crawler untuk Menghasilkan Dokumen Teks pada Domain Tertentu*”, Program Studi Teknik Informatika Universitas Tanjungpura, Pontianak, Kalimantan Barat, Vol. 5, No. 2 (2017).
- [4] Intan Ferina Irza, Zulhendra, Rfrizon, “*Analisis Perbandingan Kinerja Web Server Apache dan Nginx Menggunakan Httpperf Pada Portal Berita (Studi Kasus beritalinuk.com)*”, Vol. 5, No. 2, Juli – Desember 2017, ISSN: 2302-3295.
- [5] Eri Zuliarso, Khabib Mustofa, “*Crawling Web berdasarkan Ontology*”, Jurnal Teknologi Infomasi DINAMIK Vol. XIV, No. 2, Hal. 105-112.
- [6] B. R. Aditya, “*Penggunaan Web Crawler Untuk Menghimpun Tweets dengan Metode Pre-Processing Text Mining*”, Jurnal Infotel, Vol. 7, No. 2, ISSN : 2085-3688, e-ISSN : 2460-0997, November 2015.
- [7] P. Y. Saputra, “*Implementasi Teknik Crawling Untuk Pengumpulan Data Dari Media Sosial Twitter*”, Jurnal Dinamika Dotcom, ISSN 2086-2652, Volume 8, Nomor 2, Juli 2017.
- [8] G.Shilpy, Richhariya M. , R. Vineet, “*Real Time Text Mining on Twitter Data*”, International Journal of Computer Applications (0975 – 8887) Volume 178 – No.3, November 2017.

- [9] Sanya Goel, Mudit Bansal, Atul Kumar S, Neha Arora, “Web Crawling-based Search Engin using Pyhton”, Proceedings of The Third International Conference on Electronics Comunication and Aerospace Technology [ICECA 2019], IEEE Conference Record # 45616; IEEE Xplore ISBN: 978-1-7281-0167-5
- [10] Rahadian Irsyad, “Penggunaan Pyhton Web Framework Flask Untuk Pemula”, Laboratorium Telematika, Sekolah Teknik Elektro & Informatika, Institut Teknologi Bandung
- [11] Mike Thelwall, “ A Web Crawler Design For Data Mining”, Journal of Information Science 2001 27: 319 DOI 10.1177/016555150102700503