

**PENGEMBANGAN APLIKASI USERNAME FINDER  
BERBASIS OSINT FRAMEWORK UNTUK  
PENCARIAN NAMA PENGGUNA**



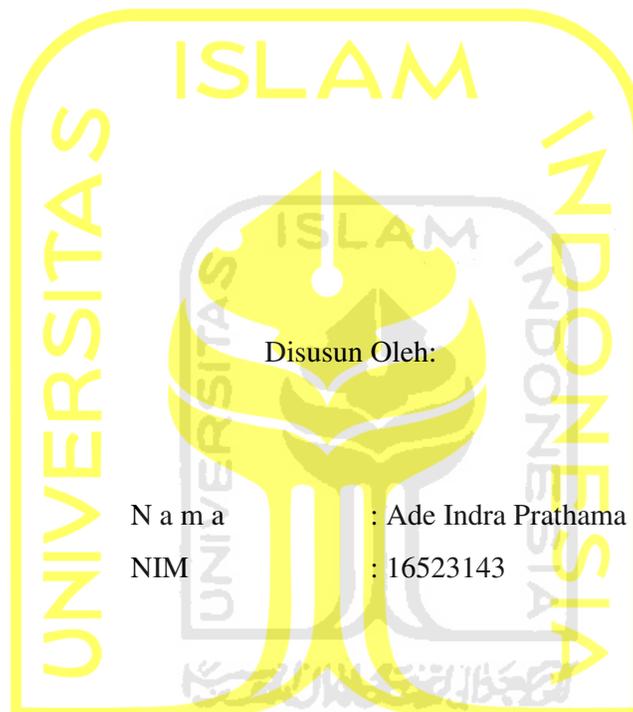
N a m a : Ade Indra Prathama  
NIM : 16523143

**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2020**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING**

**PENGEMBANGAN APLIKASI USERNAME FINDER**  
**BERBASIS OSINT FRAMEWORK UNTUK**  
**PENCARIAN NAMA PENGGUNA**

**TUGAS AKHIR**



الجمعة المباركة  
الاستاذ الاندو

Yogyakarta, 10 Juli 2020

Pembimbing,

( Fietyata Yudha, S.Kom., M.Kom. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**PENGEMBANGAN APLIKASI USERNAME FINDER**  
**BERBASIS OSINT FRAMEWORK UNTUK**  
**PENCARIAN NAMA PENGGUNA**

**TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika - Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 10 Juli 2020

Tim Penguji

Fietyata Yudha, S.Kom., M.Kom.

**Anggota 1**

Fayruz Rahma, S.T., M.Eng.

**Anggota 2**

Septia Rani, S.T., M.Cs.







Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



  
 ( Dr. Raden Teduh Dirgahayu, S.T., M.Sc. )

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ade Indra Prathama

NIM : 16523143

Tugas akhir dengan judul:

### **PENGEMBANGAN APLIKASI USERNAME FINDER BERBASIS OSINT FRAMEWORK UNTUK PENCARIAN NAMA PENGGUNA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 10 Juli 2020

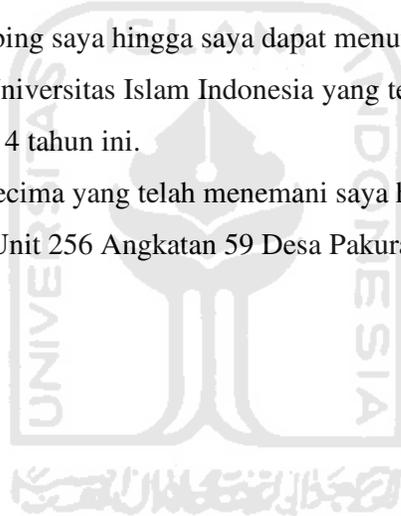


( Ade Indra Prathama )

## HALAMAN PERSEMBAHAN

Alhamdulillah rabbil 'alamin puji syukur saya panjatkan kepada Allah Subhanahu wata'ala, atas rahmat-Nya saya telah berhasil menyelesaikan tugas akhir ini sebagai syarat kelulusan di bangku perkuliahan yang telah saya jalani selama bertahun-tahun. Tugas Akhir ini saya persembahkan untuk:

1. Kedua orang tua saya Cahyono Lukman dan Uray Nur Linda yang selalu menyemangati, mendoakan, dan memberikan support untuk saya, baik selama saya menjalani perkuliahan dan mengerjakan tugas akhir ini.
2. Adik-adik saya Dea Nita Permatasari, Rio Fajar Kurniawan, dan Adrian Novan Kurniawan.
3. Bapak Fietyata Yudha, S.Kom., M.Kom. selaku dosen pembimbing, terima kasih karena telah membimbing saya hingga saya dapat menuntaskan tugas akhir ini.
4. Jurusan Informatika Universitas Islam Indonesia yang telah mewadahi saya dalam menuntut ilmu selama 4 tahun ini.
5. Teman-Teman Hexadecima yang telah menemani saya hingga saat ini.
6. Teman-Teman KKN Unit 256 Angkatan 59 Desa Pakuran.



## HALAMAN MOTO

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain, dan hanya kepada Tuhanmulah hendaknya kamu berharap”. (Q.S Al-Insyirah: 5-8)

“Mencari ilmu itu adalah wajib bagi setiap muslim laki-laki maupun muslim perempuan.”  
(Hadis Riwayat Ibnu Abdil Barr)

“Sometimes when you innovate, you make mistakes. It is best to admit them quickly and get on with improving your other innovations.” (Steve Jobs)



## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah segala puji syukur saya panjatkan kepada Allah Subhanahu wata'ala atas rahmat-Nya, saya dapat menyelesaikan tugas akhir saya yang berjudul “Pengembangan Aplikasi Open Source Intelejen Berbasis OSINT Framework untuk Pencarian Nama Pengguna”. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad Shallallahu Alaihi Wa Sallam yang telah mengantarkan manusia dari zaman kegelapan ke zaman yang terang ini. Tugas akhir ini disusun dengan tujuan memenuhi syarat untuk menyelesaikan pendidikan Strata 1 di Jurusan Informatika Universitas Islam Indonesia. Saya menyadari bahwa tugas akhir yang saya buat masih jauh dari kata sempurna dikarenakan saya masih memiliki keterbatasan pengetahuan serta pengalaman. terselesaikannya tugas akhir ini, tak lepas dari dukungan berbagai pihak yang selalu mendukung saya baik secara moril maupun materil. Oleh karena itu saya ingin mengucapkan rasa terima kasih saya kepada semua pihak yang telah membantu saya terutama kepada:

1. Allah Subhanahu wata'ala, berkat rahmat dan hidayah-Nya saya dapat menyelesaikan tugas akhir ini.
2. Kedua orang tua saya, Bapak Cahyono Lukman dan Ibu Uray Nur Linda atas dukungan dan doa hingga saat ini.
3. Bapak Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
4. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Hendrik, S.T,M.Eng, Selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
6. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., Selaku Ketua Program Studi Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Fietyata Yudha, S.Kom., M.Kom. sebagai Dosen Pembimbing yang telah banyak memberikan pengarahan, bimbingan, serta masukan kepada saya dalam menyelesaikan tugas akhir ini.
8. Dosen-dosen beserta staff Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan ilmu pengetahuan serta bantuannya hingga saat ini.

9. Semua teman-teman angkatan 2016 Hexadecima yang telah memberi dukungan kepada saya hingga saat ini.
10. Teman-teman konsentrasi penelitian JKK yang telah memberikan dukungan dan menemani perjuangan selama tahun ke-4 serta dalam menyelesaikan tugas akhir ini.
11. Dan juga pihak-pihak lain yang tidak dapat saya sebutkan satu per satu yang telah memberikan dukungan kepada saya hingga saat ini.

Saya sangat menerima kritik maupun saran dari berbagai pihak agar laporan tugas akhir ini dapat menjadi lebih baik lagi. Dan saya berharap laporan tugas akhir ini dapat bermanfaat bagi para pembaca dan semua pihak.

Yogyakarta, 10 Juli 2020



*Ade*  
( Ade Indra Prathama )

## SARI

Salah satu cara untuk melakukan proses pencarian informasi di internet adalah memanfaatkan sumber daya yang terbuka di internet. Salah satu penelitian yang mencoba memanfaatkan celah ini adalah pengembangan kerangka kerja OSINT. Proses pengembangan aplikasi berbasis OSINT Framework dengan cara mengintegrasikan 16 aplikasi pada OSINT Framework pada subbagian Username dengan tujuan menghasilkan aplikasi kompak yang dapat digunakan untuk melakukan pencarian nama pengguna pada sumber terbuka. Metode yang digunakan dalam mengintegrasikan ke-16 aplikasi OSINT Framework tersebut adalah menggunakan metode Web Scraping dan mengembangkan kode aplikasi. Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi yang bertujuan untuk memudahkan pengguna dalam mencari Username menggunakan OSINT Framework. Hasil penelitian menunjukkan bahwa, pengembangan aplikasi berbasis OSINT Framework dengan cara mengintegrasikan 16 aplikasi pada OSINT Framework pada subbagian Username dapat diimplementasikan dengan baik. Akan tetapi, terdapat 1 aplikasi kerangka kerja OSINT yang masih belum dapat diintegrasikan yaitu: Amazon Wishlists dikarenakan terdapat captcha pada aplikasi tersebut. Kesimpulan dari penelitian ini adalah, pengembangan aplikasi berbasis OSINT Framework dengan cara mengintegrasikan 16 aplikasi pada OSINT Framework pada subbagian Username hanya dapat diimplementasikan pada 15 aplikasi OSINT Framework dari total 16 aplikasi OSINT Framework, yaitu dengan metode Web Scraping dan mengembangkan kode aplikasi.

Kata Kunci : Open Source Intelligence, Nama Pengguna, Username, OSINT Framework, Kerangka Kerja OSINT.

## GLOSARIUM

Append	Menuliskan sesuatu ke dalam suatu berkas yang sudah ada tanpa menimpa isi berkas tersebut.
Attribute	Nilai atau ciri yang terdapat pada suatu object dalam pemrograman.
Bash Script	Bahasa yang berjalan di kernel linux dan unix.
Class	Cetak biru dari object dalam pemrograman.
Command Prompt	Aplikasi command line interpreter pada windows.
File log	Berkas yang dibuat untuk menyimpan log.
Graphical User Interface	Tampilan dalam suatu program yang terdapat tombol, gambar, dan ikon yang bertujuan untuk memudahkan pengguna.
Hyperlink	Menyimpan link pada suatu text sehingga dapat diklik.
Information Security	Profesi yang berkaitan dengan keamanan informasi.
Library	Kumpulan program maupun fungsi yang sudah ada yang dapat dipanggil pada program lain.
Open Source	Suatu program atau aplikasi yang source code-nya bersifat terbuka sehingga dapat diambil dan dikembangkan secara bebas.
Terminal	Aplikasi command line interpreter pada linux.
Txt	Ekstensi berkas pada komputer untuk menunjukkan bahwa berkas tersebut merupakan berkas teks biasa.
Url	Alamat dari suatu web.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR.....	vii
SARI .....	ix
GLOSARIUM .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xii
DAFTAR GAMBAR.....	xiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Langkah Penyelesaian.....	3
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1 Open Source Intelligence .....	6
2.2 OSINT Framework .....	7
2.3 Social Media Intelligence.....	9
2.4 Alat-Alat OSINT.....	9
2.5 Tantangan dalam Pengembangan Aplikasi OSINT .....	10
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>12</b>
3.1 Metode Penelitian .....	12
3.2 Web Scraping .....	12
3.3 Pengembangan Source Code Aplikasi .....	13
3.4 Diagram Alir Aplikasi.....	15
<b>BAB IV IMPLEMENTASI DAN HASIL .....</b>	<b>42</b>
4.1 Implementasi Tampilan Utama.....	42
4.2 Implementasi Web Scraping .....	46
4.2.1 Web Scraping menggunakan requests.....	47
4.2.2 Web Scraping menggunakan selenium .....	47
4.3 Ekstraksi Data dari hasil Web Scraping.....	48
4.4 Implementasi Pengembangan Source Code Aplikasi .....	48
4.4.1 Namechk.sh .....	48
4.4.2 WhatsMyName.....	49
4.5 Menampilkan Data hasil Ekstraksi .....	50
4.6 Log hasil Pencarian .....	53
4.7 Perbandingan Aplikasi Username Finder dengan Maltego.....	56
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>58</b>
5.1 Kesimpulan .....	58
5.2 Saran.....	58
DAFTAR PUSTAKA.....	59
LAMPIRAN .....	61

**DAFTAR TABEL**

Tabel 2.1 Arti Khusus pada huruf yang terdapat pada OSINT Framework .....	7
Tabel 2.2 Nama dan Fungsi tools kerangka kerja OSINT subbagian username .....	8
Tabel 2.3 Jenis-Jenis Sosial Media .....	9
Tabel 2.4 Contoh alat-alat OSINT .....	9
Tabel 4.1 Perbandingan Aplikasi Username Finder dengan Maltego .....	57



## DAFTAR GAMBAR

Gambar 1.1 Jumlah pengguna sosial media di Dunia dari tahun 2010 sampai tahun 2023 .....	2
Gambar 2.1 Ilustrasi Open Source Intelligence (OSINT).....	6
Gambar 3.1 Diagram Alir Penelitian. ....	12
Gambar 3.2 Ilustrasi Cara Kerja Web Scraping.....	13
Gambar 3.3 Hasil Pencarian Nama Pengguna menggunakan WhatsMyName. ....	14
Gambar 3.4 Hasil Pencarian Nama Pengguna menggunakan Namechk.sh.....	14
Gambar 3.5 Diagram Alir Aplikasi Username Finder secara keseluruhan.....	15
Gambar 3.6 Diagram Alir fungsi namechk.py.....	17
Gambar 3.7 Diagram Alir fungsi namechk2.py.....	18
Gambar 3.8 Diagram Alir fungsi knowem.py. ....	20
Gambar 3.9 Diagram Alir fungsi namecheckr.py.....	21
Gambar 3.10 Diagram Alir fungsi usersearch.py. ....	23
Gambar 3.11 Diagram Alir fungsi thatsthem.py.....	25
Gambar 3.12 Diagram Alir fungsi checkusernames.py.....	27
Gambar 3.13 Diagram Alir fungsi namecheckup.py. ....	29
Gambar 3.14 Diagram Alir fungsi instantusernamesearch.py. ....	30
Gambar 3.15 Diagram Alir fungsi amazonusernames.py.....	32
Gambar 3.16 Diagram Alir fungsi githubuser.py. ....	34
Gambar 3.17 Diagram Alir fungsi tinderusernames.py. ....	36
Gambar 3.18 Diagram Alir fungsi keybase.py. ....	37
Gambar 3.19 Diagram Alir fungsi mitpgpkeyserver.py. ....	39
Gambar 3.20 Diagram Alir fungsi whatsmyname.py.....	41
Gambar 4.1 Tampilan utama aplikasi Username Finder .....	43
Gambar 4.2 Kode program tampilan utama aplikasi Username Finder.....	43
Gambar 4.3 Kode program pengambilan halaman HTML menggunakan library requests.....	47
Gambar 4.4 Kode program pengambilan halaman HTML menggunakan library selenium. ..	47
Gambar 4.5 Kode program pegekstraksian data dari hasil web scraping .....	48
Gambar 4.6 Pengembangan kode program aplikasi Namechk.sh.....	48
Gambar 4.7 Fungsi untuk memanggil aplikasi Namechk.sh. ....	49
Gambar 4.8 Mengubah aplikasi WhatsMyName menjadi sebuah class.....	49
Gambar 4.9 Kode program untuk memanggil aplikasi WhatsMyName.....	50
Gambar 4.10 Contoh tampilan hasil pencarian nama pengguna.....	51

Gambar 4.11 Kode program untuk menampilkan data ke tampilan GUI. ....	51
Gambar 4.12 Tampilan hasil pencarian nama pengguna melalui Namechk.sh. ....	52
Gambar 4.13 Tampilan hasil pencarian nama pengguna melalui WhatsMyName.....	52
Gambar 4.14 Tampilan hasil pencarian nama pengguna secara keseluruhan.....	53
Gambar 4.15 Kode program untuk membuat file log berdasarkan tanggal dan waktu. ....	53
Gambar 4.16 Contoh kode program yang berfungsi untuk menyimpan ke file log. ....	54
Gambar 4.17 Isi dari file log hasil pencarian nama pengguna: Donald.....	55
Gambar 4.18 Tampilan Aplikasi Maltego pada Kali Linux. ....	57



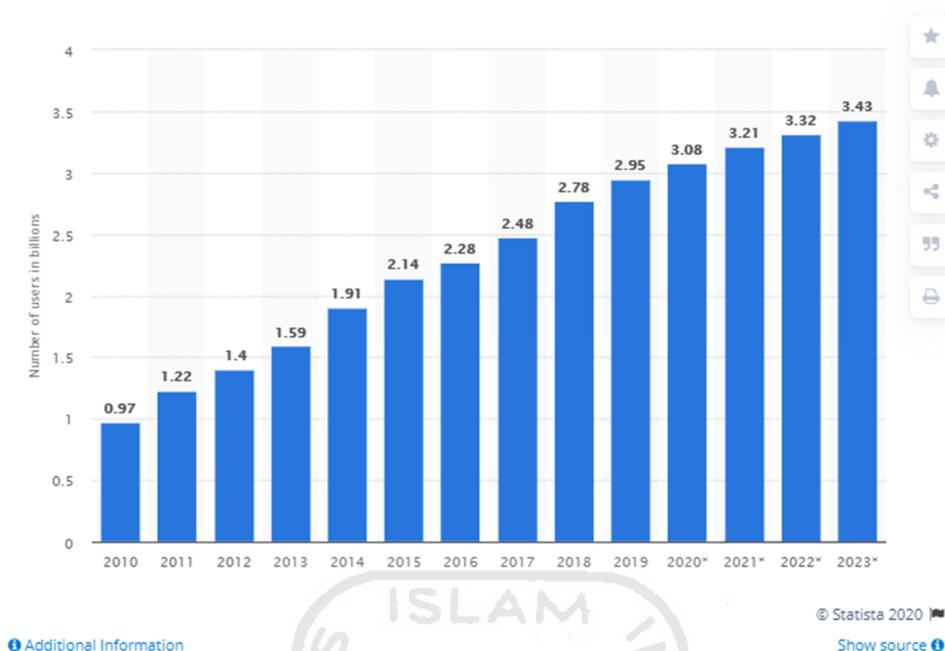
# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Di era globalisasi ini, internet telah menjadi kebutuhan sehari-hari. Hampir setiap orang membutuhkan internet di dalam kehidupan sehari-harinya. Dengan adanya sebuah internet maka pertukaran data dan informasi terjadi dengan sangat cepat. Pengguna internet di Indonesia sendiri pada tahun 2019 sudah sebanyak 171,17 juta jiwa atau sekitar 64,8 persen dari 264 juta jiwa. Jumlah pengguna internet pada tahun 2019 di Indonesia mengalami peningkatan dibandingkan pada tahun 2017 yang hanya sebanyak 54,86 persen (Pratomo, 2019).

Dengan meningkatnya jumlah pengguna internet, maka jumlah pengguna sosial media juga akan meningkat. Sosial media adalah sebuah alat komunikasi berbasis web yang dapat membuat satu orang dengan orang yang lainnya berinteraksi dan berbagi informasi. Platform-Platform sosial media biasanya memiliki fitur-fitur seperti: membuat akun pribadi, memiliki halaman profil, menambah maupun menghapus pertemanan dengan pengguna lain, dan masih banyak fitur lainnya yang mungkin satu sosial media dengan sosial media yang lainnya memiliki beragam fitur yang berbeda (Nation, 2019). Total pengguna sosial media di Indonesia yang dihitung pada Januari 2019 tercatat sebanyak 150 juta pengguna, jumlah ini mencapai 56 persen dari total penduduk Indonesia, dengan pengguna sosial media berbasis mobile yang mencapai 130 juta (“Indonesia Digital 2019 : Media Sosial,” 2019). Sedangkan total pengguna sosial media di dunia pada tahun 2019 sebanyak 2,95 miliar pengguna dan diperkirakan akan terus bertambah setiap tahunnya. Data tersebut bisa dilihat pada Gambar 1.1.



Gambar 1.1 Jumlah pengguna sosial media di Dunia dari tahun 2010 sampai tahun 2023

Sumber: (Clement, 2020)

Semakin berkembangnya internet dan sosial media, maka akan semakin banyak data-data yang tersebar di internet. Data-data yang tersebar baik di internet maupun sosial media mungkin saja berupa data publik atau data terbuka yang semua orang bisa mendapatkan data tersebut. Dengan open source intelligence (OSINT), data-data yang bersifat publik atau terbuka dapat dicari. Cara kerja open source intelligence (OSINT) adalah mengumpulkan data melalui sumber yang dapat diakses secara publik seperti jejaring sosial, media, blog dan komunitas web (Pellet, Shiaeles, & Stavrou, 2019). OSINT melibatkan pengumpulan, analisis, dan penggunaan data dari sumber terbuka untuk tujuan intelijen (Koops, Hoepman, & Leenes, 2013).

Mengembangkan OSINT adalah suatu tantangan, karena volume yang bertambah dan keanekaragaman sumber yang luar biasa. OSINT dalam bentuk digital yaitu teks dan multimedia yang dapat diakses untuk dianalisis dengan tujuan menghasilkan laporan intelijen dan estimasi intelijen. Data pada web sering memiliki format yang buruk dengan alasan terlambatnya standardisasi spesifikasi HTML dan fleksibilitas browser yang ekstrim. Bahkan kontennya sendiri tidak terdapat pengukuran akurasi ataupun kualitas (Pouchard, Dobson, & Trien, 2006). Oleh karena itu, pada penelitian ini saya akan mengembangkan Aplikasi OSINT

yang dapat melakukan pencarian nama pengguna dari sumber terbuka dengan memanfaatkan OSINT Framework pada subbab Username.

## 1.2 Rumusan Masalah

Rumusan masalah berikut akan dijawab oleh penelitian ini:

- a. Bagaimana mengintegrasikan 16 perangkat pada OSINT menjadi 1 aplikasi?
- b. Bagaimana mengembangkan aplikasi open source intelligence berbasis OSINT Framework untuk pencarian nama pengguna?

## 1.3 Batasan Masalah

Untuk menjaga fokus penelitian dalam TA ini, beberapa batasan berikut diperhatikan:

1. Aplikasi open source intelligence berbasis OSINT Framework yang akan dikembangkan hanya terfokus pada subbagian Username saja.
2. Informasi yang diekstrak oleh aplikasi ini hanya sebatas nama pengguna saja.

## 1.4 Tujuan Penelitian

Berdasarkan latar belakang masalah dan tujuan untuk mengembangkan aplikasi open source intelligence, pada Tugas Akhir (TA) ini saya akan mengembangkan aplikasi open source intelligence yang dapat mencari nama pengguna dari sumber terbuka dengan memanfaatkan OSINT Framework pada subbagian *Username*.

Kontribusi yang diharapkan dari TA ini adalah:

1. Mengintegrasikan 16 perangkat pada OSINT menjadi 1 aplikasi dengan menggunakan metode Web Scraping dan Pengembangan Source Code Aplikasi.
2. Membuat aplikasi open source intelligence yang dapat mencari dan mendapatkan nama pengguna dari sumber terbuka.

## 1.5 Langkah Penyelesaian

Untuk menjawab rumusan masalah di atas, ada beberapa tahapan yang akan saya lakukan. Tahapan-tahapan tersebut adalah:

1. Studi Literatur

Langkah pertama dalam menjawab pertanyaan ini adalah dengan mengkaji literatur-literatur mengenai apa itu open source intelligence dan cara mengembangkan sebuah aplikasi atau alat open source intelligence.

## 2. Menghubungkan aplikasi yang dibuat dengan aplikasi OSINT Framework

Aplikasi-aplikasi pada OSINT Framework subbagian Username akan dihubungkan ke dalam aplikasi yang akan dikembangkan, terdapat beberapa metode untuk menghubungkan aplikasi yang akan dikembangkan ke aplikasi OSINT Framework yaitu:

### 1. Menggunakan API

Salah satu metode untuk menghubungkan aplikasi OSINT Framework ke dalam aplikasi yang dibuat adalah dengan menggunakan API. Salah satu contoh aplikasi OSINT Framework yang menyediakan fitur ini adalah <https://namechk.com/>.

### 2. Menggunakan Web Scraping

Ada beberapa aplikasi OSINT Framework yang tidak menyediakan fitur API, maka untuk menghubungkannya perlu menggunakan metode web scraping. Web scraping adalah teknik pengambilan dokumen semi-terstruktur dari internet, secara umum halaman-halaman web dibuat menggunakan bahasa HTML atau XHTML, dan menganalisis dokumen tersebut untuk diambil data-data tertentu dari halaman tersebut. Atau sering disebut teknik mengambil konten atau sebagian data dari situs web (Syabab, 2019).

### 3. Mengembangkan kode aplikasi

Terdapat 2 aplikasi OSINT Framework subbagian username yang bersifat *open source*, yaitu `namechk.sh` yang dikembangkan menggunakan bahasa shell linux (.sh) dan `WhatsMyName` yang dikembangkan menggunakan bahasa python. *Source code* kedua aplikasi tersebut tersedia pada github dan dapat dikembangkan. Untuk menghubungkannya, dapat dilakukan dengan mengembangkan atau memodifikasi *source code* aplikasi tersebut sesuai kebutuhan.

### 4. Menggunakan URL web

Aplikasi OSINT Framework seperti Amazon Usernames, Github User, dan Tinder Usernames hanya menggunakan URL web untuk melakukan pencarian. Metode untuk menghubungkannya cukup mudah, yaitu dengan memberikan URL web yang sama dengan URL web yang digunakan pada ketiga aplikasi tersebut.

3. Mengintegrasikan 16 aplikasi yang sudah dihubungkan dengan aplikasi OSINT Framework menjadi 1 aplikasi

Setelah menghubungkan aplikasi-aplikasi tersebut, aplikasi-aplikasi tersebut akan diintegrasikan menjadi 1 aplikasi. Hal ini bertujuan agar proses pencarian username hanya memerlukan 1 kali input saja dan pencarian akan berjalan pada keenam belas aplikasi yang sudah diintegrasikan menjadi 1 aplikasi.

4. Pengujian aplikasi

Pada tahapan ini, aplikasi akan diuji dengan berbagai macam data *username* yang nantinya akan dicari menggunakan aplikasi ini. Tahapan ini berguna untuk mengetahui hasil yang didapatkan dari pengintegrasian keenam belas aplikasi OSINT Framework subbagian username.



## BAB II LANDASAN TEORI

### 2.1 Open Source Intelligence

Open Source Intelligence (OSINT) adalah sebuah intelijen yang dikumpulkan dari sumber-sumber yang terbuka secara publik. OSINT sendiri terdiri dari berbagai sumber publik, seperti (Chauhan & Panda, 2015a):

1. Publikasi Akademik: makalah penelitian, publikasi konferensi, dll.
2. Sumber Media: koran, saluran radio, televisi, dll.
3. Konten Web: situs web, media sosial, dll.
4. Data Publik: dokumen pemerintahan yang terbuka, pengumuman perusahaan publik, dll.



Gambar 2.1 Ilustrasi Open Source Intelligence (OSINT)

Sumber: (David, 2017)

Bentuk paling awal dari Open Source Intelligence melibatkan pencarian dan kombinasi yang kreatif dari informasi yang tersedia secara bebas dan mensintesiskan temuan ke dalam bentuk yang dapat ditindaklanjuti. OSINT memiliki tujuan ganda, mengumpulkan secara

kreatif, mengkodekan secara selektif, dan menggabungkan informasi yang transparan untuk mengembangkan solusi pada masalah baru (Glassman & Kang, 2012).

Akan tetapi data yang terdapat pada sumber terbuka tidak sepenuhnya bersifat publik. Pada beberapa kasus terdapat data yang dapat dikatakan terbuka akan tetapi tidak sepenuhnya terbuka. Menurut Stevan Dedijer terdapat 4 jenis informasi (Hribar, Podbregar, & Ivanuša, 2014):

1. White Information (Public Information)
2. Grey Information
3. Black Information (Classified Information)
4. Non-existing Information

## 2.2 OSINT Framework

OSINT Framework atau yang dapat disebut juga sebagai kerangka kerja OSINT adalah sebuah kumpulan tools OSINT yang dapat memudahkan dalam mengumpulkan data (SecurityTrails team, 2019). Kerangka kerja OSINT dikembangkan oleh Justin Nordine dan dapat diakses secara publik melalui website : <https://osintframework.com/>. Terdapat 32 subbagian pada kerangka kerja OSINT, masing-masing subbagian terdapat tools OSINT yang dapat digunakan untuk mencari informasi dari sumber terbuka. Pada beberapa tools, terdapat sebuah huruf yang memiliki arti khusus yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Arti Khusus pada huruf yang terdapat pada OSINT Framework

No.	Huruf	Arti
1.	T	Mengindikasikan bahwa tools harus di- <i>install</i> dan dijalankan secara lokal.
2.	D	Mengindikasikan tools menggunakan metode Google Dorking.
3.	R	Mengindikasikan tools memerlukan registrasi.
4.	M	Mengindikasikan tools menggunakan URL untuk pencariannya, dan untuk melakukan pencarian URL harus diedit secara manual.

Pada kerangka kerja OSINT subbagian *username*, terdapat 2 subbagian yaitu: Username Search Engines dan Specific Sites. Bagian “Username Search Engines” berisikan 10 tools yang berfungsi untuk mencari nama pengguna dari banyak platform, sedangkan bagian “Specific Sites” berisikan 6 tools yang berfungsi untuk mencari nama pengguna dari platform tertentu. Total dari keseluruhan tools pada kerangka kerja OSINT subbagian *username* sebanyak 16 tools. Tools tersebut dijabarkan pada Tabel 2.2.

Tabel 2.2 Nama dan Fungsi tools kerangka kerja OSINT subbagian username

No.	Nama Tools	Fungsi Tools
-----	------------	--------------

1.	Namechk	Mengetahui nama pengguna yang dicari tersedia atau tidak pada beberapa platform sosial media dan ekstensi nama domain.
2.	Namechk (T)	Mencari ketersediaan nama pengguna atau nama pengguna yang sudah terdaftar pada beberapa platform sosial media.
3.	KnowEm	Mengetahui nama pengguna yang dicari tersedia atau tidak pada beberapa platform sosial media dan ekstensi nama domain.
4.	NameCheckr	Mengetahui nama pengguna yang dicari tersedia atau tidak pada beberapa platform sosial media dan ekstensi nama domain.
5.	Usersearch.org	Mencari nama pengguna pada sosial media, situs kencan, dan forum. Usersearch.org juga dapat digunakan untuk mencari nama pengguna melalui nomor telepon dan alamat e-mail yang terdaftar pada suatu platform sosial media.
6.	WhatsMyName (T)	Mencari nama pengguna pada beberapa platform sosial media.
7.	Thats Them	Mencari informasi berdasarkan nama lengkap seseorang.
8.	Check Usernames	Mengetahui nama pengguna yang dicari tersedia atau tidak pada beberapa platform sosial media.
9.	NameCheckup	Mengetahui nama pengguna yang dicari tersedia atau tidak pada beberapa platform sosial media dan ekstensi nama domain.
10.	Instant Username Search	Mengetahui nama pengguna yang dicari tersedia atau tidak pada beberapa platform sosial media.
11.	Amazon Usernames (M)	Mencari konten pada amazon yang mengandung nama pengguna yang dicari.
12.	Amazon Wishlists	Mengetahui <i>wishlists</i> (keinginan) dari nama pengguna amazon yang dicari.
13.	Github User (M)	Mencari nama pengguna yang terdaftar pada github.
14.	Tinder Usernames (M)	Mencari nama pengguna yang terdaftar pada tinder.

15.	Keybase	Mencari nama pengguna yang terdaftar pada keybase.
16.	MIT PGP Key Server	Digunakan untuk mengekstrak, submit, dan menghapus keyserver.

### 2.3 Social Media Intelligence

Social Media Intelligence adalah sebuah intelijen yang dikumpulkan dari sosial media. Terdapat 2 jenis sosial media, yang pertama dapat diakses tanpa autentikasi dan yang kedua membutuhkan autentikasi agar dapat mengakses informasi (Chauhan & Panda, 2015a). Sosial media dikelompokkan menjadi beberapa jenis, seperti yang ditampilkan pada Tabel 2.3.

Tabel 2.3 Jenis-Jenis Sosial Media

No.	Jenis Sosial Media	Contoh
1.	Blog	Blogger
2.	Website Sosial Media	Facebook, Instagram
3.	Media Sharing Communities	Flickr
4.	Projek Kolaboratif	Wikipedia

### 2.4 Alat-Alat OSINT

Sebelum penelitian ini dilakukan, terdapat beberapa alat-alat OSINT yang sering digunakan untuk melakukan pengintaian oleh para profesional dari berbagai domain yang terkait intelijen khususnya para keamanan informasi atau yang biasa disebut *information security* (Chauhan & Panda, 2015b). Contoh dari alat-alat OSINT tersebut ditunjukkan pada Tabel 2.4.

Tabel 2.4 Contoh alat-alat OSINT

No.	Nama Alat	Penjelasan
1.	Creepy	Creepy adalah aplikasi yang dibuat menggunakan bahasa Python yang dapat mengekstrak informasi lokasi dari suatu gambar di sosial media. Creepy mendukung pencarian pada Twitter, Flickr, dan Instagram. Cara kerjanya adalah mengekstrak geolokasi berdasarkan informasi EXIF yang tersimpan pada gambar, informasi geolokasi tersedia melalui API, dan beberapa teknik lainnya.

2.	TheHarvester	TheHarvester adalah alat OSINT yang digunakan untuk mendapatkan alamat e-mail, nama karyawan, port terbuka, subdomain, host spanduk, dll. Dari sumber publik seperti search engine (Google, Bing, dan situs-situs lain seperti LinkedIn).
3.	Shodan	Shodan adalah sebuah mesin pencari berbasis web yang dapat diakses melalui url: <a href="https://www.shodan.io/">https://www.shodan.io/</a> yang berfungsi untuk melakukan pencarian terhadap spanduk/banner perangkat yang ada di internet.
4.	Search Diggity	Search Diggity adalah alat yang dikembangkan oleh Bishop Fox yang memiliki sejumlah opsi dan query basis data yang besar untuk berbagai mesin pencari yang dapat digunakan untuk mengumpulkan informasi yang berkaitan dengan target.
5.	Recon-ng	Recon-ng adalah alat OSINT yang bersifat open source yang dikembangkan oleh Tim Tomes yang sebagian besar menggunakan bahasa Python. Recon-ng terfokus pada pengintaian sumber terbuka yang berbasis web.
6.	Yahoo Pipes	Yahoo Pipes adalah aplikasi web dari Yahoo yang memberikan kebebasan kepada penggunanya untuk memilih sumber informasi yang berbeda dan mengaktifkan beberapa aturan kustom sesuai dengan kebutuhannya untuk mendapatkan hasil yang difilter.
7.	Maltego	Maltego adalah aplikasi OSINT yang paling terkenal dikarenakan memiliki kemampuan atau fitur yang unik. Selain mengekstrak data Maltego juga dapat merepresentasikan data tersebut dalam format yang mudah dipahami dan dianalisis.

## 2.5 Tantangan dalam Pengembangan Aplikasi OSINT

OSINT mempunyai tantangan dalam perlindungan privasi dan penegakan properti intelektual. Idealnya, proses teknis pengembangan alat OSINT ini dikombinasikan dengan perlindungan hukum dan etika sedemikian rupa sehingga produk yang dihasilkan memiliki desain yang sah secara hukum, dapat diterima di masyarakat (*social embedding*), dan pada saat yang sama menemukan sebuah cara yang cukup fleksibel sesuai dengan kebutuhan dari berbagai kelompok pengguna (Koops et al., 2013).

Faktor utama yang mempengaruhi nilai dari setiap data yang diambil adalah relevansi dan kredibilitas. Pengembangan Aplikasi OSINT bisa gagal karena empat masalah umum, yaitu (Weir, 2016):

1. Ketersediaan : Informasi yang diinginkan dan dicari mungkin tidak tersedia dari sumber yang dapat diakses.
2. Interpretasi : Sistem berkemungkinan salah membaca maksud dari pengguna.
3. Formulasi : Formulasi pencarian mungkin gagal dalam memberikan informasi yang diperlukan.
4. Kebingungan : Pencari informasi mungkin gagal dalam mengenali relevansi dan makna dari data yang diberikan.



## BAB III METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

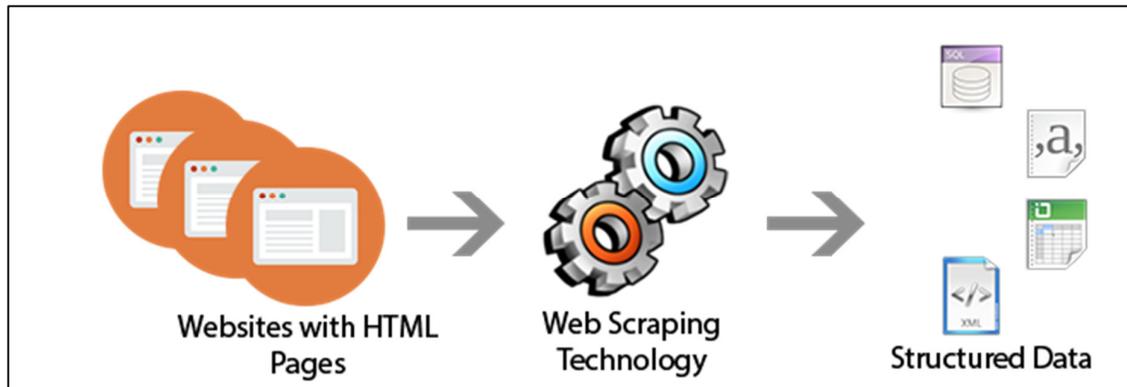
Dalam pembuatan Aplikasi Pencarian Nama Pengguna berbasis OSINT Framework ini, dilakukan tahapan berupa studi literatur. Studi literatur dilakukan untuk mengetahui metode yang nantinya akan digunakan untuk membuat Aplikasi Pencarian Nama Pengguna berbasis OSINT Framework ini. Pada tahapan ini, diputuskan bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah bahasa pemrograman *Python* dengan versi yang digunakan adalah versi 3.8 dan menggunakan modul *tkinter* yang digunakan untuk membuat tampilan *Graphical User Interface* (GUI) pada aplikasi ini. Metode-metode yang akan digunakan untuk mengembangkan aplikasi ini adalah *Web Scraping* dan mengembangkan *source code* pada aplikasi kerangka kerja OSINT. Alur metodologi penelitian pada penelitian ini, dijabarkan pada Diagram Alir Penelitian pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

### 3.2 Web Scraping

Web scraping adalah teknik pengambilan dokumen semi-terstruktur dari internet. Secara umum halaman-halaman web dibuat menggunakan bahasa HTML atau XHTML, dan dokumen tersebut dianalisis untuk diambil data-data tertentu dari halaman tersebut. Atau sering disebut teknik mengambil konten atau sebagian data dari situs web (Syabab, 2019). Proses web scraping diilustrasikan pada Gambar 3.2.



Gambar 3.2 Ilustrasi Cara Kerja Web Scraping

Sumber: (Rajagopal, 2019)

Sebagian besar metode pengintegrasian aplikasi kerangka kerja OSINT dalam membuat aplikasi pencarian nama pengguna ini adalah dengan menggunakan metode Web Scraping. Terdapat 3 *Library* yang digunakan untuk melakukan Web Scraping yaitu: BeautifulSoup, selenium, dan requests. *Library* requests digunakan untuk melakukan Web Scraping pada aplikasi kerangka kerja OSINT yang berupa website tanpa ada proses *javascript* di dalamnya. *Library* selenium adalah *library* yang biasa digunakan untuk melakukan otomatisasi *testing* terhadap sebuah website. Selain digunakan untuk otomatisasi *testing* website, selenium juga dapat digunakan untuk melakukan Web Scraping yang terdapat proses *javascript* di dalamnya. *Library* selenium memerlukan *library* webdriver yang berfungsi untuk menjalankan *browser* yang digunakan oleh *library* selenium. *Library* BeautifulSoup digunakan untuk memproses data dari halaman html web yang telah dilakukan Web Scraping.

### 3.3 Pengembangan Source Code Aplikasi

Aplikasi kerangka kerja OSINT yang dapat diintegrasikan dengan cara pengembangan *source code* aplikasi adalah namechk.sh dan WhatsMyName. WhatsMyName bersifat *open source* dan tersedia secara publik di github. WhatsMyName menggunakan linux shell command untuk menjalankan aplikasinya yang ditulis menggunakan bahasa *python*. Pada pengembangan aplikasi ini, *source code* aplikasi WhatsMyName dikembangkan dengan menjadikannya sebagai sebuah *class* sehingga dapat dipanggil di aplikasi Username Finder ini. Proses pencarian nama pengguna melalui aplikasi WhatsMyName secara langsung ditunjukkan pada Gambar 3.3.

```

kali@witchhacker: ~/De...
kali@witchhacker: ~/Desktop/WhatsMyName
File Actions Edit View Help
kali@witchhacker:~/Desktop/WhatsMyName$ sudo su
[sudo] password for kali:
root@witchhacker:/home/kali/Desktop/WhatsMyName# bash whatsmyname.sh
- 199 sites found in file.
- Looking up https://500px.com/adeindra6
- Looking up https://9gag.com/u/adeindra6
- Looking up https://about.me/adeindra6
* Skipping Angellist - Marked as not valid.
- Looking up https://www.anobii.com/adeindra6/profile
- Looking up https://appearoo.com/adeindra6
- Looking up https://bitbucket.org/api/2.0/users/adeindra6/
- Looking up http://www.authorstream.com/adeindra6/
- Looking up http://badoo.com/adeindra6/
- Looking up https://bandcamp.com/adeindra6
- Looking up https://launchpad.3signals.com/session/profile?username=adeindra6
- Looking up https://www.behance.net/adeindra6
- Looking up https://bugcrowd.com/adeindra6
- Looking up http://blip.fm/adeindra6
- Looking up http://blogmarks.net/user/adeindra6
- Looking up http://adeindra6.blogspot.com
[+] Found user at http://adeindra6.blogspot.com
- Looking up http://api.bodybuilding.com/api-proxy/bbc/get?slug=adeindra6
- Looking up https://buzerowd.com/adeindra6
- Looking up https://www.buzzfeed.com/adeindra6
- Looking up https://www.buzznet.com/author/adeindra6/
* Skipping Canva - Marked as not valid.
- Looking up https://cash.me/$adeindra6
- Looking up http://www.casdomain.com/member/adeindra6/
- Looking up https://profile.cheszburger.com/adeindra6
- Looking up https://www.chess.com/member/adeindra6
- Looking up https://discuss.codecademy.com/u/adeindra6/summary
- Looking up https://www.codementor.io/adeindra6
- Looking up http://www.colourlovers.com/Lover/adeindra6
- Looking up http://www.cruisemates.com/forum/members/adeindra6.html

```

Gambar 3.3 Hasil Pencarian Nama Pengguna menggunakan WhatsMyName

Sementara itu, namechk.sh dikembangkan menggunakan bahasa shell linux yang dapat dijalankan menggunakan terminal pada sistem operasi linux. Untuk menjalankan aplikasi namechk.sh pada sistem operasi windows digunakanlah aplikasi git bash agar dapat menjalankan perintah atau *command* linux pada sistem operasi windows. Proses pencarian nama pengguna melalui aplikasi namechk.sh secara langsung ditunjukkan pada Gambar 3.4.

```

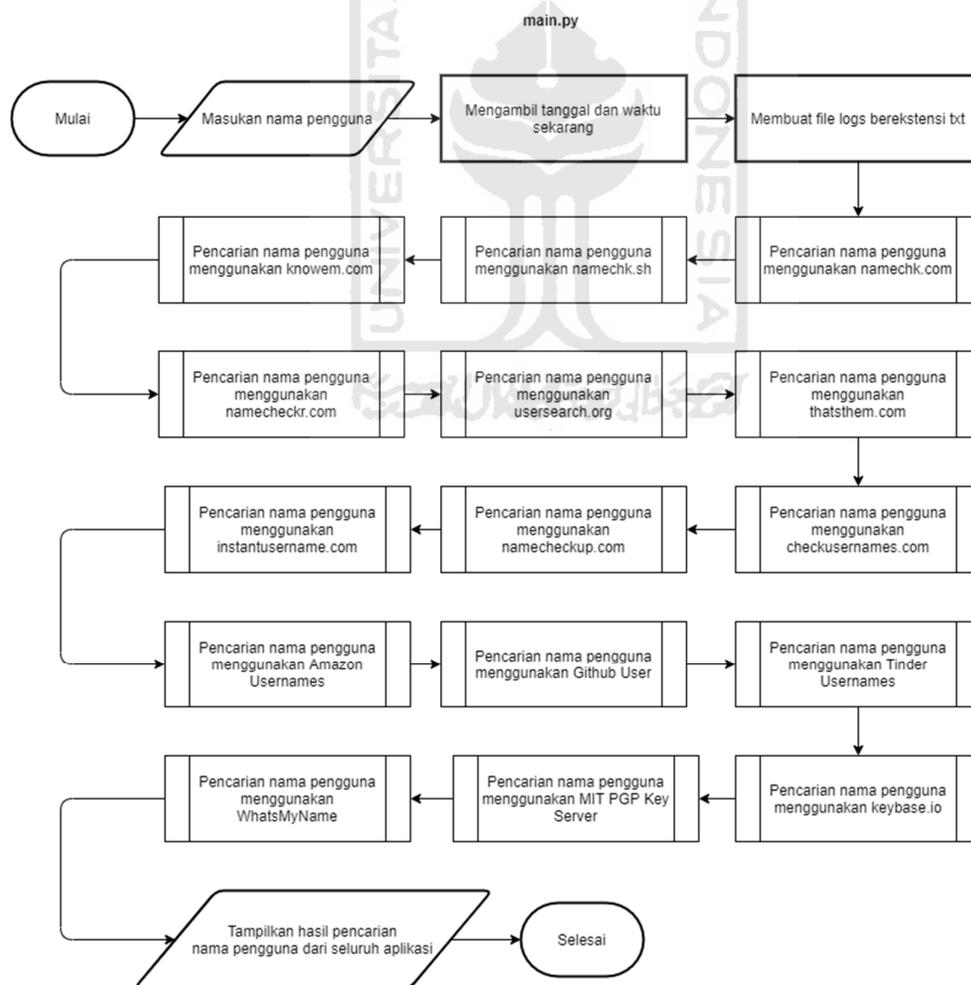
MINGW64~/Tasks/7th Term/TA/Namechk
namechk@namechk: ~/Tasks/7th Term/TA/Namechk (master)
$ bash namechk.sh donald -fu
NAMECHK.SH
osint tool based on namechk.com for checking usernames on more than 100 websites.
Developed by HATI
[+] Username found on: Facebook
[+] Profile url:
[+] Username found on: YouTube
[+] Profile url:
[+] Username found on: Twitter
[+] Profile url:
[+] Username found on: Instagram
[+] Profile url:
[+] Username found on: Blogger
[+] Profile url:
[+] Username found on: GooglePlus
[+] Profile url:
[+] Username found on: Twitch
[+] Profile url:
[+] Username found on: Reddit
[+] Profile url:
[+] Username found on: Ebay
[+] Profile url:
[+] Username found on: wordpress
[+] Profile url:
[+] Username found on: Pinterest
[+] Profile url:
[+] Username found on: Yelp
[+] Profile url:

```

Gambar 3.4 Menjalankan namechk.sh menggunakan git bash

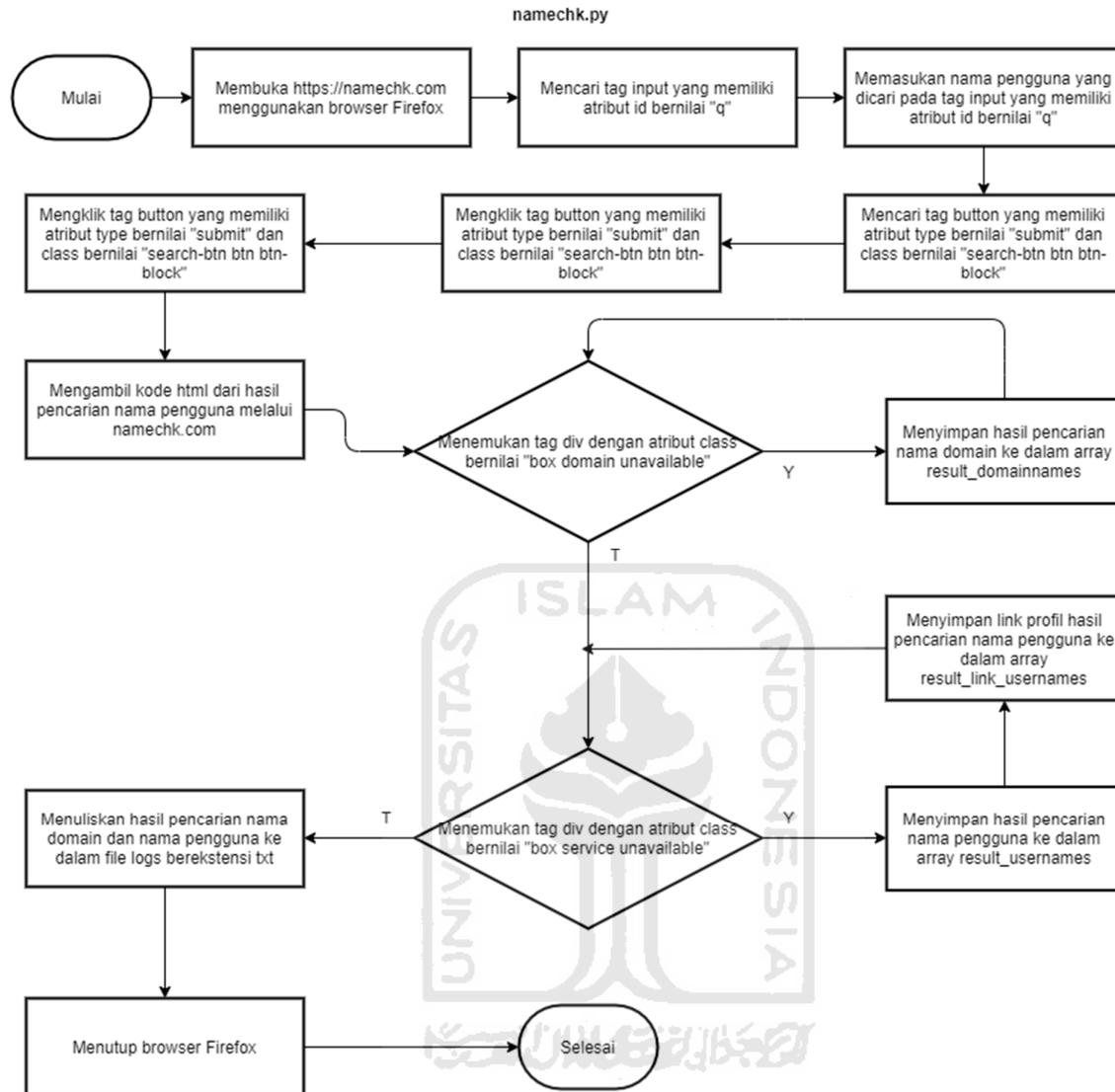
### 3.4 Diagram Alir Aplikasi

Aplikasi Username Finder terdiri dari program utama (*main program*) dan subprogram yang berupa *class*. Program utama berfungsi untuk menampilkan tampilan utama sehingga pengguna dapat memasukkan nama pengguna yang ingin dicari pada *textfield* dan mencari nama pengguna yang dimasukkan dengan menekan tombol “Search Username”. Setelah tombol “Search Username” diklik, aplikasi akan mengambil tanggal dan waktu tepat ketika tombol “Search Username” diklik. Setelah aplikasi mengambil tanggal dan waktu, aplikasi akan membuat file txt dengan nama file menggunakan tanggal dan waktu yang diambil dengan format “yyyy-mm-dd h.m.s”. Setelah itu, aplikasi akan memanggil subprogram satu per satu dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Setelah semua subprogram selesai dipanggil, aplikasi akan menampilkan hasil pencarian dari semua subprogram secara bersamaan. Proses pengembangan program utama Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.5.



Gambar 3.5 Diagram Alir Aplikasi Username Finder secara keseluruhan

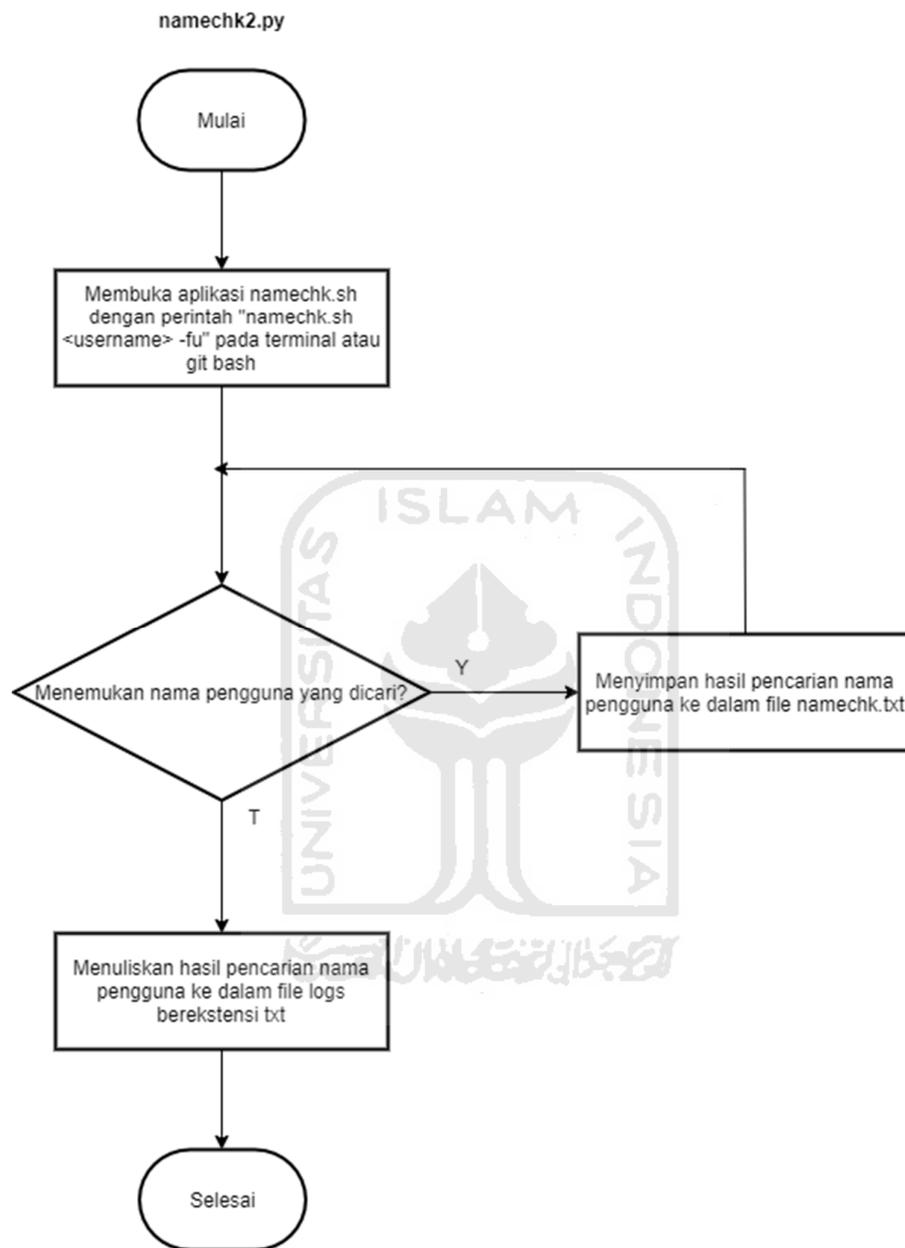
Pada subprogram namechk.com, setelah fungsi namechk dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://namechk.com> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut id bernilai “q” ( `<input id=”q”>` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut. Setelah itu, *library* selenium akan mencari *tag button* yang memiliki atribut type bernilai “submit” dan atribut class bernilai “search-btn btn btn-block” ( `<button type=”submit” class=”search-btn btn btn-block”>` ) dan mengklik tag tersebut sebanyak 2 kali untuk memulai pencarian terhadap nama pengguna yang dicari. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html namechk.com yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “box domain unavailable” ( `<div class=”box domain unavailable”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array `result_domainnames`. Setelah itu, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “box service unavailable” ( `<div class=”box service unavailable”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array `result_usernames` dan link profil yang terdapat pada atribut href tag a yang berada di dalam tag div tersebut, akan disimpan ke dalam array `result_link_usernames`. Hal ini bertujuan agar link profil dapat dijadikan *hyperlink* ketika hasil pencarian ditampilkan. Setelah itu, hasil pencarian nama domain dan nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat *RAM*. Proses pengembangan subprogram namechk.com pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.6.



Gambar 3.6 Diagram Alir fungsi namechk.py

Pada subprogram namechk.sh, setelah fungsi namechk2 dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka aplikasi namechk.sh dengan perintah “namechk.sh <username> -fu” melalui git bash apabila menggunakan sistem operasi windows atau melalui terminal apabila menggunakan sistem operasi linux. Apabila aplikasi namechk.sh menemukan nama pengguna yang dicari, hasil pencariannya akan dimasukkan ke dalam file namechk.txt. Hal ini bertujuan agar hasil pencarian nama pengguna dapat ditampilkan melalui Aplikasi Username Finder. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam file log

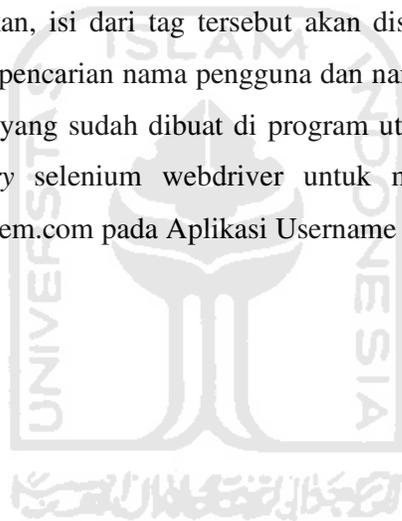
berekstensi txt yang sudah dibuat di program utama. Proses pengembangan subprogram namechk.sh pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.7.

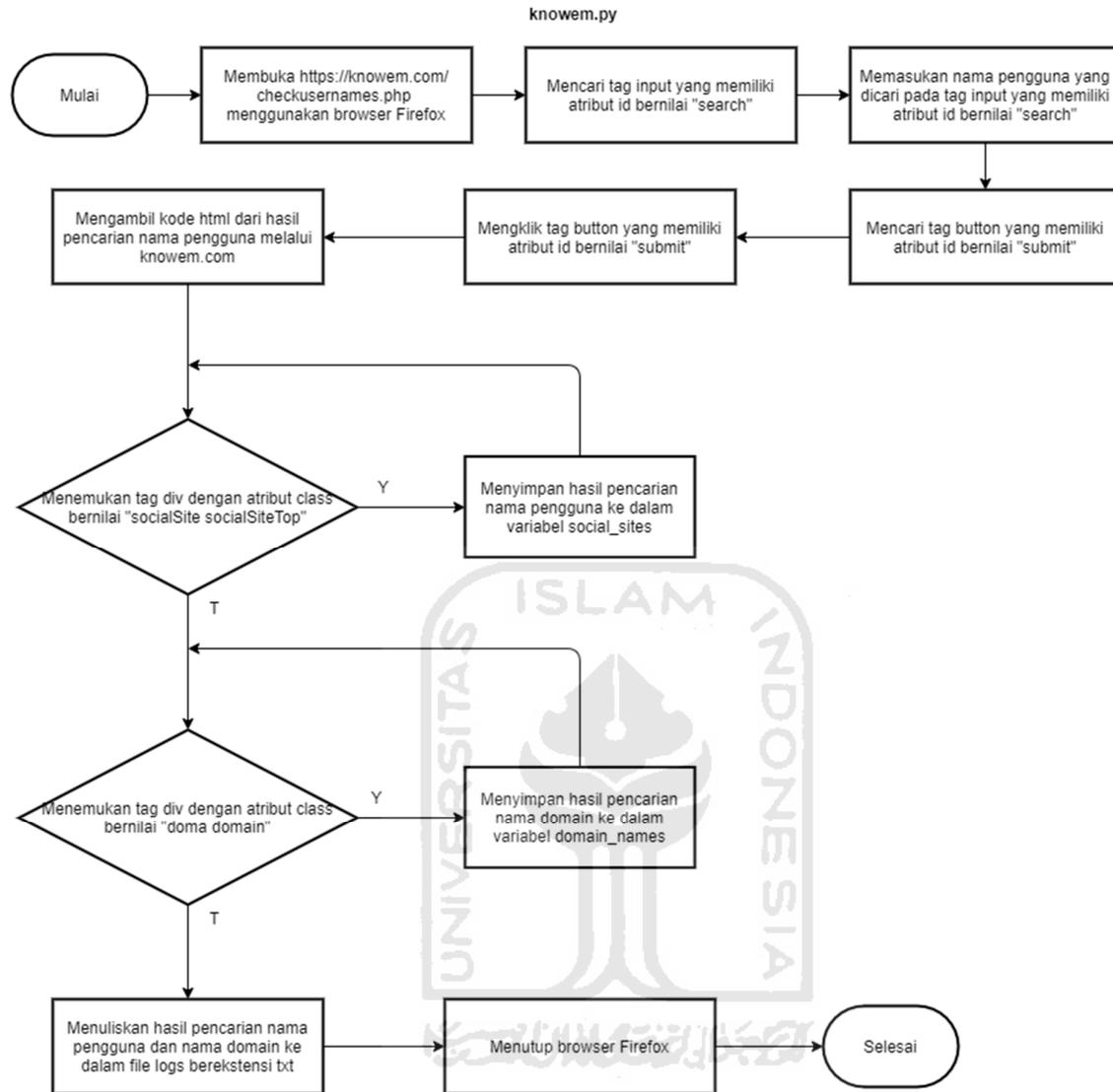


Gambar 3.7 Diagram Alir fungsi namechk2.py

Pada subprogram knowem.com, setelah fungsi knowem dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://knowem.com/checkusernames.php> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka,

*library* selenium akan mencari tag input yang memiliki atribut id bernilai “search” ( `<input id=”search”>` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut. Setelah itu, *library* selenium akan mencari *tag button* yang memiliki atribut type bernilai “submit” ( `<button type=”submit”>` ) dan mengklik tag tersebut untuk memulai pencarian terhadap nama pengguna yang dicari. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html knowem.com yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “socialSite socialSiteTop” ( `<div class=”socialSite socialSiteTop”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam variabel `social_sites`. Setelah itu, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “doma domain” ( `<div class=”doma domain”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam variabel `domain_names`. Setelah itu, hasil pencarian nama pengguna dan nama domain akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat RAM. Proses pengembangan subprogram knowem.com pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.8.

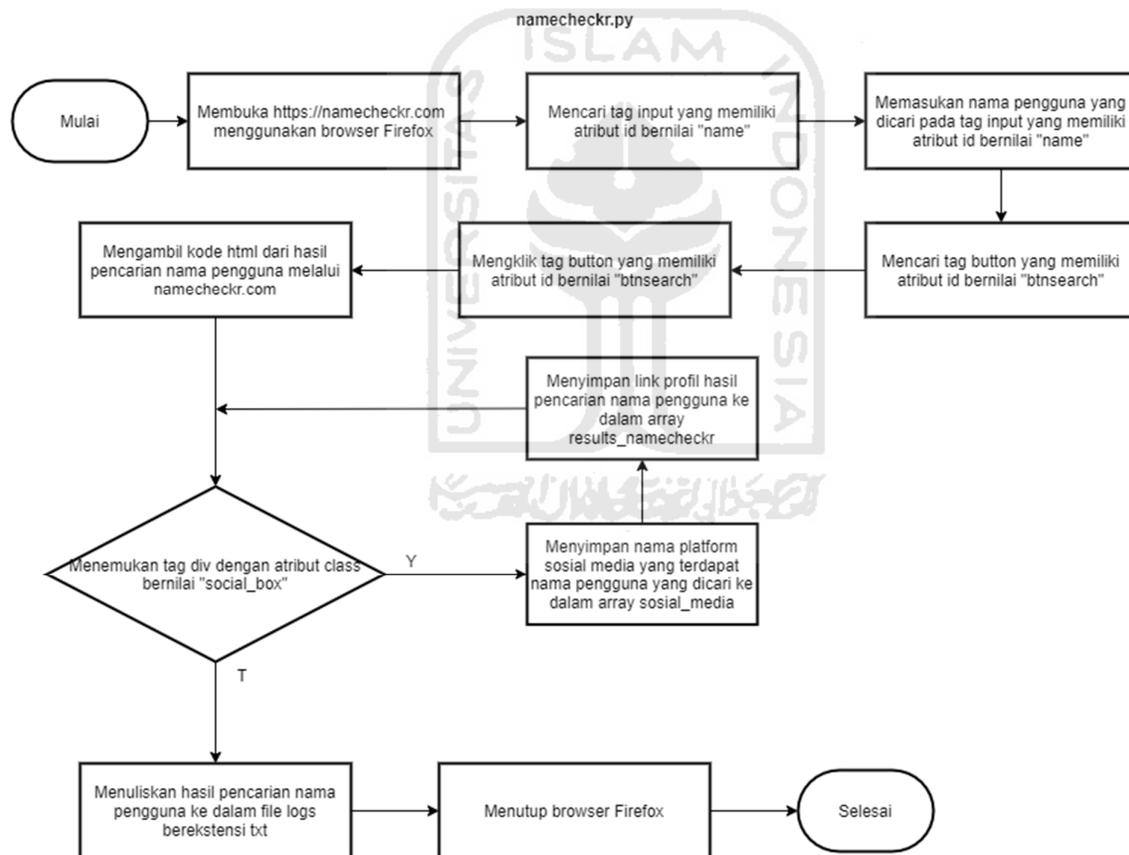




Gambar 3.8 Diagram Alir fungsi knowem.py

Pada subprogram namecheckr.com, setelah fungsi namecheckr dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://namecheckr.com> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut id bernilai “name” ( `<input id="name">` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut. Setelah itu, *library* selenium akan mencari *tag button* yang memiliki atribut id bernilai “btnsearch” dan mengklik tag tersebut untuk memulai pencarian terhadap nama pengguna yang dicari. Setelah pencarian nama pengguna selesai dilakukan, *library*

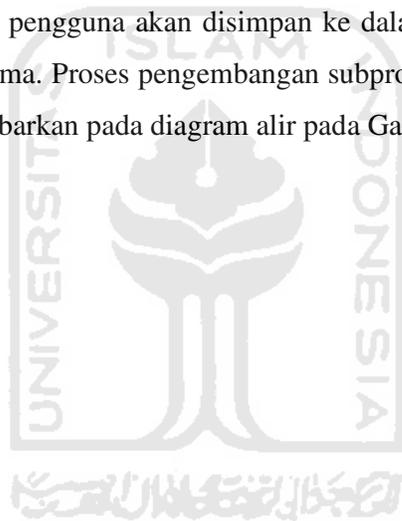
selenium akan mengambil kode html namecheckr.com yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “social\_box” ( `<div class=“social_box”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array `social_media` dan link profil yang terdapat pada atribut href tag a yang berada di dalam tag div tersebut, akan disimpan ke dalam array `results_namecheckr`. Hal ini bertujuan agar link profil dapat dijadikan *hyperlink* ketika hasil pencarian ditampilkan. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat *RAM*. Proses pengembangan subprogram namecheckr.com pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.9.

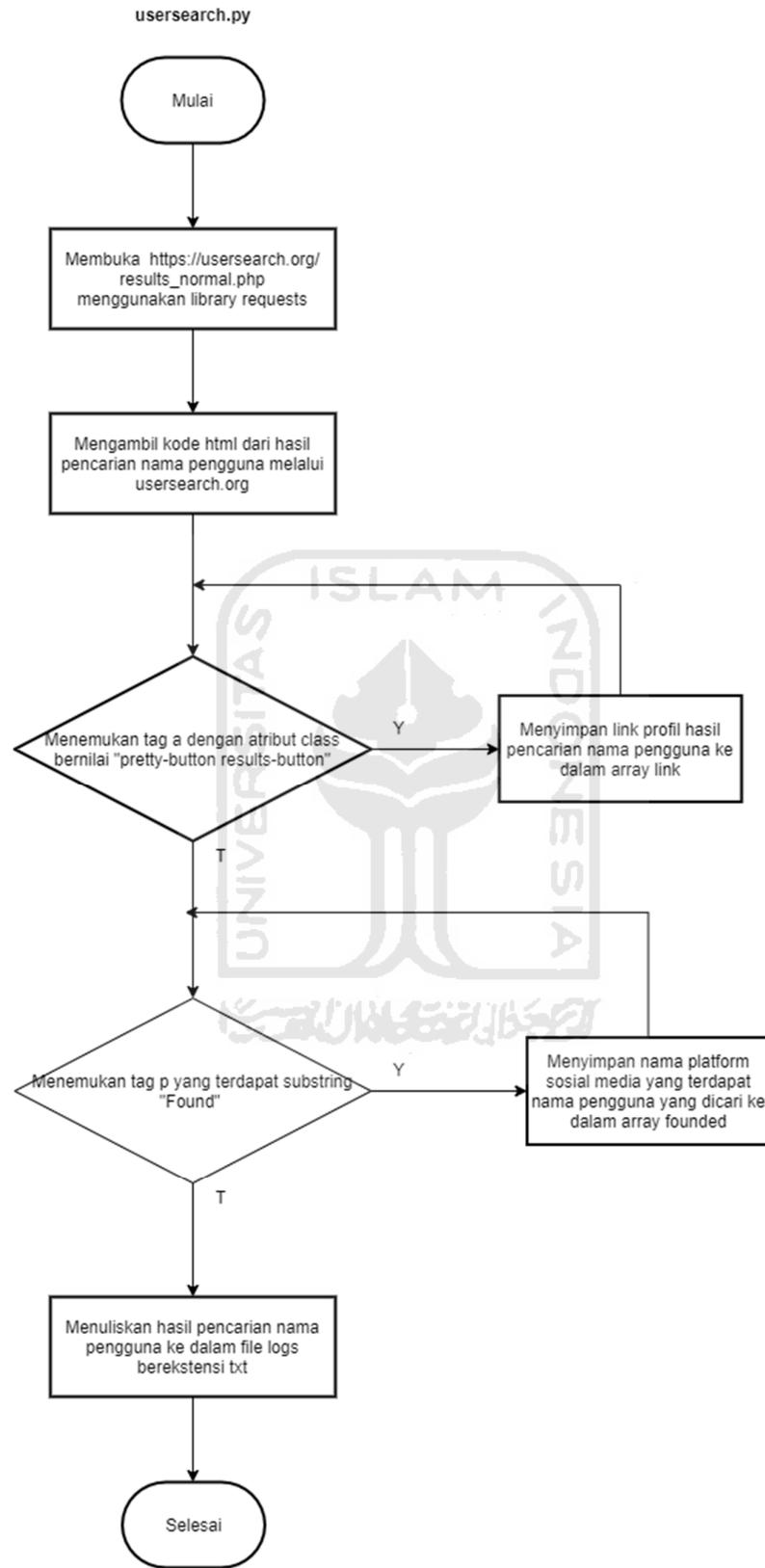


Gambar 3.9 Diagram Alir fungsi namecheckr.py

Pada subprogram usersearch.org, setelah fungsi usersearch dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka [https://usersearch.org/results\\_normal.php](https://usersearch.org/results_normal.php)

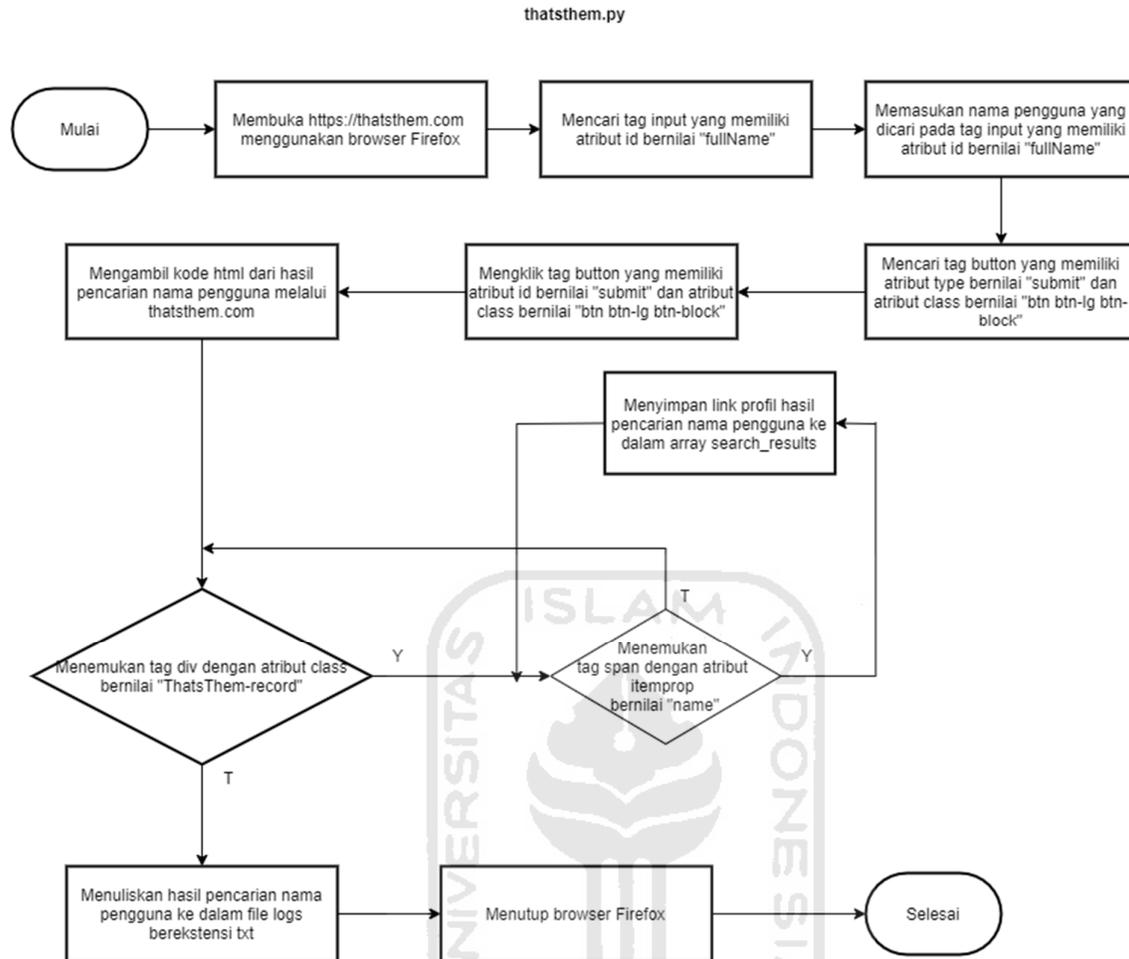
menggunakan *library* requests dengan parameter post url [https://usersearch.org/result\\_normal.php](https://usersearch.org/result_normal.php) dan nama pengguna yang dicari, sehingga pencarian nama pengguna langsung dilakukan pada saat itu juga. Setelah pencarian nama pengguna selesai dilakukan, *library* requests akan mengambil kode html usersearch.org yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag a dengan atribut class bernilai “pretty-button results-button” ( `<a class=“ pretty-button results-button”>` ) dan apabila tag tersebut ditemukan, link profil yang terdapat pada atribut href tag a tersebut, akan disimpan ke dalam array link. Hal ini bertujuan agar link profil dapat dijadikan *hyperlink* ketika hasil pencarian ditampilkan. Setelah itu, *library* BeautifulSoup akan mencari tag p yang terdapat substring “Found” ( `<p>... Found ...</p>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array founded. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Proses pengembangan subprogram usersearch.org pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.10.





Gambar 3.10 Diagram Alir fungsi usersearch.py

Pada subprogram thatsthem.com, setelah fungsi thatsthem dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://thatsthem.com> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut id bernilai “fullName” ( `<input id=“fullName”>` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut. Setelah itu, *library* selenium akan mencari *tag button* yang memiliki atribut type bernilai “submit” dan atribut class bernilai “btn btn-lg btn-block” ( `<button type=“submit” class=“btn btn-lg btn-block”>` ) dan mengklik tag tersebut untuk memulai pencarian terhadap nama pengguna yang dicari. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html thatsthem.com yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “ThatsThem-record” ( `<div class=“ThatsThem-record”>` ) dan apabila tag tersebut ditemukan, *library* BeautifulSoup akan mencari tag span dengan atribut itemprop bernilai “name” ( `<span itemprop=“name”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array `search_results`. Setelah itu, hasil pencarian nama domain dan nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat *RAM*. Proses pengembangan subprogram thatsthem.com pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.11.

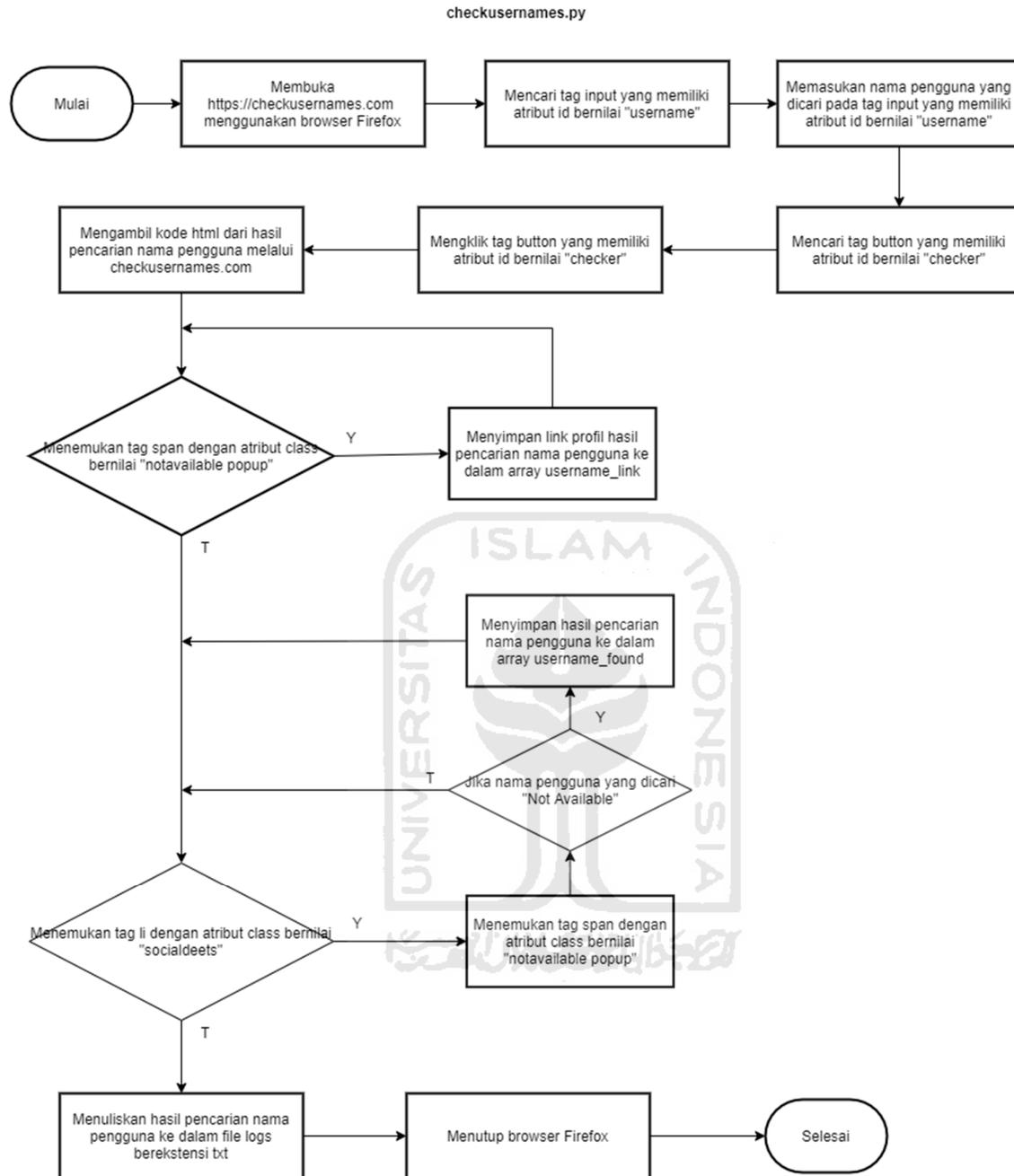


Gambar 3.11 Diagram Alir fungsi thatsthem.py

Pada subprogram checkusernames.com, setelah fungsi namechk dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://checkusernames.com/> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut id bernilai “username” ( `<input id="username">` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut. Setelah itu, *library* selenium akan mencari *tag button* yang memiliki atribut id bernilai “checker” ( `<button id="checker">` ) dan mengklik tag tersebut untuk memulai pencarian terhadap nama pengguna yang dicari. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html checkusernames.com yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag span dengan atribut class bernilai

“notavailable popup” ( `<span class=“notavailable popup”>` ) dan apabila tag tersebut ditemukan, link profil yang terdapat pada atribut href tag a yang berada di dalam tag span tersebut, akan disimpan ke dalam array `username_link`. Hal ini bertujuan agar link profil dapat dijadikan hyperlink ketika hasil pencarian ditampilkan. Setelah itu, *library* BeautifulSoup akan mencari tag li dengan atribut class bernilai “socialdeets” ( `<li class=“ socialdeets”>` ) dan tag span dengan atribut class bernilai “notavailable popup”. Apabila kedua tag tersebut ditemukan, isi dari kedua tag tersebut jika terdapat substring “Not Available” akan disimpan ke dalam array `username_found`. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat RAM. Proses pengembangan subprogram `checkusernames.com` pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.12.

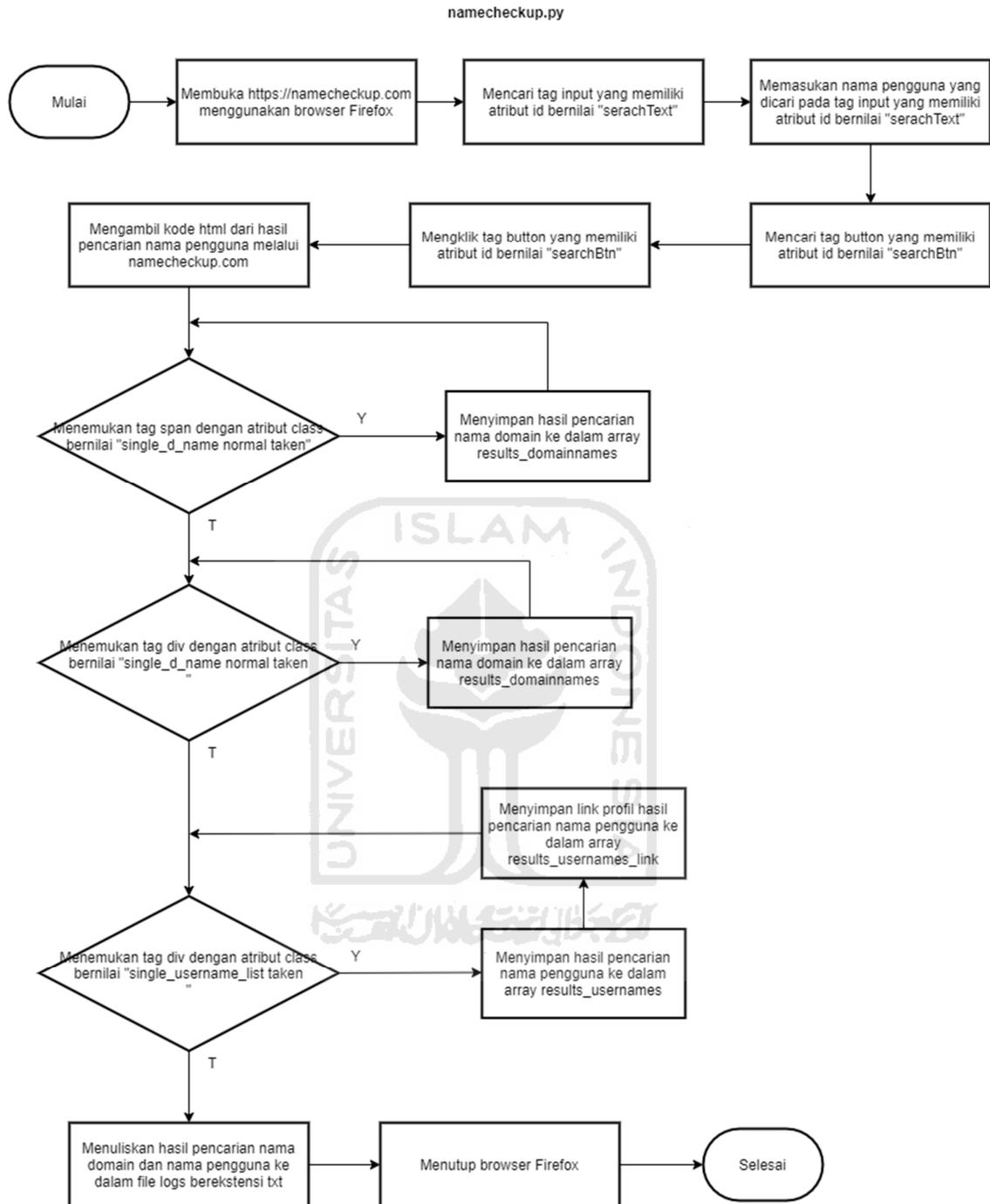




Gambar 3.12 Diagram Alir fungsi checkusernames.py

Pada subprogram namecheckup.com, setelah fungsi namecheckup dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://namecheckup.com> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut id bernilai “searchText” ( <input

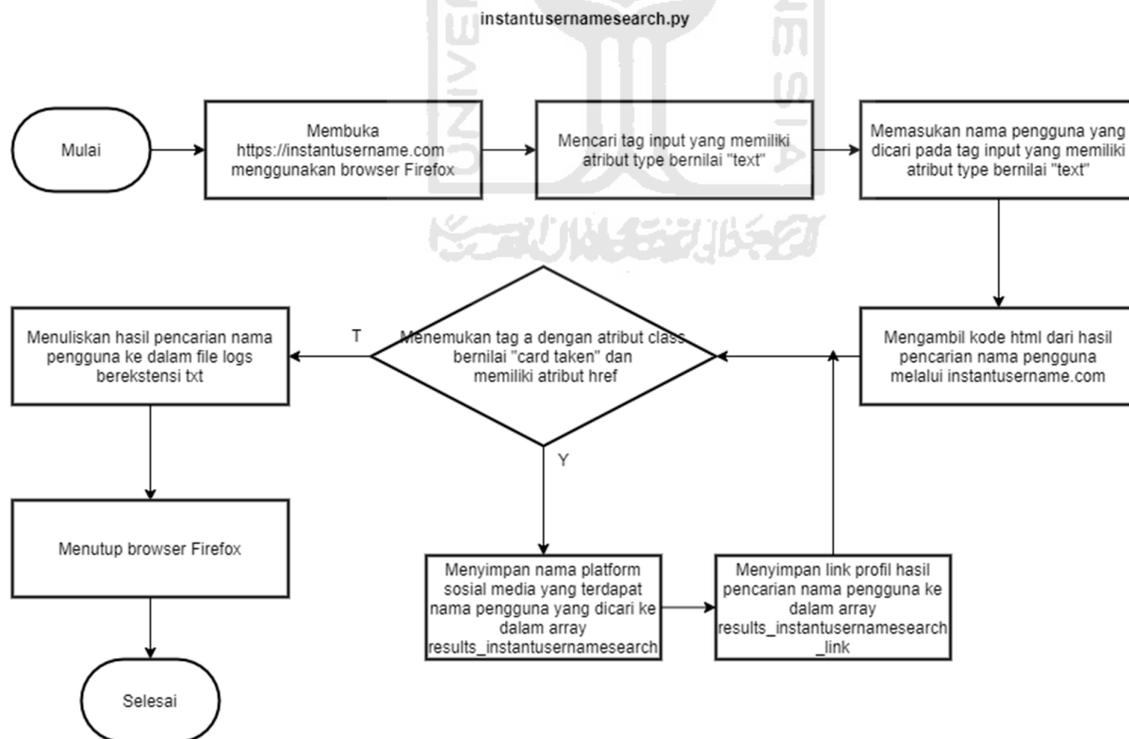
id="searchText"> ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut. Setelah itu, *library* selenium akan mencari *tag button* yang memiliki atribut id bernilai "searchBtn" ( <button id="searchBtn"> ) dan mengklik tag tersebut untuk memulai pencarian terhadap nama pengguna yang dicari. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html namecheckup.com yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag span dengan atribut class bernilai "single\_d\_name normal taken" ( <span class="single\_d\_name normal taken"> ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array results\_domainnames. Setelah itu, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai "single\_d\_name normal taken" ( <div class="single\_d\_name normal taken"> ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array results\_domainnames. Setelah itu, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai "single\_username\_list taken" ( <div class="single\_username\_list taken"> ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array results\_usernames dan link profil yang terdapat pada atribut href tag a yang berada di dalam tag div tersebut, akan disimpan ke dalam array results\_usernames\_link. Hal ini bertujuan agar link profil dapat dijadikan *hyperlink* ketika hasil pencarian ditampilkan. Setelah itu, hasil pencarian nama domain dan nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat RAM. Proses pengembangan subprogram namecheckup.com pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.13.



Gambar 3.13 Diagram Alir fungsi namecheckup.py

Pada subprogram `instantusername`, setelah fungsi `instantusername` dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka `https://instantusername.com` menggunakan Browser Firefox melalui *library* selenium

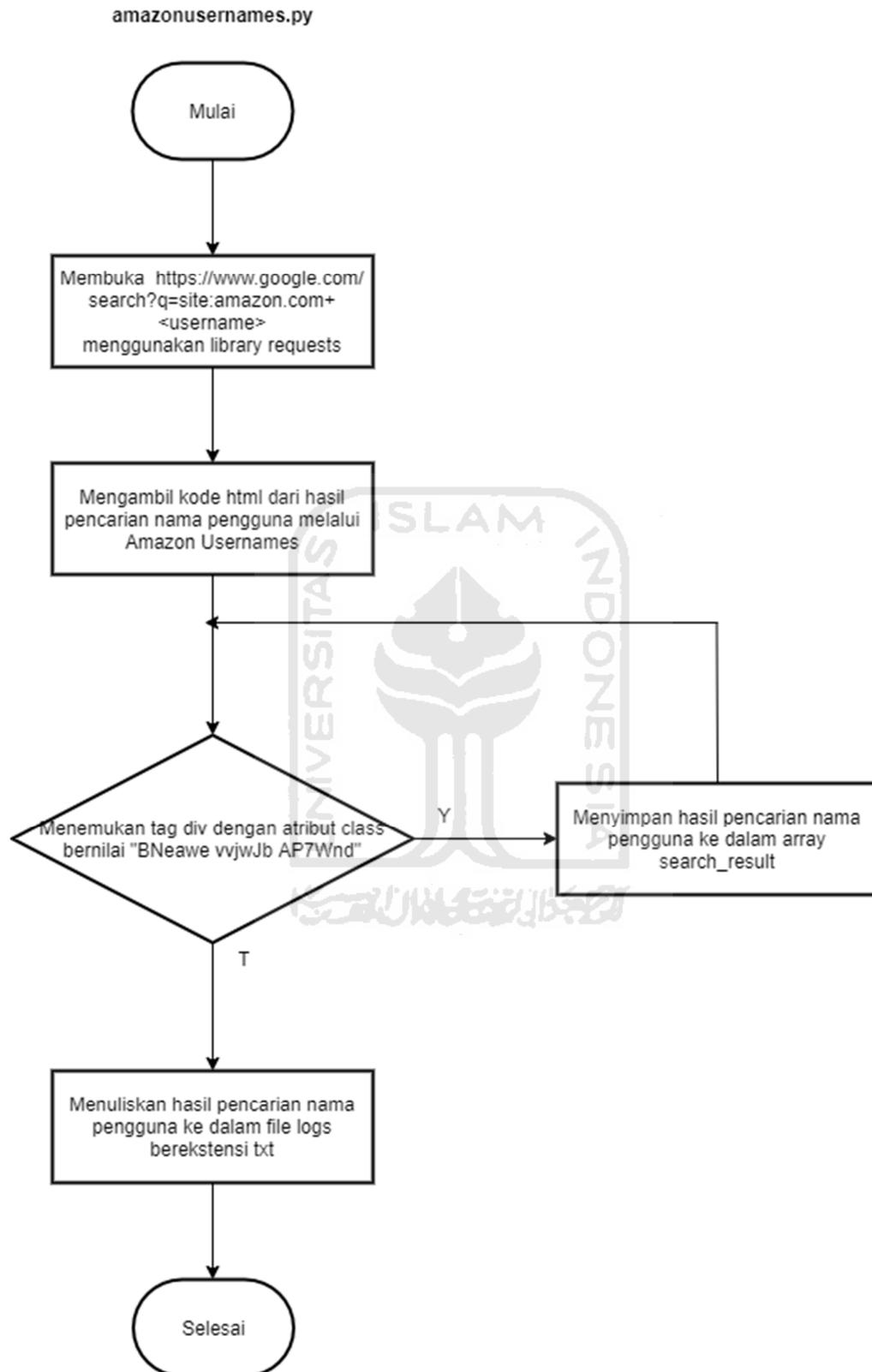
webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut type bernilai "text" ( `<input type="text">` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut, secara otomatis *instantusername* akan langsung melakukan pencarian nama pengguna. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html *instantusername.com* yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag a dengan atribut class bernilai "card taken" dan memiliki atribut href ( `<a class="card taken" href="">` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut yang berisikan nama platform sosial media akan disimpan ke dalam array `results_instantusername` dan link profil yang terdapat pada atribut href tag a akan disimpan ke dalam array `results_instantusername_link`. Hal ini bertujuan agar link profil dapat dijadikan hyperlink ketika hasil pencarian ditampilkan. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat *RAM*. Proses pengembangan subprogram *instantusername* pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.14.



Gambar 3.14 Diagram Alir fungsi *instantusername*.py

Pada subprogram amazonusernames, setelah fungsi amazonusernames dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://www.google.com/search?q=site:amazon.com+<username>> menggunakan *library* requests. Setelah link tersebut berhasil dibuka, *library* requests akan mengambil kode html <https://www.google.com/search?q=site:amazon.com+<username>>. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “BNeawe vvjwJb AP7Wnd” ( `<div class=“BNeawe vvjwJb AP7Wnd”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array `search_result`. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Proses pengembangan subprogram amazonusernames pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.15.

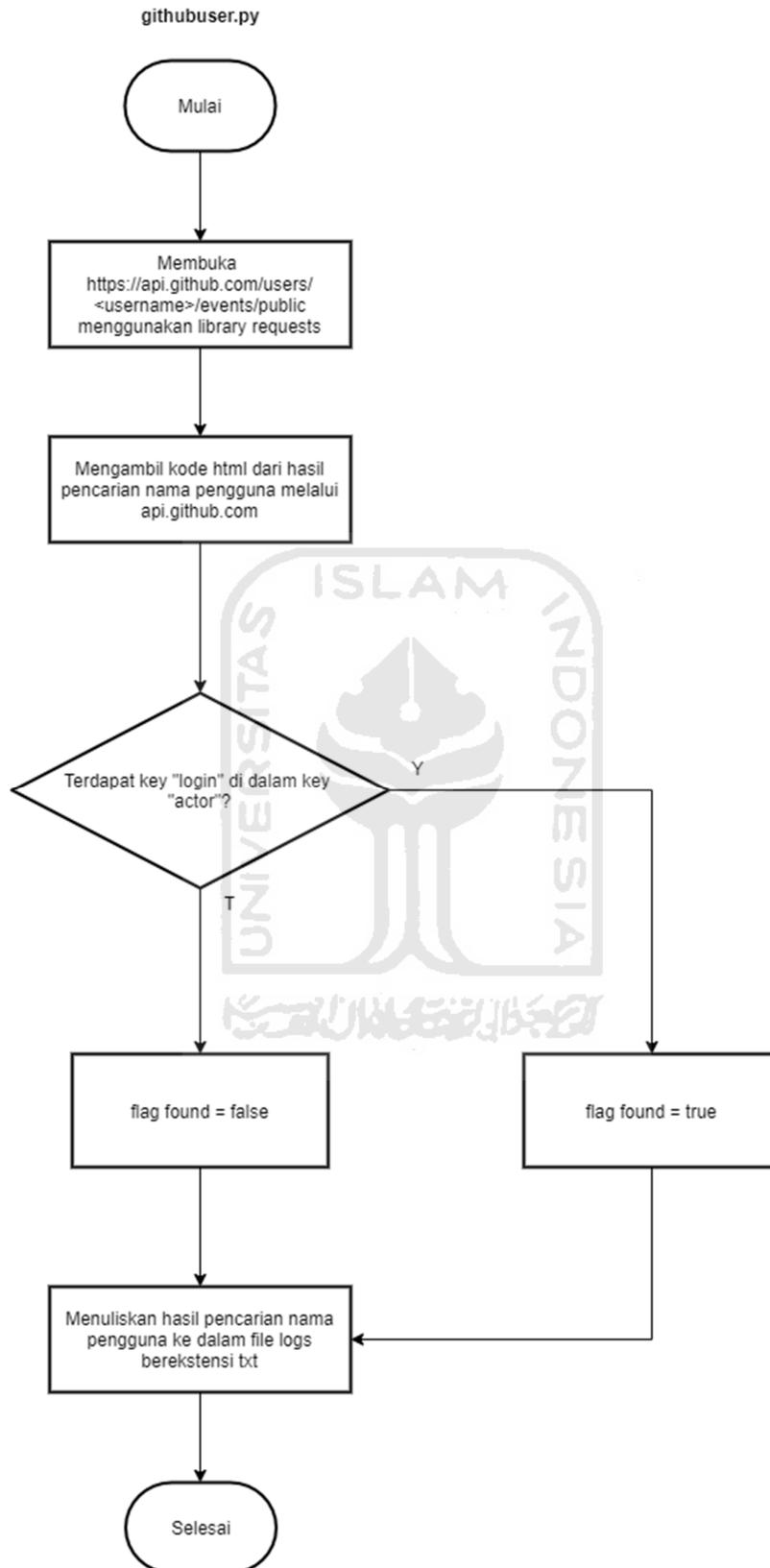




Gambar 3.15 Diagram Alir fungsi amazonusernames.py

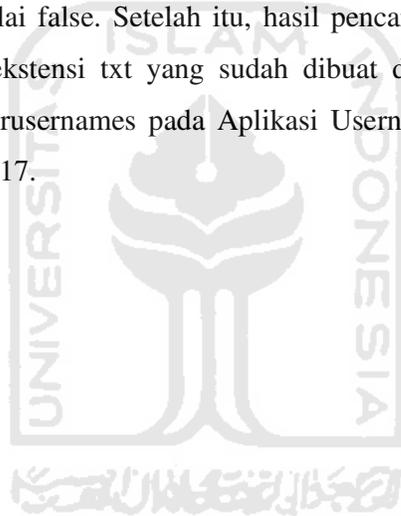
Pada subprogram githubuser, setelah fungsi githubuser dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://api.github.com/users/<username>/events/public> menggunakan *library requests*. Setelah link tersebut berhasil dibuka, *library requests* akan mengambil kode html <https://api.github.com/users/<username>/events/public>. Setelah kode html diambil, Aplikasi akan mencari key “login” yang berada di dalam key “actor” dan apabila key tersebut ditemukan, variabel found akan diberi nilai true. Sedangkan, apabila key “login” yang berada di dalam key “actor” tidak ditemukan, maka variabel found akan diberikan nilai false. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Proses pengembangan subprogram githubuser pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.16.

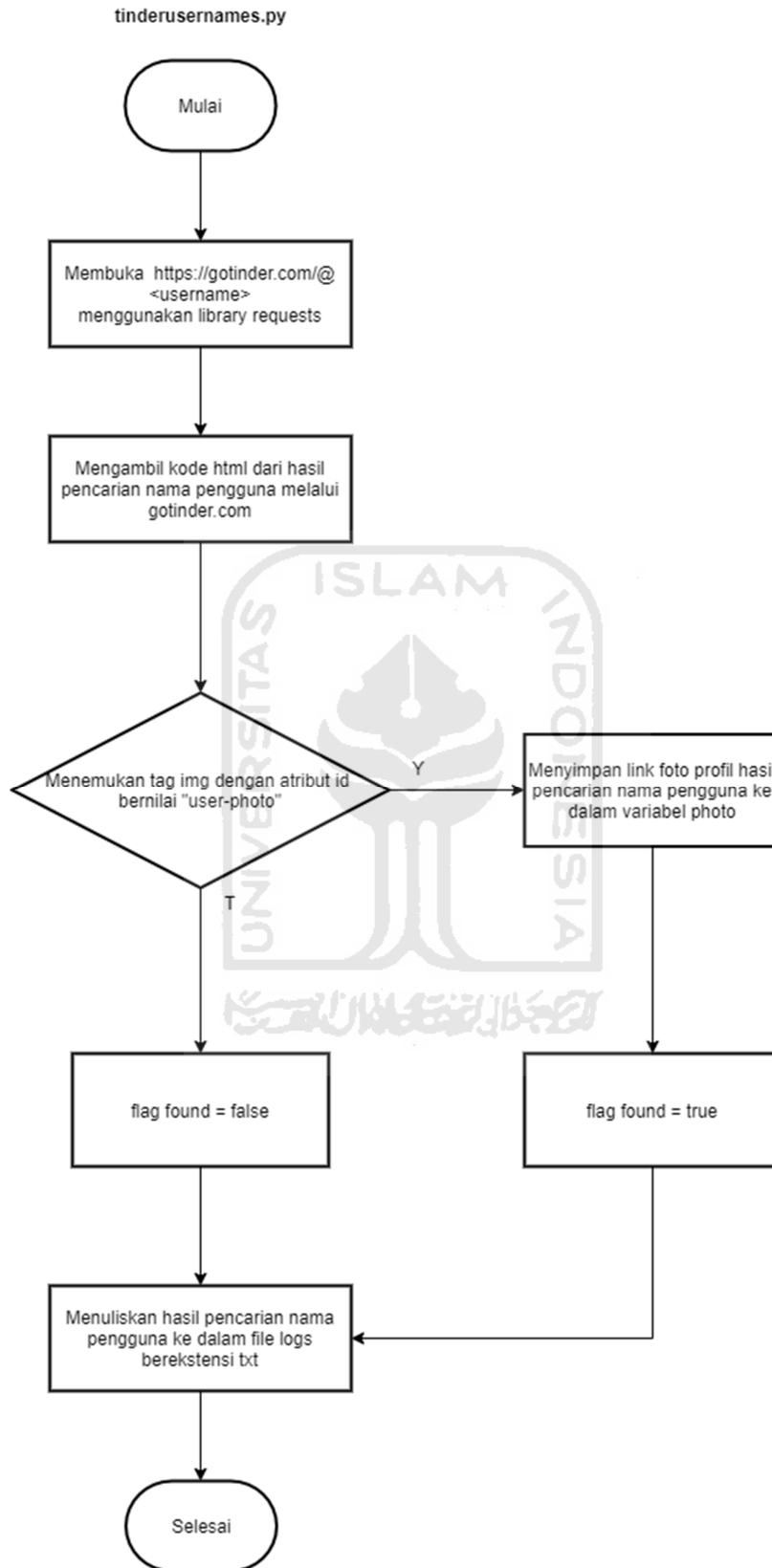




Gambar 3.16 Diagram Alir fungsi githubuser.py

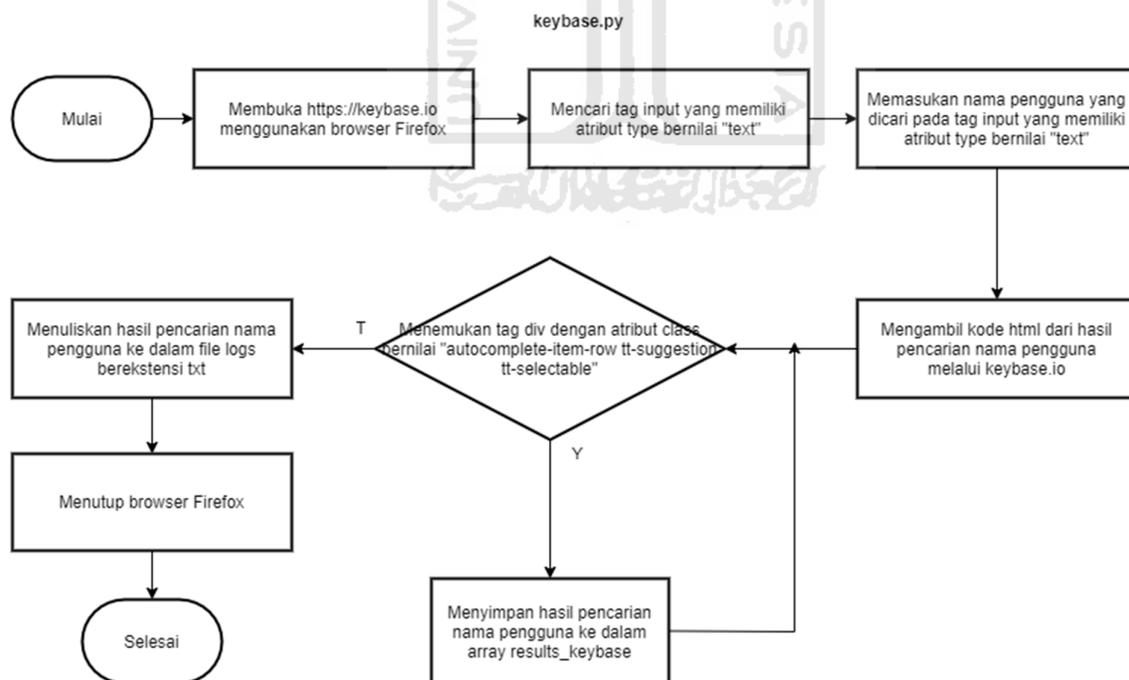
Pada subprogram `tinderusernames`, setelah fungsi `tinderusernames` dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://www.gotinder.com/@<username>> menggunakan *library requests*. Setelah link tersebut berhasil dibuka, *library requests* akan mengambil kode html <https://www.gotinder.com/@<username>>. Setelah kode html diambil, *library BeautifulSoup* akan mencari tag `img` dengan atribut `id` bernilai “user-photo” ( `<img class=“user-photo”>` ) dan apabila tag tersebut ditemukan, link profil yang terdapat pada atribut `href` tag `a` tersebut, akan disimpan ke dalam variabel `photo` dan memberi nilai `true` pada variabel `found`. Hal ini bertujuan agar link profil dapat dijadikan *hyperlink* ketika hasil pencarian ditampilkan. Apabila tag `img` dengan atribut `id` bernilai “user-photo” ( `<img class=“user-photo”>` ) tidak ditemukan, maka variabel `found` akan diberikan nilai `false`. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi `txt` yang sudah dibuat di program utama. Proses pengembangan subprogram `tinderusernames` pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.17.





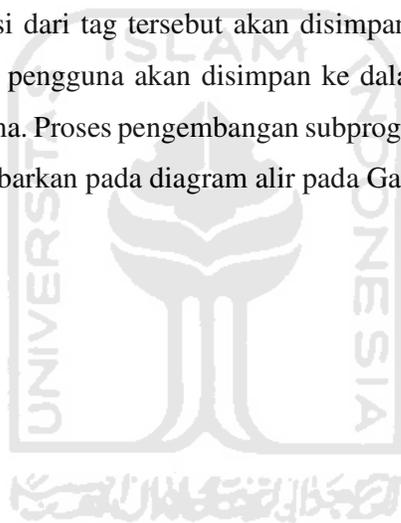
Gambar 3.17 Diagram Alir fungsi tinderusernames.py

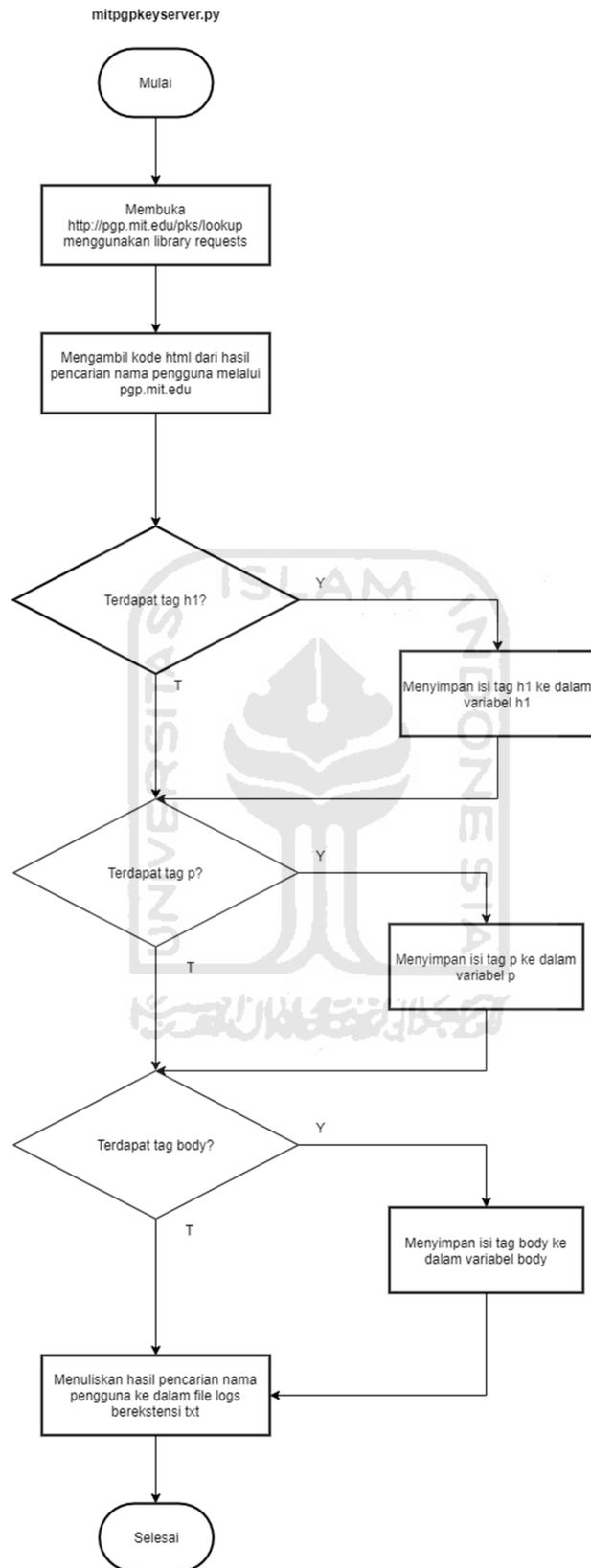
Pada subprogram keybase, setelah fungsi keybase dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <https://keybase.io> menggunakan Browser Firefox melalui *library* selenium webdriver. Setelah berhasil dibuka, *library* selenium akan mencari tag input yang memiliki atribut type bernilai “text” ( `<input type=“text”>` ) dan memasukkan nama pengguna yang didapat dari parameter nama pengguna ke dalam tag tersebut, secara otomatis keybase akan langsung melakukan pencarian nama pengguna. Setelah pencarian nama pengguna selesai dilakukan, *library* selenium akan mengambil kode html keybase.io yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag div dengan atribut class bernilai “autocomplete-item-row tt-suggestion tt-selectable” ( `<div class=“ autocomplete-item-row tt-suggestion tt-selectable”>` ) dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam array `results_keybase`. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi txt yang sudah dibuat di program utama. Kemudian, menutup Browser Firefox melalui *library* selenium webdriver untuk menghemat *RAM*. Proses pengembangan subprogram keybase pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.18.



Gambar 3.18 Diagram Alir fungsi keybase.py

Pada subprogram `mitpgpkeyserver`, setelah fungsi `mitpgpkeyserver` dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan membuka <http://pgp.mit.edu/pks/lookup> menggunakan *library* requests dengan parameter `get url http://pgp.mit.edu/pks/lookup` dan nama pengguna yang dicari, sehingga pencarian nama pengguna langsung dilakukan pada saat itu juga. Setelah pencarian nama pengguna selesai dilakukan, *library* requests akan mengambil kode html `mitpgpkeyserver` yang telah selesai melakukan pencarian nama pengguna yang dicari. Setelah kode html diambil, *library* BeautifulSoup akan mencari tag `h1` dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam variabel `h1`. Setelah itu, *library* BeautifulSoup akan mencari tag `p` dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam variabel `p`. Setelah itu, *library* BeautifulSoup akan mencari tag `body` dan apabila tag tersebut ditemukan, isi dari tag tersebut akan disimpan ke dalam variabel `body`. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam *file log* berekstensi `txt` yang sudah dibuat di program utama. Proses pengembangan subprogram `mitpgpkeyserver` pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.19.

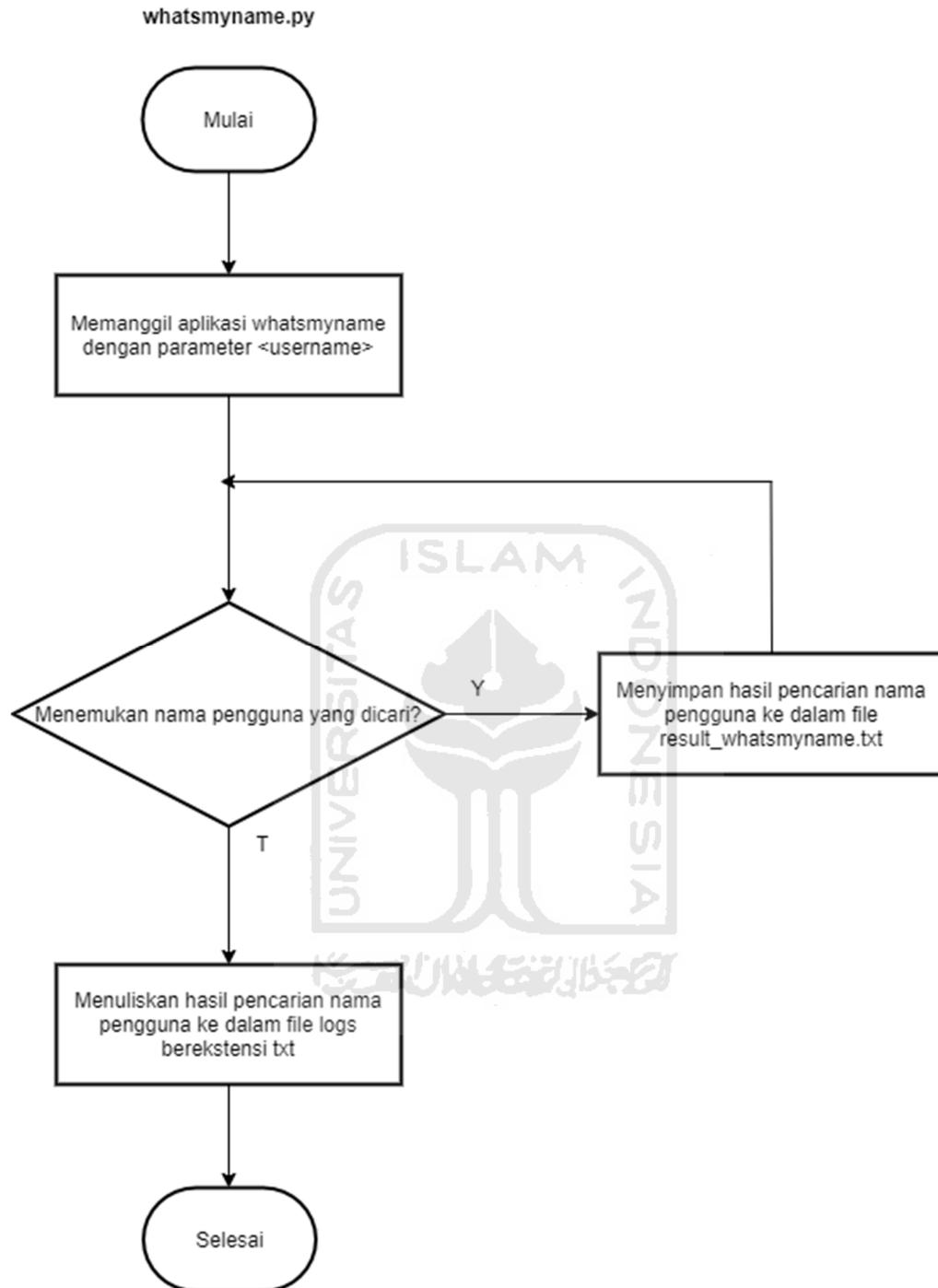




Gambar 3.19 Diagram Alir fungsi mitpgpkeyserver.py

Pada subprogram `whatsmyname`, setelah fungsi `whatsmyname` dipanggil dengan parameter nama pengguna dan tanggal dan waktu yang diambil ketika tombol “Search Username” diklik. Aplikasi Username Finder akan memanggil aplikasi `whatsmyname` yang telah dimodifikasi menjadi fungsi dengan parameter nama pengguna yang dicari. Apabila aplikasi `whatsmyname` menemukan nama pengguna yang dicari, maka hasil pencariannya akan dimasukkan ke dalam file `result_whatsmyname.txt`. Hal ini bertujuan agar hasil pencarian nama pengguna dapat ditampilkan melalui Aplikasi Username Finder. Setelah itu, hasil pencarian nama pengguna akan disimpan ke dalam file log berekstensi `txt` yang sudah dibuat di program utama. Proses pengembangan subprogram `whatsmyname` pada Aplikasi Username Finder, digambarkan pada diagram alir pada Gambar 3.20.



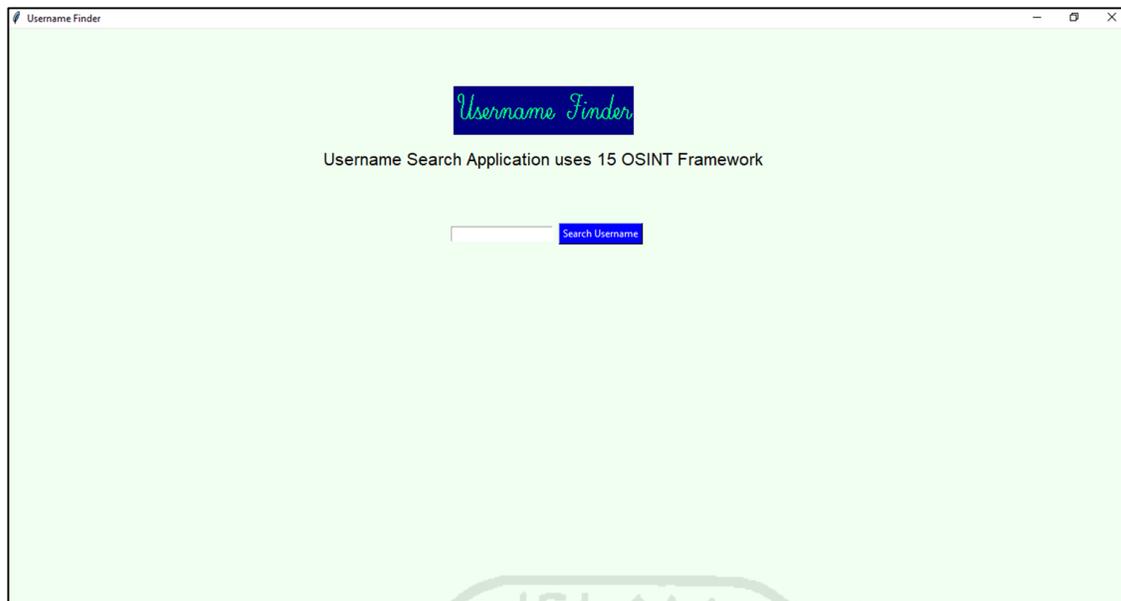


Gambar 3.20 Diagram Alir fungsi whatsmyname.py

## BAB IV IMPLEMENTASI DAN HASIL

### 4.1 Implementasi Tampilan Utama

Tampilan utama aplikasi Username Finder ini dibuat cukup sederhana. Pada tampilan utama hanya terdapat *textfield* dan tombol “Search Username” untuk melakukan pencarian. Terdapat juga beberapa label yang hanya berfungsi sebagai dekorasi tampilan agar tampilan terlihat lebih menarik. Pengguna hanya perlu menuliskan nama pengguna yang ingin dicari pada *textfield* yang telah disediakan dan menekan tombol “Search Username”. Setelah menekan tombol “Search Username”, proses pencarian nama pengguna akan berjalan dengan cara memanggil fungsi-fungsi pencarian nama pengguna dari setiap aplikasi kerangka kerja OSINT yang sudah diintegrasikan dengan aplikasi Username Finder ini. Proses pencarian nama pengguna akan berlangsung sekitar 15 menit, hal ini dikarenakan aplikasi Username Finder melakukan pencarian menggunakan aplikasi kerangka kerja OSINT yang sudah diintegrasikan secara satu per satu. Setiap aplikasi kerangka kerja OSINT yang diintegrasikan, membutuhkan waktu pencarian yang berbeda-beda. Pada metode web scraping yang menggunakan *library* requests waktu yang dibutuhkan cukup sebentar (sekitar 30 detik), sedangkan pada metode web scraping yang menggunakan *library* selenium dan pengembangan *source code* aplikasi, waktu yang dibutuhkan cukup lama (sekitar 1-2 menit). Tampilan utama aplikasi Username Finder ini dapat dilihat pada Gambar 4.1, serta kode program tampilan utama dapat dilihat pada Gambar 4.2.



Gambar 4.1 Tampilan utama aplikasi Username Finder

```

import tkinter as Username_Finder
import time
import datetime
import webbrowser
import namechk
import namechk2
import namecheckr
import knowem
import namecheckup
import instantusernamesearch
import checkusernames
import thatsthem
import usersearch
import whatsmyname
import amazonusernames
import amazonwishlists
import githubuser
import tinderusernames
import keybase
import mitpgpkeyserver

def on_mousewheel(event):
    shift = (event.state & 0x1) != 0
    scroll = -1 if event.delta > 0 else 1
    if (shift):
        mainmenu.xview_scroll(scroll, "units")
    else:
        mainmenu.yview_scroll(scroll, "units")

root = Username_Finder.Tk()
w,h = root.winfo_screenwidth(),root.winfo_screenheight()
root.geometry("%dx%d+0+0" % (w,h))
root.title("Username Finder")
frame = Username_Finder.Frame(root,width=3840,height=2160)

```

```

frame.pack(expand=True,fill='both')
mainmenu = Username_Finder.Canvas(frame, width = 3840, height = 2160,
scrollregion=(0,0,3840,3840), bg="honeydew")
vbar = Username_Finder.Scrollbar(frame,orient='vertical')
vbar.pack(side='right',fill='y')
vbar.config(command=mainmenu.yview)
hbar = Username_Finder.Scrollbar(frame,orient='horizontal')
hbar.pack(side='bottom',fill='x')
hbar.config(command=mainmenu.xview)
mainmenu.bind_all("<MouseWheel>",on_mousewheel)
mainmenu.config(width=3840,height=2160)
mainmenu.config(yscrollcommand=vbar.set)
mainmenu.config(xscrollcommand=hbar.set)
mainmenu.pack(side='left',expand=True,fill='both')

username_finder_label = Username_Finder.Label(root, text="Username Finder")
username_finder_label.config(font=("Script",35,"bold"),fg="springgreen",bg=
"navy")
mainmenu.create_window(650,100,window=username_finder_label)

username_finder_label2 = Username_Finder.Label(root, text="Username Search
Application uses 15 OSINT Framework")
username_finder_label2.config(font=("Helvetica",16),bg="honeydew")
mainmenu.create_window(650,160,window=username_finder_label2)

textfieldcari = Username_Finder.Entry(root)
mainmenu.create_window(600, 250, window=textfieldcari)

def callldomain(url_domain):
    webbrowser.open_new(url_domain)

def callback(url_username):
    webbrowser.open_new(url_username)

def searchUsername():
    username = textfieldcari.get()

    label = Username_Finder.Label(root, text="Searching Username : " +
username + " ...", bg="honeydew")
    mainmenu.create_window(650, 280, window=label)

    waktu_pencarian = datetime.datetime.now().strftime("%Y-%m-%d %H.%M.%S")
    log = open("logs\\" + waktu_pencarian + ".txt","w")
    log.close()

    summary_domain = open("summary_domain.txt","w")
    summary_domain.close()

    summary_username = open("summary_username.txt","w")
    summary_username.close()

    # Osint Framework no.1 (Ready)
    namechk.namechk(username,waktu_pencarian)

    # Osint Framework no.2 (Ready)
    namechk2.namechk2(username,waktu_pencarian)

    # Osint Framework no.3 (Ready)
    knowem.knowem(username,waktu_pencarian)

```

```

# Osint Framework no.4 (Ready)
namecheckr.namecheckr(username,waktu_pencarian)

# Osint Framework no.5 (Ready)
usersearch.usersearch(username,waktu_pencarian)

# Osint Framework no.6 moved due network bug

# Osint Framework no.7 (Ready)
thatsthem.thatsthem(username,waktu_pencarian)

# Osint Framework no.8 (Ready)
checkusernames.checkusernames(username,waktu_pencarian)

# Osint Framework no.9 (Ready)
namecheckup.namecheckup(username,waktu_pencarian)

# Osint Framework no.10 (Ready)
instantusernamesearch.instantusernamesearch(username,waktu_pencarian)

# Osint Framework no.11 (Ready)
amazonusernames.amazonusernames(username,waktu_pencarian)

# Osint Framework no.12 (Detected as Robot)
#amazonwishlists.amazonwishlists(username)

# Osint Framework no.13 (Ready)
githubuser.githubuser(username,waktu_pencarian)

# Osint Framework no.14 (Ready)
tinderusernames.tinderusernames(username,waktu_pencarian)

# Osint Framework no.15 (Ready)
keybase.keybase(username,waktu_pencarian)

# Osint Framework no.16 (Ready)
mitpgpkeyserver.mitpgpkeyserver(username,waktu_pencarian)

# Osint Framework no.6 (Ready)
whatsmyname.whatsmyname(username,waktu_pencarian)

log = open("logs\\" + waktu_pencarian + ".txt","a")

title1 = Username_Finder.Label(root,text="Domain Names
Found",bg="honeydew")
title2 = Username_Finder.Label(root,text="Usernames
Found",bg="honeydew")
mainmenu.create_window(150,350,window=title1)
mainmenu.create_window(400,350,window=title2)

d = open("summary_domain.txt","r")
u = open("summary_username.txt","r")
read_domain = d.read().splitlines()
read_username = u.read().splitlines()

log.write("[ ----- Summary Domain Names Found ----- ]\n")
start = 400
for domain in read_domain:

```

```

        domain_label
Username_Finder.Label(root,text=domain,fg="blue",cursor="hand2",bg="honeyde
w")
        domain_label.pack
        domain_label.bind("<Button-1>",lambda e,index = domain :
calldomain(index))
        start = start + 20
        mainmenu.create_window(150,start>window=domain_label)
        log.write(domain + "\n")

        log.write("[ ----- Summary Usernames Found ----- ]\n")
        start = 400
        width_p = 300
        width_l = 500
        for user in read_username:
            platform = user.split(":")[0]
            platform = platform.strip()
            link = user.split(":")[1] + ":" + user.split(":")[2]
            link = link.replace(" ", "")
            username_label = Username_Finder.Label(root,text=platform + " :
",bg="honeydew")
            link_label
Username_Finder.Label(root,text=link,fg="blue",cursor="hand2",bg="honeydew"
)
            link_label.pack
            link_label.bind("<Button-1>",lambda e,index = link :
callback(index))
            start = start + 20
            mainmenu.create_window(width_p,start>window=username_label)
            mainmenu.create_window(width_l,start>window=link_label)
            log.write(platform + " : " + link + "\n")
            if(start >= 2400):
                start = 400
                width_p = width_p + 400
                width_l = width_l + 400

        log.close()

buttoncari = Username_Finder.Button(text='Search Username',
command=searchUsername, bg="blue", fg="white", relief="raised")
mainmenu.create_window(720, 250, window=buttoncari)

root.mainloop()

```

Gambar 4.2 Kode program tampilan utama aplikasi Username Finder

## 4.2 Implementasi Web Scraping

Teknik Web Scraping yang diimplementasikan pada aplikasi Username Finder dibagi menjadi 2 berdasarkan *library* yang digunakan, yaitu: requests dan selenium, 2 *library* tersebut digunakan untuk menjalankan aplikasi berbasis web kerangka kerja OSINT. Selain 2 *library* tersebut, diperlukan juga 1 *library* bernama BeautifulSoup untuk melakukan ekstraksi data dari hasil menjalankan aplikasi berbasis web kerangka kerja OSINT baik melalui requests maupun selenium.

#### 4.2.1 Web Scraping menggunakan requests

Proses web scraping menggunakan *library* requests cukup sederhana, dikarenakan *library* ini hanya membuka halaman web yang dituju lalu menyimpan kode HTML dari halaman web tersebut. *Library* requests tidak cocok digunakan apabila pencarian nama pengguna pada aplikasi berbasis web kerangka kerja OSINT menggunakan *javascript*. Contoh kode program untuk melakukan pengambilan halaman HTML menggunakan *library* requests ditunjukkan pada Gambar 4.3.

```
url = "https://www.gotinder.com/@" + username
results = requests.get(url)
content = results.content
```

Gambar 4.3 Kode program pengambilan halaman HTML menggunakan library requests

#### 4.2.2 Web Scraping menggunakan selenium

Proses web scraping menggunakan *library* selenium sedikit kompleks. Selain itu, selenium menggunakan *library* webdriver yang digunakan untuk menjalankan browser yang nantinya akan menjalankan proses *javascript* pada saat pencarian nama pengguna. Cara kerja web scraping dengan *library* selenium adalah dengan membuka *url* aplikasi kerangka kerja OSINT menggunakan *library* webdriver, lalu nama pengguna dicari menggunakan aplikasi kerangka kerja OSINT secara otomatis. Proses otomatisasi ini dilakukan menggunakan *library* selenium dengan cara memasukkan nama pengguna yang di-input melalui aplikasi Username Finder ke dalam aplikasi web kerangka kerja OSINT dengan cara menggunakan *id,name,class*, atau *xpath* dari tag `<input type="text">` yang digunakan untuk melakukan pencarian nama pengguna. Kemudian, tombol untuk melakukan pencarian akan diklik secara otomatis dengan cara menggunakan *library* selenium untuk mencari tombol *submit* dengan cara menggunakan *class* atau *attribute* type yang bernilai *submit* dari tag input. Setelah itu, pencarian nama pengguna berjalan dari *browser* yang telah ditentukan melalui *library* webdriver, kemudian hasil dari pencarian tersebut akan diambil kode HTML nya. Pada metode ini, terdapat jeda waktu beberapa detik (`time.sleep()`) setelah melakukan klik tombol *submit*. Hal ini bertujuan untuk memastikan proses *javascript* pencarian nama pengguna sudah selesai dilakukan. Contoh kode program untuk melakukan pencarian nama pengguna sekaligus pengambilan halaman HTML menggunakan *library* selenium ditunjukkan pada Gambar 4.4.

```
driver = webdriver.Firefox()

driver.get(url)
driver.find_element_by_id("q").send_keys(username)
driver.find_element_by_xpath("//button[@type='submit' and
@class='search-btn btn btn-block']").click()
time.sleep(30)
```

```

driver.find_element_by_xpath("//button[@type='submit' and
@class='search-btn btn btn-block']").click()
time.sleep(10)
content = driver.page_source
soup = BS(content, "html.parser")

```

Gambar 4.4 Kode program pengambilan halaman HTML menggunakan library selenium

### 4.3 Ekstraksi Data dari hasil Web Scraping

Hasil dari web scraping tidak langsung ditampilkan, akan tetapi dilakukan ekstraksi data terlebih dahulu untuk memilah kode HTML yang berisikan hasil pencarian nama pengguna untuk ditampilkan pada aplikasi ini. Proses ekstraksi data ini menggunakan *library* BeautifulSoup. Cara kerja *library* BeautifulSoup ini adalah dengan mencari *tag* dari halaman HTML yang sudah di web scraping, seperti <div> atau <span>. Dalam pencarian *tag* juga dapat menspesifikasikan pencarian, yaitu dengan cara menambahkan *attribute* seperti *class, id, name*, dll ketika hasil pencarian memiliki nilai *attribute* khusus yang terdapat pada *tag* halaman HTML yang menampung hasil pencarian nama pengguna tersebut. Salah satu contoh kode program yang melakukan ekstraksi data hasil pencarian nama pengguna ditunjukkan pada Gambar 4.5.

```

for found in soup.find_all("div", {"class" : "box domain
unavailable"}):
    dn = username + found.text
    results_domainnames.append(dn)

for found in soup.find_all("div", {"class" : "box service unavailable"}):
    ul = found.find("a")
    results_usernames.append(found.text + " : ")
    results_link_usernames.append(ul["href"])

```

Gambar 4.5 Kode program pengeksraksian data dari hasil web scraping

## 4.4 Implementasi Pengembangan Source Code Aplikasi

Tidak semua aplikasi kerangka kerja OSINT berbasis web. Terdapat 2 aplikasi kerangka kerja OSINT yang bukan merupakan aplikasi berbasis web, yaitu: Namechk.sh dan WhatsMyName. Kedua aplikasi ini merupakan aplikasi *open source* yang kode programnya dapat diunduh dari github.

### 4.4.1 Namechk.sh

Aplikasi Namechk.sh adalah sebuah *bash script* yang dapat dijalankan menggunakan *terminal* pada sistem operasi linux. Untuk menjalankan aplikasi Namechk.sh pada sistem operasi windows, diperlukan sebuah aplikasi salah satunya adalah git bash. Agar aplikasi ini dapat dijalankan melalui aplikasi Username Finder, maka aplikasi Namechk.sh perlu dilakukan sedikit pengembangan pada kode aplikasi tersebut.

```

...
echo "" > namechk.txt

```

```

...
found(){
    if [ $verify == 0 ];then
        profile=$(cat verify | sed -e 's/[{}]"|'/'/g' | cut -d , -f 5 | cut -
d ":" -f 2,3)
        echo -e "\e-----\e"
        echo -e " [+] Username found on:    ${service[$cont]}" >> namechk.txt
        echo -e " [+] Profile url:         \e$profile\e"
        export ${service[$cont]}
    fi
}

```

Gambar 4.6 Pengembangan kode program aplikasi Namechk.sh

Pengembangan kode program aplikasi Namechk.sh bertujuan agar aplikasi tersebut dapat menangkap input dari aplikasi Username Finder dan mengeluarkan hasil output ke aplikasi Username Finder. Berdasarkan Gambar 4.6, terdapat penambahan kode yaitu: `echo "" > namechk.txt` yang bertujuan untuk membuat berkas txt kosong dengan nama namechk.txt. Dan juga terdapat perubahan kode yaitu dibagian: `echo -e " [+] Username found on: ${service[$cont]}" >> namechk.txt` yang bertujuan untuk memindahkan *output* hasil pencarian nama pengguna dari yang tadinya ditampilkan pada *terminal* atau *git bash* menjadi disimpan ke dalam berkas namechk.txt.

```

p = subprocess.Popen(["namechk.sh",username,"-fu"],shell=True) #Windows
#p = subprocess.Popen(["./namechk.sh",username,"-fu"]) #Linux
p.wait()

```

Gambar 4.7 Fungsi untuk memanggil aplikasi Namechk.sh

Setelah melakukan pengembangan dan perubahan pada kode aplikasi Namechk.sh, aplikasi Namechk.sh dipanggil dengan menggunakan *library* `subprocess.Popen` seperti yang tertera pada Gambar 4.7. Parameter yang digunakan untuk memanggil aplikasi Namechk.sh adalah `-fu`, yang bertujuan untuk menemukan nama pengguna yang dicari. Setelah proses pencarian selesai, aplikasi Username Finder akan menampilkan hasil pencarian melalui berkas txt yang berisikan hasil pencarian pada aplikasi Namechk.sh.

#### 4.4.2 WhatsMyName

Aplikasi WhatsMyName adalah aplikasi dikembangkan menggunakan bahasa *python* dan dijalankan menggunakan *command prompt* atau *terminal* pada sistem operasi linux. Agar aplikasi ini dapat dijalankan melalui aplikasi Username Finder, maka aplikasi WhatsMyName diubah menjadi sebuah *class*. Setelah aplikasi WhatsMyName diubah menjadi *class*, aplikasi WhatsMyName dapat dipanggil oleh aplikasi Username Finder.

```

def search(user_name):

```

Gambar 4.8 Mengubah aplikasi WhatsMyName menjadi sebuah class

Pengembangan kode program aplikasi WhatsMyName bertujuan agar aplikasi tersebut dapat menangkap *input* dari aplikasi Username Finder dan mengeluarkan hasil *output* ke aplikasi Username Finder. Berdasarkan Gambar 4.8, aplikasi WhatsMyName tersebut diubah menjadi sebuah *class* sehingga aplikasi tersebut dapat dipanggil oleh aplikasi Username Finder.

```

web_accounts_list_checker.search(username)
    f = open("result_whatsmyname.txt", "r")
    title = tk.Label(whatsmyname, text="Usernames found from whatsmyname for:
" + username)
    whatsmynamemenu.create_window(500, 50, window=title)
    content = f.read().splitlines()
    start = 80
    for l in content:
        found_label = tk.Label(whatsmyname, text=l[:17]) #Windows
        #found_label = tk.Label(whatsmyname, text=l[5:23]) #Linux
        link = l[18:] #Windows
        #link = l[23:-4] #Linux
        link_label = tk.Label(whatsmyname, text=link, fg="blue", cursor="hand2")
    tk.Label(whatsmyname, text=link, fg="blue", cursor="hand2")
        link_label.pack
        link_label.bind("<Button-1>", lambda e, url = link : callback(url))
        start = start + 20
        whatsmynamemenu.create_window(400, start, window=found_label)
        whatsmynamemenu.create_window(600, start, window=link_label)

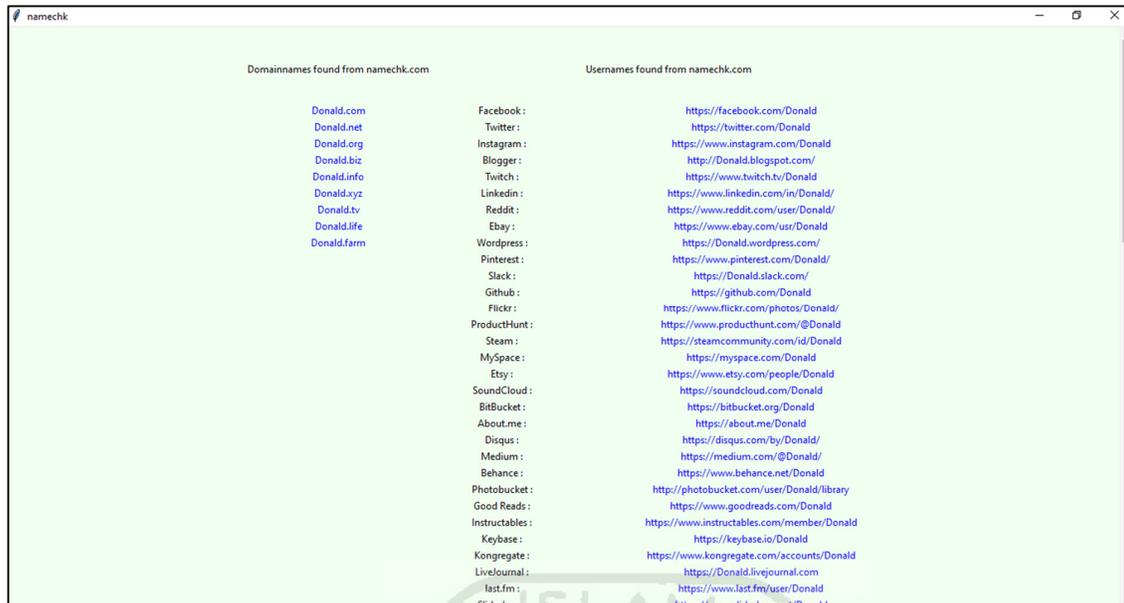
```

Gambar 4.9 Kode program untuk memanggil aplikasi WhatsMyName

Berdasarkan Gambar 4.9, setelah aplikasi WhatsMyName diubah menjadi *class*, aplikasi WhatsMyName dipanggil dengan parameter *input* yang berupa nama pengguna yang dicari, kemudian hasil dari pencarian disimpan kedalam berkas *txt*. Setelah itu, isi dari berkas *txt* diekstrak dan ditampilkan ke dalam aplikasi Username Finder.

#### 4.5 Menampilkan Data hasil Ekstraksi

Setelah proses integrasi selesai dilakukan dan data sudah berhasil diekstrak. Langkah selanjutnya adalah menampilkan data. Aplikasi “Username Finder” ini akan menampilkan data dengan tampilan *Graphical User Interface* (GUI). Proses menampilkan data pada aplikasi ini adalah dengan cara membuat jendela baru pada setiap aplikasi kerangka kerja OSINT yang diintegrasikan. Contoh tampilan data hasil pencarian nama pengguna melalui namechk ditunjukkan pada Gambar 4.10, serta kode program untuk tampilan data tersebut ditunjukkan pada Gambar 4.11.



Gambar 4.10 Contoh tampilan hasil pencarian nama pengguna

```

start = 80
for i in range(len(results_domainnames)):
    label_result_domainnames = tk.Label(namechk, text=results_domainnames[i], fg="blue", cursor="hand2")
    label_result_domainnames.pack
    label_result_domainnames.bind("<Button-1>", lambda e, index = i : calldomain(index))
    start = start + 20
namechkmenu.create_window(400, start, window=label_result_domainnames)

start = 80
label_result_link_usernames = []
for j in range(len(results_usernames)):
    label_result_usernames = tk.Label(namechk, text=results_usernames[j])
    label_result_link_usernames.append(tk.Label(namechk, text=results_link_usernames[j], fg="blue", cursor="hand2"))
    label_result_link_usernames[j].pack
    label_result_link_usernames[j].bind("<Button-1>", lambda e, index = j : callback(index))
    start = start + 20
    namechkmenu.create_window(600, start, window=label_result_usernames)
namechkmenu.create_window(900, start, window=label_result_link_usernames[j])

```

Gambar 4.11 Kode program untuk menampilkan data ke tampilan GUI

Untuk aplikasi kerangka kerja OSINT yang diintegrasikan melalui metode pengembangan *source code* aplikasi, hasil pencarian akan ditampilkan juga dengan cara membuat jendela baru. Proses penampilan data pada metode ini adalah dengan cara

mengekstrak *file* txt data hasil pencarian nama pengguna dan menampilkan data tersebut ke bentuk tampilan *Graphical User Interface* (GUI). Contoh tampilan data hasil pencarian nama pengguna melalui *namechk.sh* ditunjukkan pada Gambar 4.12, serta contoh tampilan data hasil pencarian nama pengguna melalui *WhatsMyName* ditunjukkan pada Gambar 4.13.

```

namechk2
Username found from namechk.sh for: Donald

[+] Username found on: Facebook
[+] Username found on: YouTube
[-] Username found on: Twitter
[-] Username found on: Instagram
[-] Username found on: Blogger
[+] Username found on: GooglePlus
[-] Username found on: Twitch
[-] Username found on: Reddit
[+] Username found on: Ebay
[-] Username found on: Wordpress
[+] Username found on: Pinterest
[+] Username found on: Yelp
[+] Username found on: Slack
[+] Username found on: Github
[-] Username found on: Basecamp
[-] Username found on: Tumblr
[+] Username found on: Flickr
[-] Username found on: Pandora
[+] Username found on: ProductHunt
[-] Username found on: Steam
[-] Username found on: MySpace
[+] Username found on: Foursquare
[+] Username found on: OkCupid
[-] Username found on: Vimeo
[-] Username found on: UStream
[+] Username found on: Etsy
[-] Username found on: SoundCloud
[-] Username found on: BitBucket
[-] Username found on: Meetup
[+] Username found on: Fotolog
[-] Username found on: Vine
[-] Username found on: PayPal
[+] Username found on: Dribbble
[+] Username found on: Imgur
[-] Username found on: Tracky
[+] Username found on: Flipboard
[+] Username found on: Vk
[+] Username found on: kik
[+] Username found on: Codecademy
[-] Username found on: Roblox
[+] Username found on: Greavatar
[+] Username found on: Trip
[-] Username found on: Pastebin
[+] Username found on: Coinbase
[-] Username found on: BlipFM
[-] Username found on: Wikipedia
[-] Username found on: Ello
[-] Username found on: StreamMe
[-] Username found on: IFTTT
[-] Username found on: WebCredit
[-] Username found on: CodeMentor
[-] Username found on: Soupio
[+] Username found on: Fiverr
[+] Username found on: Trakt
[-] Username found on: Hacknews
[-] Username found on: five0px
[-] Username found on: Spotify
[-] Username found on: POF
  
```

Gambar 4.12 Tampilan hasil pencarian nama pengguna melalui Namechk.sh

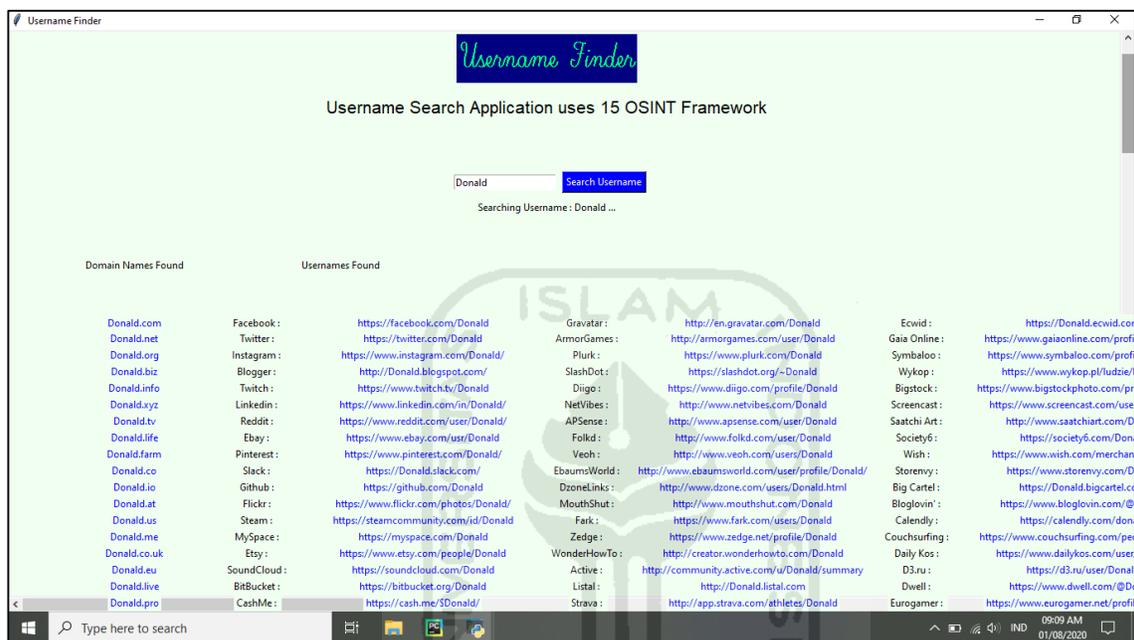
```

whatsmyname
Usernames found from whatsmyname for: Donald

[-] Found user at https://500px.com/Donald
[-] Found user at https://about.me/Donald
[-] Found user at https://ask.fm/Donald
[-] Found user at http://www.authorstream.com/Donald/
[-] Found user at https://bandcamp.com/Donald
[-] Found user at https://launchpad37signals.com/session/profile?username=Donald
[-] Found user at https://www.behance.net/Donald
[-] Found user at http://blip.fm/Donald
[-] Found user at http://Donald.blogspot.com
[-] Found user at http://api.bodybuilding.com/api-proxy/bbc/get?slug=Donald
[-] Found user at https://www.dailymotion.com/Donald
[-] Found user at https://www.diigo.com/interact_api/load_profile_info?name=Donald
[-] Found user at https://disqus.com/by/Donald/
[-] Found user at https://www.etsy.com/user/Donald
[-] Found user at https://www.etsy.com/people/Donald
[-] Found user at https://www.fanpop.com/fans/Donald
[-] Found user at https://www.flickr.com/photos/Donald/
[-] Found user at https://flipboard.com/@Donald
[-] Found user at https://www.freesound.org/people/Donald/
[-] Found user at https://friendfinder.com/profile/Donald
[-] Found user at https://www.furaffinity.net/user/Donald
[-] Found user at https://www.gamespot.com/profile/Donald/
[-] Found user at https://connect.garmin.com/modern/profile/Donald
[-] Found user at https://www.geocaching.com/profile/?u=Donald
[-] Found user at https://giphy.com/channel/Donald
[-] Found user at https://api.github.com/users/Donald
[-] Found user at https://gitlab.com/Donald
[-] Found user at https://news.ycombinator.com/user?id=Donald
[-] Found user at https://rfttt.com/p/Donald
[-] Found user at https://imageshack.com/user/Donald
[-] Found user at https://inkbunny.net/Donald
  
```

Gambar 4.13 Tampilan hasil pencarian nama pengguna melalui WhatsMyName

Untuk memudahkan pengguna dalam mencari nama pengguna, aplikasi Username Finder juga menampilkan hasil pencarian secara keseluruhan. Hasil pencarian secara keseluruhan ini bersifat unik atau tidak menampilkan duplikasi hasil pencarian pada platform yang sama. Hasil pencarian secara keseluruhan ditampilkan pada tampilan utama aplikasi Username Finder seperti yang ditunjukkan pada Gambar 4.14.



Gambar 4.14 Tampilan hasil pencarian nama pengguna secara keseluruhan

#### 4.6 Log hasil Pencarian

Setelah melakukan pencarian nama pengguna, Aplikasi Username Finder akan menyimpan *log* hasil pencarian yang pernah dilakukan ke dalam *file* txt. *File log* akan dihasilkan setiap melakukan pencarian nama pengguna. Berdasarkan kode program pada Gambar 4.15, ketika melakukan pencarian nama pengguna, aplikasi Username Finder akan mengambil tanggal dan waktu saat pencarian dilakukan. Tanggal dan waktu yang diambil, akan dimasukkan ke dalam variabel dan dijadikan *parameter* ketika memanggil fungsi pencarian nama pengguna dari setiap aplikasi kerangka kerja OSINT. Proses pembuatan *file log* adalah dengan cara membuka *file log* dengan mode *write* atau membuat *file* baru. Apabila *file* sudah ada, maka *file* yang sudah ada akan ditimpa.

```
waktu_pencarian = datetime.datetime.now().strftime("%Y-%m-%d %H.%M.%S")
log = open("logs\\" + waktu_pencarian + ".txt", "w")
```

```
log.close()
```

Gambar 4.15 Kode program untuk membuat file log berdasarkan tanggal dan waktu

Di setiap fungsi pencarian nama pengguna dari setiap aplikasi kerangka kerja OSINT, selain ditampilkan ke layar, hasil pencarian juga disimpan ke dalam *file log* dengan nama *file* sesuai dengan tanggal dan waktu ketika pencarian mulai dilakukan. Proses penyimpanan dilakukan dengan cara membuka *file log* dengan mode *append* atau menuliskan baris pada *file* yang sudah ada. Pada Gambar 4.16 ditunjukkan contoh kode program yang berfungsi untuk membuat *file log* hasil pencarian nama pengguna melalui namechk.

```
log = open("logs\\" + waktu_pencarian + ".txt","a")
log.write("[ ----- Namechk ----- ]\n")
log.write("--- Domain Names ---\n")

start = 80
for i in range(len(results_domainnames)):
    label_result_domainnames = tk.Label(namechk, text=results_domainnames[i], fg="blue", cursor="hand2", bg="honeydew")
    log.write(results_domainnames[i] + "\n")
    label_result_domainnames.pack
    label_result_domainnames.bind("<Button-1>", lambda e, index = i : calldomain(index))
    start = start + 20

namechkmenu.create_window(400, start, window=label_result_domainnames)

log.write("--- Usernames ---\n")

start = 80
label_result_link_usernames = []
for j in range(len(results_usernames)):
    label_result_usernames = tk.Label(namechk, text=results_usernames[j], bg="honeydew")
    label_result_link_usernames.append(tk.Label(namechk, text=results_link_usernames[j], fg="blue", cursor="hand2", bg="honeydew"))
    log.write(results_usernames[j] + results_link_usernames[j] + "\n")
    label_result_link_usernames[j].pack
    label_result_link_usernames[j].bind("<Button-1>", lambda e, index = j : callback(index))
    start = start + 20
    namechkmenu.create_window(600, start, window=label_result_usernames)

namechkmenu.create_window(900, start, window=label_result_link_usernames[j])
log.close()
```

Gambar 4.16 Contoh kode program yang berfungsi untuk menyimpan ke file log

Isi dari *file log* akan kurang lebih sama dengan hasil pencarian yang ditampilkan ke layar ketika pencarian telah selesai dilakukan. Setiap *log* hasil pencarian akan dipisahkan berdasarkan aplikasi kerangka kerja OSINT yang digunakan untuk mencari nama pengguna yang dicari. Hasil pencarian nama pengguna secara keseluruhan (yang tidak mengandung duplikasi hasil) juga disimpan ke dalam *file log* sehingga pengguna lebih mudah menemukan nama pengguna yang pernah dicari sebelumnya seperti yang dicontohkan pada Gambar 4.17.

```

...
[ ----- Summary Domain Names Found ----- ]
Patrick.com
Patrick.net
Patrick.org
Patrick.co
Patrick.biz
Patrick.io
Patrick.at
Patrick.us
Patrick.me
Patrick.co.uk
Patrick.eu
Patrick.info
Patrick.xyz
Patrick.ca
Patrick.be
Patrick.tv
Patrick.la
Patrick.life
Patrick.ch
Patrick.ms
Patrick.in
Patrick.club
Patrick.tech
Patrick.site
Patrick.online
Patrick.space
Patrick.digital
Patrick.today
Patrick.pro
Patrick.shop
Patrick.cc
Patrick.vip
Patrick.de
Patrick.ltd
Patrick.tk
Patrick.nl
Patrick.tw
Patrick.pl
Patrick.work
Patrick.tools
[ ----- Summary Usernames Found ----- ]
Facebook : https://facebook.com/Patrick
YouTube : https://www.youtube.com/Patrick
Twitter : https://twitter.com/Patrick
Instagram : https://www.instagram.com/Patrick/

```



```

Blogger : http://Patrick.blogspot.com/
Twitch : https://www.twitch.tv/Patrick
Linkedin : https://www.linkedin.com/in/Patrick/
Shopify : https://Patrick.myshopify.com
Reddit : https://www.reddit.com/user/Patrick/
Ebay : https://www.ebay.com/usr/Patrick
Pinterest : https://www.pinterest.com/Patrick/
Slack : https://Patrick.slack.com/
Github : https://github.com/Patrick
Flickr : https://www.flickr.com/photos/Patrick/
ProductHunt : https://www.producthunt.com/@Patrick
Steam : https://steamcommunity.com/id/Patrick
MySpace : https://myspace.com/Patrick
Foursquare : https://foursquare.com/Patrick
Etsy : https://www.etsy.com/people/Patrick
SoundCloud : https://soundcloud.com/Patrick
BitBucket : https://bitbucket.org/Patrick
CashMe : https://cash.me/$Patrick/
DailyMotion : https://www.dailymotion.com/Patrick
About.me : https://about.me/Patrick
Disqus : https://disqus.com/by/Patrick/
Medium : https://medium.com/@Patrick/
Behance : https://www.behance.net/Patrick
Photobucket : http://photobucket.com/user/Patrick/library
Coderwall : https://coderwall.com/Patrick
Good Reads : https://www.goodreads.com/Patrick
Instructables : https://www.instructables.com/member/Patrick
Keybase : https://keybase.io/Patrick
Kongregate : https://www.kongregate.com/accounts/Patrick
LiveJournal : https://Patrick.livejournal.com
last.fm : https://www.last.fm/user/Patrick
Slideshare : https://www.slideshare.net/Patrick
Dribbble : https://dribbble.com/Patrick
Flipboard : https://flipboard.com/@Patrick
Vk : https://vk.com/Patrick
Kik : http://kik.me/Patrick
Gravatar : https://en.gravatar.com/Patrick
Trip : https://www.skyscanner.com/trip/user/Patrick
Pastebin : https://pastebin.com/u/Patrick
blip.fm : https://blip.fm/Patrick
...

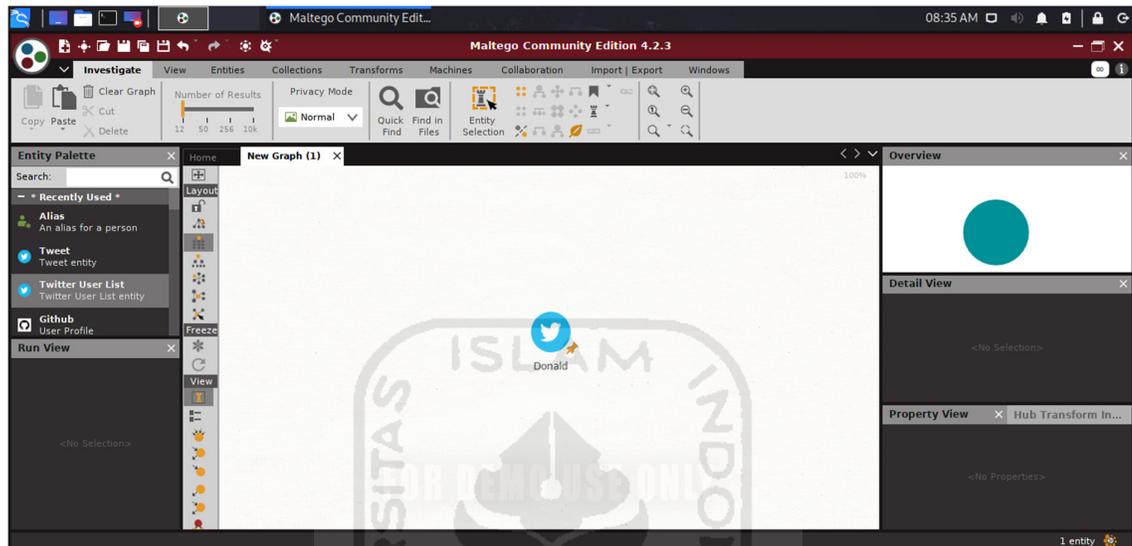
```

Gambar 4.17 Contoh isi dari file log hasil pencarian nama pengguna: Donald

#### 4.7 Perbandingan Aplikasi Username Finder dengan Maltego

Maltego adalah sebuah aplikasi OSINT yang tidak hanya menyediakan platform untuk mengekstrak data saja, tetapi juga dapat menampilkan data dengan format yang mudah dimengerti dan mudah dianalisis (Chauhan & Panda, 2015b). Maltego memungkinkan pengguna untuk melakukan transformasi, transformasi adalah suatu fungsi yang memetakan satu entitas ke entitas lainnya. Entitas adalah sesuatu yang mungkin ingin diselidiki oleh pengguna. Contohnya adalah domain, website, alamat e-mail, individu, nama server, lokasi dan nomor telepon (Bradbury, 2011). Maltego terdiri dari 2 versi, yaitu: versi *commercial* (berbayar) dan versi *community* (gratis). Untuk menggunakan Maltego, pengguna

diharuskan membuat akun terlebih dahulu. Pada aplikasi Maltego, terdapat banyak item hub transformasi yang berasal dari penyedia transformasi pihak ketiga sehingga memerlukan API Key dari penyedia tersebut agar dapat menggunakan transformasinya. Aplikasi Maltego sudah ter-*install* secara otomatis pada sistem operasi Kali Linux, contoh tampilannya dapat dilihat pada Gambar 4.18.



Gambar 4.18 Tampilan Aplikasi Maltego pada Kali Linux

Terdapat beberapa perbedaan antara Aplikasi Username Finder dengan Maltego, perbedaan tersebut dapat dilihat pada Tabel 4.1.

Tabel 4.1 Perbandingan Aplikasi Username Finder dengan Maltego

Perbedaan	Username Finder	Maltego
Tampilan	Sederhana	Rumit
Memerlukan login / registrasi	Tidak	Ya
Memerlukan API Key	Tidak	Ya
Teknik Pencarian	Sekaligus pada beberapa platform	Terpisah pada setiap platform
Fungsi Penggunaan	Hanya mencari nama pengguna	Dapat juga digunakan untuk melakukan pengintaian ketika <i>penetration testing</i> .

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa:

1. Metode yang digunakan untuk mengintegrasikan 16 perangkat pada OSINT menjadi 1 aplikasi adalah: Web Scraping dan Pengembangan *Source Code* Aplikasi. Namun, Aplikasi Username Finder belum dapat mengintegrasikan 1 aplikasi kerangka kerja OSINT yaitu Amazon Wishlists dikarenakan terdapat captcha pada aplikasi tersebut.
2. Aplikasi yang dikembangkan sudah dapat mencari dan mendapatkan nama pengguna dari sumber terbuka. Walaupun tampilannya masih cukup sederhana.

#### 5.2 Saran

Dari Aplikasi Username Finder yang merupakan penggabungan dari 16 aplikasi kerangka kerja OSINT, masih terdapat beberapa kekurangan dan kelemahan yang dapat dikembangkan lebih lanjut. Kekurangan dan kelemahan Aplikasi Username Finder yaitu:

- a. Masih belum dapat mengintegrasikan 1 aplikasi kerangka kerja OSINT, yaitu: Amazon Wishlists dikarenakan terdapat captcha pada aplikasi tersebut.
- b. Belum dapat menampilkan foto pengguna dari nama pengguna yang dicari pada bagian pencarian melalui tinderusernames. Hanya dapat menampilkan link fotonya saja
- c. Tampilan hasil pencarian masih terlalu sederhana. Belum ada logo sosial media, grafik, tabel, dll. Dan juga hasil pencarian dengan membuat jendela atau *window* baru pada setiap aplikasi kerangka kerja OSINT yang diintegrasikan dirasa kurang efektif.
- d. Proses pencarian masih lambat, untuk pengembangan yang akan datang, dibutuhkan algoritma atau *library* yang dapat melakukan pencarian nama pengguna dengan lebih cepat. Hal ini dikarenakan aplikasi Username Finder ini melakukan pencarian secara satu-persatu pada setiap aplikasi.

## DAFTAR PUSTAKA

- Bradbury, D. (2011). In plain view : open source intelligence. *Computer Fraud & Security*, 2011(4), 5–9. [https://doi.org/10.1016/S1361-3723\(11\)70039-2](https://doi.org/10.1016/S1361-3723(11)70039-2)
- Chauhan, S., & Panda, N. K. (2015a). Open Source Intelligence and Advanced Social Media Search. In *Hacking Web Intelligence* (pp. 15–32). <https://doi.org/10.1016/B978-0-12-801867-5.00002-1>
- Chauhan, S., & Panda, N. K. (2015b). *OSINT Tools and Techniques*. <https://doi.org/10.1016/B978-0-12-801867-5.00006-9>
- Clement, J. (2020). Number of social network users worldwide from 2010 to 2023(in billions). Retrieved July 2, 2020, from <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/>
- David, A. (2017). SEEING THE FOREST THROUGH THE TREES: THE VALUE OF OSINT FOR THE U.S. NAVY. Retrieved July 5, 2020, from <http://cimsec.org/seeing-forest-trees-value-osint-u-s-navy/34897>
- Glassman, M., & Kang, M. J. (2012). Computers in Human Behavior Intelligence in the internet age : The emergence and evolution of Open Source Intelligence ( OSINT ). *Computers in Human Behavior*, 28(2), 673–682. <https://doi.org/10.1016/j.chb.2011.11.014>
- Hribar, G., Podbregar, I., & Ivanuša, T. (2014). *International Journal of Intelligence and OSINT: A “ Grey Zone ”? OSINT: A “ Grey Zone ”?* (July), 37–41. <https://doi.org/10.1080/08850607.2014.900295>
- Indonesia Digital 2019 : Media Sosial. (2019). Retrieved December 2, 2019, from <https://websindo.com/indonesia-digital-2019-media-sosial/>
- Koops, B.-J., Hoepman, J.-H., & Leenes, R. (2013). Open-source intelligence and privacy by design. *Computer Law & Security Review*, 29(6), 676–688.
- Nation, D. (2019). What Is Social Media? Retrieved December 2, 2019, from <https://www.lifewire.com/what-is-social-media-explaining-the-big-trend-3486616>
- Pellet, H., Shiaeles, S., & Stavrou, S. (2019). Localising social network users and profiling their movement. *Computers and Security*, 81, 49–57. <https://doi.org/10.1016/j.cose.2018.10.009>
- Pouchard, L. C., Dobson, J. M., & Trien, J. P. (2006). *A Framework for the Systematic Collection of Open Source Intelligence*. 102–107.
- Pratomo, Y. (2019). APJII: Jumlah Pengguna Internet di Indonesia Tembus 171 Juta Jiwa. Retrieved December 2, 2019, from <https://tekno.kompas.com/read/2019/05/16/03260037/apjii-jumlah-pengguna-internet->

di-indonesia-tembus-171-juta-jiwa

Rajagopal, S. K. P. (2019). Web Scraping using Python. Retrieved July 5, 2020, from <https://dev.to/prsharankumar/web-scraping-using-python-2ip6>

SecurityTrails team. (2019). OSINT Framework: The Perfect Cybersecurity Intel Gathering Tool. Retrieved November 18, 2019, from <https://securitytrails.com/blog/osint-framework>

Syabab, A. (2019). Apa itu web scraping? Retrieved November 18, 2019, from <https://pesonainformatika.com/other-notes/apa-itu-web-scraping/>

Weir, G. R. S. (2016). The Limitations of Automating OSINT : Understanding the Question , Not the Answer. In *Automating Open Source Intelligence*. <https://doi.org/10.1016/B978-0-12-802916-9/00009-9>



LAMPIRAN

