

SKRIPSI

«JUDUL BAHASA INDONESIA»



Harry Senjaya Darmawan

NPM: 2017730067

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN

«tahun»

UNDERGRADUATE THESIS

«JUDUL BAHASA INGGRIS»



Harry Senjaya Darmawan

NPM: 2017730067

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY

«tahun»

LEMBAR PENGESAHAN

«JUDUL BAHASA INDONESIA»

Harry Senjaya Darmawan

NPM: 2017730067

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

«pembimbing utama/1»

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

«JUDUL BAHASA INDONESIA»

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» «tahun»



Harry Senjaya Darmawan
NPM: 2017730067

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	3
2.1 JSOUP	3
2.1.1 JSOUP	3
2.1.2 Connection	3
2.1.3 Response	4
2.1.4 Document	4
2.1.5 Elements	5
2.1.6 Element	5
2.2 Java Swing	5
DAFTAR REFERENSI	7
A KODE PROGRAM	9
B HASIL EKSPERIMEN	11

DAFTAR GAMBAR

B.1 Hasil 1	11
B.2 Hasil 2	11
B.3 Hasil 3	11
B.4 Hasil 4	11

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Setiap dosen wali memiliki data mengenai mahasiswa walinya. Namun, walaupun dosen wali memiliki data mengenai mahasiswa walinya, dosen wali juga perlu melakukan pemeriksaan data mahasiswa walinya, terutama data akademiknya secara berkala. Dengan berbagai kesibukan yang dialami oleh para dosen wali dan mahasiswa, ditambah dengan situasi Indonesia saat ini yang menyebabkan perkuliahan dilakukan secara daring, akan sangat sulit bagi dosen wali untuk menemui mahasiswa wali. Hal ini menyebabkan dosen wali kesulitan mengamati perkembangan mahasiswa walinya.

Maka dari itu, pada skripsi ini akan dibuat sebuah perangkat lunak yang berupa *screen saver* yang dapat menampilkan data akademik mahasiswa wali secara acak. Dengan menggunakan perangkat lunak tersebut, dosen wali dapat tetap mengamati perkembangan mahasiswa walinya, paling tidak secara akademik.

Dikarenakan terbimbing tidak memiliki akses ke SIAKAD untuk mengakses data mahasiswa wali, namun terbimbing memiliki akses ke Student Portal maka, terbimbing mensimulasikan dengan Student Portal, dan kemudian Pembimbing mengubah aksesnya ke SIAKAD. Pembimbing dan terbimbing menyepakati struktur kelas yang akan digunakan yaitu struktur kelas SIAModels yang tersedia pada Maven Public Repository [1].

Teknologi yang dapat dimanfaatkan untuk mengambil data mahasiswa yaitu *library* JSOUP. JSOUP dapat digunakan untuk melakukan *web scraping*, sehingga pengambilan data mahasiswa tidak memerlukan API (*Application Programming Interface*). Teknologi lainnya yang dapat dimanfaatkan yaitu Java Swing. Java Swing dapat digunakan untuk mengonversi aplikasi tersebut menjadi *screen saver*.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada skripsi ini adalah sebagai berikut:

- Bagaimana cara memanfaatkan JSOUP untuk mengambil data mahasiswa?
- Bagaimana cara memanfaatkan Java Swing untuk mengonversi aplikasi tersebut menjadi *screen saver*?

1.3 Tujuan

Tujuan yang ingin dicapai dari penulisan skripsi ini sebagai berikut:

- Memanfaatkan JSOUP untuk mengambil data mahasiswa.
- Memanfaatkan Java Swing untuk mengonversi aplikasi tersebut menjadi *screen saver*.

1.4 Batasan Masalah

Dikarenakan terbimbing tidak memiliki akses ke SIAKAD untuk mengakses data mahasiswa wali, namun terbimbing memiliki akses ke Student Portal maka, terbimbing mensimulasikan dengan Student Portal, dan kemudian Pembimbing mengubah aksesnya ke SIAKAD.

1.5 Metodologi

Langkah-langkah yang akan dilakukan dalam melakukan penelitian ini yaitu:

1. Melakukan studi mengenai JSOUP.
2. Melakukan studi mengenai cara mengonversi aplikasi menjadi *screen saver*.
3. Mempelajari struktur kelas SIAModels.
4. Menganalisis IF Student Portal dan Student Portal UNPAR.
5. Merancang struktur kelas aplikasi.
6. Mendesain antarmuka aplikasi.
7. Mengimplementasikan JSOUP untuk mengambil data mahasiswa.
8. Mengonversi aplikasi menjadi *screen saver* dengan menggunakan Java Swing.
9. Melakukan pengujian dan eksperimen.
10. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Dokumen dibagi ke dalam beberapa bab dengan sistematika pembahasan sebagai berikut:

- Bab 1. Pendahuluan, membahas tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metode penelitian dan sistematika pembahasan mengenai skripsi.
- Bab 2. Landasan Teori, membahas landasan dari teori-teori yang berhubungan serta mendukung penelitian, meliputi JSOUP, dan Java Swing.
- Bab 3. Analisis, menjelaskan tentang cara memanfaatkan JSOUP, dan cara memanfaatkan Java Swing.
- Bab 4. Perancangan, membahas perancangan antarmuka, diagram kelas beserta deskripsi kelas dan fungsinya.
- Bab 5. Implementasi dan pengujian, membahas hasil-hasil implementasi dan pengujian secara fungsional dan eksperimental.
- Bab 6. Kesimpulan dan saran, membahas kesimpulan yang diperoleh dari penelitian ini dan saran untuk pengembangan berikutnya.

BAB 2

LANDASAN TEORI

Bab Landasan Teori ini berisi teori-teori yang menjadi dasar penelitian ini, meliputi JSOUP, dan Java Swing.

2.1 JSOUP

Menurut Bo Zhao[2], *web scraping* juga dikenal sebagai ekstraksi atau pemanenan web, adalah teknik untuk mengekstrak data dari *World Wide Web* (WWW) dan menyimpannya ke sistem file atau basis data untuk diambil atau dianalisis. Salah satu teknologi yang dapat dimanfaatkan untuk melakukan *web scraping* yaitu *library* Java JSOUP. JSOUP adalah *library* Java untuk mengerjakan dokumen HTML yang menyediakan API yang baik untuk mengekstraksi, memanipulasi data, dan menyelesaikan pembersihan data awal menggunakan metode terbaik dari *Document Object Model* (DOM), *Cascading Style Sheets* (CSS), dan metode lain yang mirip dengan jQuery. JSOUP mengimplementasikan spesifikasi WHATWG HTML5, dan mem-parsing HTML ke DOM yang sama seperti yang dilakukan *browser* modern. Berikut adalah layanan utama yang tersedia di JSOUP [3]:

1. *Scrape* dan *parse* HTML dari URL, *file*, atau string.
2. Mencari dan ekstrak data menggunakan traversal DOM dan CSS *selector*.
3. Memanipulasi elemen HTML, atribut HTML, dan teks.
4. Membersihkan konten yang dikirim oleh pengguna yang menggunakan *safe white-lists* untuk mencegah serangan XSS.
5. Menghasilkan HTML yang rapi.

Subbab-subbab berikut menjelaskan beberapa kelas dari JSOUP.

2.1.1 JSOUP

Kelas ini merupakan inti untuk mengakses fungsi jsoup. Salah satu fungsi yang dimiliki kelas ini adalah sebagai berikut:

- **public static Connection connect(String url)**
Berfungsi untuk membuat koneksi baru dengan suatu situs web.
Parameter: URL situs web dengan protokol HTTP atau HTTPS.
Kembalian: koneksi dengan situs web.

2.1.2 Connection

Kelas ini merupakan *interface* yang menyediakan pengambilan data dari situs web. Beberapa fungsi yang dimiliki kelas ini adalah sebagai berikut:

- **Connection cookies(Map<String,String> cookies)**
Berfungsi untuk menambahkan *cookie*.
Parameter:
 - **cookies** Map dari *cookie*.**Kembalian:** koneksi yang sama tetapi sudah diubah.

- **Connection data(String key, String value)**
 Berfungsi untuk menambahkan parameter data yang bisa dikirim melalui metode HTTP GET atau POST.
Parameter:
 - **key** kunci data.
 - **value** nilai data.**Kembalian:** koneksi yang sama tetapi sudah diubah.
- **Connection method(Connection.Method method)**
 Berfungsi untuk mengatur metode permintaan HTTP, GET atau POST. Metode pengiriman secara *default* adalah GET.
Parameter:
 - **method** metode pengiriman permintaan HTTP.**Kembalian:** koneksi yang sama tetapi sudah diubah.
- **Connection timeout(int millis)**
 Berfungsi untuk mengatur batas waktu *request*. Batas waktu nol akan dianggap sebagai batas waktu yang tak terhingga.
Parameter:
 - **millis** batas waktu dalam milidetik.**Kembalian:** koneksi yang sama tetapi sudah diubah.
- **Connection validateTLSCertificates(boolean value)**
 Berfungsi untuk mengatur pemeriksaan sertifikat TLS untuk permintaan HTTPS. Nilai **true** untuk memeriksa dan nilai **false** untuk tidak memeriksa.
Parameter:
 - **value** status pemeriksaan sertifikat TLS.**Kembalian:** koneksi yang sama tetapi sudah diubah.
- **Connection.Response execute()**
 Berfungsi untuk mengirim permintaan HTTP.
Kembalian: objek Response.

2.1.3 Response

Kelas ini merepresentasikan permintaan HTTP. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **Map<String,String> cookies()**
Method ini berfungsi untuk mendapatkan seluruh *cookies*.
Kembalian: seluruh *cookies*.
- **Document parse()**
 Berfungsi untuk mengurai *body* jawaban menjadi dokumen.
Kembalian: koneksi yang sama tetapi sudah diubah.
- **String body()**
 Berfungsi untuk mendapatkan *body* jawaban dalam bentuk *string*.
Kembalian: *body* jawaban dalam bentuk *string*.

2.1.4 Document

Kelas ini merepresentasikan dokumen HTML. Salah satu *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Elements select(String cssQuery)**
Method ini diturunkan dari kelas Element, berfungsi untuk menemukan elemen HTML yang sesuai dengan kueri CSS.
Parameter:
 - **cssQuery** kueri CSS berupa CSS Selector.

Kembalian: elemen-elemen HTML yang sesuai dengan kueri CSS.

2.1.5 Elements

Kelas ini merepresentasikan kumpulan elemen HTML. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Elements select(String query)**
Berfungsi untuk menemukan elemen-elemen yang sesuai dalam *list* elemen.
Parameter:
 - **query** kueri CSS berupa CSS Selector.**Kembalian:** elemen-elemen yang sudah diseleksi sesuai kueri.
- **public String val()**
Berfungsi untuk mendapatkan nilai dari elemen pertama.
Kembalian: nilai elemen.
- **public String text()**
Method Berfungsi untuk mendapatkan kombinasi teks dari seluruh elemen yang sesuai.
Kembalian: seluruh teks dalam *string*.

2.1.6 Element

Kelas ini merepresentasikan sebuah elemen HTML yang berisikan *tag*, atribut, dan anak elemen. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Element child(int index)**
Berfungsi untuk mendapatkan anak elemen berdasarkan nomor indeks.
Parameter:
 - **index** nomor index.**Kembalian:** anak elemen.
- **public Element children()**
Berfungsi untuk mendapatkan seluruh anak elemen.
Kembalian: seluruh anak elemen.
- **public String className()**
Berfungsi untuk mendapatkan nama kelas elemen.
Kembalian: nama kelas elemen.
- **public String text()**
Berfungsi untuk mendapatkan teks dari elemen.
Kembalian: teks dalam *string*.

2.2 Java Swing

Java Swing adalah toolkit *graphical user interface* (GUI) generasi berikutnya yang dibuat oleh Sun Microsystems untuk memungkinkan pengembangan perusahaan di Java [4]. Pada Java, istilah *frame* digunakan untuk apa yang umumnya disebut '*window*'. Pada Swing, *frame* tersebut direalisasikan oleh kelas `JFrame`. *Frame* adalah area persegi panjang dengan *title bar* di atasnya. *Title bar* juga berisi tombol untuk menutup, memaksimalkan atau membuat ikon bingkai. Di bawah *title bar* adalah area di mana komponen grafis lebih lanjut dapat ditambahkan [5]. `JFrame` tersebut akan dimanfaatkan untuk membuat tampilan dari aplikasi *screen saver*.

DAFTAR REFERENSI

- [1] Alfadian, P. (2020) SIA Models. [Online; diakses 19-Oktober-2020].
- [2] Zhao, B. (2017) Web Scraping.
- [3] Cokrowibowo, S., Nur, N., dan Irmayanti (2020) Web page extraction and classification using jsoup and naïve bayes. *IOP Conference Series: Materials Science and Engineering*, **875**, 012089.
- [4] Loy, M., Eckstein, R., Wood, D., Elliott, J., dan Cole, B. (2002) *Java Swing, Second Edition*, 2 edition. O'Reilly Media, Inc.
- [5] Fischer, P. (2005) *Introduction to Graphical User Interfaces with Java Swing*. Addison-Wesley.

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Kode A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```


LAMPIRAN B

HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4