

BAB IV

HASIL DAN PEMBAHASAN

1.1 Batasan Masalah

Implementasi klasifikasi judul TA otomatis ini bertujuan untuk menentukan konsentrasi yang sesuai dengan judul yang diserahkan mahasiswa Teknik Informatika UII secara otomatis. Berdasarkan hasil analisis dan perancangan sistem yang telah dilakukan, maka dilakukan implementasi sistem dengan algoritma *Naïve Bayes Classifier* ke dalam bentuk program dengan menggunakan bahasa pemrograman PHP dan *framework* CodeIgniter.

1.1.1 Perangkat Lunak Pengembangan

Perangkat lunak yang digunakan adalah perangkat lunak dengan spesifikasi berikut:

1. Windows 10 Professional 64bit digunakan sebagai sistem operasi laptop yang digunakan.
2. *Framework* CodeIgniter sebagai pembangun *website* yang menggunakan bahasa pemrograman PHP.
3. *Library* PHP-*Classifier* sebagai mesin learning untuk proses pengklasifikasian.
4. MySQL digunakan sebagai basisdata penyimpanan data yang diperlukan.
5. Mozilla Firefox digunakan sebagai standart *browser* untuk menjalankan program.
6. Visual Studio *Code* digunakan sebagai *text editor* pembangun sistem.
7. Github digunakan untuk kerja kolaborasi dengan pengembang KPTA lainnya.

1.1.2 Perangkat Keras Pengembangan

Perangkat lunak yang digunakan dengan spesifikasi berikut:

1. Laptop Lenovo Intel Core i3-3120M CPU 2.50GHz.
2. RAM 4GB.
3. *Harddisk* dengan kapasitas 500GB.
4. Sistem *input* seperti *mouse* dan *keyboard*.

1.2 Implementasi Sistem

Implementasi sistem merupakan tahap menggabungkan sistem klasifikasi dengan sistem KPTA FTI UII sehingga siap digunakan.

1.2.1 Pengumpulan Data

Data yang digunakan pada penelitian ini sebagian besar didapat dari data judul TA Jurusan Teknik Informatika UII sedangkan untuk data pendukung didapat dari *Google Scholar*. Setelah data sudah terkumpul kemudian dibagi menjadi dua yaitu data *training* dan data *testing* dengan perbandingan 2/3 untuk data *training* dan 1/3 untuk data *testing*.

1.2.2 Text Preprocessing

Text Preprocessing merupakan tahap untuk mengubah bentuk data yang belum terstruktur menjadi data yang terstruktur (Imbar et al., 2014). Tujuan dari *text preprocessing* adalah mempersiapkan teks menjadi data yang akan mengalami pengolahan pada tahapan berikutnya (Andini, 2013). Tahapan proses ini adalah sebagai berikut:

Case Folding dan Remove Punctuation

Pada tahap ini semua huruf dalam satu kalimat akan diubah menjadi huruf kecil dan menghapus tanda baca. Implementasi *code* dapat dilihat pada Gambar 4.1.

```

$str = $this->input->post('judul');
    $str = str_replace("'", "", $str); $str = str_replace("-", "", $str);
    $str = str_replace("&", "", $str); $str = str_replace("(", "", $str);
    $str = str_replace("=", "", $str);
    $str = str_replace(".", "", $str);
    $str = str_replace(", ", "", $str);
    $str = str_replace(":", "", $str);
    $str = str_replace("; ", "", $str);
    $str = str_replace("!", "", $str);
    $str = str_replace("?", "", $str);
    $str = str_replace("/", "", $str);
    $str = str_replace("1", " ", $str);
    $str = str_replace("2", " ", $str);
    $str = str_replace("3", " ", $str);
    $str = str_replace("4", " ", $str);
    $str = str_replace("5", " ", $str);
    $str = str_replace("6", " ", $str);
    $str = str_replace("7", " ", $str);
    $str = str_replace("8", " ", $str);
    $str = str_replace("9", " ", $str);
    $str = str_replace("0", " ", $str);
    $judul=strtolower($str);

```

Gambar 4.1 *Code Case Folding dan Remove Punctuation***Tokenizing**

Pada tahap ini dilakukan pemotongan *string input* berdasarkan tiap kata dalam kalimat. Spasi digunakan untuk memisahkan antar kata tersebut. Implementasi *code* dapat dilihat pada Gambar 4.2.

```
$token=(explode(" ", $judul));
```

Gambar 4.2 *Code Tokenizing***Stopword Removal**

Pada tahap ini, dilakukan penghapusan kata yang ada pada *database stopwords*. Implementasi *code* dapat dilihat pada Gambar 4.3.

```
$jumlah_token=count($token);
    for ($st=0; $st < $jumlah_token ; $st++) {
        $stoplist=$this->ci->m_stoplist-
>get_count_by_kata_tidak_singkatan($token[$st]);
        if ($stoplist>0) {
            $judul = str_replace($token[$st], "", $judul);
        }
    }
    $judul=$this->trim_string($judul);
```

Gambar 4.3 *Code Stopword Removal***1.2.3 Pembobotan Kata(TF-IDF)**

Tahap ini pembobotan kata digunakan untuk mendapatkan suatu topik atau kata kunci dari kumpulan judul TA. Nilai bobot suatu kata (*term*) menyatakan kepentingan bobot tersebut dalam merepresentasikan judul. Data untuk pembobotan kata diperoleh dari jumlah kata muncul dalam semua dokumen. Implementasi *code* dapat dilihat pada Gambar 4.4.

```
$count = isset($this->tokens[$token][$subject]) ?
$this->tokens[$token][$subject] : 0;
$scores[$subject] += log( ($count + 1) / ($subject_data['count_tokens'] + $this-
>total_tokens));}$scores[$subject]=$subject_data['prior_value'] +
$scores[$subject];
    $total_score += $scores[$subject];
```

Gambar 4.4 *Code* Pembobotan Kata

1.2.4 Mengubah Judul Menjadi *Title Case*

Tahap ini melakukan perubahan pada judul yang dimasukan mahasiswa dengan mengubah format judul menjadi *title case*. Implementasi *code* dapat dilihat pada Gambar 4.5.

```

$title_case=$this->remove_punctuation_tc($judul);
    $token_tc=(explode(" ", $title_case));
    $jumlah_token_tc=count($token_tc);
    $judul_title_case="";
    for ($stc=0; $stc < $jumlah_token_tc ; $stc++) {
        $stoplist_tc          =          $this->ci->m_stoplist-
>get_count_by_kata_jenis($token_tc[$stc], KATA_SAMBUNG);
        if ($stoplist_tc<=0) {
            $huruf_depan = substr($token_tc[$stc], 0, 1);
            $huruf_depan=strtoupper($huruf_depan);
            $kata = substr($token_tc[$stc], 1);

            $judul_title_case=$judul_title_case.$huruf_depan.$kata." ";
        }
        else{
            $judul_title_case=$judul_title_case.$token_tc[$stc]."
";
        }
    }
    for ($stc=0; $stc < $jumlah_token_tc ; $stc++) {
        $stoplist_tc2          =          $this->ci->m_stoplist-
>get_count_by_kata_jenis($token_tc[$stc], KATA_SINGKATAN);
        if ($stoplist_tc2>0) {
            $huruf_depan = substr($token_tc[$stc], 0, 1);
            $huruf_depan=strtoupper($huruf_depan);
            $kata = substr($token_tc[$stc], 1);
            $singkatan=strtoupper($huruf_depan.$kata);
            $judul_title_case          =
str_replace($huruf_depan.$kata,$singkatan,$judul_title_case);
        }
    }
    $judul_title_case=$this->trim_string($judul_title_case);

```

Gambar 4.5 *Code Title Case*

1.2.5 Perbandingan Kemiripan Teks Skripsi

Tahap ini melakukan perbandingan data proposal dan arsip TA pada sistem untuk dihitung kemiripannya. Implementasi *code* dapat dilihat pada Gambar 4.6.

```

$string1=$this->remove_punctuation($string1);
$string2=$this->remove_punctuation($string2);
$string1 = strtolower(strip_tags($string1));
$string2 = strtolower(strip_tags($string2));
$arr_string1=(explode(" ", $string1));
$arr_string2=(explode(" ", $string2));
$vec1 = array_count_values($arr_string1);
$vec2 = array_count_values($arr_string2);
$hasil = $this->_dotProduct($vec1, $vec2) / ($this->_absVector($vec1) * $this->_absVector($vec2));
$hasil=$hasil*100;
$persentase=round($hasil,2);

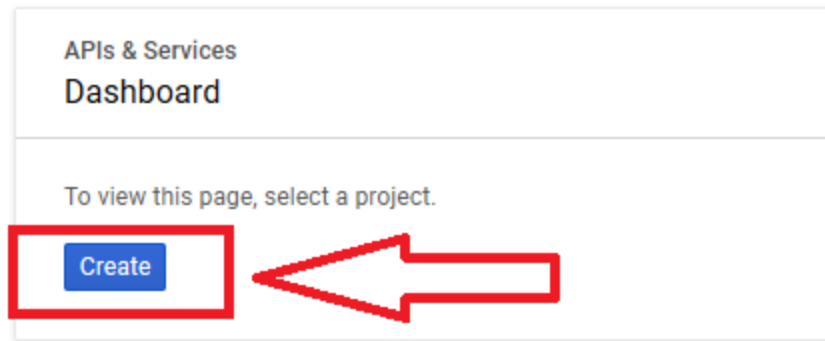
```

Gambar 4.6 *Code Similar text*

1.2.6 Konfigurasi Single Sign On Google (SSO)

Daftar Google Developer

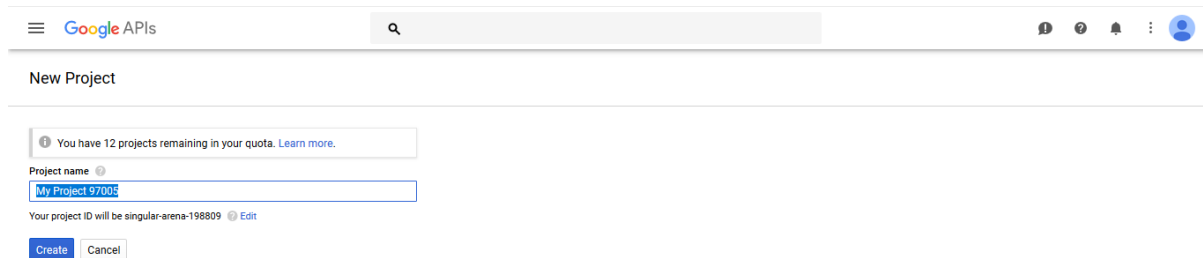
Pertama buka halaman Google *developer* <https://console.developers.google.com>, pastikan email yang tertaut adalah email untuk mendaftarkan id SSO Google. Setelah terbuka halaman *dashboard* Google *developer* pilih *create* tampilan seperti Gambar 4.7.



Gambar 4. 7 *Create Project SSO*

Memberi Nama *Project*

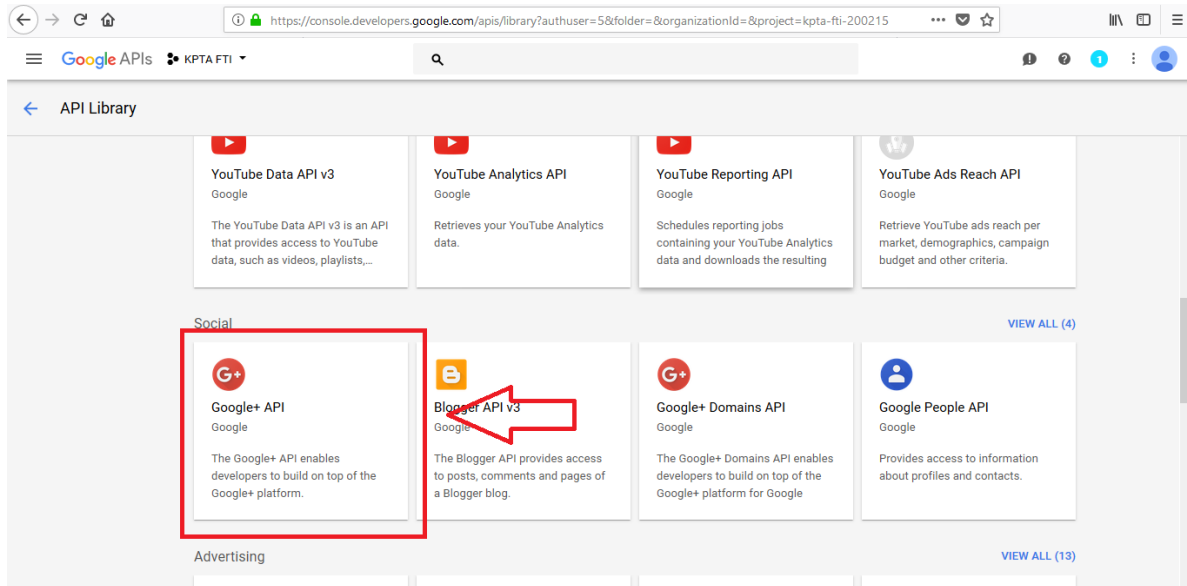
Setelah melakukan langkah pada Gambar 4.7 kemudian memberi nama untuk *project* SSO dengan “KPTA FTI UII”. Tampilan halaman seperti Gambar 4.8.



Gambar 4. 8 Memberi Nama *Project*

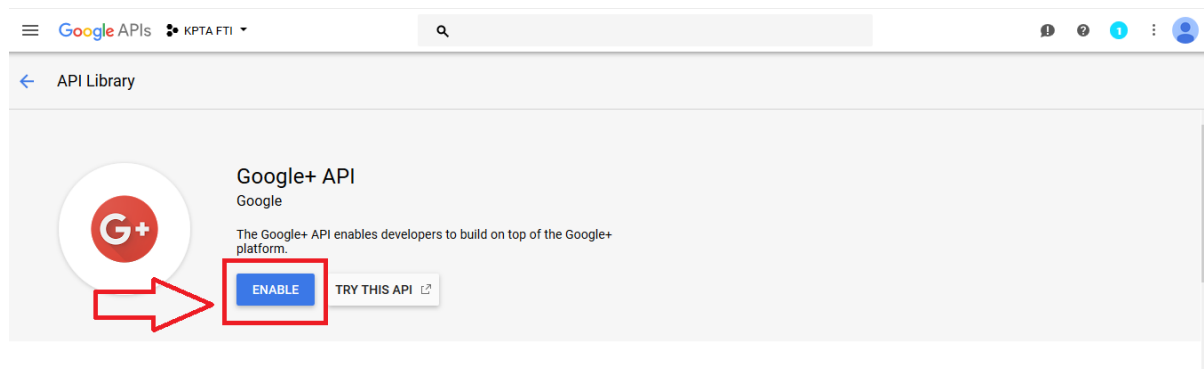
Install *Library G+ API*

Setelah melakukan langkah pada Gambar 4.8 kemudian akan kembali ke halaman *dashboard* Google *developer*. Selanjutnya pilih menu *API Library* dan setelah halaman termuat cari *library* Google+ API. Tampilan halaman seperti Gambar 4.9.



Gambar 4. 9 Pilih *Library* G+ API

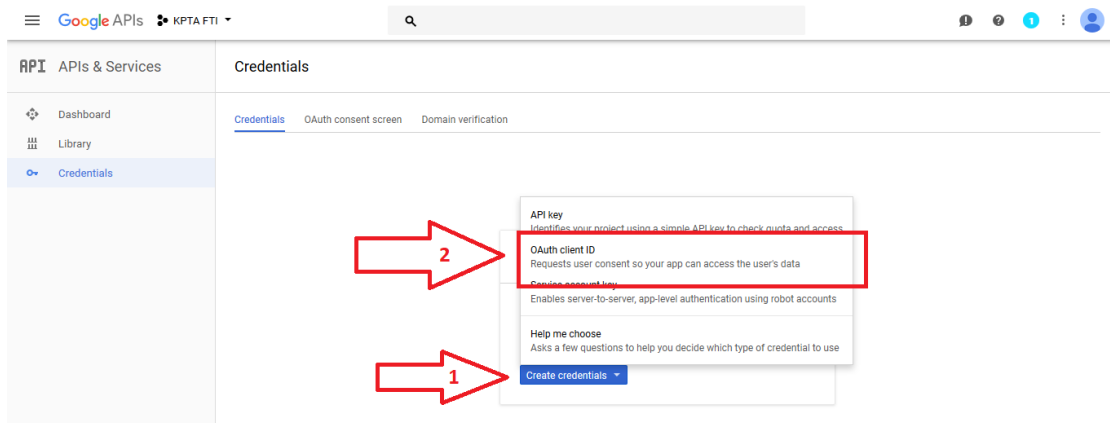
Setelah memilih Google+ API, akan langsung di halaman aktivasi *library*, setelah halaman termuat lalu pilih *enable* untuk mengaktifkan Google+ API. Tampilan halaman seperti Gambar 4.10.



Gambar 4. 10 Install *Library* G+ API

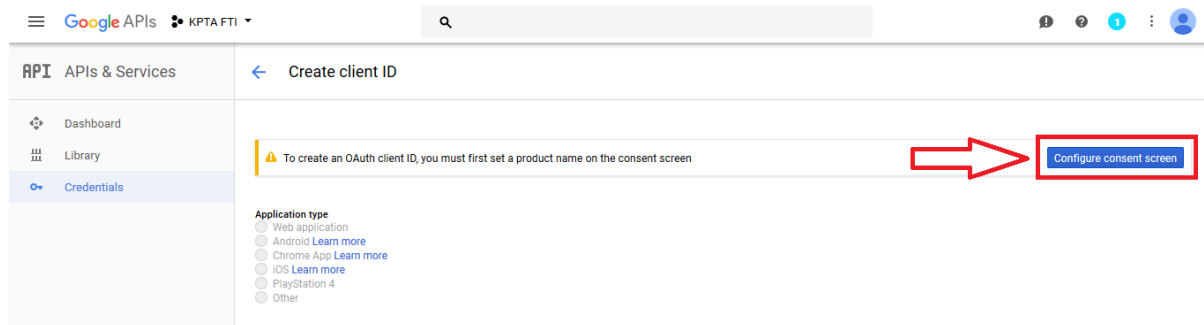
Buat *Credentials*

Setelah aktivasi *library* kemudian membuat *Credentials* pilih menu *Credentials*, kemudian pilih *create credentials* dan pilih *Oauth client ID*. Tampilan seperti Gambar 4.11.



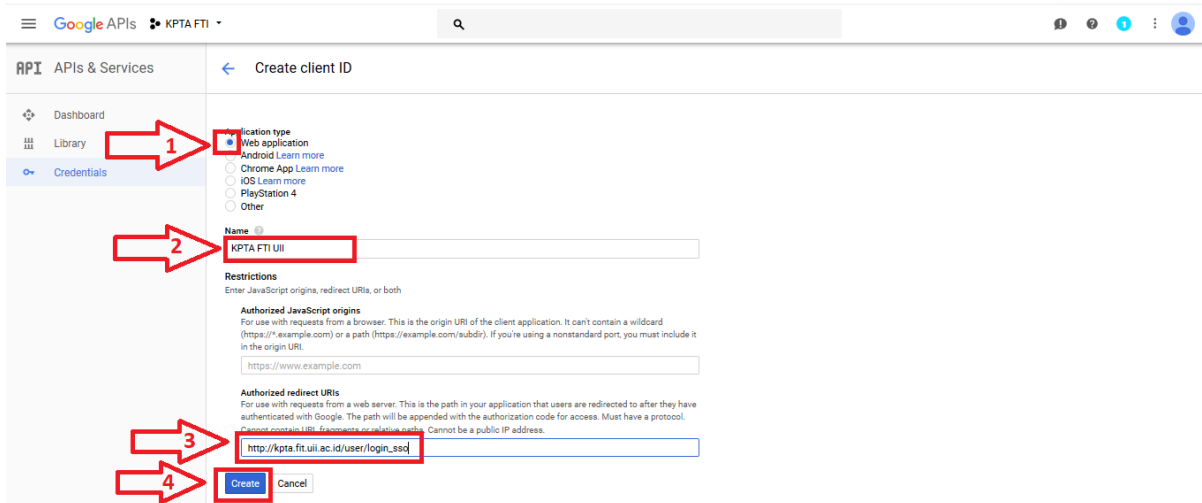
Gambar 4. 11 Buat *Credentials*

Setelah memilih OAuth *client ID* kemudian pilih *Configure Consent Screen* untuk melakukan konfigurasi. Tampilan seperti Gambar 4.12.



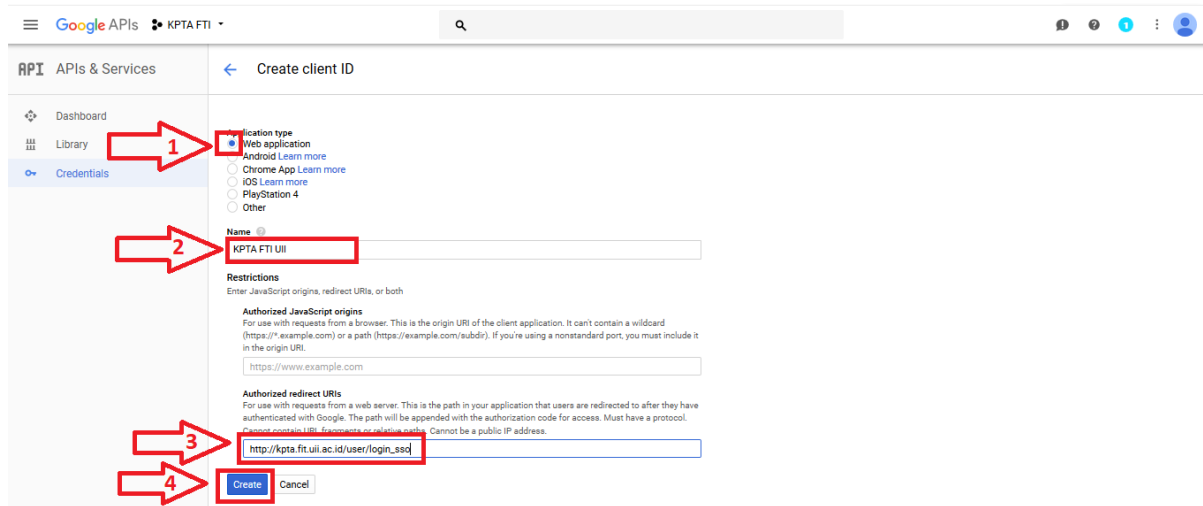
Gambar 4. 12 Pilih *Configure Consent Screen*

Setelah memilih *Configure Consent Screen* akan muncul halaman konfigurasi. Pastikan email yang tertaut adalah email yang sesuai kemudian isi bagian nama saja selebihnya biarkan kosong, lalu pilih *save*. Tampilan seperti Gambar 4. 13



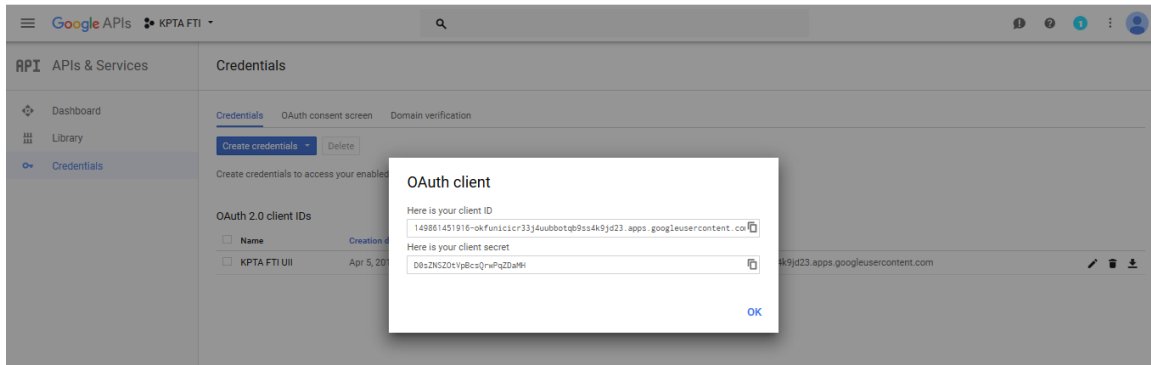
Gambar 4. 13 *Configure Consent Screen*

Setelah melakukan *Configure Consent Screen* kemudian pilih *web application*, kemudian menginputkan nama dan url untuk melakukan *callback*. Selanjutnya pilih *create*. Tampilan seperti gambar 4.14.



Gambar 4. 14 *Buat Credentials Web Application*

Setelah membuat *Credentials* akan muncul *Client ID* dan *Secret ID*, kemudian ID tersebut di implementasikan di *code* sistem KPTA.



Gambar 4. 15 *Client ID & Secret ID OAuth*

Konfigurasi *Client ID & Secret OAuth*

Tahap ini untuk melakukan konfigurasi *Client ID & Secret ID OAuth* pada sistem. Implementasi *code* dapat dilihat pasda Gambar 4.16.

```

$client = new Google_Client();
    $client-
>setClientId('1498614519160kfunicicr33j4uubbotqb9ss4k9jd23.apps.googleuserco
ntent.com');
$client->setClientSecret("D0sZNSZ0tVpBcsQrwPqZDaMH");
$client->setRedirectUri("kpta.fit.uui.ac.id/user/login_sso");
$client->setScopes(array(
    "https://www.googleapis.com/auth/userinfo.email",
    "https://www.googleapis.com/auth/userinfo.profile",
    "https://www.googleapis.com/auth/plus.me"
));

```

Gambar 4. 16 Implementasi *Client ID & Secret ID OAuth*

1.2.7 Login Single Sign On (SSO) Gmail UII

Tahap ini adalah melakukan pengecekan apakah email yang dipakai adalah email UII atau bukan, jika email termasuk UII maka akan dicek lagi apakah email termasuk email mahasiswa, dosen, karyawan FTI UII.

```

if (isset($_SESSION['access_token']) && $_SESSION['access_token']){
    $profile = $_SESSION['access_profile'];
    $cek_email=$profile['emails']['0']['value'];
    $user_iii=substr($cek_email,-9);
    $panjang_email=strlen($cek_email);
    if ($user_iii=="iii.ac.id") {
        if ($panjang_email==27) {
            $username=substr($cek_email,0,8);
            $result = $this->m_user->check_usersso($username);

            if (isset($result)) {
                $this->load->model("m_menu");
                $user = $this->m_user->get_by_id($result->id_user, $result->id_role);

                $user->login = md5(true);
                $user->menu = $this->m_menu->select($result->id_role);

                $this->session->set_userdata(md5("simson"), $user);

                redirect("kerja/mahasiswa");
            } else {
                $this->session->set_flashdata("message",
                "<div class='alert alert-danger alert-dismissible mg-b-sm'>\n"
                . "<button type='button' class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>\n"
                . "Error:<br/>Email Tidak Terdaftar</div>");

                redirect("user/login");
            }
        }
    }
} else {
    $this->session->set_flashdata("message",
    "<div class='alert alert-danger alert-dismissible mg-b-sm'>\n"

```

```

        ."<button                type='button'
class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>\n"
        . "Error:<br/>Email                Tidak
Terdaftar</div>");
        redirect("user/login");
    }}}

```

Gambar 4. 17 Login Single Sign On Gmail UII

1.2.8 Training Data

Data *training* yang sudah dilakukan tahap *text preprocessing* dan telah diklasifikasikan secara manual disimpan kedalam *database*. Setelah itu data yang sudah diklasifikasikan secara manual dibuat model probabilitiknya yang nantinya akan disimpan kembali ke dalam *database* yang akan digunakan pada tahap *testing* sebagai model klasifikasi judul TA. Pada penelitian ini PHP-*Classifier* dijadikan *library* dalam pembuatan sistem klasifikasi judul TA. PHP-*Classifier* adalah *machine learning library* untuk bahasa pemrograman PHP. *Library* ini berfungsi untuk proses klasifikasi dan evaluasi. Penelitian ini memanfaatkan metode *Naïve Bayes Classifier*, metode ini didasarkan pada teorema *Bayes* dan tidak memerlukan parameter apapun dalam konstruktor.

1.2.9 Testing Data

Tahap ini menggunakan data *testing* sebagai data masukan yang tentu saja sudah melalui tahap *text preprocessing*. Tahap *testing* juga memanfaatkan PHP-*Classifier* sebagai *library*.

1.3 Hasil Implementasi Sistem

Hasil implementasi ini membahas tentang klasifikasi judul TA otomatis dan alur sistem yang telah dirumuskan di bab sebelumnya.

1.3.1 Implementasi Halaman Form Daftar TA Awal

Halaman *form* daftar TA awal merupakan halaman *form* untuk memasukan judul TA. Setelah mahasiswa memilih tampilkan rekomendasi pada halaman *form* daftar TA awal ini akan melakukan klasifikasi judul yang dimasukan dengan menggunakan algoritma *Naïve Bayes Classification*, hasil perhitungan akan ditampilkan pada halaman *form* daftar TA rekomendasi konsentrasi. Implementasi halaman *form* daftar TA awal dapat dilihat pada Gambar 4.18.

Gambar 4.18 Halaman *Form* Daftar TA awal

1.3.2 Implementasi Halaman *Form* Daftar TA Rekomendasi Konsentrasi

Halaman *form* daftar TA rekomendasi konsentrasi merupakan halaman *form* untuk menampilkan tiga rekomendasi dengan nilai tertinggi berdasarkan proses perhitungan dengan algoritma *Naïve Bayes Classification* dari judul TA yang diserahkan. Pada implementasi ini juga terdapat proses mengubah judul menjadi *Title Case*. Kemudian mahasiswa wajib memilih salah satu dari tiga rekomendasi tersebut. Implementasi halaman *form* daftar TA rekomendasi awal dapat dilihat pada Gambar 4.19.

Gambar 4.19 Halaman *Form* Daftar TA Rekomendasi Konsentrasi

1.3.3 Implementasi Halaman Kemiripan Proposal

Halaman kemiripan proposal merupakan halaman *form* untuk melakukan persetujuan proposal dan pada halaman ini terdapat insdikator kemiripan proposal dengan data arsip proposal pada sistem. Implementasi halaman dokumen similaritas dapat dilihat pada Gambar 4.20.

No	Judul TA	Kemiripan
1	APLIKASI GAME TABEL PERIODIK UNSUR KIMIA IPA SMA	49.77 %
2	Game Edukasi Berbasis Android untuk Pengenalan Bahasa Jepang bagi Anak-Anak	48.1 %
3	PENGEMBANGAN APLIKASI FLASHCARD UNTUK PENGENALAN KARAKTER WAYANG BERBASIS ANDROID	47 %
4	Game Puzzle Logika Untuk Memotivasi Belajar Matematika Anak	45.83 %
5	Digital Game Based Learning dengan Teknik	42.93 %

Gambar 4.20 Halaman Dokumen Smilaritas

1.4 Pengujian Sistem

1.4.1 Klasifikasi Judul

Pengujian sistem klasifikasi judul TA dilakukan untuk mengetahui tingkat akurasi klasifikasi judul TA yang dilakukan secara manual dengan klasifikasi judul TA yang dilakukan oleh sistem dengan menggunakan *Naïve Bayes Classifier*. Metode evaluasi klasifikasi yang digunakan dalam penelitian ini adalah *Holdout*. Dalam metode ini data dibagi menjadi dua kelompok data, yaitu data *training* dan data *testing*, secara acak. Secara khusus dua pertiga dari data dialokasikan dalam kelompok data *training*, dan sepertiga sisanya ke dalam kelompok data *testing*. Data *training* digunakan untuk memperoleh model, dan akurasinya diestimasi menggunakan data *testing*. Pengujian lain dilakukan dengan menggunakan *confusion matrix* yaitu sebuah matrik dari prediksi yang akan dibandingkan dengan kelas yang asli dari data masukan. Pengujian sistem menggunakan sepertiga dari jumlah seluruh data yang berjumlah Sebanyak 1.083 data judul skripsi digunakan dalam penelitian ini. Data judul skripsi kemudian dibagi menjadi dua

bagian yaitu 862 judul sebagai data *training* dan sebanyak 224 judul sebagai data testing. Dari ke-822 data training, masing-masing kategori judul memiliki jumlah data yang berbeda JKK 184 data, ITSC 177 data, RPL 78 data, Multimedia 203 data, SI 140 data, dan Medis 40 data. Dari model klasifikasi yang dihasilkan, kemudian diuji dengan menggunakan 224 data dimana SI 31 data, RPL 21 data, Multimedia 51 data, JKK 42 data, ITSC 42 data, Medis 37 data. Hasil pengujian yang diperoleh pada pengujian ini akan dijelaskan dalam Tabel 4.1.

Table 4.1 Hasil Evaluasi Klasifikasi Judul Tugas Akhir

		<i>Predicted Class</i>						<i>Precision</i>
		<i>SI</i>	<i>RPL</i>	<i>SC</i>	<i>JJK</i>	<i>MULMED</i>	<i>MEDIS</i>	
<i>Actual Class</i>	<i>SI</i>	28	0	3	1	0	0	87.5 %
	<i>RPL</i>	1	14	2	1	3	0	66.67%
	<i>SC</i>	0	0	39	1	2	0	92.86%
	<i>JJK</i>	1	0	4	35	2	0	83.33%
	<i>MULMED</i>	3	1	0	0	46	0	92 %
	<i>MEDIS</i>	2	0	9	2	2	22	59.46%
	<i>Recall</i>	80.0%	93.33%	68.43%	87.50%	83.64%	100%	

Setelah sistem melakukan klasifikasi, hasil perhitungan *precision* untuk Sistem Informasi sebesar 87,5 %, Rekayasa Perangkat Lunak sebesar 66.67%, Informatika Teori dan Sistem Cerdas sebesar 92,86%, Jaringan dan Keamanan Komputer sebesar 83.33%, Multimedia sebesar 92% dan Informatika Medis sebesar 59.46%. Hasil perhitungan *recall* untuk Sistem Informasi sebesar 80 %, Rekayasa Perangkat Lunak sebesar 93.33%, Informatika Teori dan Sistem Cerdas sebesar 68.43%, Jaringan dan Keamanan Komputer sebesar 87.50%, Multimedia sebesar 83.64% dan Informatika Medis sebesar 100%.

Dari hasil pengujian table 4.1 hasil *precision* rendah pada konsentrasi Rekayasa Perangkat Lunak dan Informatika Medis. Rendahnya *precision* disebabkan karena beberapa kata kunci pada judul *testing* lebih mengarah ke konsentrasi lain dikarenakan masih kurangnya data judul dan data *training* sedangkan pada Informatika Medis kurangnya *precision* disebabkan adanya kemiripan judul dengan Informatika Teori dan Sistem Cerdas serta Sistem Informasi sementara untuk data *training* kedua konsentrasi tersebut lebih besar sehingga sistem mengarahkannya kedua

konsentrasi tersebut. beberapa contoh kemiripan kata kunci antara lain sistem pakar, sistem pendukung keputusan, dan sistem informasi.

Nilai *recall* tertinggi terdapat di Informatika Medis ini dikarenakan tidak ada judul dari konsentrasi lain yang salah terdeteksi sebagai informatika medis sehingga nilai *recall* Informatika Medis sangat tinggi yaitu 100%. Nilai *recall* terendah didapat oleh Informatika Teori dan Sistem Cerdas meskipun nilai *precision* cukup tinggi tapi tidak dengan nilai *recall*-nya ini disebabkan karena judul konsentrasi lain masih terdeteksi sebagai Informatika Teori dan Sistem Cerdas sehingga membuat nilai *recall*-nya rendah.

Rerata perhitungan *precision* pada klasifikasi Judul TA adalah 80,30% hasil ini menunjukkan bahwa sistem sudah bisa memberikan rekomendasi konsentrasi judul TA dengan ketepatan yang tinggi. Rerata perhitungan *recall* pada klasifikasi Judul TA adalah 85,48% hasil ini menunjukkan sistem sudah berhasil menemukan kembali informasi judul TA yang salah diklasifikasikan.

1.4.2 Pengujian Kemiripan Proposal

Metode evaluasi kemiripan proposal adalah dengan cara mencocokkan perhitungan manual dengan perhitungan sistem , berikut contoh data yang diuji:

- a. *Test* 1 data 1 sebanyak 5 kata dengan data 2 sebanyak 5 kata dengan kata mirip 0 kata
- b. *Test* 2 data 1 sebanyak 4 kata dengan data 2 sebanyak 5 kata dengan kata mirip 3 kata
- c. *Test* 3 data 1 sebanyak 5 kata dengan data 2 sebanyak 4 kata dengan kata mirip 2 kata
- d. *Test* 4 data 1 sebanyak 15 kata dengan data 2 sebanyak 15 kata dengan kata mirip 7 kata
- e. *Test* 5 data 1 sebanyak 10 kata dengan data 2 sebanyak 15 kata dengan kata mirip 8 kata
- f. *Test* 6 data 1 sebanyak 15 kata dengan data 2 sebanyak 10 kata dengan kata mirip 10 kata
- g. *Test* 7 data 1 sebanyak 25 kata dengan data 2 sebanyak 25 kata dengan kata mirip 21 kata
- h. *Test* 8 data 1 sebanyak 15 kata dengan data 2 sebanyak 25 kata dengan kata mirip 21 kata
- i. *Test* 9 data 1 sebanyak 25 kata dengan data 2 sebanyak 25 kata dengan kata mirip 25 kata

Perhitungan Kemiripan *Data Test*

Hasil pengujian yang diperoleh pada pengujian ini akan dijelaskan dalam Tabel 4.2.

Table 4. 2 Perhitungan Kemiripan

Data	Perhitungan	
	Manual	Sistem
<i>Test 1</i>	00.00 %	00.00 %
<i>Test 2</i>	33.81 %	33.81 %
<i>Test 3</i>	20.00 %	20.00 %
<i>Test 4</i>	43.84 %	43.84 %
<i>Test 5</i>	61.36 %	61.36 %
<i>Test 6</i>	76.70 %	76.70 %
<i>Test 7</i>	80.83 %	80.83 %
<i>Test 8</i>	69.57 %	69.57 %
<i>Test 9</i>	100 %	100 %

Bedasarkan hasil perhitungan kemiripan antara sistem dan manual hasilnya semuanya sama. Sehingga bisa disimpulkan untuk perhitungan kemiripan di KPTA UII sudah akurat.