



TUGAS AKHIR - SM141501

**PENINGKATAN KUALITAS HASIL PENCARIAN
TANPA MELANGGAR PRIVASI PENGGUNA PADA
PENELUSURAN WEB TERPERSONALISASI
MENGUNAKAN ALGORITMA *GREEDY***

**REGITA PRAMESTI NUR CAHYANI
NRP 0611144000027**

**Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2018**



FINAL PROJECT - SM141501

***QUALITY IMPROVEMENT OF SEARCH RESULT
WITHOUT VIOLATING USER PRIVACY ON THE
PERSONALIZED WEB SEARCH USING GREEDY
ALGORITHM***

**REGITA PRAMESTI NUR CAHYANI
NRP 0611144000027**

**Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

LEMBAR PENGESAHAN

PENINGKATAN KUALITAS HASIL Pencarian Tanpa Melanggar Privasi Pengguna pada Penelusuran Web Terpersonalisasi Menggunakan Algoritma Greedy

QUALITY IMPROVEMENT OF SEARCH RESULT WITHOUT VIOLATING USER PRIVACY ON THE PERSONALIZED WEB SEARCH USING GREEDY ALGORITHM

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
REGITA PRAMESTI NUR CAHYANI
NRP. 06111440000027

Menyetujui,
Dosen Pembimbing,


Dr. Budi Setiyono, S.Si, MT
NIP. 19720207 199702 1 001

Mengetahui,
Kepala Departemen Matematika
FMKSD ITS


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, 30 Juli 2018

PENINGKATAN KUALITAS HASIL PENCARIAN TANPA MELANGGAR PRIVASI PENGGUNA PADA PENELUSURAN WEB TERPERSONALISASI MENGUNAKAN ALGORITMA *GREEDY*

Nama Mahasiswa : Regita Pramesti Nur Cahyani
NRP : 0611144000027
Departemen : Matematika
Dosen Pembimbing : Dr. Budi Setiyono, S.Si, MT

Abstrak

Penelusuran Web Terpersonalisasi (PWS) memungkinkan pengguna layanan mesin pencari web (*Web Search Engine*) menggunakan personalisasi data untuk mendapatkan informasi yang relevan dengan preferensinya. Pada salah satu jenis PWS, yaitu PWS berbasis profil (*User Profile*), penyedia layanan harus mempertimbangkan dua efek yang kontradiktif selama proses pencarian. Pada saat proses pencarian, penyedia layanan akan berusaha memperbaiki kualitas pencarian dengan memanfaatkan utilitas personalisasi profil pengguna. Di sisi lain, jika privasi pengguna terkendali dan terekam oleh mesin pencari web (*Web Search Engine*), kemungkinan yang terjadi adalah data-data tersebut akan disalahgunakan oleh pihak ketiga. Kebutuhan penanganan masalah kontradiktif tersebut sangat diperlukan. Oleh karena itu, sistem yang dibuat memanfaatkan algoritma *Greedy* untuk melindungi privasi pengguna selama proses pencarian berlangsung tanpa mempengaruhi kualitas hasil pencarian. Hasil uji coba memperlihatkan bahwa dengan adanya perlindungan privasi pada penelusuran web terpersonalisasi yang dibuat, sistem mampu memberikan akurasi 75,89% dan 77,78% untuk setiap jenis *Keyword* pada relevansi keinginan pengguna terhadap hasil pencarian. Sistem telah memberikan hasil yang kualitasnya sama seperti mesin pencari web terpersonalisasi yang tidak memiliki perlindungan privasi.

Kata Kunci : PWS, *Web Search Engine*, algoritma *Greedy*

***QUALITY IMPROVEMENT OF SEARCH RESULT
WITHOUT VIOLATING USER PRIVACY ON THE
PERSONALIZED WEB SEARCH USING GREEDY
ALGORITHM***

Name of Student : Regita Pramesti Nur Cahyani
NRP : 0611144000027
Department : Mathematics
Supervisor : Dr. Budi Setiyono, S.Si, MT

Abstract

Personalized Web Search (PWS) enables Web search engine service users to use personalized data to get information relevant to their preference. In one of the PWS types, the profile-based PWS (User Profile), the service provider must consider two contradictory effects during the search process. During the search process, service providers will attempt to improve the quality of search by utilizing the user profile personalization utility. On the other hand, if the user's privacy is controlled and recorded by the web search engine (Web Search Engine), chances are that the data will be misused by a third party. The need for handling such contradictory problems is necessary. Therefore, the created system utilizes the Greedy algorithm to protect the user's privacy during the search process without affecting the quality of the search results. The trial results show that with privacy protection on the personalized web search being created, the system is able to provide high insurance 75,89% and 77,78% on the relevance of the user's desire for the search results. The system has given the same quality results as a personalized web search engine that has no privacy protection.

Keyword : PWS, Web Search Engine, Greedy Algorithm

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridho-Nya penulis dapat menyelesaikan Tugas Akhir dengan judul

**“PENINGKATAN KUALITAS HASIL PENCARIAN
TANPA MELANGGAR PRIVASI PENGGUNA PADA
PENULUSURAN WEB TERPERSONALISASI
MENGUNAKAN ALGORITMA *GREEDY*”**

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Matematika, Fakultas Matematika, Komputasi, dan Sains Data, Institut Teknologi Sepuluh Nopember, Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Drs. Sentot Didik Surjanto, M.Si selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
3. Dr. Budi Setiyono, S.Si, MT selaku Dosen Pembimbing yang telah membimbing penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
4. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
5. Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS.
6. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
7. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a yang tak terhingga.
8. Teman-teman angkatan 2014 yang selalu memberikan dukungan.

9. Semua pihak yang tidak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

TUGAS AKHIR - SM141501	i
FINAL PROJECT - SM141501	iii
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan Tugas Akhir	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu	7
2.2 Penelusuran Web Terpersonalisasi (PWS)..	8
2.3 Mesin Pencari Web (<i>Web Search Engine</i>) ..	9
2.4 Profil Pengguna	10
2.5 Persyaratan Privasi yang Disesuaikan.....	11
2.6 Personalisasi Berbasis Profil (<i>Profile-Based Personalization</i>).....	11
2.7 Framework	12
2.7.1 CodeIgniter	12
2.7.2 Model-View-Controller (MVC).....	14
2.7.3 Framework-UPS (User Customizable Privacy preserving Search).....	15
2.8 Generalisasi Profil Pengguna	16
2.9 Algoritma <i>Greedy</i>	17
2.10 Database	19
2.10.1 <i>Database Management Sistem (DBMS)</i>	19
2.10.2 MySQL	19
2.11 <i>Unified Modeling Language (UML)</i>	20

2.12	<i>Application Programming Interface (API)</i>	21
2.13	<i>XAMPP Server</i>	23
2.14	<i>Hypertext Preprocessor (PHP)</i>	23
2.15	<i>JavaScript Object Notation (JSON)</i>	24
BAB III METODE PENELITIAN		25
3.1	Studi Literatur	25
3.2	Pengumpulan Data	25
3.3	Analisis dan Perancangan Sistem	25
3.4	Implementasi Sistem	26
3.5	Analisis Hasil Pengujian	27
3.6	Kesimpulan dan Saran	27
BAB IV ANALISIS DAN PERANCANGAN SISTEM		29
4.1	Analisis Sistem	29
4.1.1	Deskripsi Sistem Secara Umum	29
4.1.2	Analisis Kebutuhan Sistem	30
4.2	Perancangan Sistem	31
4.2.1	Gambaran Sistem	32
4.2.2	Perancangan Alur Sistem	51
4.2.3	Perancangan Alur Penggunaan API	55
4.2.4	Skema Database	59
4.2.5	Perancangan Halaman Antarmuka (<i>Interface</i>)	60
BAB V IMPLEMENTASI DAN PENGUJIAN		65
5.1	Implementasi Sistem	65
5.1.1	Implementasi <i>CodeIgniter</i>	65
5.1.2	Implementasi Login Pengguna	68
5.1.3	Implementasi <i>Perlindungan Privasi</i>	69
5.1.4	Implementasi Algoritma <i>Greedy</i>	70
5.1.5	Implementasi Akuisisi Data	73
5.1.6	Implementasi <i>Filter</i> Data	73
5.1.7	Implementasi Proses Personalisasi Data Pengguna	73
5.1.8	Implementasi <i>Database</i>	73
5.1.9	Implementasi Halaman antarmuka (<i>Interface</i>)	82

5.2	Pengujian Aplikasi	89
5.2.1	Skenario Uji Coba.....	90
5.2.2	Pengujian Kesesuaian <i>Keyword</i> dengan Hasil yang Ditampilkan.....	93
5.2.3	Pengujian Keakuratan Hasil dengan Relevansi Keinginan Pengguna	99
5.2.4	Pengujian Peningkatan Kualitas Hasil Pencarian Tanpa Melanggar Persyaratan Privasi Pengguna	106
5.2.5	Perbandingan Aplikasi	112
5.1.4	Hasil Uji Coba	121
BAB VI PENUTUP		125
6.1	Kesimpulan.....	125
6.2	Saran	126
DAFTAR PUSTAKA		127

DAFTAR GAMBAR

Gambar 2.1 Contoh Profil Pengguna.....	10
Gambar 2.2 Contoh <i>Taxonomy Repository</i>	10
Gambar 2.3 Diagram data pada sistem.....	13
Gambar 2.4 Diagram MVC Codeigniter.....	15
Gambar 2.5 Contoh <i>Use Case Diagram</i>	21
Gambar 2.6 Contoh <i>Swimlane Diagram</i>	22
Gambar 2.7 Skema konektivitas API antar software.....	22
Gambar 3.1 Blok diagram alur implementasi sistem.....	28
Gambar 4.1 Desain arsitektur sistem.....	33
Gambar 4.2 Desain proses login pengguna.....	35
Gambar 4.3 Desain penentuan parameter.....	36
Gambar 4.4 Perbedaan hasil pencarian.....	37
Gambar 4.5 Skema Algoritma <i>Greedy</i>	42
Gambar 4.6 Desain akuisisi data.....	43
Gambar 4.7 Desain filter data.....	45
Gambar 4.8 Metadata <i>laptop</i>	47
Gambar 4.9 Metadata <i>apple</i>	48
Gambar 4.10 Deskripsi <i>similar_text()</i>	48
Gambar 4.11 <i>Use Case</i> pencarian informasi web menggunakan sistem dengan mengisi data <i>login/query</i>	53
Gambar 4.12 <i>Use case</i> pemrosesan data <i>login/ data query</i> pencarian yang dimasukkan pengguna menggunakan sistem.....	55
Gambar 4.13 <i>Swimlane</i> sistem.....	57
Gambar 4.14 Parameter <i>cx</i>	58
Gambar 4.15 Skema <i>database</i>	59
Gambar 4.16 Halaman awal.....	61
Gambar 4.17 Halaman registrasi.....	61
Gambar 4.18 Halaman login.....	62
Gambar 4.19 Halaman pencarian.....	62
Gambar 4.20 Halaman hasil.....	63
Gambar 5.1 Kode pada <i>autoload.php</i>	64
Gambar 5.2 Kode pada <i>config.php</i>	64
Gambar 5.3 Kode pada <i>routes.php</i>	65

Gambar 5.4 Kode pada <i>config.php</i>	65
Gambar 5.5 Kode pada <i>database.php</i>	66
Gambar 5.6 Kode pada <i>index.php</i>	67
Gambar 5.7 Kode implementasi registrasi pengguna pada controller	68
Gambar 5.8 Kode implementasi login pengguna pada controller	69
Gambar 5.9 Kode implementasi login pengguna pada model....	70
Gambar 5.10 Kode implementasi Greedy untuk privasi	72
Gambar 5.11 Kode implementasi akuisisi data.....	73
Gambar 5.12 Kode implementasi filtering data.....	74
Gambar 5.13 Potongan kode <i>gethistory</i> (privasi umum).....	75
Gambar 5.14 Potongan kode <i>gethistory</i> privasi (privasi khusus)	75
Gambar 5.15 Potongan kode insert keyword (privasi umum)....	76
Gambar 5.16 Potongan kode insert privasi (privasi khusus).....	76
Gambar 5.17 Potongan kode insert data dari server ke database	77
Gambar 5.18 Potongan kode insert data yang sudah difilter ke database.....	77
Gambar 5.19 Potongan kode mendapatkan semua data pada tabel di database.....	78
Gambar 5.20 Potongan kode mendapatkan judul website untuk proses filtering data.....	78
Gambar 5.21 Potongan kode mendapatkan semua data riwayat pencarian pengguna berdasarkan privasi.....	79
Gambar 5.22 Potongan kode mendapatkan semua data berdasarkan id pengguna dan privasi khusus.....	79
Gambar 5.23 Potongan kode pengecekan usernam	80
Gambar 5.24 Tabel data untuk menyimpan data hasil filtering ..	81
Gambar 5.25 Tabel datalain untu menyimpan data image.....	81
Gambar 5.26 Tabel datasite untuk menyimpan data website	81
Gambar 5.27 Tabel keyword untuk menyimpan data privasi umum	82
Gambar 5.28 Tabel privasi untuk menyimpan data privasi khusus	82
Gambar 5.29 Tabel login untuk menyimpan data login pengguna	82

Gambar 5.30 Struktur database tugasakhir.....	82
Gambar 5.31 Kode <i>homecontroller.php</i>	83
Gambar 5.32 Kode <i>usercontroller.php</i>	84
Gambar 5.33 Kode <i>testcontroller.php</i>	84
Gambar 5.34 Implementasi halaman awal	85
Gambar 5.35 Implementasi halaman registrasi.....	85
Gambar 5.36 Implementasi halaman login.....	86
Gambar 5.37 Implementasi halaman pencarian.....	86
Gambar 5.38 Implementasi halaman hasil website.....	87
Gambar 5.39 Implementasi halaman hasil image	87
Gambar 5.40 Grafik kenaikan jumlah data hasil pencarian	101
Gambar 5.41 Grafik kenaikan jumlah data hasil sesuai preferensi pengguna pencarian	102
Gambar 5.42 Hasil pencarian “ <i>laptop</i> ” pada katagori website di sistem yang dibuat.....	104
Gambar 5.43 Hasil pencarian “ <i>laptop</i> ” pada katagori image di sistem yang dibuat.....	105
Gambar 5.44 Hasil pencarian “ <i>apple smartphones</i> ” pada katagori wesite di sistem yang dibuat	105
Gambar 5.45 Hasil pencarian “ <i>apple smartphones</i> ” pada katagori image di sistem yang dibuat.....	106
Gambar 5.46 Hasil pencarian “buku cerita anak” pada katagori website di sistem yang dibuat	106
Gambar 5.47 Hasil pencarian “ <i>buku cerita anak</i> ” pada katagori image di sistem yang dibuat.....	107
Gambar 5.48 Parameter untuk pencarian “ <i>laptop</i> ” ke server pada sistem yang dibuat.....	107
Gambar 5.49 Parameter untuk pencarian “ <i>apple smartphones</i> ” ke server pada sistem yang dibuat.....	107
Gambar 5.50 Parameter untuk pencarian “ <i>buku cerita anak</i> ” ke server pada sistem yang dibuat.....	108
Gambar 5.51 Data login pengguna yang masuk ke sistem yang dibuat.....	108
Gambar 5.52 Hasil pencarian “ <i>laptop</i> ” pada katagori website di Google.....	109

Gambar 5.53 Hasil pencarian “ <i>laptop</i> ” pada katagori image di Google.....	109
Gambar 5.54 Hasil pencarian “ <i>apple smartphones</i> ” pada katagori website di Google.....	110
Gambar 5.55 Hasil pencarian “ <i>apple smartphones</i> ” pada katagori image di Google.....	110
Gambar 5.56 Hasil pencarian “ <i>buku cerita anak</i> ” pada katagori website di Google.....	111
Gambar 5.57 Hasil pencarian “ <i>buku cerita anak</i> ” pada katagori image di Google.....	111
Gambar 5.58 Parameter untuk pencarian “ <i>laptop</i> ” ke server pada Google.....	112
Gambar 5.59 Parameter untuk pencarian “ <i>apple smartphones</i> ” ke server pada Google.....	112
Gambar 5.60 Parameter untuk pencarian “ <i>buku cerita anak</i> ” ke server pada Google.....	112
Gambar 5.61 Data login pengguna yang masuk ke <i>Google</i>	112

DAFTAR TABEL

Tabel 5.1 Skenario keyword untuk pengguna 1.....	91
Tabel 5.2 Skenario keyword untuk pengguna 2.....	92
Tabel 5.3 Skenario keyword yang saling berkaitan untuk pengguna.....	92
Tabel 5.4 Kesesuaian informasi yang ditampilkan dengan jenis keyword pada pengguna 1.....	93
Tabel 5.5 Jumlah data yang diterima pada pengguna 1.....	94
Tabel 5.6 Jumlah data yang sesuai keinginan pengguna yang diterima pada pengguna 1.....	95
Tabel 5.7 Kesesuaian informasi yang ditampilkan dengan jenis keyword pada pengguna 2.....	95
Tabel 5.8 Jumlah data yang diterima pada pengguna 2.....	96
Tabel 5.9 Jumlah data yang sesuai keinginan pengguna yang diterima pada pengguna 2.....	96
Tabel 5.10 Kesesuaian informasi yang ditampilkan dengan relevansi keinginan pengguna pada pengguna 1.....	97
Tabel 5.11 Jumlah data yang diterima pada pengguna 1.....	98
Tabel 5.12 Jumlah data sesuai keinginan pengguna yang diterima pada pengguna 1.....	99
Tabel 5.13 Kesesuaian informasi yang ditampilkan dengan relevansi keinginan pengguna pada pengguna 2.....	99
Tabel 5.14 Jumlah data yang diterima pada pengguna 2.....	100
Tabel 5.15 Jumlah data sesuai keinginan pengguna yang diterima pada pengguna 2.....	100
Tabel 5.16 Jumlah data yang diterima pada pengguna	101
Tabel 5.17 Jumlah data sesuai keinginan pengguna yang diterima.....	102
Tabel 5.18 Hasil uji coba.....	122

BAB I

PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir ini. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Dalam perkembangan teknologi yang canggih seperti sekarang ini, orang pada umumnya akan melakukan pencarian terhadap informasi yang dibutuhkan melalui mesin pencari web (*web search engine*). Mesin pencari web (*web search engine*) adalah sistem perangkat lunak yang memungkinkan untuk mencari informasi di web (*www*), informasi yang dicari dapat berupa gambar, dokumen atau jenis file lainnya [1]. Seringkali ketika seorang pengguna melakukan pencarian informasi di web, mesin pencari mengalami kegagalan saat mengembalikan hasil pencarian. Hasil yang ditampilkan tidak relevan atau tidak sesuai dengan kebutuhan pengguna. Hasil yang tidak relevan tersebut sebagian besar disebabkan oleh banyaknya variasi konteks dan latar belakang pengguna, serta ketidakjelasan teks [2].

Penelusuran Web Terpersonalisasi (PWS) adalah salah satu teknik pencarian umum yang ditujukan untuk memberikan peningkatan kualitas hasil pencarian, dan disesuaikan dengan kebutuhan individual pengguna [2]. Pada salah satu jenis PWS, yaitu PWS berbasis profil, penyedia layanan harus mempertimbangkan dua efek yang kontradiktif selama proses pencarian. Pada saat proses pencarian, penyedia layanan akan berusaha memperbaiki kualitas pencarian dengan memanfaatkan

utilitas personalisasi profil pengguna. Di sisi lain, jika privasi pengguna terkendali dan terekam oleh mesin pencari web (*web search engine*), kemungkinan yang terjadi adalah data-data tersebut akan disalahgunakan oleh pihak ketiga. Ketika pihak ketiga memiliki data privasi pengguna, data-data tersebut digunakan untuk menggolongkan minat pengguna dan dijadikan sasaran untuk publikasi, dan mangsa pasar yang terkadang merisihkan pengguna itu sendiri. Bahkan dampak terbesar yang mungkin adalah data digunakan sebagai perantara tindakan kriminal yang ditujukan kepada pengguna tersebut. Sehingga penyedia layanan juga harus menyembunyikan isi privasi dalam profil pengguna untuk menghindari resiko terkendalinya privasi pengguna.

Kebutuhan penanganan masalah kontradiktif tersebut sangat diperlukan, mengingat pengguna mesin pencari web (*web search engine*) saat ini semakin meningkat. Para pengguna saat ini banyak menuntut keuntungan dari kecanggihan teknologi, tanpa harus mengorbankan sesuatu yang merugikan mereka. Penyedia layanan harus berusaha memenuhi keinginan pengguna layanan, melalui personalisasi dengan hanya mengorbankan sedikit dan kurang sensitif bagian dari profil pengguna, yaitu profil umum. Hal ini diharapkan dapat melindungi privasi pengguna tanpa mengorbankan kualitas pencarian hasil terpersonalisasi.

Sistem yang diusulkan menghasilkan profil lengkap pengguna, dan digunakan untuk mendapatkan personalisasi yang lebih baik. Berdasarkan hal tersebut, ketika pengguna memasukkan permintaan, sistem akan memberikan saran yang sesuai dengan kebutuhan pengguna. Pada penelitian terdahulu telah diusulkan PWS *framework* yaitu UPS (*User customizable privacy preserving search*) yang dapat menggeneralisasikan profil

untuk setiap *query* sesuai dengan persyaratan privasi pengguna menggunakan algoritma *Greedy*. Proses generalisasi secara *run time* ditujukan untuk menyeimbangkan antara dua metrik prediktif yang mengevaluasi kegunaan personalisasi dan risiko privasi dalam mengekspos profil umum [4][5]. UPS terdiri dari server mesin pencari yang terpercaya dan beberapa pengguna. Algoritma *Greedy* merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya [3].

Oleh karena itu, pada Tugas Akhir ini akan dibahas mengenai menyediakan perlindungan privasi sisi pengguna (*Client-side Privacy*) dalam penelusuran web terpersonalisasi (PWS). Setiap PWS menangkap profil pengguna dalam hierarkis taksonomi. Sistem ini akan melakukan generalisasi online dengan algoritma *Greedy* pada profil pengguna untuk melindungi privasi pengguna. Selain itu, sistem juga akan berusaha memperbaiki kualitas pencarian dengan utilitas personalisasi profil pengguna, sehingga stabilitas kualitas pencarian meningkat, hasil yang didapat sesuai dengan permintaan dan persyaratan privasi, tanpa harus takut privasinya terancam.

1.2 Rumusan Masalah

Rumusan masalah pada Tugas Akhir ini adalah bagaimana cara membangun perangkat lunak sistem perlindungan privasi pengguna tanpa mengorbankan kualitas pencarian pada mesin pencari web (*web search engine*) dalam Penelusuran Web Terpersonalisasi (PWS) berbasis profil yang sudah ada.

1.3 Batasan Masalah

Pada penelitian ini, penulis membuat batasan masalah sebagai berikut :

1. Penelusuran Web Terpersonalisasi (PWS) yang digunakan merupakan jenis PWS berbasis profil pengguna.
2. Data yang digunakan merupakan metadata dari *Google API (Application Programming Interface) Custom Search*.
3. Metadata yang dapat diakuisisi sebanyak maksimal 10 data untuk setiap *Query*.
4. Menggunakan *Framework CodeIgniter*.
5. Implementasi kerja sistem menggunakan sistem pada *framework UPS (User customizable privacy preserving search)*
6. Menggunakan DBMS (*Data Base Management Sistem*) yaitu MYSQL yang membantu proses penyimpanan data.
7. Implementasi menggunakan *script PHP dan SQL*.
8. *Query/Keyword* masukkan, disusun berdasarkan kata yang dianggap pengguna bersifat tidak sensitif ditaruh diawal kalimat dan kemudian disusul dengan kata yang bersifat sensitif bagi pengguna

1.4 Tujuan

Tujuan pembuatan Tugas Akhir ini adalah Membangun perangkat lunak sistem perlindungan privasi pengguna tanpa mengorbankan kualitas pencarian pada mesin pencari web (*web search engine*) dalam Penelusuran Web Terpersonalisasi (PWS) berbasis profil yang sudah ada.

1.5 Manfaat

Manfaat yang bisa diperoleh dari Penelitian ini adalah :

1. Untuk masyarakat dapat membantu memberikan mesin pencari web (*web search engine*) yang lebih berkualitas.
2. Untuk pendidikan perguruan tinggi dapat dimanfaatkan sebagai referensi implementasi *Web Search* menggunakan

metode Penelusuran Web Terpersonalisasi (PWS) berbasis profil pengguna untuk menghasilkan stabilitas kualitas pencarian, kualitas pencarian tinggi serta sesuai dengan pertanyaan dan persyaratan privasi yang diajukan pengguna, tanpa harus takut privasinya terancam.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang materi-materi yang mendukung Tugas Akhir ini, antara lain penelitian terdahulu, penelusuran web terpersonalisasi, mesin pencari web, profil pengguna, persyaratan privasi yang disesuaikan, personalisasi berbasis profil, *Framework*, generalisasi profil pengguna, algoritma *Greedy*, *Database*, *Unified Modeling Language (UML)*, *Application Programming Interface (API)*, *XAMPP*, *PHP* dan *JSON*.

BAB III METODOLOGI PENELITIAN

Pada bab ini dibahas tentang langkah – langkah dan metode yang digunakan untuk menyelesaikan Tugas Akhir ini.

BAB IV PERANCANGAN IMPLEMENTASI

Pada bab ini akan menguraikan bagaimana tahapan-tahapan dalam perancangan implementasi.

BAB V IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi dari algoritma *Greedy* untuk perlindungan privasi pengguna saat melakukan pencarian informasi dengan mesin pencari web, dan proses personalisasi data pengguna sehingga hasil yang ditampilkan sesuai atau relevan dengan preferensi/keinginan pengguna. Setelah itu, dilakukan analisis terhadap hasil implementasi. Analisis ini bertujuan untuk melihat apakah algoritma tersebut dapat diterapkan untuk menyelesaikan Peningkatan kualitas pencarian tanpa melanggar privasi pengguna pada pencarian web terpersonalisasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang dijelaskan dibagi menjadi beberapa subbab yaitu penelitian terdahulu, penelusuran web terpersonalisasi, mesin pencari web, profil pengguna, persyaratan privasi yang disesuaikan, personalisasi berbasis profil, *Framework*, generalisasi profil pengguna, algoritma *Greedy*, *Database*, *Unified Modeling Language (UML)*, *Application Programming Interface (API)*, *XAMPP*, *PHP* dan *JSON*.

2.1 Penelitian Terdahulu

Pada penelitian terdahulu Ghilby Varghese Jaison, dan Charlse M Varghese telah mengembangkan sistem perlindungan privasi pada penelusuran web terpersonalisasi (PWS) menggunakan enkripsi homomorfik [6]. Penelitian ini dilakukan karena sebenarnya Penelusuran Web terpersonalisasi (PWS) sudah dianggap efektif dalam meningkatkan kualitas pencarian. Namun, bukti menunjukkan bahwa ketakutan pengguna untuk mengungkapkan informasi pribadi mereka selama pencarian telah menjadi penghalang utama bagi berkembangnya PWS secara luas. Oleh karena itu, dibuatlah sistem pendukung perlindungan privasi pada PWS menggunakan server indeks yang dienkripsi dengan enkripsi homomorfik. Sistem ditujukan untuk mencapai PWS menggunakan teknik berdasarkan klik-log. Fokus utama sistem adalah untuk privasi dari istilah-istilah yang dicari pada server dan server penyadap. Penelitian ini merepresentasikan sebuah perlindungan privasi *framework* berdasarkan enkripsi homomorfik untuk PWS. Sistem ini berpotensi diadopsi oleh

PWS yang dapat mengoperasikan server enkripsi homomorfik. Hasil penelitian menunjukkan bahwa sistem dapat mencapai kualitas pencarian yang baik, tanpa mempengaruhi privasi pengguna serta menegaskan keefektifan dan keefisienan dari metode yang digunakan.

Selain PWS *framework* yang menggunakan enkripsi homomorfik, pada penelitian lain juga telah dibahas mengenai penggunaan PWS *framework* yaitu UPS (*User customizable privacy preserving search*) oleh Kavitha V dan Uma Maheswari T [7] dalam menyelesaikan permasalahan yang sama. Sistem yang dibangun akan memberikan proses generalisasi profil pengguna berdasarkan pengaturan persyaratan privasi pengguna. Sehubungan dengan itu, untuk memberikan dukungan pada proses generalisasi profil pengguna secara *run time*, dua algoritma efektif *Greedy DP* dan *Greedy IL* telah dikembangkan. Sistem memberikan metode yang murah kepada pengguna dan memberikan kestabilan pada profil pengguna. Hal ini digunakan untuk mencegah pemaparan profil yang tidak perlu. Sebelum menggeneralisasi profil secara *run time*, pengguna harus memutuskan apakah akan mempersonalisasi permintaan di UPS atau tidak.

2.2 Penelusuran Web Terpersonalisasi (PWS)

Penelitian sebelumnya tentang perlindungan privasi PWS tidak menguntungkan. Sistem PWS yang ada saat ini memiliki banyak masalah [1]. Pembuatan profil run-time tidak didukung pada PWS berbasis profil yang ada. "Satu ukuran cocok untuk semua" merupakan strategi yang digunakan.

Pendekatan ini akan membahayakan privasi pengguna. Pendekatan yang lebih baik adalah membuat keputusan online

yang dapat memilih tingkat pemaparan profil pengguna dan apakah menerapkan personalisasi *Query*.

Penyesuaian persyaratan privasi tidak dipertimbangkan dalam metode yang ada. Hal ini menyebabkan ketidakseimbangan perlindungan ke beberapa privasi pengguna. Satu pengguna mendapatkan perlindungan lebih, sementara yang lainnya tidak memiliki perlindungan. *Surprisal* berdasarkan teori informasi, adalah metrik yang digunakan untuk mendeteksi topik sensitif. Metrik ini menganggap ketertarikan dengan sedikit dukungan dokumen merupakan kesensitifan yang tinggi. Contoh yang salah akan menyesatkan anggapan tersebut: Jika jumlah dokumen yang lebih banyak dilihat dengan *query* "gender", metrik tersebut mungkin menyimpulkan bahwa "gender" sangat umum.

Interaksi pengguna iteratif diperlukan untuk membuat hasil personalisasi dalam banyak teknik PWS. Penyempurnaan hasil pencarian dilakukan dengan beberapa metrik yang membutuhkan beberapa interaksi pengguna seperti rangking, rangking rata-rata, peringkat skor. Antarmuka ini tidak praktis untuk pembuatan profil run time.

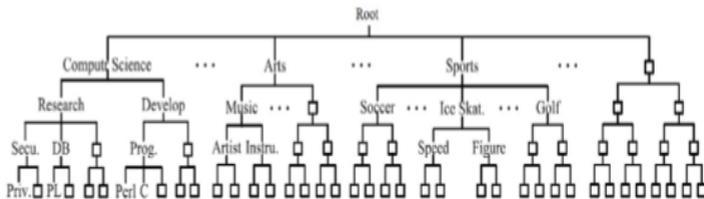
2.3 Mesin Pencari Web (*Web Search Engine*)

Mesin pencari web (*web search engine*) adalah sistem perangkat lunak yang memungkinkan untuk mencari informasi di web (*www*), informasi yang dicari dapat berupa gambar, dokumen atau jenis file lainnya [1]. Seringkali ketika seorang pengguna melakukan pencarian informasi di web, mesin pencari mengalami kegagalan saat mengembalikan hasil. Hasil yang ditampilkan tidak relevan atau tidak sesuai dengan kebutuhan pengguna. Hasil yang tidak relevan tersebut sebagian besar disebabkan oleh banyaknya variasi konteks dan latar belakang pengguna, serta

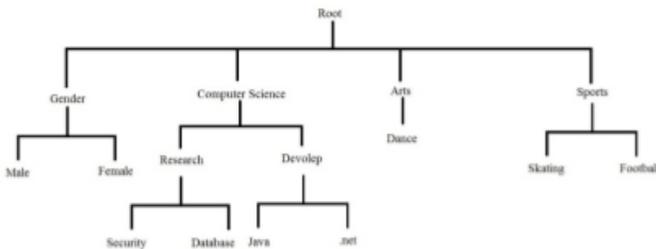
ketidakjelasan teks [2]. Sebagian besar mesin pencari dijalankan oleh perusahaan swasta yang menggunakan algoritma kepemilikan dan basis data tertutup, di antaranya yang paling populer adalah safari *Google, MSN Search dan Yahoo!*.

2.4 Profil Pengguna

Konsisten dengan banyak karya masa lalu dalam pencarian web yang dipersonalisasi, setiap profil pengguna di UPS (User Customizable Privacy preserving Search) mengadopsi struktur hirarkis. Diagram profil pengguna sampel diilustrasikan pada Gambar 2.1 , yang dibangun berdasarkan sampel taksonomi repositori pada Gambar 2.2 [6].



Gambar 2.1. Contoh Profil Pengguna [6]



Gambar 2.2. Contoh *Taxonomy Repository* [6]

2.5 Persyaratan Privasi yang Disesuaikan

Persyaratan privasi yang disesuaikan dapat ditentukan dengan sejumlah node-sensitif (topik) dalam profil pengguna. Pengungkapannya (ke server) mengenakan kompromi tentang privasi pengguna. Dalam contoh profil yang ditunjukkan pada Gambar 2.1, simpul sensitif $S = \{\text{Dewasa; Keamanan; Skating}\}$; Berbayang warna abu-abu di H. Perlu dicatat bahwa kekhawatiran privasi pengguna mungkin berbeda dari satu topik sensitif ke topik yang lain. Pada contoh di atas, pengguna mungkin ragu untuk berbagi kepentingan pribadinya hanya untuk menghindari berbagai iklan. Dengan demikian, pengguna mungkin masih mentolerir pemaparan kepentingan tersebut untuk diperdagangkan demi utilitas personalisasi yang lebih baik. Namun, pengguna mungkin tidak akan pernah membiarkan minat lain pada topik gender untuk diungkapkan. Karena nilai sensitifitas secara tegas menunjukkan kekhawatiran keamanan pengguna, strategi penghematan perlindungan yang paling jelas adalah mencabut sub-pohon di semua hub yang nilai sensitifitasnya lebih menonjol daripada batas. Sistem semacam itu disinggung karena pelanggaran. Bagaimanapun, pelanggaran adalah jauh dari cukup untuk melawan musuh yang lebih halus [6].

2.6 Personalisasi Berbasis Profil (*Profile-Based Personalization*)

Banyak representasi profil tersedia dalam literatur untuk memfasilitasi strategi personalisasi yang berbeda. Teknik sebelumnya menggunakan daftar istilah atau vektor atau tas kata untuk mewakili profil mereka. Namun, karya terbaru membangun profil dalam struktur hirarkis karena kemampuan deskriptifnya yang lebih kuat, skalabilitas yang lebih baik, dan efisiensi akses yang lebih tinggi. Sebagian besar representasi hirarkis dibangun

dengan hirarki atau grafik topik tertimbang yang ada, seperti ODP, Wikipedia dan sebagainya. Selain itu juga dikerjakan pembangunan profil hierarkis secara otomatis melalui analisis frekuensi-waktu pada data pengguna [8].

2.7 Framework

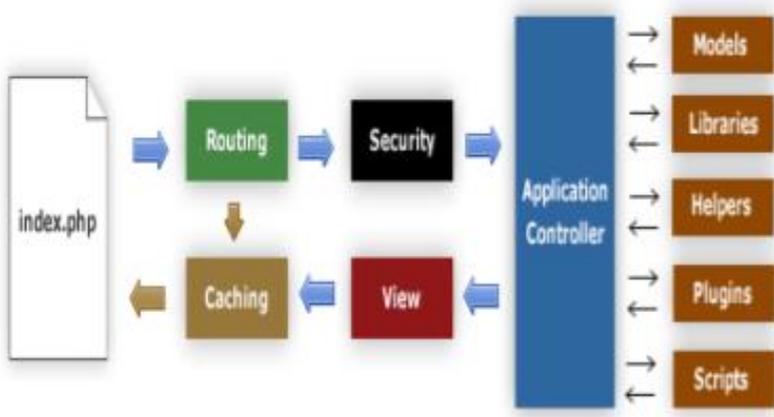
Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi website, harus mengikuti aturan dari framework tersebut [9]. Framework dapat diartikan sebagai komponen pemrograman yang siap pakai kapan saja, sehingga seorang programmer tidak perlu membuat script yang sama untuk permasalahan yang sama. Framework itu sendiri memiliki fungsi yang berbeda-beda serta cara pemakaiannya juga berbeda-beda. Untuk pemrograman aplikasi berbasis web ada beberapa framework yang menyediakan fungsi-fungsi yang mudah digunakan, salah satunya adalah CodeIgniter.

2.7.1 CodeIgniter

CodeIgniter adalah sebuah framework aplikasi web yang bersifat open source yang berfungsi untuk membangun aplikasi dengan bahasa pemrograman PHP secara dinamis. CodeIgniter memiliki banyak fitur yang membuat framework ini berbeda, salah satunya adalah dokumentasi yang lengkap yang mencakup seluruh aspek dalam framework. CodeIgniter dibangun menggunakan konsep MVC (Model-View-Controller) development pattern. CodeIgniter juga mampu berjalan pada lingkungan shared hosting karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang cepat [10]. Proses data mengalir pada sistem yang menggunakan menggunakan framework CodeIgniter dapat dilihat pada Gambar 2.3

Keterangan :

1. Index.php berfungsi sebagai front controller, menginisialisasi base resource untuk menjalankan CodeIgniter.
2. Router memeriksa HTTP request untuk menentukan apa yang harus dilakukan.
3. Jika Cache aktif, maka hasilnya akan langsung dikirimkan ke browser dengan mengabaikan aliran data normal.
4. Security berfungsi sebagai keamanan dimana sebelum controller dimuat, HTTP request dan data yang akan dikirim ke user akan di filter.
5. Controller memuat model, core libraries, plugins, helpers, dan semua resource yang diperlukan untuk proses request.
6. View yang dihasilkan akan dikirim ke user. Jika chace aktif, maka view akan disimpan sebagai chace sehingga pada request selanjutnya langsung ditampilkan.



Gambar 2.3 Diagram data pada sistem [11]

2.7.2 Model-View-Controller (MVC)

MVC adalah pattern atau pola pemrograman yang memisahkan desain, data, dan proses. MVC menstrukturisasi aplikasi dengan cara tersebut untuk mempromosikan penggunaan kembali dari kode program [10]. Adapun komponen-komponen MVC antara lain:

1. Model

Model berhubungan dengan data dan interaksi ke database atau webservice. Model merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.

2. View

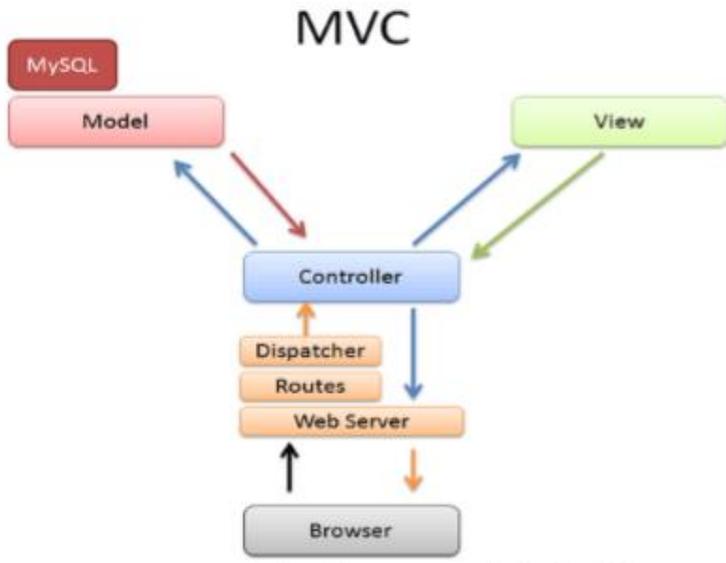
View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user seperti halaman web, rss, javascript dan lain-lain. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. view dapat dikatakan sebagai halaman antarmuka website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan tidak terdapat kode perintah untuk melakukan koneksi ke database. View hanya dikhususkan untuk menampilkan data-data hasil dari Model dan Controller.

3. Controller

Controller bertindak sebagai penghubung model dan view. Controller inilah didalamnya terdapat class dan fungsi yang memproses permintaan dari view ke dalam struktur data di dalam

model. Controller juga berisi kode perintah untuk mengakses database karena tugas mengakses data telah diserahkan kepada model. Tugas Controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke database, menyediakan penanganan error, mengerjakan proses logika dari aplikasi serta melakukan validasi terhadap input.

Alur diagram dari framework CodeIgniter dengan MVC dapat dilihat pada Gambar 2.4



Gambar 2.4 Diagram MVC CodeIgniter [11]

2.7.3 Framework-UPS (User Customizable Privacy preserving Search)

User customizable privacy preserving search (UPS) memberikan penggeneralisasian profil berdasarkan pengaturan

persyaratan privasi pengguna. Membangun demarkasi metrik yang bertentangan (predikat), yaitu perangkat lunak personalisasi dan ancaman privasi, untuk hierarkis profil pengguna [7]. Sehubungan dengan itu, untuk memberikan dukungan pada proses generalisasi profil pengguna secara *run time*, algoritma efektif telah dikembangkan. Sistem memberikan metode yang murah kepada pengguna dan memberikan kestabilan pada profil pengguna. Hal ini digunakan untuk mencegah pemaparan profil yang tidak perlu. Sebelum menggeneralisasi profil secara *run time*, pengguna harus memutuskan apakah akan mempersonalisasi permintaan di UPS atau tidak.

Kelemahan metode UPS ini adalah :

1. Tidak melampaui kesendirian penggunaan tunggal secara tepat.
2. Tidak mendapatkan keseluruhan topik pendukung privasi.
3. Proporsi periode sistem meningkat dalam sistem ini.
4. Tingkat akurasi lebih rendah bila dibandingkan dengan sistem lainnya.

2.8 Generalisasi Profil Pengguna

Kaitannya dengan mengatasi masalah dengan pelarangan, sistem yang ada mengusulkan sebuah Teknik. Sistem mendeteksi dan menghilangkan satu set node X, yang sensitif dari H, sehingga risiko privasi diperkenalkan dengan mengekspos G selalu terkendali. Untuk kejelasan deskripsi, diasumsikan bahwa semua sub-pohon H yang berakar pada simpul di X tidak saling tumpang tindih. Proses ini disebut generalisasi, dan output G adalah profil umum. Teknik generalisasi tampaknya dapat dilakukan selama pemrosesan *offline* tanpa melibatkan kueri

pengguna [6]. Namun, tidak praktis melakukan generalisasi *offline* karena dua alasan:

1. Output dari generalisasi *offline* mungkin berisi banyak cabang topik, yang tidak relevan dengan *query*. Solusi yang lebih fleksibel memerlukan generalisasi *online*, dan bergantung pada *query*. Generalisasi *online* tidak hanya menghindari pengungkapan privasi yang tidak perlu, namun juga menghilangkan topik yang tidak relevan dengan *query* saat ini.
2. Penting untuk memantau utilitas personalisasi selama generalisasi. Namun, generalisasi yang terlalu berlebihan dapat menyebabkan ambiguitas dalam personalisasi, dan akhirnya menyebabkan hasil pencarian yang buruk. Pemantauan utilitas hanya mungkin dilakukan jika generalisasi dilakukan saat *run time*.

2.9 Algoritma Greedy

Algoritma *Greedy* adalah sebuah proses matematika yang secara rekursif membangun satu set objek dari bagian penyusun terkecil. Rekursi adalah pendekatan pemecahan masalah dimana solusi untuk masalah tertentu bergantung pada solusi untuk kasus yang lebih kecil dari masalah yang sama. Algoritma *Greedy* mencari solusi sederhana dan mudah diterapkan untuk masalah multi-langkah kompleks dengan menentukan langkah selanjutnya yang akan memberikan manfaat paling nyata. Algoritma semacam itu disebut serakah (*Greedy*) karena sementara solusi optimal untuk setiap instance yang lebih kecil akan menghasilkan output langsung, algoritma tersebut tidak mempertimbangkan masalah yang lebih besar secara keseluruhan. Begitu keputusan dibuat, tidak pernah dipikirkan ulang. Keuntungan menggunakan algoritma *Greedy* adalah solusi untuk contoh masalah yang lebih

kecil dapat langsung dan mudah dimengerti. Kelemahannya adalah sangat mungkin solusi jangka pendek yang paling optimal dapat menyebabkan hasil jangka panjang terburuk. Algoritma *Greedy* sering digunakan dalam jaringan bergerak *ad hoc* untuk secara efisien merutekan paket dengan jumlah hop paling sedikit dan kemungkinan penundaan terpendek. Algoritma ini juga digunakan dalam pembelajaran mesin, *business intelligence* (BI), *artificial intelligence* (AI) [12]. Pseudo-code dari algoritma *Greedy* secara umum dapat dilihat sebagai berikut :

```

procedure greedy(input C: himpunan_kandidat;
                  output S : himpunan_solusi)
{ menentukan solusi optimum dari persoalan
  optimasi dengan algoritma greedy
  Masukan: himpunan kandidat C
  Keluaran: himpunan solusi S
}

```

Deklarasi

```
x : kandidat;
```

Algoritma:

```

S ← {}           { inisialisasi S
dengan kosong }
  while (belum SOLUSI(S)) and (C ≠ {} )
do
  x ← SELEKSI(C); { pilih sebuah
kandidat dari C}
  C ← C - {x}     { elemen himpunan
kandidat berkurang satu }
  if LAYAK(S ∪ {x}) then
    S ← S ∪ {x}
  endif
endwhile
{SOLUSI(S) sudah diperoleh or C = {} }

```

2.10 Database

Database adalah kumpulan item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware dan software guna melakukan manipulasi untuk kegunaan tertentu [13]. Database juga merupakan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil query dari database disebut Database Management Sistem (DBMS). Salah satu perangkat lunak yang sering digunakan adalah MySQL.

2.10.1 Database Management Sistem (DBMS)

Database Management Sistem adalah seperangkat program komputer yang didesain untuk mengatur sebuah basis data yang disimpan secara terstruktur, dan melakukan operasi-operasi data sesuai dengan permintaan penggunanya yang bertujuan untuk mengakses dan mengelola data tersebut [13].

2.10.2 MySQL

MySQL adalah sistem manajemen database yang bersifat open source. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia. MySQL merupakan sistem manajemen database yang bersifat relasional. Artinya data-data yang dikelola dalam database akan diletakan pada beberapa tabel yang terpisah sehingga proses manipulasi data akan menjadi cepat [14].

MySQL dapat digunakan untuk mengelola database mulai dari yang kecil hingga yang sangat besar. MySQL juga dapat menjalankan perintah-perintah Structure Query Language (SQL) untuk mengelola database relasional yang ada didalamnya.

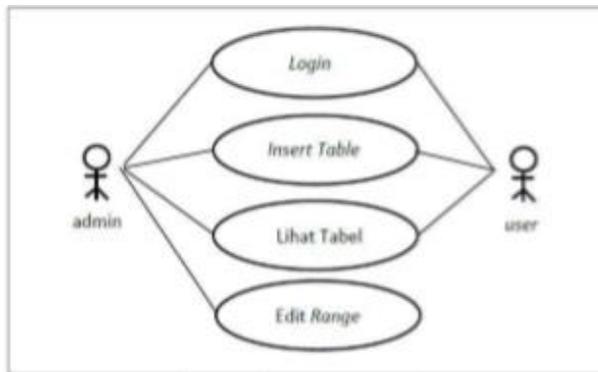
2.11 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak [15]. *Unified Modeling Language (UML)* adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.

UML dapat memvisualisasikan, menetapkan, membuat dan mendokumentasikan aplikasi pada software yang akan dibangun. Saat sistem software menjadi lebih besar dan lebih kompleks, maka diperlukan penyederhanaan untuk lebih mudah dalam penggunaan [16]. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantic. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentukbentuk tersebut dapat dikombinasikan. Ada beberapa diagram yang terdapat pada pemodelan UML namun hanya beberapa diagram yang sering digunakan yaitu sebagai berikut :

1. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan semua bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan berinteraksi dengan sistem aplikasi [15]. Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.5.



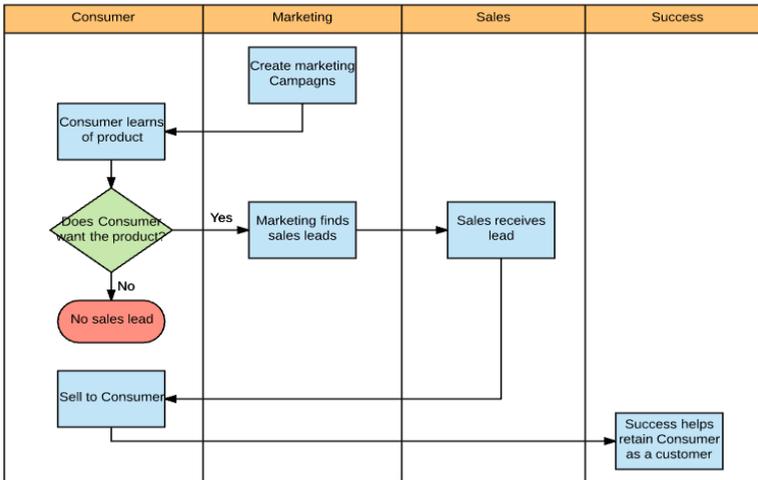
Gambar 2.5 Contoh *Use Case Diagram* [16]

2. *Swimlane Diagram*

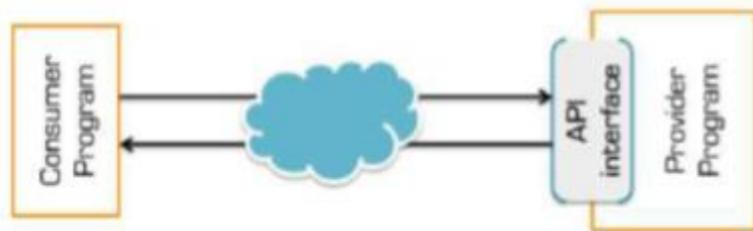
sebuah diagram flow adalah proses yang menggambarkan interaksi dari beberapa bagian yang berbeda yang terlibat dalam sebuah lini proses bisnis. Diagram ini menggunakan format jalur hubungan (*Swimlane*), adapun menggambarkannya dilakukan dengan cara menampilkan *Stakeholder* pada baris diagram serta kerangka waktu pada kolom diagram, dan kemudian aktivitasnya ditampilkan menggunakan symbol *Flowchart* [17]. Contoh *swimline diagram* dapat dilihat pada Gambar 2.6.

2.12 *Application Programming Interface (API)*

API merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain [13]. Skema dapat dilihat pada Gambar 2.7



Gambar 2.6 Contoh *Swimlane Diagram* [16]



Gambar 2.7 Skema konektivitas API antar software [13]

Secara struktural, API merupakan spesifikasi dari suatu data structure, objects, functions, beserta parameter-parameter yang diperlukan untuk mengakses resource dari aplikasi tersebut. Seluruh spesifikasi tersebut membentuk suatu interface yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain, dan API dapat digunakan dengan berbagai bahasa programming,

ataupun hanya dengan menggunakan URL (*Uniform Resource Identifier*) yang telah disediakan oleh suatu *website* [13].

2.13 XAMPP Server

XAMPP adalah program aplikasi pengembangan yang berguna untuk pengembangan *website* berbasis PHP dan MySQL. XAMPP berfungsi sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl [17]. *Web server Apache* berfungsi untuk simulasi pengembangan *website*. Melalui aplikasi ini, *developer* dapat menguji aplikasi secara langsung dari komputer tanpa perlu terkoneksi dengan internet. XAMPP juga dilengkapi fitur manajemen *database* PHPMyAdmin, sehingga pengembangan web berbasis *database* dapat dilakukan dengan mudah [17].

2.14 Hypertext Preprocessor (PHP)

PHP adalah bahasa pemrograman *server-side* yang didesain untuk pengembangan web. Disebut bahasa pemrograman *server-side* karena PHP diproses pada komputer server. Hal ini berbeda dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*. Untuk membuat halaman web, PHP bukanlah bahasa pemrograman yang wajib digunakan. *Website* dapat dibuat hanya dengan menggunakan HTML tetapi *website* yang dihasilkan merupakan *website* statis dimana konten dan halaman web bersifat tetap. Dengan menggunakan PHP, *website* yang dibuat dapat berupa *website* dinamis yang artinya tampilan konten tergantung situasi. *Website* dinamis juga dapat menyimpan data ke dalam *database*, membuat halaman yang berubah-ubah sesuai input user, dan lain-lain [18].

2.15 *JavaScript Object Notation (JSON)*

JavaScript Object Notation (JSON) merupakan suatu format pertukaran data yang dirancang menjadi format yang mudah dibaca dan ditulis oleh manusia, serta mudah untuk dihasilkan dan diproses oleh komputer [19]. Format pertukaran data JSON tidak bergantung pada bahasa pemrograman yang dipakai, namun susunan data-nya menggunakan aturan yang mirip dengan bahasa pemrograman C. JSON diperkirakan dapat menguraikan data seratus kali lebih cepat dibandingkan dengan XML di *modern browser*.

BAB III METODE PENELITIAN

Pada bab ini dijelaskan objek, peralatan, dan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan setiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Studi Literatur

Pada tahap ini akan dilakukan pencarian jurnal-jurnal ilmiah, pencarian buku dan sumber referensi lainnya. Mempelajari dan memahami konsep Penulusuran Web Terpersonalisasi (PWS), Personalisasi berbasis profil, *CodeIgniter*, konsep sebuah *Google API*, bagaimana mengimplementasikan Algoritma *Greedy* untuk mendapatkan metadata dari server yang menyediakannya, serta penyaringan berdasarkan privasi pengguna dari *metadata* yang telah didapat dengan menggunakan *database MYSQL*.

3.2 Pengumpulan Data

Pada tahap ini akan dilakukan pengumpulan data profil pengguna yang dibutuhkan. Data yang dibutuhkan didasarkan pada hasil *Query/Keyword* pertanyaan dan persyaratan yang diajukan pengguna saat sedang melakukan pencarian dengan mesin pencari web (*web search engine*) yang kemudian akan membangun suatu *Taxonomy Repository*. Data hasil *Query* tersebut kemudian diintegrasikan sebagai parameter dalam mengambil *metadata* pada *Google API* untuk dijadikan sumber data.

3.3 Analisis dan Perancangan Sistem

Pada tahapan ini akan dilakukan analisis sistem, analisis kebutuhan sistem, perancangan algoritma, perancangan antarmuka sistem, dan perancangan *database*. Sistem ini memiliki masukan ke *database*, masukan tersebut adalah *metadata* berupa teks yang disediakan oleh server. *Metadata* ini diperoleh melalui *keyword* yang tidak mengandung privasi dengan menggunakan algoritma yang ada, sehingga *metadata* yang diinginkan sesuai dengan *keyword* dari pengguna tersebut

3.4 Implementasi Sistem

Pada tahap ini akan dilakukan implementasi sistem seperti pada blok diagram yang di tunjukkan pada Gambar 3.1. Tahap implementasi pada Gambar 3.1 dikelompokkan menjadi :

1. Pengumpulan Data Pengguna

Data implisit diperoleh dari aktivitas pengguna yang terekam pada *Web Server Logs* melalui *cookies* atau *session* dan *Query* yang dimasukkan sebagai *Keyword*. Data eksplisit diperoleh dari lembar pendaftaran (*Login*) dan rating kuisioner.

2. Generalisasi Profil Pengguna

Generalisasi profil pengguna untuk setiap *query* disesuaikan dengan persyaratan privasi yang diajukan. Profil pengguna didapatkan dari *query* atau *keyword* yang dimasukkan pengguna. Kemudian dari masukan tersebut akan dilakukan generalisasi profil menggunakan rancangan implementasi Algoritma *Greedy* sehingga menghasilkan profil umum pengguna yang sudah terpisah dengan privasi pengguna.

3. Pengambilan Data ke Server (Google)

Selanjutnya, profil umum pengguna yang didapat pada proses generalisasi akan ditambahkan dengan *Taxonomi Repository* untuk kemudian dijadikan data *query* umum sebagai parameter yang dikirimkan ke *Server Google API Custom Search* dalam mengambil *metadata*.

4. Proses Pencarian yang Disesuaikan

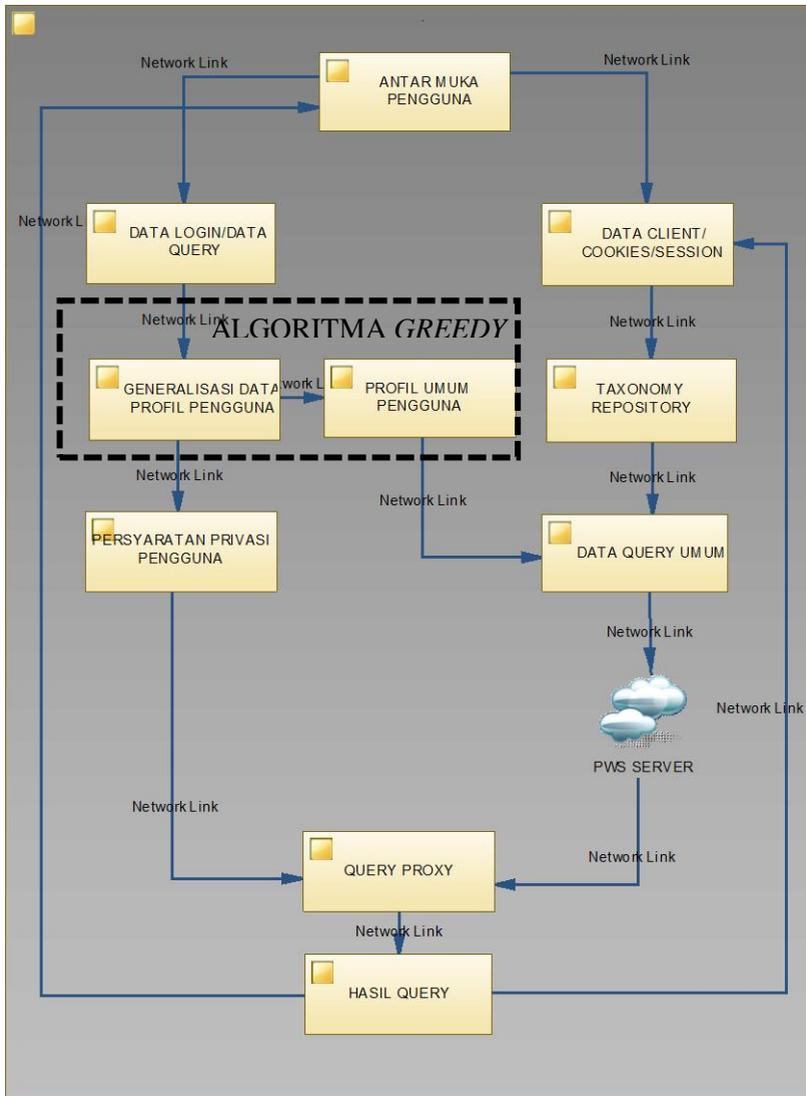
Metadata yang diperoleh dari *server* akan disimpan di *database*. Setelah itu data-data tersebut akan disaring lagi dengan persyaratan privasi pengguna sehingga menghasilkan *query proxy*. Dari hasil *query proxy* kemudian akan di susun kembali menghasilkan hasil *query* yang relevan sesuai dengan permintaan pengguna yang kemudian akan ditampilkan di antar muka pengguna.

3.5 Analisis Hasil Pengujian

Tahap terakhir dari penelitian ini adalah menganalisis hasil implementasi yang telah dilakukan. Pengujian dinyatakan berhasil jika keseluruhan fungsi dapat berjalan dengan baik. Apabila ada beberapa fungsi yang tidak berjalan dengan baik maka dari analisis ini diharapkan diketahui penyebab tidak berjalannya suatu fungsi, sehingga kekurangan sistem dapat diperbaiki.

3.6 Kesimpulan dan Saran

Tahap penarikan kesimpulan merupakan proses dimana dilakukan penarikan kesimpulan terhadap hasil yang telah dicapai, serta diberikan saran sebagai masukan dan perbaikan untuk penelitian selanjutnya.



Gambar 3.1. Blok Diagram Alur Implementasi Sistem

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai analisis dan perancangan sistem dimulai dari deskripsi sistem dan kebutuhan sistem, menyusun kembali hasil pencarian, penggunaan API, penggunaan DBMS yaitu MYSQL untuk penyimpanan dan pengolahan data, serta penjelasan mengenai alur sistem yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1 Analisis Sistem

Pada sub bab ini menjelaskan proses analisis dalam membangun sistem pencarian web. Analisis sistem terbagi menjadi 2, yaitu deskripsi sistem dan analisis kebutuhan sistem. Deskripsi sistem merupakan penjelasan umum sistem secara keseluruhan. Analisis kebutuhan sistem merupakan analisis terhadap kebutuhan sistem seperti perangkat keras dan perangkat lunak.

4.1.1 Deskripsi Sistem Secara Umum

Sistem yang akan dibangun merupakan sistem pada Pencarian Web Terpersonalisasi (PWS). Pada sistem ini difungsikan untuk memberikan informasi hasil pencarian yang relevan dengan apa yang diinginkan pengguna sistem sesuai dengan *Query/Keyword* atau kata kunci yang dimasukkan. Hasil yang ditampilkan sistem kepada pengguna merupakan hasil pencarian yang sudah melalui proses pencarian dengan mengutamakan perlindungan privasi pengguna dalam proses personalisasi data pengguna itu sendiri. Sistem memanfaatkan DBMS untuk pengolahan data yang telah didapat dan menggunakan *framework Codeigniter* untuk membangun aplikasi

berbasis web. Melalui penyimpanan data pada *database*, sistem juga melakukan pengolahan riwayat pencarian pengguna guna memaksimalkan proses personalisasi. Sumber data diperoleh melalui pengambilan *metadata Google API*. *API* yang dimanfaatkan merupakan *Google API Custome Search*.

Masukkan sistem berupa data *Login/data Query/Keyword* atau kata kunci yang dimasukkan pengguna. Data *Login* digunakan untuk memaksimalkan proses personalisasi. *Query/Keyword* dijadikan sebagai profil pengguna yang kemudian digunakan untuk menentukan bagian mana saja yang sensitive (privasi khusus) dan tidak sensitive (privasi umum) pengguna. Proses penentuan tersebut ditujukan untuk perlindungan privasi pengguna dalam pengambilan (akuisisi) data dari *Server*. Data yang didapat dari *Server* kemudian akan di simpan di *database*. Proses selanjutnya yaitu pengolahan data sebelum ditampilkan kembali ke pengguna melalui halaman antarmuka (*Interface*). Kenaikan Kualitas yang dimaksud didalam Tugas Akhir ini adalah kesesuaian hasil pencarian yang ditampilkan ke pengguna dengan preferensi/keinginan pengguna dibandingkan dengan hasil yang ditampilkan.

4.1.2 Analisis Kebutuhan Sistem

Aplikasi ini dibangun dengan menggunakan *software XAMPP* yang fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), menggunakan *software Notepad ++* sebagai editor *script PHP*, aplikasi *Firefox* untuk menampilkan halaman *interface* dan *DBMS (Data Base Management Sistem)* yang digunakan adalah *MYSQL*.

Ada beberapa aspek yang dibutuhkan sistem dalam membangun sebuah aplikasi pencarian yang menjadi dasar untuk perancangan sistem adalah sebagai berikut :

1. Sistem mampu menangkap *Query* yang diberikan pengguna dan menjadikannya sebagai profil pengguna untuk kemudian digeneralisasi dengan Algoritma *Greedy*
2. Sistem mampu memisahkan antara persyaratan privasi pengguna (privasi khusus) dengan data profil umum pengguna (privasi umum)
3. Sistem mampu melakukan *Query* pengambilan data dari *Server*, yaitu *Google*.
4. Sistem mampu melakukan proses personalisasi data pengguna
5. Sistem mampu mengelompokkan data berdasarkan 2 kategori yaitu *Website* dan *Image*.
6. Sistem mampu menyimpan data ke database berdasarkan kategori.
7. Sistem mampu menyaring kembali data yang diberikan oleh masing-masing server sesuai dengan kebutuhan berdasarkan kesamaan judul website dengan privasi khusus pengguna
8. Sistem mampu menampilkan kembali data yang telah dimasukkan ke database dalam bentuk informasi dari masing-masing kategori.

4.2 Perancangan Sistem

Berdasarkan tujuan dari Tugas Akhir ini yaitu membangun sistem perlindungan privasi pengguna tanpa mengorbankan kualitas pencarian pada mesin pencari web (*Web Search Engine*) dalam Penelusuran Web Terpersonalisasi (PWS) berbasis profil yang sudah ada, maka akan dibutuhkan perancangan sistem. Sistem yang akan dibangun menggunakan *Server Google*. *Server* tersebut

digunakan karena memberikan layanan API secara gratis dimana syarat untuk penggunaan API adalah memiliki *API Key*.

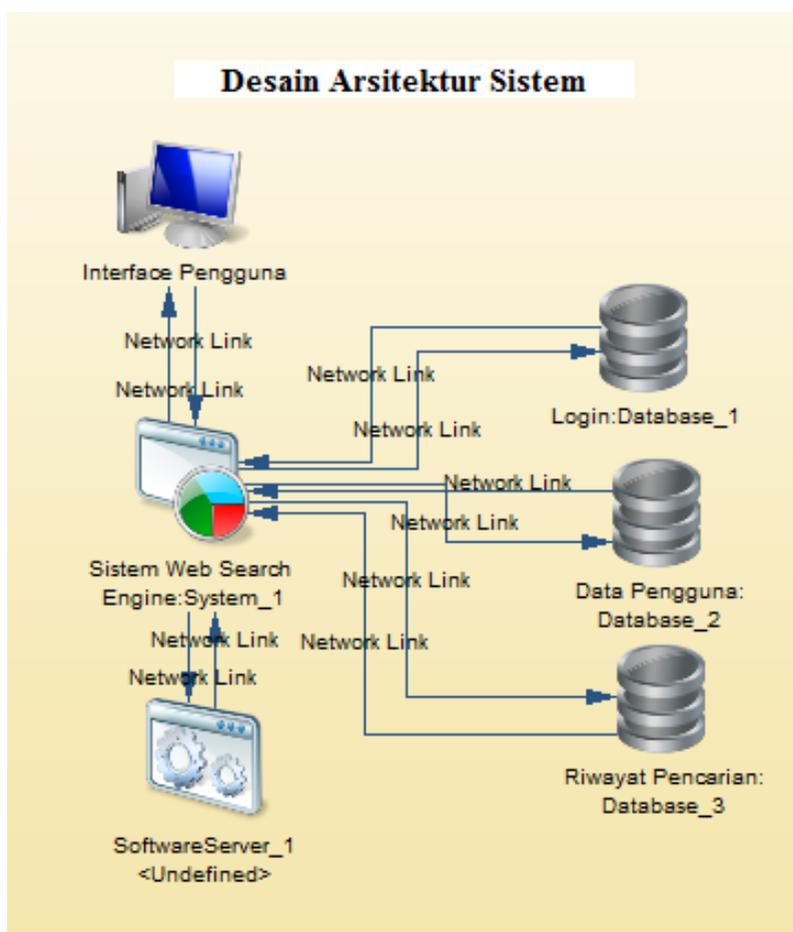
Data yang dapat diambil (akuisisi) dari server dikelompokkan menjadi 2 kategori yaitu *Website* dan *Image*. Sistem menggunakan perangkat lunak manajemen basis data yaitu MySQL yang fungsinya untuk mengelola data yang telah diambil dari server. Tujuannya untuk menyatukan data dengan kategori yang sama dimana data yang akan dimasukkan kedalam database disaring terlebih dahulu. Hal ini dilakukan karena hasil output dari sistem nantinya hanya berupa informasi yang penting seperti judul, *Uniform Resource Identifier (URI)*, deskripsi, *Link Image*, dan *Link Thumbnail*. Setelah semua informasi masuk kedalam database, informasi yang ada pada database diolah dan kemudian ditampilkan sesuai dengan kategori masing-masing sesuai keinginan user. Kemudian selanjutnya perancangan sistem dilanjutkan pada tahap yang meliputi gambaran sistem, perancangan alur sistem, perancangan alur penggunaan API, perancangan *Database*, dan perancangan halaman antarmuka (*interface*).

4.2.1 Gambaran Sistem

Berikut gambaran tentang arsitektur dari sistem aplikasi pencarian yang berupa proses berjalannya sistem secara umum yang disajikan pada Gambar 4.1. proses berjalannya sistem terbagi menjadi 4 yaitu :

Pengumpulan data pengguna, generalisasi profil pengguna, pengambilan data ke server, dan proses pencarian yang disesuaikan. Sistem melakukan generalisasi menggunakan Algoritma *Greedy* terhadap *Query/Keyword* yang digunakan

sebelum mengakuisisi data. Data yang diambil berupa teks dengan format JSON dan kemudian disimpan ke dalam database.



Gambar 4.1. Desain Arsitektur Sistem

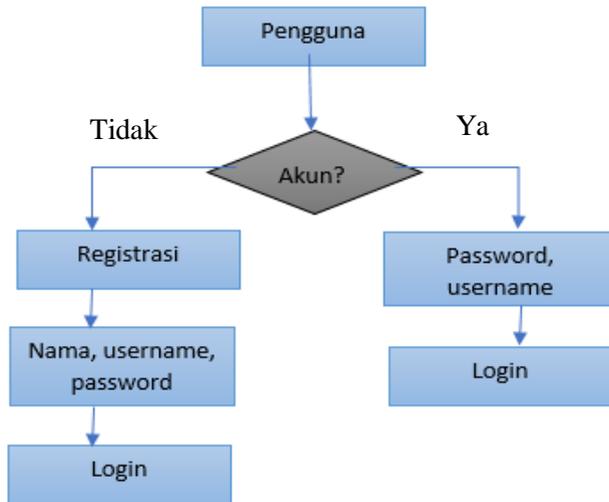
Selanjutnya metadata dari *Google API* yang telah diakuisisi dikelompokkan berdasarkan kategori yang telah ditentukan. Data yang tersimpan di database disaring lagi dengan privasi pengguna sehingga diperoleh data yang sesuai dengan keinginan pengguna untuk kemudian di tampilkan di halaman *Interface*. Pada tampilan hasil terdapat 2 kategori pencarian yang telah didapat yaitu *Website* dan *Image*. Perancangan dibagi menjadi beberapa tahapan sebagai berikut :

4.2.1.1 Desain Proses Login Pengguna

Sistem yang dibuat ditujukan untuk peningkatan kualitas personalisasi. Proses personalisasi akan lebih baik jika menggunakan sistem data personal. Sistem data personal yang dimaksud disini adalah sistem yang membuat setiap data memiliki pemilik, sehingga setiap pengguna memiliki datanya masing-masing dan tidak akan bertukar dengan pengguna lainnya. Hal ini bertujuan untuk membuat relevansi data pencarian meningkat. Sistem yang dibuat akan mengetahui keinginan atau ketertarikan dari pengguna sistem lewat personalisasi data dengan memanfaatkan data Login pengguna. Desain dapat dilihat pada Gambar 4.2.

4.2.1.2 Desain Proses Pencarian Data

Saat melakukan pencarian, dibutuhkan suatu parameter yang digunakan untuk menentukan apa yang akan dicari dan apa yang akan ditampilkan di halaman *Interface*. Proses pencarian *Web Search Engine* pada umumnya akan menggunakan *Query/Keyword* yang dimasukkan sebagai parameter secara utuh tanpa melalui proses yang lainnya. Berbeda dengan sistem yang akan dibuat pada Tugas Akhir ini. Sistem yang dibuat pada Tugas Akhir ini akan menjadikan *Query/Keyword* masukkan sebagai hal utama untuk menentukan hasil pencarian yang akan ditampilkan ke pengguna

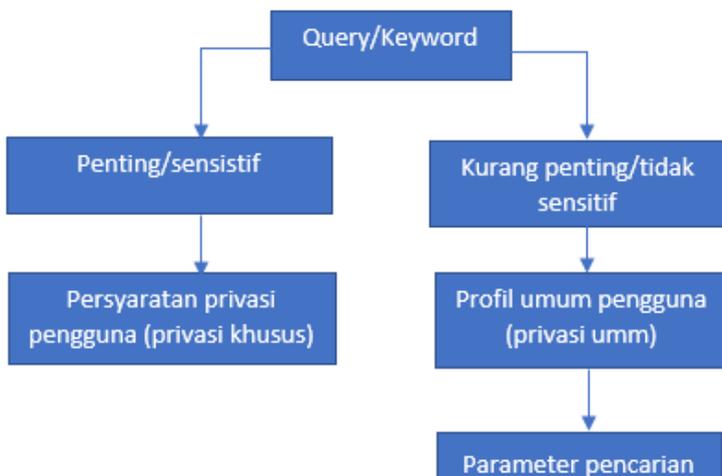


Gambar 4.2. Desain Proses Login Pengguna

tanpa harus menjadikan semua masukan sebagai parameter pencarian data ke *Server*. Pada proses pencarian data ini, sistem yang dibuat akan mengadaptasi desain alur dari *Framework User customizable privacy preserving search* (UPS). Ide awal dari *Framework* ini adalah membuat keputusan bagian mana saja dari profil pengguna yang akan disembunyikan sementara oleh sistem, dan bagian mana saja yang akan di rambahkan sebagai parameter ke *Server* sehingga privasi pengguna terlindungi. Desain pencarian data terbagi menjadi desain penentuan parameter, akuisisi data dan filter data.

4.2.1.3 Desain Penentuan Parameter Pengambilan Data API

Saat proses pengambilan data ke server dibutuhkan parameter tertentu. Berikut desain penentuan parameter data API dapat dilihat pada Gambar 4.3.



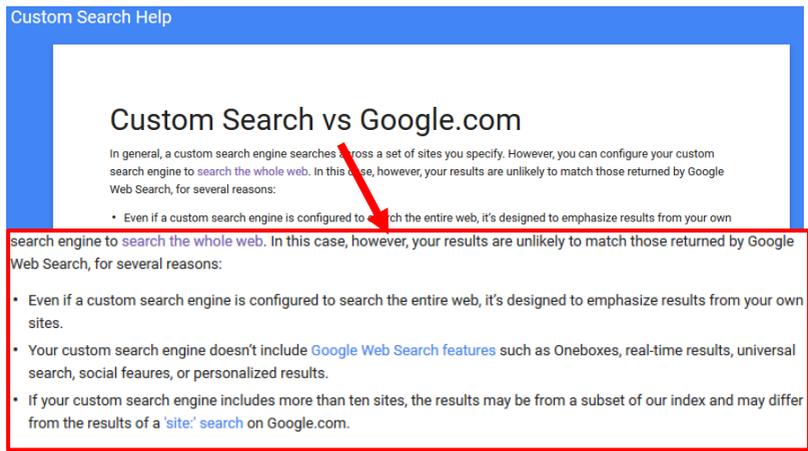
Gambar 4.3. Desain Penentuan Parameter

Pada sistem ini, parameter yang diberikan ke server untuk menangkap data bukan merupakan *Query/Keyword* yang dimasukkan, melainkan data *Query* umum pengguna. Sistem tidak menggunakan *Query/Keyword* secara langsung karena diperlukan adanya perlindungan privasi pengguna sebelum data pengguna dirambahkan ke *Server*. Sistem akan melakukan generalisasi profil pengguna dengan Algoritma *Greedy* dari data *Login/data Query/Keyword* yang menghasilkan profil umum pengguna (privasi umum), kemudian profil umum akan diakumulasi dengan *Taxonomy Repository* untuk mendapatkan data *Query* umum pengguna. Sehingga parameter pencarian yang digunakan sistem

untuk mengambil data di *Server* merupakan bagian yang tidak sensitif (privasi umum) dari profil pengguna, sedangkan bagian yang sensitif (privasi khusus) pengguna akan disembunyikan sementara oleh sistem untuk nantinya akan dilakukan penyaringan kembali. Data diambil dari *Google API Custome Search*.

1. *Google API*

Google menyediakan banyak jenis API yang dapat dimanfaatkan oleh *Developer*. Pada Tugas Akhir ini sistem yang dirancang akan menggunakan *Google API Custome Search*. *Google API Custome Search* merupakan satu-satunya API *Google* yang dapat digunakan untuk mendapatkan data semua *Website* hasil pencarian mesin pencari web (*Web Search Engine*), sama seperti data saat kita mencari di mesin pencari *Google*. Tetapi terdapat beberapa perbedaan mengenai hasil pencarian menggunakan *Google API Custome Search* dengan *Google.com*. Perbedaan dapat dilihat pada Gambar 4.4.



Gambar 4.4. Perbedaan Hasil Pencarian [20]

Hasil pencarian menggunakan *Google Custome Search* tidak selalu sama dengan hasil pencarian dari *Google Web Search*. Hal ini terjadi karena beberapa alasan yaitu :

1. Meskipun *Custome Search Engine* berkonfigurasi dengan keseluruhan web, tetapi itu didesain untuk menekankan pencarian terhadap web site kita sendiri.
2. *Custome Search Engine* tidak menyertakan fitur-fitur dalam *Google Web Search* seperti *Oneboxes*, hasil *Real-Time*, pencarian umum, fitur social, dan hasil pencarian terpersonalisasi.

4.2.1.4 Desain Implementasi Algoritma Greedy

Proses ini dimulai saat data *Query/Keyword* yang dimasukkan pengguna sudah ditangkap oleh sistem. Data *Query/Keyword* tersebut akan dijadikan sistem sebagai profil pengguna. Dari data tersebut kemudian sistem akan menjeneralisasi profil pengguna sehingga sistem dapat memisahkan data profil umum/ bagian tidak sensitif pengguna (privasi umum) dengan persyaratan privasi/ bagian sensitif pengguna (privasi khusus). Setelah didapat data privasi, sistem akan memberikan keputusan bagian mana yang akan dijadikan parameter pencarian saat mengambil data ke *Server* dan bagian mana yang akan disimpan sistem untuk sementara. Implementasi Algoritma *Greedy* berdasarkan pengelompokkan *Keyword* secara umum. Dimana kebiasaan orang pada umumnya ketika mencari informasi, *Keyword* yang dimasukkan pastilah memiliki motif. Hal ini yang biasa disebut dengan *Search Intent*.

Skema umum Algoritma *Greedy* :

1. Himpunan kandidat

Himpunan kandidat pada penelitian ini adalah himpunan yang merepresentasikan jumlah kata pada *keyword* atau *query* masukan. Jumlahnya paling sedikit satu kata.

2. Himpunan solusi

Total kata yang dipilih dari bagian tidak sensitif (umum) dan bagian sensitif (khusus) memiliki panjang *array* yang tepat sama dengan panjang array *Query/Keyword* masukkan.

3. Fungsi seleksi

Pada fungsi selektif, pilihlah *array* dengan indeks terkecil dari himpunan kandidat. Aturan pada fungsi ini adalah sebagai berikut:

- a. Jika panjang *array* adalah 1, maka pilih semua *array* untuk jadi bagian tidak sensitif (umum).
- b. Jika panjang *array* adalah 2, maka pilih indeks ke-0 untuk menjadi bagian tidak sensitif (umum) dan indeks ke-1 untuk menjadi bagian sensitif (khusus).
- c. Jika panjang *array* adalah n , maka pilih indeks ke-0 dan indeks ke-1 untuk menjadi bagian tidak sensitif (umum). Sedangkan sisanya menjadi bagian sensitif (khusus)

4. Fungsi layak

Fungsi layak berfungsi untuk memeriksa apakah *array* dapat dipisahkan menjadi bagian tidak sensitif (umum) dan bagian sensitif (khusus) dengan mempertimbangkan panjang *array*.

5. Fungsi objektif

Fungsi objektifnya adalah jumlah bagian sensitif (khusus) maksimum dan jumlah bagian tidak sensitif (umum) minimum.

Pseudocode Algoritma Greedy dapat dilihat sebagai berikut:

Algoritma menentukan privasi umum dan khusus
(input datakeyword : himpunan array, i :panjang array)

{membagi array yang memiliki panjang A menjadi bagian tidak sensitif (umum) dan bagian sensitif (khusus). Tetapi bagian tidak sensitif minimum dan bagian sensitif maksimum}

Deklarasi

Datakeyword \leftarrow {himpunan array kata}

i \leftarrow panjang array datakeyword

privasi_umum \leftarrow array (bagian tak sensitif (privasi umum))

privasi_khusus \leftarrow array (bagian sensitif (privasi khusus))

Algoritma

Parameter \leftarrow Keyword/Query masukkan

Datakeyword \leftarrow array dari Keyword

i \leftarrow panjang array

privasi_umum \leftarrow {}

privasi_khusus \leftarrow {}

if (i!=0){ then

 if (i==1){ then

 privasi_umum[0] \leftarrow datakeyword[0];

 }

 else if (i==2){ then

```

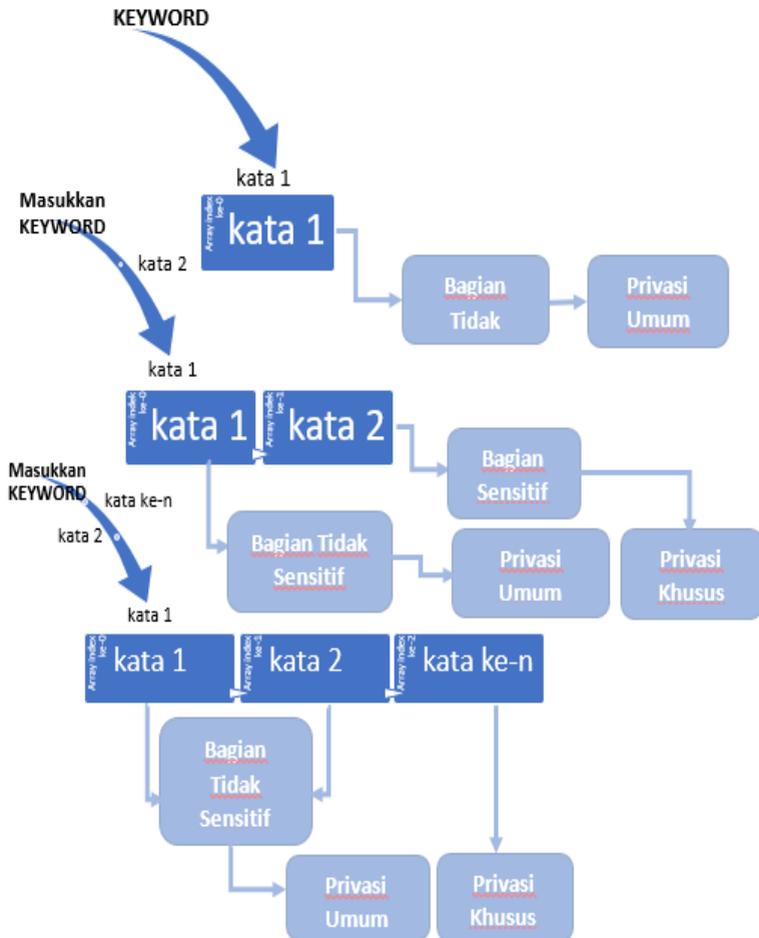
        privasi_umum[0] ← datakeyword[0];
        privasi_khusus[0] ← datakeyword[1];
    }
    else { then
        privasi_umum[0] ← datakeyword[0] + datakeyword[1];
        for (x=2; x <= i; x++){
            then
            privasi_khusus[0] ← datakeyword[x];
        }
    }
    While (panjang privasi_umum + panjang
    privasi_khusus == panjang array datakeyword){
        Privasi_umum['database_keyword'] ← privasi_umum[0];
        Privasi_khusus['database_privasi'] ← privasi_khusus[0];
    }
}endif

```

Sistem yang dirancang akan berjalan seperti yang dijelaskan pada Gambar 4.5.

4.2.1.5 Desain Pengambilan (Akuisisi) Data

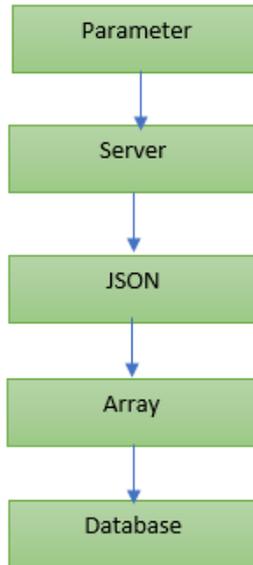
Setelah mencari dan mendapatkan data dengan menggunakan *Query* umum (privasi umum) pengguna, data yang didapat disimpan sementara didalam sistem dengan mengubah format data sebelumnya yaitu JSON menjadi *array*. Informasi data



Gambar 4.5. Skema Algoritma Greedy

diakuisisi sesuai kebutuhan yaitu data mengenai judul, URI, deskripsi, *Link Image*, dan *Link Thumbnail*. Lalu data yang telah

diakuisisi kemudian dimasukkan ke *Database*. Berikut desain proses akuisisi data dapat dilihat pada Gambar 4.6 dan Algoritma akuisisi data yang digunakan data dapat dilihat sebagai berikut:



Gambar 4.6. Desain Akuisisi Data

Algoritma Akuisisi Data

{mengambil data dari *Server* dan menyimpannya ke *Database*}

Deklarasi

datakeyword \leftarrow string (privasi umum)

id_user \leftarrow integer (id pengguna)

data_api \leftarrow array (data yang diakuisisi)

Deskriptif

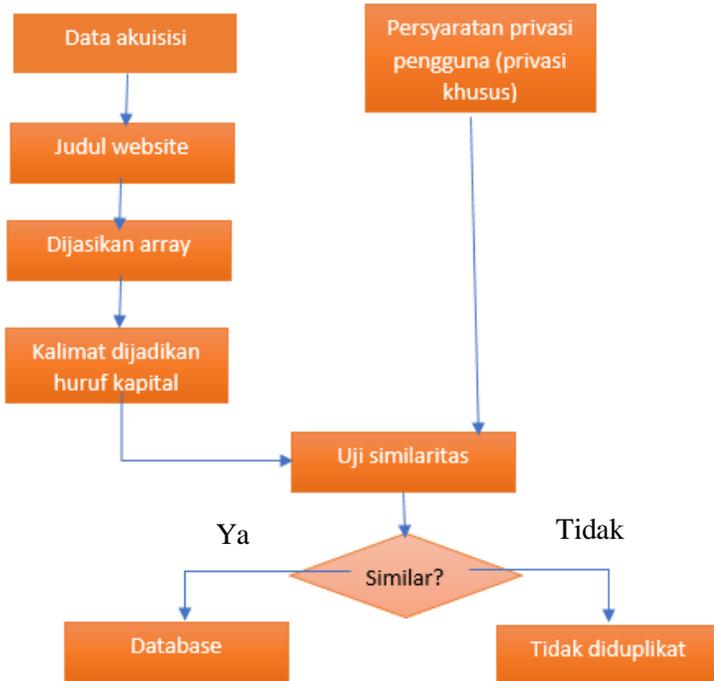
1. Mulai

2. Menjadikan privasi umum ke dalam bentuk *string* dan masukkan ke *datakeyword*
3. Merambahkan *datakeyword* sebagai parameter pengambilan data ke *Server*
4. Menangkap *Metadata* dari *Server* berupa JSON
5. Men-decode JSON ke dalam bentuk *array*
6. Mengambil data sesuai kebutuhan yaitu judul, deskripsi, URI, *Link Image*, dan *Link Thumbnail*
7. Menyimpan data ke dalam *Database*
8. Selesai

4.2.1.6 Desain Penyaringan (Filter) Data

Proses *Filtering* data dimulai ketika data yang diakuisisi telah dimasukkan ke *Database*. Data tersebut akan disaring lagi dengan mencari kesamaan antara judul website dengan bagian sensitif dari profil pengguna (privasi khusus)/persyaratan privasi pengguna yang sebelumnya telah di sembunyikan sistem untuk sementara, sehingga hasil yang nanti ditampilkan sudah relevan dengan apa yang diinginkan pengguna. Berikut desain proses filter data yang digunakan dapat dilihat pada Gambar 4.7.

Hasil akuisisi data yang dibutuhkan disimpan ke dalam *array* dimana setiap data yang ada didalam *array* diberikan label untuk membedakan data yang masuk tersebut. Kemudian data *array* di filter dan dimasukkan ke table yang lain untuk memudahkan pengolahan riwayat pencarian pengguna serta tampilan data di halaman *Interface*. Algoritma *filter* data yang digunakan pada sistem yang dibuat dijelaskan sebagai berikut:



Gambar 4.7. Desain Filter Data

Algoritma Filtering Data

{menyaring data pada database yang similar dengan privasi khusus pengguna berdasarkan judul website}

Deklarasi

upper ← array (privasi khusus dijadikan huruf besar)

```
upper1 ← array (judul Website dijadikan huruf
besar)
```

```
data1 ← array (data filtering berupa keterangan
Website)
```

```
data2 ← array (data filtering berupa keterangan
Image)
```

```
datafilter ← array (data hasil filtering)
```

Algoritma

```
upper ← privasi khusus dijadikan huruf besar
```

```
upper1 ← judul Website dijadikan huruf besar
```

```
if (pengujian similaritas upper dan upper1
dijadikan presentase){
```

```
    if (presentase > 10){ then
```

```
        data1 ← data hasil filtering berupa
                keterangan Website
```

```
        data2 ← data hasil filtering berupa
                keterangan Image
```

```
        datafilter ← data1+data2; }endif
```

```
    }endif
```

Sebagai contoh jika memasukkan Query/Keyword “*laptop*” (yang dirambahkan ke *Google API* yaitu “*laptop*”) dan “*apple smartphones*” (yang dirambahkan ke *Google API* yaitu “*apple*”, sedangkan “*smartphones*” akan disembunyikan sementara oleh sistem) maka akan diperoleh *metadata* seperti pada Gambar 4.8 dan 4.9.

```

kind: "customsearch#search"
url:
  type: "application/json"
  template: "https://www.googleapis.com/customsearch/v1?q={searchTerms}&num={count}&start={startIndex}&lr={googleHost}&c2coff={disableCnTwTranslation}&hq={hq}&hl={hl}&siteSearch={siteSearch}&siteSearchTerms={orTerms}&relatedSite={relatedSite}&dateRestrict={dateRestrict}&lowRange={lowRange}&imgType={imgType}&imgColorType={imgColorType}&imgDominantColor={imgDominantColor}&alt=json"
queries:
  request:
    0:
      title: "Google Custom Search - apple"
      totalResults: "766000000"
      searchTerms: "apple"
      count: 10
      startIndex: 1
      inputEncoding: "utf8"
      outputEncoding: "utf8"
      safe: "off"
      cx: "00505199285557048902:l66ryr_og-q"

```

Gambar 4.8. Metadata *Laptop*

```

kind: "customsearch#search"
url:
  type: "application/json"
  template: "https://www.googleapis.com/customsearch/v1?q={searchTerms}&num={count}&start={startIndex}&lr={googleHost}&c2coff={disableCnTwTranslation}&hq={hq}&hl={hl}&siteSearch={siteSearch}&siteSearchTerms={orTerms}&relatedSite={relatedSite}&dateRestrict={dateRestrict}&lowRange={lowRange}&imgType={imgType}&imgColorType={imgColorType}&imgDominantColor={imgDominantColor}&alt=json"
queries:
  request:
    0:
      title: "Google Custom Search - laptop"
      totalResults: "297000000"
      searchTerms: "laptop"
      count: 10
      startIndex: 1
      inputEncoding: "utf8"
      outputEncoding: "utf8"
      safe: "off"
      cx: "00505199285557048902:l66ryr_og-q"

```

Gambar 4.9. Metadata *Apple*

Proses Filtering menggunakan `Similar_Text` dari *Library PHP*. Konsep dari `similar_text` yaitu dengan melewati referensi sebagai argumen ketiga, `similar_text ()` akan menghitung kesamaan dalam persen, dengan membagi hasil `similar_text ()` dengan rata-rata panjang string yang diberikan kali 100. Hasil akan mengembalikan jumlah karakter yang cocok di kedua string. Jumlah karakter yang cocok dihitung dengan mencari substring umum terpanjang pertama, dan kemudian melakukan ini untuk prefiks dan sufiks, secara rekursif. Panjang semua substring umum ditambahkan. Deskripsi `similar_text` dapat dilihat pada Gambar 4.10.

A screenshot of a code editor or documentation window. The title bar at the top is labeled "Description" in a purple font. Below the title bar, the PHP function signature for `similar_text` is displayed in a monospaced font with syntax highlighting: `int similar_text (string $first , string $second [, float &$percent])`.

```
Description

int similar_text ( string $first , string $second [, float &$percent ] )
```

Gambar 4.10. Deskripsi `similar_text()`

4.2.1.7 Desain Proses Personalisasi Pengguna

Sistem bekerja pada Penelusuran Web Terpersonalisasi (PWS) sehingga diperlukan adanya proses personalisasi untuk menaikkan kualitas hasil pencarian. Proses personalisasi ini berhubungan dengan data *Login* pengguna serta riwayat pencarian pengguna itu sendiri. Setiap pengguna memiliki tanda pengenal (*ID Login*) yang berbeda-beda. Setiap data yang diakuisisi akan tersimpan di database sesuai dengan tanda *ID Login* masing-masing pengguna.

Hal ini akan memudahkan sistem dalam proses akumulasi data riwayat pencarian. Sistem juga akan lebih mudah membedakan kepemilikan data, sehingga data yang ditampilkan nantinya akan lebih relevan dan sesuai dengan keinginan pengguna yang berbeda-beda.

Misalkan data *Query/Keyword* sudah digeneralisasi sehingga menghasilkan profil umum pengguna dan persyaratan privasi pengguna (privasi khusus). Penambahan data profil umum pengguna dengan *Taxonomy Repository* (riwayat pencarian pengguna) yang menghasilkan data *Query* umum (privasi umum). Algoritma personalisasi untuk sistem yang dibuat dijelaskan sebagai berikut:

Algoritma Personalisasi Data Pengguna

{mempersonalisasi data pengguna sehingga hasil pencarian lebih relevan dengan keinginan masing-masing pengguna}

Deklarasi

```
Datahistory ← array      datakeyword ← array
jumlah_data ← integer (panjang array datahistory)
id_user ← integer        cek1 ← array
privasi_umum ← array     privasi_khusus ← array
nilai ← integer (panjangcek1)
i ← integer (panjang Query/Keyword masukkan)
datafilter ← array
```

Algoritma

```
privasi_umum ← datakeyword
id_user ← id pengguna
datahistory ← get history keyword pengguna dengan
privasi_umum dan id_user pada database keyword
(privasi umum)
jumlah_data ← panjang array datahistory
if (jumlah_data ← 0) { then
```

```

cek1 ← get history keyword dengan
      privasi_umum dan id_user pada
      database privasi (privasi khusus)
nilai ← panjang cek1
if (nilai<0){then
  if (i<1){
    insert data ke database Website
    dan Image
    print data dari database
    Website dan Image
  }
  else {
    if (pengujian similaritas judul
    site dan privasi khusus){
      if (presentase >10){
        insert data ke
        database data filter
        print data dari
        database data filter
      }
    }
  }
else{ then
  if (i<1){
    insert data ke database Website
    dan Image
    print data ke database Website
    dan Image ditambah database
    data filter
  }
  else{
    insert data ke database data filter
    print data dari database data filter
    (berdasarkan privasi umum dan
    privasi khusus)
  }
}
}
else {then
  if (i<1){then
    print data_api;
  }
}

```

```

    }
else{
    insert data ke database data filter
    print data dari database data filter
}
else{then
    print data ke database data filter
}
}
}

```

4.2.2 Perancangan Alur Sistem

Perancangan alur sistem dibuat menggunakan UML (*Unified Modeling Language*) metodologi untuk membangun sistem OOP (*Oriented Object Programming*) yang diantaranya membuat *Use Case Diagram*, dan *Swimline Diagram*. Secara garis besar alur sistem di bagi dalam tiga tahapan, yaitu perlindungan privasi, proses personalisasi, pengambilan, penyaringan dan penyimpanan data, serta skema database. Berikut penjelasan mengenai metodologi untuk membangun sistem :

4.2.2.1 Use Case Diagram

Use Case Diagram merupakan pemodelan untuk menggambarkan perilaku sistem yang akan dibuat. Sebuah Use Case mempresentasikan sebuah interaksi antara pengguna atau sistem lainnya dengan sistem.

4.2.2.1.1 Use Case : Pencarian Informasi Web Menggunakan Sistem Dengan Mengisi Data Login/Data Query Oleh Pengguna

Primary Actor : Pengguna

Goal in Context : Pengguna melakukan pencarian tentang informasi yang dibutuhkan melalui sistem mesin pencari web yang dibuat dengan terlebih dahulu memasukkan data *Login/ data Query*.

Precondition : Pengguna masuk ke sistem

Trigger : Pengguna memasukkan data *Login/ data Query*

Scenario :

- Pengguna melakukan *Login/Registrasi* ke sistem
- Pengguna memasukkan data *Login/ data Query*
- Sistem memproses data *Query* masukkan
- Sistem menampilkan data hasil pencarian
- Pengguna melihat hasil pencarian melalui halaman *Interface*

Exception :

- Password salah : actor memasukkan kembali password yang benar
- Password tidak dikenali : sistem akan memberikan waktu tunggu untuk mereset ulang password

Frequency of Use : 24 Hours

Channel to Actor : Halaman *Interface*

Secondary Actor : Sistem

Channel to Secondary Actor : Halaman *Interface*

Use Case dapat dilihat pada Gambar 4.11

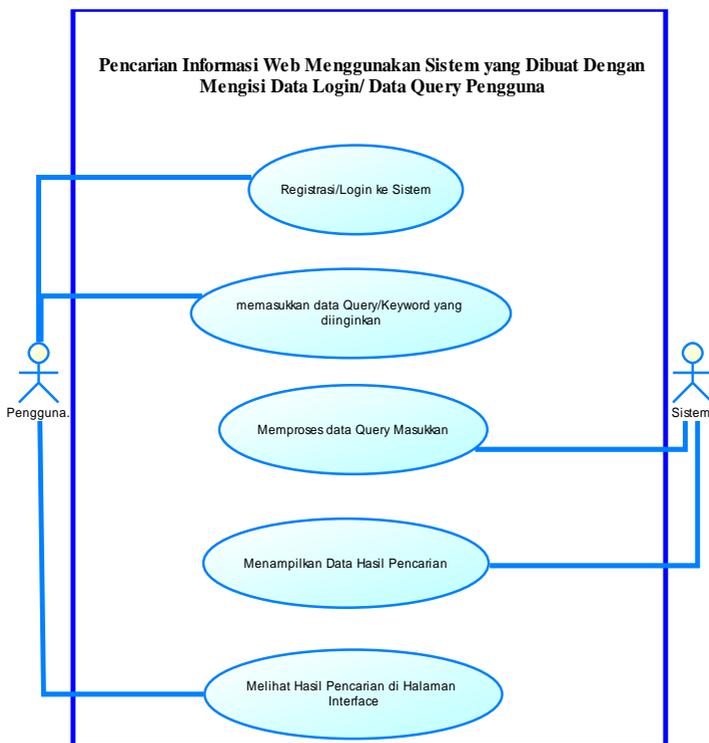
4.2.2.1.2 Use Case : Pemrosesan Data *Login/Data Query* Pencarian Yang Dimasukkan Pengguna Menggunakan Sistem

Primary Actor : Sistem

Goal in Context : Sistem akan memproses data *Login/* data *Query* pencarian dan menggolah profil serta riwayat pencarian pengguna sehingga mendapatkan hasil pencarian yang sesuai dengan keinginan pengguna (relevan).

Precondition : Pengguna memasukkan data *Login/* data *Query*.

Triger : Sistem menangkap data *Login/* data *Query* yang dimasukkan masukkan pengguna.



Gambar 4.11. Use Case Pencarian Informasi Web Menggunakan Sistem Dengan Mengisi Data Login/Data Query

Scenario :

- Pengguna memasukkan data *Login*/ data *Query*
- Sistem menggeneralisasi data yang masuk menggunakan Algoritma *Greedy*
- Sistem menyembunyikan persyaratan privasi pengguna (privasi khusus)
- Sistem merambahkan data *Query* umum (privasi umum) pengguna ke Server
- Sistem mengakuisisi data dari Server
- Sistem menyimpan data yang diakuisisi ke dalam *Database*
- Sistem menyaring data pada *Database* dengan persyaratan privasi (privasi khusus) pengguna
- Sistem mere-rank hasil pencarian dan menampilkannya ke Halaman *Interface* pengguna
- Pengguna melihat hasil pencarian pada halaman *Interface*.

Exception :

- Password salah : actor memasukkan kembali password yang benar
- Password tidak dikenali : sistem akan memberikan waktu tunggu untuk mereset ulang password

Frequency of Use : 24 Hours

Channel to Actor : Halaman *Interface*

Secondary Actor : Pengguna

Channel to Secondary Actor : Halaman *Interface*

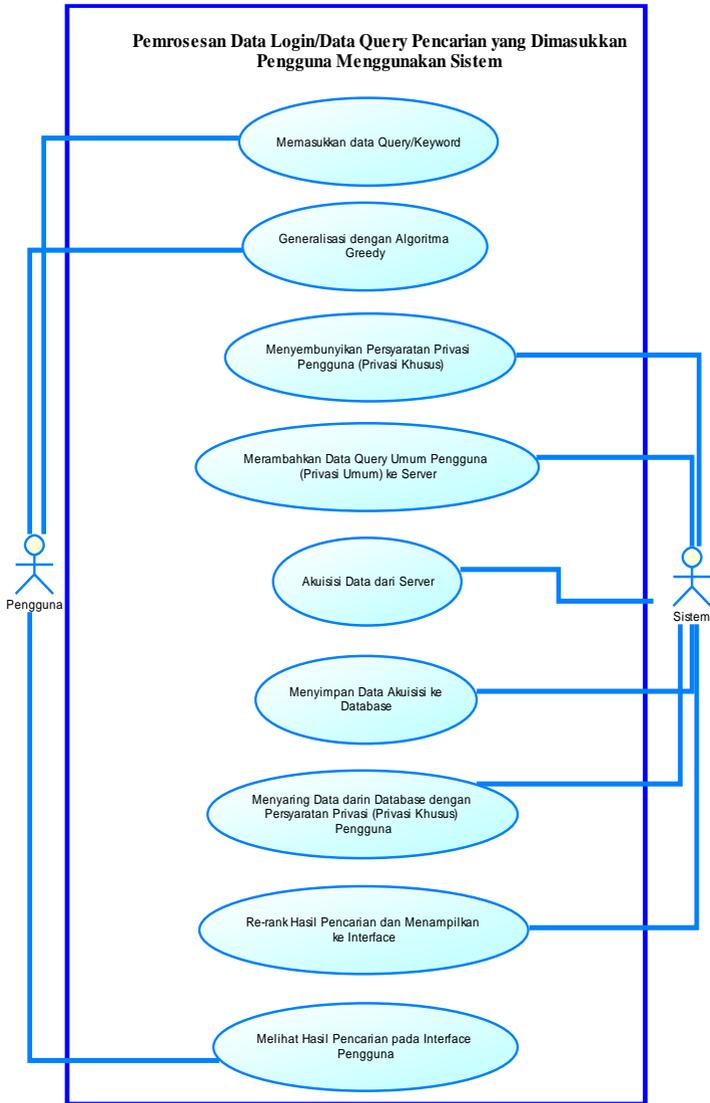
Use Case dapat dilihat pada Gambar 4.12

4.2.2.2 *Swimline Diagram*

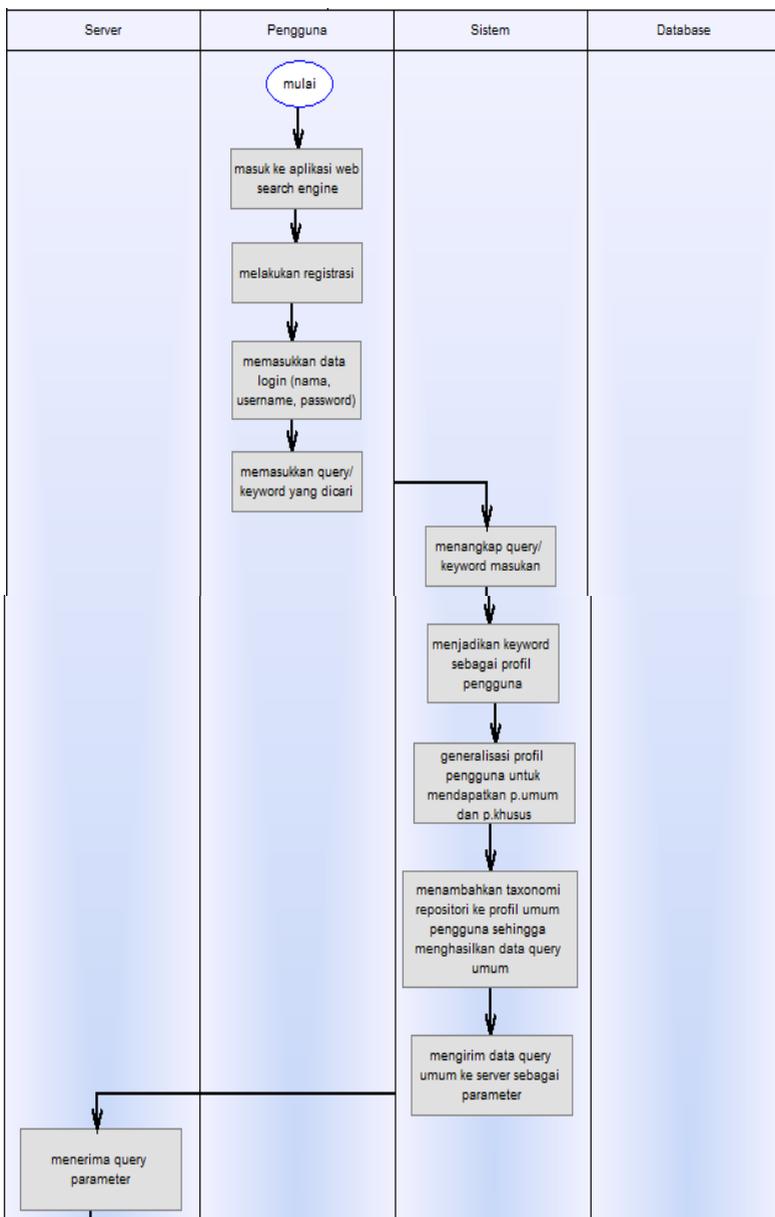
Diagram ini menggunakan format jalur hubungan (*swimline*). Dalam *swimline diagram* ini terdapat beberapa *Stakeholder* yaitu pengguna, sistem dan *Server* yang didasarkan pada hasil analisis kebutuhan pengguna. Hasil Diagram ini dapat dilihat pada Gambar 4.13.

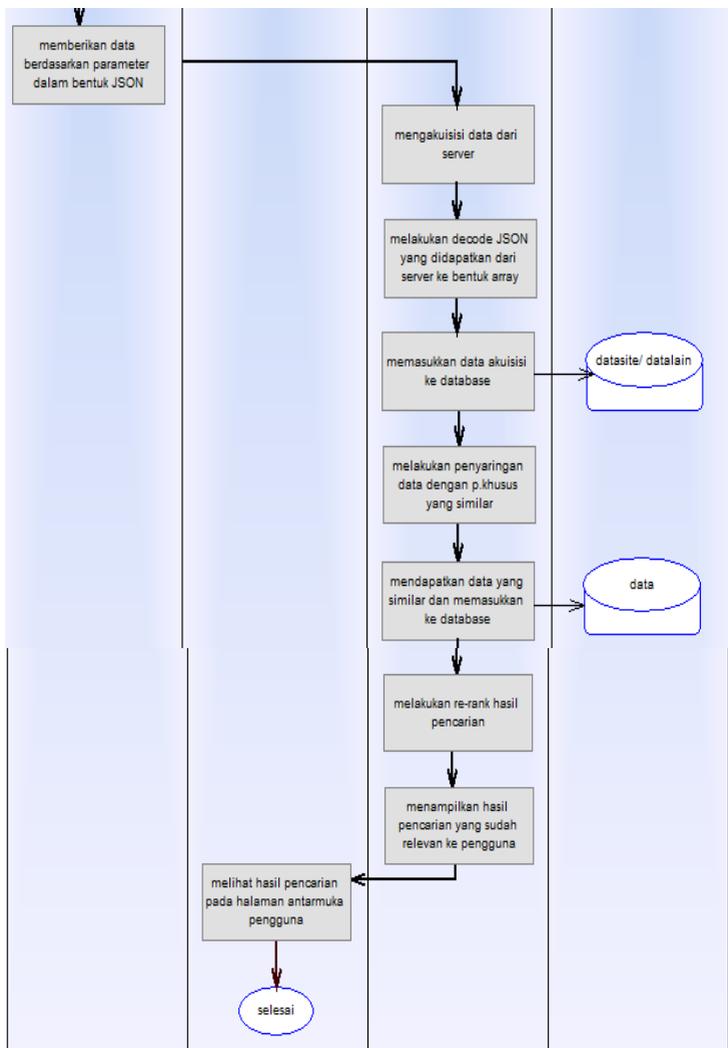
4.2.3 Perancangan Alur Penggunaan API

API merupakan sebuah fungsi siap pakai yang disediakan oleh *Server*. Syarat menggunakan API adalah memiliki *API key*. *API key* (*Application Programming Interface key*) adalah kode atau kunci yang digunakan untuk mengakses layanan yang ingin digunakan. Cara mendapatkan *API key* dari layanan yang ingin digunakan berbeda-beda dan juga penggunaan API dari setiap layanan berbeda. API yang digunakan adalah *Google API Custome Search* yang fungsinya memberikan informasi mengenai semua site. Pencarian site dapat dilakukan dengan mengirimkan permintaan HTTP GET dan hanya memerlukan parameter *q* yang merupakan sebuah keyword untuk mencari informasi site. Format metadata yang didapat yaitu JSON. Adapun parameter tambahan yang diperlukan untuk mencari informasi site dengan *Google API Custome Search* akan ditampilkan pada Gambar 4.14.



Gambar 4.12. Use Case Pemrosesan Data *Login/Data Query* Pencarian Yang Dimasukkan Pengguna Menggunakan Sistem





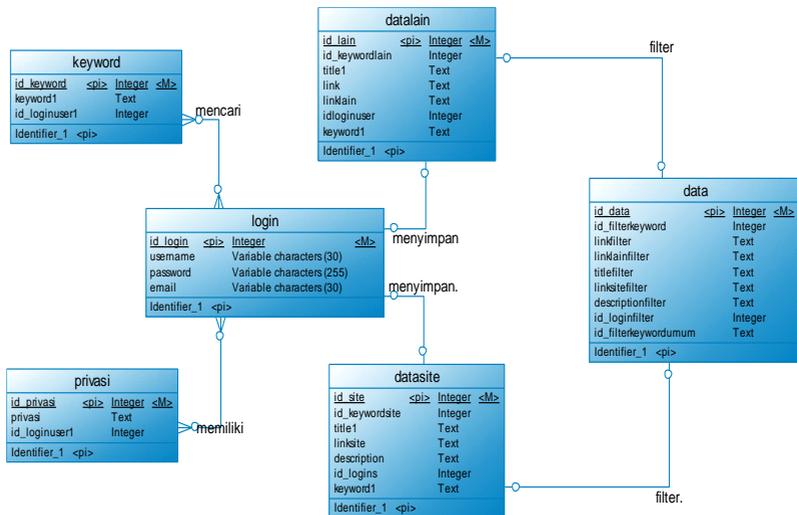
Gambar 4.13. *Swimlane* Sistem

cx	
Description	Required. The <code>cx</code> parameter specifies a unique code that identifies a custom search engine. You must specify a Custom Search Engine using the <code>cx</code> parameter to retrieve search results from that CSE.
	To find the value of the <code>cx</code> parameter, go to Control Panel > Codes tab of your CSE and you will find it in the text area under 'Paste this code in the page where you'd like your search box to appear. The search results will be shown on a Google-hosted page.'
Examples	<code>q=Frodo&cx=00255077836266642015:u-scht7a-8i</code>

Gambar 4.14. Parameter `cx` [20]

4.2.4 Skema Database

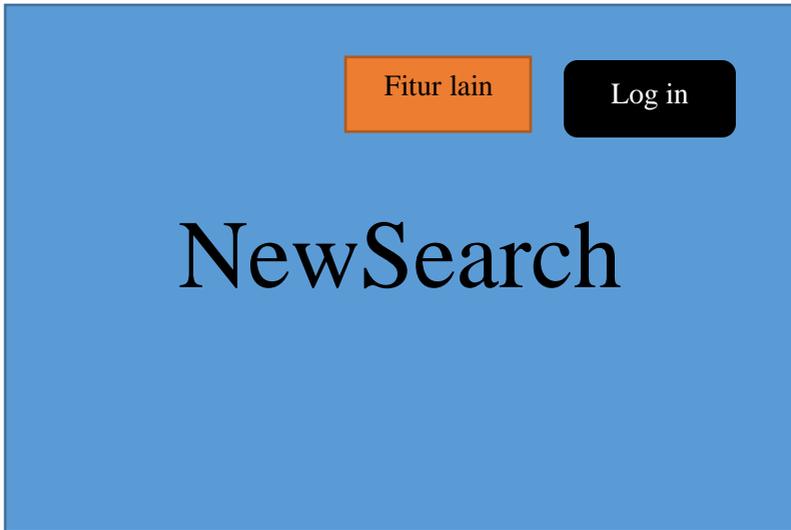
Berdasarkan kebutuhan sistem mesin pencari web yang akan dibangun, maka rancangan skema *Database* secara fisik dapat dilihat pada Gambar 4.15.



Gambar 4.15. Skema *Database*

4.2.5 Perancangan Halaman Antarmuka (*Interface*)

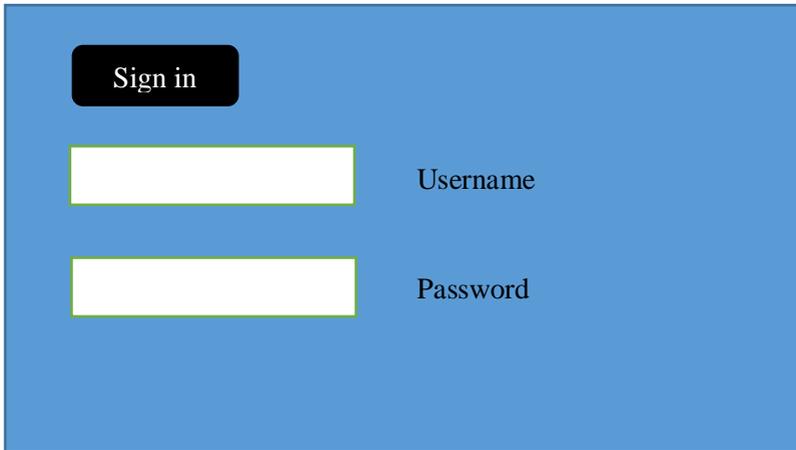
Halaman antarmuka aplikasi terbagi menjadi 4, yaitu halaman awal, halaman *Login/Registrasi*, halaman pencarian dan halaman hasil. Pada halaman awal yang merupakan tampilan pertama aplikasi berisi tombol *Login* dan fitur dimana tombol *Login* digunakan untuk masuk ke halaman *Login/Registrasi* sedangkan fitur berisi tombol-tombol tambahan yang membuat aplikasi lebih menarik. Desain halaman awal aplikasi terdapat pada Gambar 4.16. Pada halaman *Login/Registrasi* berisi *Form* data *Login* yang akan diisi oleh pengguna saat pengguna ingin menggunakan sistem. Desain halaman *Login/Registrasi* aplikasi terdapat pada Gambar 4.17 dan Gambar 4.18. Pada halaman pencarian akan berisi kolom pencarian dan tombol *Logout* dimana kolom pencarian berfungsi untuk menangkap data *Query/Keyword* yang dimasukkan oleh pengguna sedangkan tombol *Logout* digunakan jika pengguna ingin keluar dari sistem. Desain halaman pencarian aplikasi terdapat pada Gambar 4.19. Selanjutnya halaman hasil merupakan halaman antarmuka yang menampilkan hasil dari pencarian yang dilakukan berdasarkan data *Query/Keyword* yang dimasukkan oleh pengguna. Pada halaman hasil dibagi menjadi hasil pencarian *Website* dan hasil pencarian *Image*. Disini yang akan ditampilkan yaitu judul, *Uniform Resource Identifier (URI)*, deskripsi, *Link Image*, dan *Link Thumbnail* dari informasi yang dihasilkan. Lalu pada kiri atas tampilan terdapat kolom pencarian agar pengguna yang ingin mencari lagi hanya perlu memasukkan data tanpa perlu kembali ke halaman pencarian serta tombol *Logout*. Desain antarmuka halaman hasil dapat dilihat pada Gambar 4.20.



4.16. Halaman awal

The image shows a blue rectangular area representing a registration page. At the top left, there is a black button labeled "Sign up". Below this, there are three input fields, each with a green border, arranged vertically. To the right of each input field is a label: "Nama" for the first field, "Username" for the second, and "Password" for the third.

Gambar 4.17. Halaman Registrasi



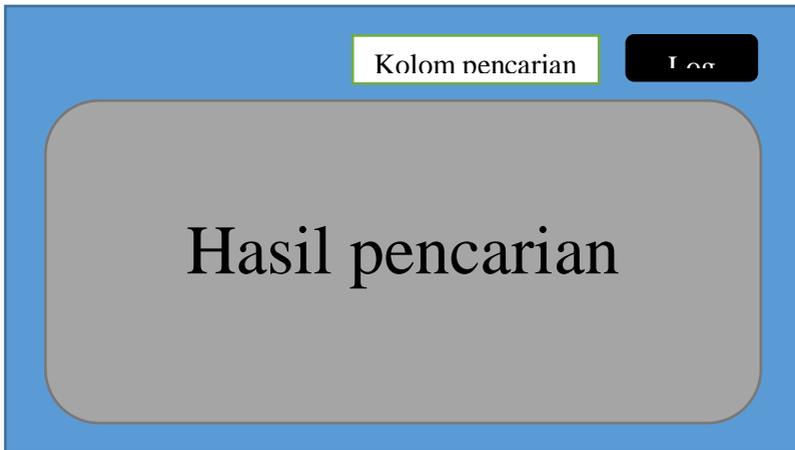
A login form on a blue background. At the top left is a black button with the text "Sign in". Below it are two white input fields with green borders. The first field is labeled "Username" and the second is labeled "Password".

Gambar 4.18. Halaman Login



A search page on a blue background. At the top right is a black button with the text "Log out". In the center is the text "NewSearch" in a large, bold, serif font. Below it is a white input field with a green border containing the text "Kolom pencarian".

Gambar 4.19. Halaman Pencarian



Gambar 4.20. Halaman Hasil

BAB V

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan mengenai tahapan implementasi dan pengujian sistem mesin pencari web (*Web Search Engine*). Untuk implementasi dilakukan berdasarkan analisis dan perancangan sistem yang telah dibuat pada bab sebelumnya. Pada tahap implementasi dan pengujian yang akan dilakukan meliputi tahapan implementasi *Framework Codeigniter* untuk aplikasi web, implementasi perlindungan privasi, implementasi *Database*, implementasi API, implementasi personalisasi, implementasi antarmuka (*Interface*) dan uji coba terhadap sistem mesin pencari web (*Web Search Engine*).

5.1 Implementasi Sistem

Implementasi dilakukan berdasarkan analisis dan perancangan aplikasi yang telah dibuat pada bab sebelumnya.

5.1.1 Implementasi CodeIgniter

Codeigniter adalah sebuah *Framework* aplikasi web yang bersifat *Open Source* yang berfungsi untuk membangun aplikasi dengan *Script PHP*. Sebelum menggunakan *Framework Codeigniter*, langkah awal yang dilakukan adalah melakukan konfigurasi dengan menggunakan fungsi *config*.

5.1.1.1 Konfigurasi Autoload

Autoload dalam adalah konfigurasi untuk mengatur pemanggilan *Class* apa saja yang akan panggil secara otomatis. Artinya dengan memasukkan nama *Class* yang akan dipanggil, *Class* tersebut tidak perlu dipanggil jika ingin digunakan dalam

Controller. Seperti Gambar 5.1. Secara otomatis akan memanggil *Class Database* dan *Session*.

```
<?php
defined('BASEPATH') OR exit('No direct
script access allowed');
$autoload['libraries'] = array('database',
'session');
}
```

Gambar 5.1 Kode pada *autoload.php*

5.1.1.2 Konfigurasi *Config*

Konfigurasi ini berguna untuk mengarahkan ke alamat yang diakses. Setiap menggunakan *base_url* maka akan selalu mengarah ke *http://localhost/SearchEngine*. Konfigurasi dapat dilihat pada Gambar 5.2.

```
<?php
defined('BASEPATH') OR exit('No direct script
access allowed');
$config['base_url'] =
'http://localhost/SearchEngine';
}
```

Gambar 5.2 Kode pada *config.php*

5.1.1.3 Konfigurasi *Routes*

Konfigurasi ini berisi *Controller* utama sebagai tampilan utama *Website*. Setiap mengakses aplikasi, sistem memanggil *Homecontroller.php* untuk menampilkan halaman utama.

```
<?php
defined('BASEPATH') OR exit('No direct script
access allowed');
$route['default_controller'] =
'Homecontroller';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
}
```

Gambar 5.3 Kode pada *routes.php*

5.1.1.4 Konfigurasi *Session*

Konfigurasi ini bertujuan untuk membatasi waktu *Update*. Pada aplikasi pencarian yang dibuat ini diatur untuk melakukan pencarian maksimal selama 60 detik atau 1 menit. Artinya, jika waktu pencarian melebihi 1 menit, maka pencarian akan dihentikan. Konfigurasi *Session* dapat dilihat pada Gambar 5.4.

```
<?php
defined('BASEPATH') OR exit('No direct script
access allowed');
$config['sess_time_to_update'] = 60;
}
```

Gambar 5.4 Kode pada *config.php*

5.1.1.5 Konfigurasi *Database*

Konfigurasi ini digunakan untuk membuat koneksi ke *Database*. Database yang digunakan adalah “tugasakhir”, dapat dilihat pada Gambar 5.5.

```

<?php
defined('BASEPATH') OR exit('No direct
script access allowed');
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'tugasakhir',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => TRUE,
);

```

Gambar 5.5 Kode pada *database.php*

5.1.1.6 Konfigurasi *Error Reporting*

Konfigurasi ini berguna untuk mengatasi *Error Report* yang muncul setelah melakukan pencarian. Jika data yang dicari kosong maka akan terjadi error sebanyak perulangan dari data yang kosong dan akan merusak tampilan halaman antarmuka. Konfigurasi dapat dilihat pada Gambar 5.6.

5.1.2 Implementasi Login Pengguna

Login merupakan langkah pertama yang harus dilakukan pengguna jika ingin menggunakan sistem ini dalam mencari informasi yang dibutuhkan. Pada bab sebelumnya sudah dibahas mengenai *Login* pengguna. Berikut adalah kode untuk membangun halaman *Login* sistem dapat lihat pada Gambar 5.7, 5.8

```

switch (ENVIRONMENT) {
    case 'development':
        error_reporting(-1);
        ini_set('display_errors', 0);
        break;
    case 'testing':
    case 'production':
        ini_set('display_errors', 0);
        if (version_compare(PHP_VERSION,
'5.3', '>=')) {
            error_reporting(E_ALL &
~E_NOTICE & ~E_DEPRECATED & ~E_STRICT &
~E_USER_NOTICE & ~E_USER_DEPRECATED);
        }
        else {
            error_reporting(E_ALL &
~E_NOTICE & ~E_STRICT & ~E_USER_NOTICE);
        }
        break;
    default:
        header('HTTP/1.1 503 Service
Unavailable.', TRUE, 503);
        echo 'The application environment
is not set correctly.';
        exit(1); // EXIT_ERROR }}

```

Gambar 5.6 Kode pada *index.php*

5.1.3 Implementasi *Perlindungan Privasi*

Implementasi perlindungan privasi adalah proses dimana sistem yang dibuat dapat mengelola data *Login/data Query/Keyword* yang merupakan profil pengguna. Proses ini bertujuan untuk memisahkan bagian sensitif (privasi khusus) dan bagian tak sensitif (privasi umum) pengguna. Privasi umum akan dijadikan parameter pengambilan data ke *Server* sedangkan

privasi khusus akan di simpan sementara oleh sistem. Impelementasi dapat dilihat pada Gambar 5.9.

```

// Check for user login process
public function user_login_process() {
// new user registrtration (sign up)
    $this->form_validation->set_rules('username',
'username', 'required');
    $this->form_validation-
>set_rules('email', 'email', 'required|valid_email');
    $this->form_validation-
>set_rules('password', 'password', 'required');
    $this->form_validation-
>set_rules('password_conf', 'password', 'required|matches[pa
ssword]');
    if($this->form_validation->run() == FALSE) {
        $this->load->view('login_view');
    }else{
        $data['username'] = $this->input-
>post('username');
        $data['email'] = $this->input-
>post('email');
        $data['password'] = md5($this->input-
>post('password'));

        $this->Loginmodel-
>registration_insert($data);
        $this->session-
>set_userdata('username', $data['username']);
        // $this->load->view('home', $data);
        redirect('homecontroller2', 'refresh');
    }
}

```

Gambar 5.7. Kode Implementasi Registrasi Pengguna Pada Controller

5.1.4 Implementasi Algoritma *Greedy*

Berdasarkan Pseudocode Algoritma *Greedy* yang sudah dibahas pada bab sebelumnya, berikut kodenya dapat dilihat pada Gambar 5.10

```
public function user_login(){
// user already exist or not (sigh in)
$valid = $this->form_validation;
    $valid-
>set_rules('username','username','required');
    $valid-
>set_rules('password','password','required');
    $data['username'] = $this->input-
>post('username');
    $data['password'] = md5($this->input->
post('password'));

    if($valid->run()) {
        $this->Loginmodel->login($data);
        $this->session->
set_userdata('username',$data['username']);

        $this->load->view('home',$data);
    }
}
public function logout(){
    $this->simple_login->logout();

    redirect('Homecontroller');
}
}
```

Gambar 5.8. Kode Implementasi Login Pengguna Pada Controller

```

public function registration_insert($data) {

// Query to check whether username already exist or not
$condition = "username =" . "'" . $data['username'] .
"";
$this->db->select('*');
$this->db->from('login');
$this->db->where($condition);
$this->db->limit(1);
$query = $this->db->get();
if ($query->num_rows() == 0) {

// Query to insert data in database
    $this->db->insert('login', $data);
    //$this->session-
>set_userdata('username', $data['username']);
    $idBaru = $this->db->insert_id();
    return $idBaru;

}

public function login($data) {

$condition = "username =" . "'" . $data['username'] . "'
AND " . "password =" . "'" . $data['password'] . "'";
$this->db->select('*');
$this->db->from('login');
$this->db->where($condition);
$this->db->limit(1);
$query = $this->db->get();
if ($query->num_rows() == 1) {
return true;
} else {
return false;
}

public function getHistory($username){
    $query= $this->db-
>get_where('login', array('username'=>$username));
    return $query->result_array();
}

```

Gambar 5.9. Kode Implementasi Login Pengguna Pada Model

5.1.5 Implementasi Akuisisi Data

Berdasarkan Algoritma akuisisi data yang sudah dibahas pada bab sebelumnya, berikut kodenya dapat dilihat pada Gambar 5.11

5.1.6 Implementasi *Filter* Data

Berdasarkan Algoritma *Filter* data yang sudah dibahas pada bab sebelumnya, berikut kodenya dapat dilihat pada Gambar 5.12

5.1.7 Implementasi Proses Personalisasi Data Pengguna

Berdasarkan Algoritma Personalisasi data yang sudah dibahas pada bab sebelumnya, kode pada Gambar 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.22, 5.23.

5.1.8 Implementasi *Database*

Implementasi *Database* adalah proses pembuatan tabel-tabel yang diperlukan untuk menyimpan data yang akan diambil menggunakan API. Pembuatan tabel ini sesuai dengan perancangan *Database* sebelumnya. Berikut ini *Strukture Table* dari datasite (*Website*), datalain (*Image*), data (hasil filtering), privasi (privasi khusus), keyword (privasi umum) dan login (data *Login*).

1. *Strukture Table* datasite dapat dilihat pada Gambar 5.24. Tabel ini berisi id_site sebagai *Primary Key*, id_jenisite, id_keywordsite, title, linksite, description, id_logins dan keyword
2. *Strukture Table* datalain dapat dilihat pada Gambar 5.25. Tabel ini berisi id_lain sebagai *Primary Key*, id_keywordlain, link, linklain, id_loginuser dan keyword

```

//PENGGUNAAN ALGORITMA GREEDY UNTUK
GENERALISASI PROFIL PENGGUNA BERDASARKAN
KEYWORD

        $parameter =
$_POST["keyword"];
        $datakeyword = array();
        $privasi_umum = array();
        $privasi_khusus = array();
        $dataHistory=array();
        $datakeyword = explode("
", $parameter);
        $i = count($datakeyword);
        if ($i > 0){
            if ($i==1){

                $privasi_umum['keyword']
=$datakeyword[0];
                }
                elseif ($i==2){
                    $privasi_umum
['keyword']= $datakeyword[0];
                    $privasi_khusus
['privasi']=$datakeyword[1];
                }
                else{
                    $privasi_umum
['keyword']=$datakeyword[0]."
".$datakeyword[1];

                    for ($x=2;$x<=count($datakeyword);$x+
+){

                        $privasi_khusus['privasi'] .=
$datakeyword[$x]." ";
                    }}}

```

Gambar 5.10. Kode Implementasi Greedy untuk Privasi

```

$data =
json_decode(file_get_contents("https://www.googleapis.com
/customsearch/v1?key=AIzaSyCEwBSirwyOUVLQAYXtIT-
szcfirueP2_I&cx=005051992855557048902:lc6yry_og-
q&q=".rawurlencode($datakeyword)));
//MEMASUKKAN MASING_MASING DATA KE TABEL DALAM DATABASE
foreach ($data->items as $now) {
$datalain=array();
$data = array();
if($now->title){
    $data['title'] = $now->title;
    $datalain['title']=$now->title;}
else{
    $data['title'] = "";
    $datalain['title'] = "";}
if($now->link){
    $data['linksite'] = $now->link;}
else{
    $data['linksite'] = "";}
if($now->snippet){
    $data['description'] = $now->snippet;}
else{
    $data['description'] = "";}
if($now->pagemap->cse_thumbnail[0]->src){
    $datalain['linklain'] = $now->pagemap-
>cse_thumbnail[0]->src;}
else{
    $datalain['linklain'] = "";}
if($now->pagemap->cse_image[0]->src){
    $datalain['link'] = $now->pagemap->cse_image[0]-
>src;}
else{
    $datalain['link'] = "";}
$data['id_jenissite'] = 1;
$data['id_keywordsite'] = $idBaru;
$data['keyword']=$privasi_umum['keyword'];
$datalain['keyword']=$privasi_umum['keyword'];
$data['id_logins']=$idUser;
$datalain['id_keywordlain']=$idBaru;
$datalain['id_loginuser']=$idUser;
$this->Soemodel->insert_data_api($data);
$this->Imagemodel->insertData($datalain);
}

```

Gambar 5.11. Kode Implementasi Akuisisi Data

```

$upper = strtoupper($privasi_khusus
['privasi']);
$upper1 = strtoupper($data['title']);
if(similar_text($upper,$upper1,$percent)){
    if($percent>10){
        $data1=$this->Soemode-
>getalldata($data['title']);
        $data2=$this->Soemod-
>getalldata1($data['title']);
//MENDUPLIKAT DATA
$datafilter['id_jenis']=1;
$datafilter['id_filterkeyword']=$privasi_khus
us['privasi'];
$datafilter['id_filterkeyworddumum']=$privasi_
umum['keyword'];
$datafilter['titlefilter']=$data['title'];
$datafilter['linkfilter']=$data2['link'];
$datafilter['linklainfilter']=$data2['linklai
n'];
$datafilter['linksitefilter']=$data1['linksit
e'];
$datafilter['descriptionfilter']=$data1['desc
ription'];
$datafilter['id_loginfilter']=$data1['id_logi
ns'];
$this->Soemodel-
>insert_data_filter($datafilter);
    }
}
}

```

Gambar 5.12. Kode Implementasi Filtering Data

```

public function
gethistory($umum,$id_umum) {
    $query = $this->db-
>get_where('keyword', array('keyword' =>
$umum,'id_loginkuser' => $id_umum));
    return $query->result_array();
}

```

Gambar 5.13. Potongan Kode *gethistory Keyword* (Privasi Umum)

```

public function
gethistory1($khusus,$id_khusus) {
    $query = $this->db-
>get_where('privasi', array('privasi' =>
$khusus,'id_loginpuser' => $id_khusus));
    return $query->result_array();
}

```

Gambar 5.14. Potongan Kode *gethistory Privasi* (Privasi Khusus)

3. *Strukture Table* data dapat dilihat pada Gambar 5.26. Tabel ini berisi *id_data* sebagai *Primary Key*, *id_jenis*, *id_filterkeyword*, *linkfilter*, *linklainfilter*, *titlefilter*, *linksitefilter*, *descriptionfilter*, *id_loginfilter* dan *id_filterkeywordumum*

```
public function insertKeyword($data) {
    if($this->db-
>insert("keyword", $data)) {
        $idBaru = $this->db-
>insert_id();
        return $idBaru;
    }
    return 0;
}
```

Gambar 5.15. Potongan Kode *Insert Keyword* (Privasi Umum)

```
public function insertPrivasi($data) {
    if($this->db-
>insert("privasi", $data)) {
        $idprivasi= $this->db-
>insert_id();
        return $idprivasi;
    }
    return 0;
}
```

Gambar 5.16. Potongan Kode *Insert Privasi* (Privasi Khusus)

```
public function insert_data_api($data) {  
    if ($this->db->insert("datasite", $data)) {  
        return true;  
    }  
}
```

Gambar 5.17. Potongan Kode *Insert Data* dari *Server* ke *Database*

```
public function insert_data_filter($data)  
{  
    if($this->db->insert("data", $data)) {  
        $idprivasi= $this->db->insert_id();  
        return $idprivasi;  
    }  
    return 0;  
}
```

Gambar 5.18. Potongan Kode *Insert Data* yang Sudah di *Filter* ke *Database*

```

public function getalldata($title){
    $query=$this->db-
>get_where('datasite',array('title'=>$tit
le));
    return $query->row_array();
}

```

Gambar 5.19. Potongan Kode Mendapatkan Semua Data pada Tabel di *Database*

```

public function getdata($data){
    $this->db->select('title');
    $this->db-
>from('datasite');
    $this->db->where('keyword',
$data);
    $query = $this->db->get();
    return $query-
>result_array();
}

```

Gambar 5.20. Potongan Kode Mendapatkan Judul *Website* untuk Proses *Filtering* Data

```

public function getDataUserGrup($string){
    $this->db->select('*');
    $this->db->from('data');
    $this->db->
>where('id_filterkeyword', $string);
    $query = $this->db->get();
    return $query->result_array();
}

```

Gambar 5.21. Potongan Kode Mendapatkan Semua Data Riwayat Pencarian Pengguna Berdasarkan Privasi

```

public function
getDataApi($idk, $kkhusus){
    $query = $this->db->
>get_where('data',
array('id_loginfilter'=>$idk, 'id_filterke
yword' => $kkhusus));
    return $query->result_array();
}

```

Gambar 5.22. Potongan Kode Mendapatkan Semua Data Berdasarkan Id Pengguna dan Privasi Khusus

```

public function CheckUsername(){
    $user=$this->session-
>userdata('username');
    $cek=$this->Loginmodel-
>gethistory($user);
    $idUser= $cek[0]['id_login'];
    //$idUser=$this->CI->session-
>userdata('id_login', uniqid(rand()));
    return $idUser;
}

```

Gambar 5.23. Potongan Kode Pengecekan Username

4. *Strukture Table* privasi dapat dilihat pada Gambar 5.27. Tabel ini berisi id_privasi sebagai *Primary Key*, privasi dan id_loginpuser
5. *Strukture Table* keyword dapat dilihat pada Gambar 5.28. Tabel ini berisi id_keyword sebagai *Primary Key*, keyword dan id_loginkuser
6. *Strukture Table* login dapat dilihat pada Gambar 5.29. Tabel ini berisi id_login sebagai *Primary Key*, username, password dan email

Struktur database yang terdiri dari table-table diatas ditunjukkan pada Gambar 5.30.

5.1.9 Implementasi Halaman antarmuka (*Interface*)

Implementasi halaman antarmuka pengguna yang terdiri dari halaman awal, halaman *Login*, halaman pencarian dan

#	Nama	Tipe	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id_data	int(50)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	2 id_jenis	int(50)			Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	3 id_filterkeyword	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	4 linkfilter	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	5 linklainfilter	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	6 titlefilter	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	7 linksitefilter	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	8 descriptionfilter	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	9 id_loginfilter	int(50)			Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya
<input type="checkbox"/>	10 id_filterkeyworddum	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Lainnya

Gambar 5.24. Tabel Data untuk Menyimpan Data Hasil *Filtering*

#	Nama	Tipe	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id_lain	int(50)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 id_keywordlain	int(11)			Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 title	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	4 link	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	5 linklain	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	6 id_loginuser	int(50)			Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	7 keyword	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya

Gambar 5.25. Tabel Datalain untuk Menyimpan Data *Image*

#	Nama	Tipe	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id_site	int(50)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 id_jenissite	int(50)			Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 id_keywordsite	int(50)			Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	4 title	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	5 linksite	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	6 description	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	7 id_logins	int(50)			Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	8 keyword	text	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya

Gambar 5.26. Tabel Datasite untuk Menyimpan Data *Website*

#	Nama	Tipe	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id_keyword	int(50)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	2	keyword	text	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	3	id_loginuser	int(50)		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya

Gambar 5.27. Tabel Keyword untuk Menyimpan Data Privasi Umum

#	Nama	Tipe	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id_privasi	int(50)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	2	privasi	text	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	3	id_loginuser	int(50)		Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya

Gambar 5.28. Tabel Privasi untuk Menyimpan Data Privasi Khusus

#	Nama	Tipe	Penyortiran	Atribut	Tak Terilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id_login	int(50)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	2	username	varchar(30)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	3	password	varchar(255)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya
<input type="checkbox"/>	4	email	varchar(30)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Utama Unik Indeks Lainnya

Gambar 5.29. Tabel Login untuk Menyimpan Data Login Pengguna

Tabel	Tindakan	Baris	Tipe	Penyortiran	Ukuran	Beban
<input type="checkbox"/>	data	23	InnoDB	latin1_swedish_ci	16 KB	-
<input type="checkbox"/>	datalain	190	InnoDB	latin1_swedish_ci	80 KB	-
<input type="checkbox"/>	datasite	190	InnoDB	latin1_swedish_ci	96 KB	-
<input type="checkbox"/>	keyword	76	InnoDB	latin1_swedish_ci	16 KB	-
<input type="checkbox"/>	login	13	InnoDB	latin1_swedish_ci	16 KB	-
<input type="checkbox"/>	privasi	3	InnoDB	latin1_swedish_ci	16 KB	-
6 tabel	Jumlah	445	InnoDB	latin1_swedish_ci	240 KB	0 B

Gambar 5.30. Struktur Database tugasakhir

halaman hasil yang sudah dibahas pada halaman sebelumnya dapat dilihat pada Gambar 5.34, 5.35, 5.36, 5.37, 5.38 dan 5.39 . Ketika pengguna melakukan akses pada halaman awal, pengguna akan langsung terhubung dengan *Homecontroller* yang selanjutnya diteruskan ke fungsi *view* untuk *view*. Kode untuk *Homecontroller* halaman awal dapat dilihat pada Gambar 5.31. Terlihat pada gambar tersebut, *Homecontroller* langsung terhubung ke *display.php*. Selanjutnya untuk *Login*, halaman akan terhubung dengan *Usercontroller* untuk menampilkan *view Login/Register*. Kode untuk *Usercontroller* halaman awal dapat dilihat pada Gambar 5.32. Kemudian akan diteruskan ke *Homecontroller2* untuk *view* ke halaman pencarian dan *Testcontroller* untuk *view* ke halaman hasil. Kode untuk *Testcontroller* halaman awal dapat dilihat pada Gambar 5.33.

```
<?php
class HomeController extends CI_Controller{
    function __construct() {
        parent::__construct();
        $this->load->helper('url');
    }
    public function index()
    {
        $data['title'] = 'Home Search';
        $this->load->view(
'display',$data );
    }
}
```

Gambar 5.31. Kode *HomeController.php*

```

<?php
class Usercontroller extends CI_Controller {
public function __construct() {
parent::__construct();
public function index() {
    $data['title'] = 'Login or Register';
    // $data['page'] = 'login_view';
    $this->load->view('login view', $data);}}}?>

```

Gambar 5.32. Kode *Usercontroller.php*

```

<?php
defined('BASEPATH') OR exit('No direct script
access allowed');

class Testcontroller extends CI_Controller {
    function __construct() {
        parent::__construct();

        $this->load-
>library('session','form_validation');

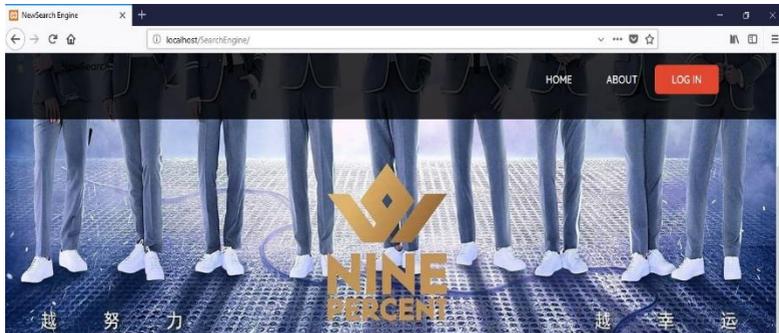
        $this->load->helper('url','form');
        $this->load->model('Soemodel');

        $this->load->model('Imagemodel');
        $this->load->model('Loginmodel');

        $this->load->view('Ajax',$array);}

```

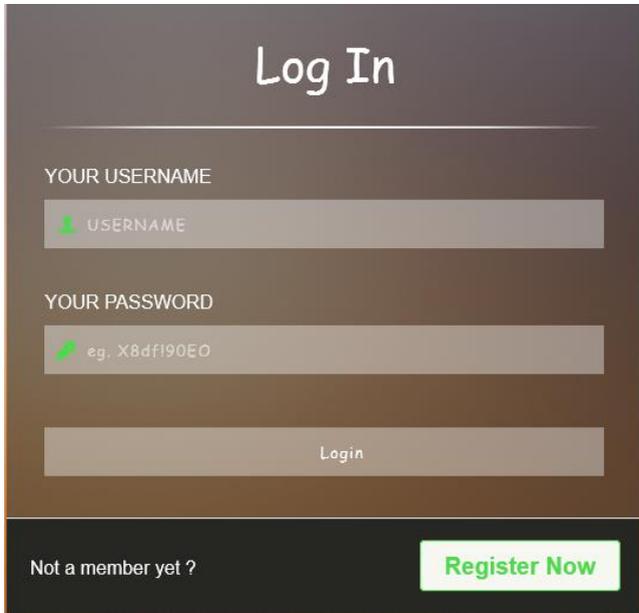
Gambar 5.33. Kode *Testcontroller.php*



NewSearch adalah sebuah mesin pencari yang memanfaatkan Application Programming Interface (API) berupa Google Custom Search API yang digunakan untuk melakukan pencarian dengan meningkatkan kualitas pencarian tanpa mengganggu privasi pengguna.

Gambar 5.34. Implementasi Halaman Awal

Gambar 5.35. Implementasi Halaman Registrasi



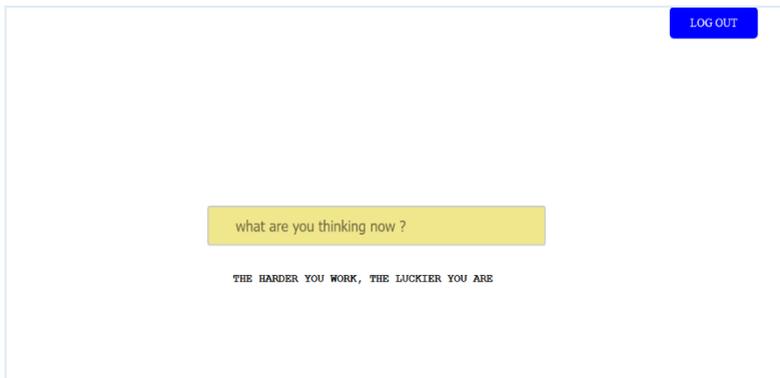
Log In

YOUR USERNAME

YOUR PASSWORD

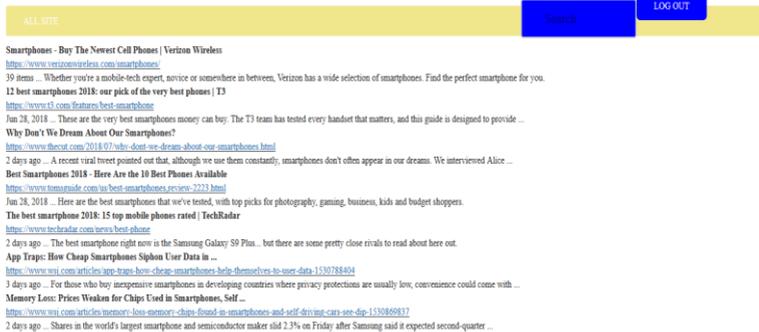
Not a member yet ?

Gambar 5.36. Implementasi Halaman Login

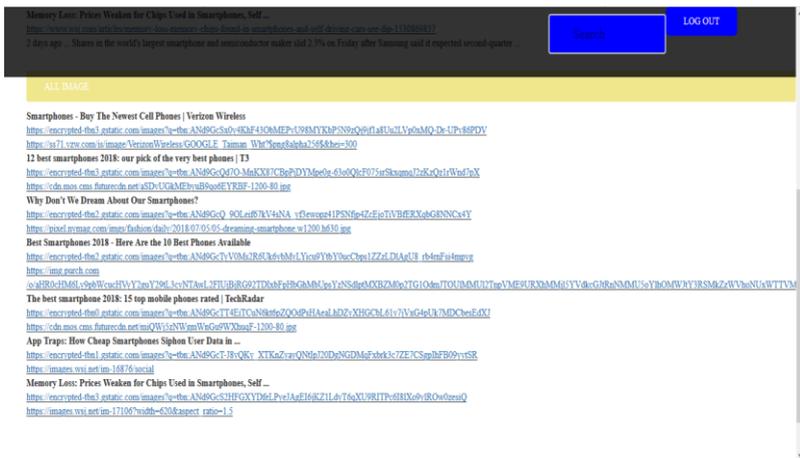


THE HARDER YOU WORK, THE LUCKIER YOU ARE

Gambar 5.37. Implementasi Halaman Pencarian



Gambar 5.38. Implementasi Halaman Hasil Website



Gambar 5.39. Implementasi Halaman Hasil Image

5.2 Pengujian Aplikasi

Pada sub bab ini akan dibahas mengenai uji coba aplikasi pencarian yang telah diimplementasikan sebelumnya. Uji coba terhadap aplikasi pencarian ini dilakukan dengan menganalisis tingkat kecocokan pencarian yang berdasarkan kesesuaian keyword,

keakuratan dari hasil pencarian dan membandingkan hasil pencarian pada aplikasi dalam penelitian dengan mesin pencari *Google*. Perangkat yang digunakan adalah Adobe Dreamweaver dengan bahasa pemrograman PHP, MySQL sebagai DBMS, dan *Google API Custome Search* untuk mendapatkan data mengenai *Site*. Pada uji coba ini, terdapat batasan yaitu *Keyword* terdiri dari satu kata, dua kata dan tiga kata.

5.2.1 Skenario Uji Coba

Query/Keyword yang dimasukkan terdiri dari satu kata, dua kata, tiga kata atau lebih. *Query/Keyword* yang dimasukkan dapat dituliskan dalam berbagai Bahasa tetapi hasil pencarian akan mengembalikan deskripsi dalam Bahasa Inggris, kecuali pencarian menggunakan Bahasa Indonesia.

Pada uji coba ini, pengguna melakukan pencarian dengan *Query/Keyword* yang sesuai dengan preferensi/keinginan dan *Query/Keyword* terdiri dari satu kata, dua kata dan 3 kata. Hasil pencarian yang ditampilkan ketika memasukkan *Query/Keyword* adalah hasil pencarian yang belum terpersonalisasi tetapi sudah dilakukan proses peningkatan kualitas, kemudian pada pencarian berikutnya *Query/Keyword* akan dipersonalisasi sehingga kualitas pencarian mengalami peningkatan karena hasil pencarian sesuai atau relevan dengan keinginan pengguna. Kenaikan Kualitas yang dimaksud didalam Tugas Akhir ini adalah kesesuaian hasil pencarian yang ditampilkan ke pengguna dengan preferensi/keinginan pengguna.

Skenario *Query/Keyword* yang dicari oleh pengguna 1 beserta preferensi/keinginannya yang selanjutnya digunakan untuk relevansi data dijelaskan pada Tabel 5.1

Tabel 5.1 Skenario *Keyword* untuk pengguna 1

Pengguna	<i>Keyword</i>	Preferensi
1	Laptop	Laptop
	Apple Smartphones	Brand Apple dengan tipe Smartphone
	Laptop Asus 2018	Laptop Asus 2018
	Brand Laptop Popular	Brand Laptop yang Sedang Popular
	Samsung Smartphones	Brand Samsung dengan tipe Smartphone
	Clothes	Baju
	Sepatu	Film, dan lain-lain
	Indomie Goreng	Indomie Goreng
	Buku Cerita Anak	Buku cerita untuk anak-anak
	Pensil	Pensil Alat Tulis
	Piala Dunia	Piala Dunia Sepakbola
	Film Perahu Kertas	Film Perahu Kertas

Skenario *Query/Keyword* yang dicari oleh pengguna 2 beserta preferensi/keinginannya yang selanjutnya digunakan untuk relevansi data dijelaskan pada Tabel 5.2

Skenario *Query/Keyword* yang saling berkaitan digunakan untuk uji kualitas hasil pencarian pada proses pencarian oleh pengguna dijelaskan pada Tabel 5.3 .

Preferensi/keinginan pengguna yang digunakan pada uji coba ini yaitu berkenaan dengan alat elektronik dan *Keyword* yang dilihat kualitas nya adalah “*New Iphone Apple*”.

Tabel 5.2 Skenario *Keyword* untuk pengguna 2

Pengguna	<i>Keyword</i>	Preferensi
1	Windows	Sistem Operasi
	Grammar English	Aturan bahasa inggris
	Asian Games 2018	Asian Games 2018
	Pisang	Makanan atau buah
	Berita terbaru	Berita terbaru
	Sekolah terbaik ASEAN	Sekolah terbaik di ASEAN

Tabel 5.3 Skenario *Keyword* yang saling berkaitan untuk pengguna

Percobaan	<i>Keyword</i>
1	New Iphone Apple
2	Apple Smartphones
3	New Iphone Apple
4	Smartphones Apple
5	Apple
6	New Iphone Apple
7	Samsung Smartphones
8	Macbook Terbaru Apple
9	Android Versus Apple
10	New Iphone Apple

5.2.2 Pengujian Kesesuaian *Keyword* dengan Hasil yang Ditampilkan

Skenario yang dilakukan adalah dengan melakukan percobaan memasukkan *Query/Keyword* pada halaman antarmuka pencarian yang terdiri dari satu kata, dua kata dan tiga kata dengan menggunakan 2 jenis *Keyword* yaitu Bahasa Indonesia dan Bahasa Inggris sesuai dengan *Tabel 5.1* dan *Tabel 5.2*. Kemudian akan dilihat kesesuaian antara informasi yang diterima atau ditampilkan pada halaman hasil dengan *Keyword* yang digunakan. Pada setiap kategori dan semua data yang telah diterima akan dilihat isi dari data tersebut sehingga bisa diketahui apakah data yang diterima sesuai dengan *Keyword*. Selanjutnya dibuat persentase untuk hasil yang sesuai dengan menggunakan persamaan berikut:

$$\text{Persentase Kesesuaian} = \left(\frac{x}{n}\right)100\% \dots\dots\dots(1)$$

$$\text{Rata-rata Persentase } \textit{Keyword} = \left(\frac{\sum_{i=1}^j p_i}{j}\right)\dots\dots\dots(2)$$

$$\text{Rata-rata Persentase Keseluruhan} = \left(\frac{\sum_{i=1}^k q_i}{k}\right)\dots\dots\dots(3)$$

Dengan penjelasan,

x = Data yang sesuai perkategori

n = Jumlah data yang diterima perkategori

p_i = Persentase kategori ke- i

j = Jumlah kategori

q_i = Persentase keyword ke- i

k = Jumlah keyword

Berikut hasil uji coba dengan menggunakan *Keyword* sesuai Tabel 5.1 dapat dilihat pada Tabel 5.4.

Tabel 5.4 Kesesuaian Informasi yang Ditampilkan dengan Jenis *Keyword* pada Pengguna 1

No	Jenis Keyword	Keyword	Presentase Kesesuaian		
			Website	Image	Rata-rata
1	Bahasa Inggris	Laptop	100%	94,12 %	97,06 %
		Apple Smartphones	100%	25%	62,5%
		Laptop Asus 2018	100%	100%	100%
		Brand Laptop Popular	100%	100%	100%
		Samsung Smartphones	100%	70%	85%
		Clothes	90%	85,71 %	87,86 %
Total Keseluruhan					88,74 %
2	Bahasa Indonesia	Sepatu	100%	100%	100%
		Indomie Goreng	100%	87,5%	93,75 %
		Buku Cerita Anak	100%	88,89 %	94,45 %
		Pensil	100%	60%	80%
		Piala Dunia	100%	100%	100%
		Film Perahu Kertas	80%	77,78 %	78,89 %
Total Keseluruhan					91,18 %

Dari hasil Tabel 5.4 dapat diberikan analisis bahwa penggunaan *Keyword* Berbahasa Inggris dan Bahasa Indonesia menghasilkan pencarian yang sesuai. Pencarian menggunakan *Keyword* Bahasa Inggris memiliki prosentase rata-rata kesesuaian dengan hasil yang ditampilkan yaitu 88,74% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Sedangkan pencarian menggunakan *Keyword* Bahasa Indonesia memiliki prosentase rata-rata kesesuaian dengan hasil yang ditampilkan yaitu 91,18% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Jumlah data yang diterima dapat dilihat pada Tabel 5.5 dan jumlah data yang sesuai dengan keinginan pengguna dapat dilihat pada Tabel 5.6

Tabel 5.5 Jumlah Data yang Diterima pada Pengguna 1

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa Inggris	Laptop	10	17	27
		Apple Smarphones	10	12	22
		Laptop Asus 2018	5	10	15
		Brand Laptop Popular	10	20	30
		Samsung Smartphones	20	30	50
		Clothes	10	14	24
Total Keseluruhan					168

2	Bahasa Indonesia	Sepatu	10	20	30
		Indomie Goreng	8	16	24
		Buku Cerita Anak	10	18	28
		Pensil	10	20	30
		Piala Dunia	7	14	21
		Film Perahu Kertas	10	18	28
Total Keseluruhan					161

Tabel 5.6 Jumlah Data Sesuai Keinginan Pengguna yang Diterima pada Pengguna 1

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa Inggris	Laptop	10	16	26
		Apple Smartphones	10	3	13
		Laptop Asus 2018	5	10	15
		Brand Laptop Popular	10	20	30
		Samsung Smartphones	20	21	41
		Clothes	9	12	21
Total Keseluruhan					146

2	Bahasa Indonesia	Sepatu	10	20	30
		Indomie Goreng	8	14	24
		Buku Cerita Anak	10	16	26
		Pensil	10	12	22
		Piala Dunia	7	14	21
		Film Perahu Kertas	8	14	22
Total Keseluruhan					145

Berikut hasil uji coba dengan menggunakan *Keyword* sesuai Tabel 5.2 dapat dilihat pada Tabel 5.7.

Tabel 5.7 Kesesuaian Informasi yang Ditampilkan dengan Jenis *Keyword* pada Pengguna 2

No	Jenis Keyword	Keyword	Presentase Kesesuaian		
			Website	Image	Rata-rata
1	Bahasa Inggris	Windows	100%	100%	100%
		Grammar English	87,5%	81,81%	84,66%
		Asian Games 2018	100%	100%	100%
Total Keseluruhan					94,89%
2	Bahasa Indonesia	Pisang	100%	100%	100%
		Berita Terbaru	80%	78,95%	79,48%
		Sekolah Terbaik Asean	88,89%	88,24%	88,57%
Total Keseluruhan					89,35%

Dari hasil Tabel 5.7 dapat diberikan analisis bahwa penggunaan *Keyword* Berbahasa Inggris dan Bahasa Indonesia menghasilkan pencarian yang sesuai. Pencarian menggunakan *Keyword* Bahasa Inggris memiliki prosentase rata-rata kesesuaian dengan hasil yang ditampilkan yaitu 94,89% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Sedangkan pencarian menggunakan *Keyword* Bahasa Indonesia memiliki prosentase rata-rata kesesuaian dengan hasil yang ditampilkan yaitu 89,35% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Jumlah data yang diterima dapat dilihat pada Tabel 5.8 dan jumlah data yang sesuai dengan keinginan pengguna dapat dilihat pada Tabel 5.9.

Tabel 5.8 Jumlah Data yang Diterima pada Pengguna 2

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa Inggris	Windows	10	20	30
		Grammar English	8	11	19
		Asian Games 2018	4	7	11
Total Keseluruhan					60
2	Bahasa Indonesia	Pisang	10	20	30
		Berita Terbaru	10	19	29
		Sekolah Terbaik Asean	9	17	26
Total Keseluruhan					85

Tabel 5.9 Jumlah Data sesuai keinginan pengguna yang Diterima pada Pengguna 2

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa	Windows	10	20	30
		Grammar English	7	9	16
	Inggris	Asian Games 2018	4	7	11
Total Keseluruhan					57
2	Bahasa	Pisang	10	20	30
		Berita Terbaru	8	15	23
	Indonesia	Sekolah Terbaik Asean	8	15	23
Total Keseluruhan					76

5.2.3 Pengujian Keakuratan Hasil dengan Relevansi Keinginan Pengguna

Skenario yang dilakukan adalah dengan melakukan percobaan memasukkan *Query/Keyword* pada halaman antarmuka pencarian yang terdiri dari satu kata, dua kata dan tiga kata dengan menggunakan 2 jenis *Keyword* yaitu Bahasa Indonesia dan Bahasa Inggris sesuai dengan *Tabel 5.1* dan *Tabel 5.2*. Kemudian akan dilihat keakuratan antara informasi yang diterima atau ditampilkan pada halaman hasil dengan relevansi keinginan pengguna. Pada setiap kategori dan semua data yang telah diterima akan dilihat isi dari data tersebut sehingga bisa diketahui

apakah data yang diterima sesuai dengan preferensi atau keinginan pengguna. Selanjutnya dibuat persentase untuk hasil yang sesuai dengan menggunakan persamaan berikut:

$$\text{Persentase Kesesuaian} = \left(\frac{x}{n}\right)100\% \dots\dots\dots(1)$$

$$\text{Rata-rata Persentase Keyword} = \left(\frac{\sum_{i=1}^j p_i}{j}\right)\dots\dots\dots(2)$$

$$\text{Rata-rata Persentase Keseluruhan} = \left(\frac{\sum_{i=1}^k q_i}{k}\right)\dots\dots\dots(3)$$

Dengan penjelasan,

x = Data yang sesuai perkategori

n = Jumlah data yang diterima perkategori

p_i = Persentase kategori ke- i

j = Jumlah kategori

q_i = Persentase keyword ke- i

k = Jumlah keyword

Berikut hasil uji coba dengan menggunakan *Keyword* sesuai Tabel 5.1 dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kesesuaian informasi yang Ditampilkan dengan Relevansi Keinginan Pengguna 1

No	Jenis Keyword	Keyword	Presentase Kesesuaian		
			Website	Image	Rata-rata
1	Bahasa Inggris	Laptop	100%	90%	95%
		Apple Smartphones	60%	25%	42,5%
		Laptop Asus 2018	100%	66,67%	83,34%
		Brand Laptop Popular	80%	80%	80%
		Samsung Smartphones	70%	63,33%	66,67%
		Clothes	90%	85,71%	87,86%
Total Keseluruhan					75,89%
2	Bahasa Indonesia	Sepatu	100%	100%	100%
		Indomie Goreng	50%	50%	50%
		Buku Cerita Anak	90%	88,89%	89,45%
		Pensil	70%	60%	65%
		Piala Dunia	83,33%	83,33%	83,33%
		Film Perahu Kertas	80%	77,78%	78,89%
Total Keseluruhan					77,78%

Dari hasil Tabel 5.10 dapat diberikan analisis bahwa penggunaan *Keyword* Berbahasa Inggris dan Bahasa Indonesia menghasilkan pencarian yang akurat. Pencarian menggunakan *Keyword* Bahasa Inggris memiliki prosentase rata-rata keakuratan hasil yang ditampilkan dengan relevansi keinginan pengguna yaitu 75,89% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Sedangkan pencarian menggunakan *Keyword* Bahasa Indonesia memiliki prosentase rata-rata keakuratan hasil yang ditampilkan dengan relevansi keinginan pengguna yaitu 77,78% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Jumlah data yang diterima dapat dilihat pada Tabel 5.11 dan jumlah data yang sesuai dengan keinginan pengguna dapat dilihat pada Tabel 5.12

Tabel 5.11 Jumlah Data yang Diterima pada Pengguna 1

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa Inggris	Laptop	10	20	30
		Apple Smartphones	10	12	22
		Laptop Asus 2018	6	6	12
		Brand Laptop Popular	10	20	30
		Samsung Smartphones	20	30	50
		Clothes	10	14	24
Total Keseluruhan					168

2	Bahasa	Sepatu	10	20	30
		Indomie Goreng	8	16	24
		Buku Cerita Anak	10	18	28
	Indonesia	Pensil	10	20	30
		Piala Dunia	6	12	18
		Film Perahu Kertas	10	18	28
		Total Keseluruhan			

Tabel 5.12 Jumlah Data sesuai keinginan pengguna yang Diterima pada Pengguna 1

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa Inggris	Laptop	10	18	28
		Apple Smartphones	6	3	9
		Laptop Asus 2018	6	4	10
		Brand Laptop Popular	8	16	24
		Samsung Smartphones	14	19	33
		Clothes	9	12	21
Total Keseluruhan					125

2	Bahasa Indonesia	Sepatu	10	20	30
		Indomie Goreng	4	8	12
		Buku Cerita Anak	9	16	25
		Pensil	7	12	19
		Piala Dunia	5	10	15
		Film Perahu Kertas	8	14	22
		Total Keseluruhan			

Berikut hasil uji coba dengan menggunakan *Keyword* sesuai Tabel 5.2 dapat dilihat pada Tabel 5.13.

Tabel 5.13 Kesesuaian informasi yang Ditampilkan dengan Relevansi Keinginan Pengguna 2

No	Jenis Keyword	Keyword	Presentase Kesesuaian		
			Website	Image	Rata-rata
1	Bahasa	Windows	90%	90%	90%
		Grammar English	50%	54,54%	52,27%
	Inggris	Asian Games 2018	100%	100%	100%
Total Keseluruhan					80,76%
2	Bahasa Indonesia	Pisang	80%	80%	80%
		Berita Terbaru	80%	78,95%	79,48%
		Sekolah Terbaik Asean	77,78%	76,47%	77,13%
Total Keseluruhan					78,87%

Dari hasil Tabel 5.13 dapat diberikan analisis bahwa penggunaan *Keyword* Berbahasa Inggris dan Bahasa Indonesia menghasilkan pencarian yang akurat. Pencarian menggunakan *Keyword* Bahasa Inggris memiliki prosentase rata-rata keakuratan hasil yang ditampilkan dengan relevansi keinginan pengguna yaitu 80,76% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Sedangkan pencarian menggunakan *Keyword* Bahasa Indonesia memiliki prosentase rata-rata keakuratan hasil yang ditampilkan dengan relevansi keinginan pengguna yaitu 78,87% untuk pencarian yang sudah melingkupi satu kata, dua kata dan tiga kata. Jumlah data yang diterima dapat dilihat pada Tabel 5.14 dan jumlah data yang sesuai dengan keinginan pengguna dapat dilihat pada Tabel 5.15

Tabel 5.14 Jumlah Data yang Diterima pada Pengguna 2

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa Inggris	Windows	10	20	30
		Grammar English	8	11	19
		Asian Games 2018	4	7	11
Total Keseluruhan					50
2	Bahasa Indonesia	Pisang	10	20	30
		Berita Terbaru	10	19	29
		Sekolah Terbaik Asean	9	17	26
Total Keseluruhan					85

Tabel 5.15 Jumlah Data sesuai keinginan pengguna yang Diterima pada Pengguna 2

No	Jenis Keyword	Keyword	Kemunculan Data		
			Website	Image	Total
1	Bahasa	Windows	9	18	27
		Grammar English	4	6	10
	Inggris	Asian Games 2018	4	7	11
Total Keseluruhan					48
2	Bahasa	Pisang	8	16	24
		Berita Terbaru	8	15	23
	Indonesia	Sekolah Terbaik Asean	7	13	20
		Total Keseluruhan			

5.2.4 Pengujian Peningkatan Kualitas Hasil Pencarian Tanpa Melanggar Persyaratan Privasi Pengguna

Skenario yang dilakukan adalah dengan melakukan percobaan memasukkan *Query/Keyword* sesuai dengan *Tabel 5.3*. Kemudian akan dilihat kualitas hasil pencarian yaitu dengan membandingkan antara informasi yang diterima atau ditampilkan pada halaman hasil dengan keinginan pengguna. Pada setiap data yang telah diterima akan dilihat isi dari data tersebut sehingga bisa diketahui apakah data yang diterima sesuai dengan keinginan pengguna dan mengingat adanya personalisasi data pengguna itu sendiri, maka hasil pencarian juga selalu meningkat. Berikut hasil

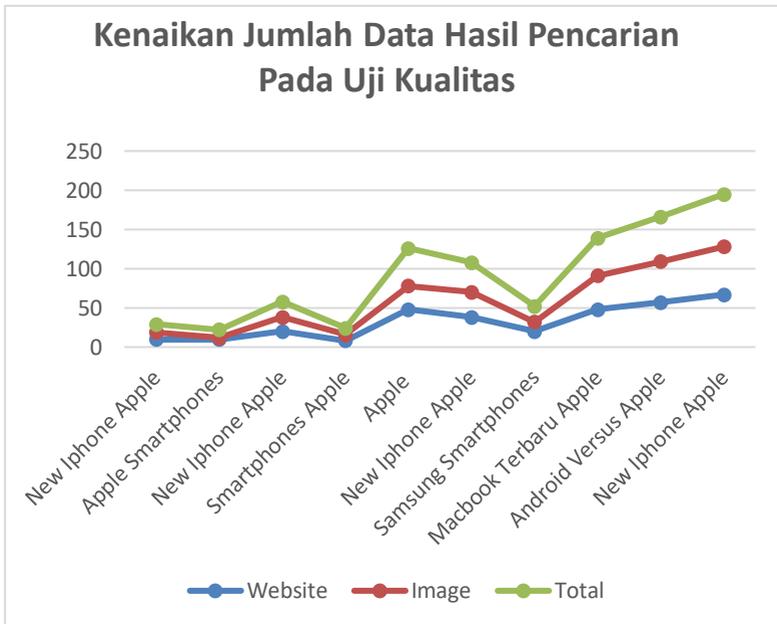
uji coba dengan menggunakan *Keyword* sesuai Tabel 5.2 dapat dilihat pada Tabel 5.16 dan 5.17. serta penyajian grafik dapat dilihat pada Gambar 5.40 dan 5.41.

Tabel 5.16 Jumlah Data yang Diterima

Percobaan ke-n	Keyword	Data		
		Website	Image	Total
1	New Iphone Apple	10	19	29
Total Keseluruhan				29
2	Apple Smartphones	10	12	22
3	New Iphone Apple	20	38	58
Total Keseluruhan				80
4	Smartphones Apple	8	16	24
5	Apple	48	78	126
6	New Iphone Apple	38	70	108
Total Keseluruhan				258
7	Samsung Smartphones	20	32	52
8	Macbook Terbaru Apple	48	91	139
9	Android Versus Apple	57	109	166
10	New Iphone Apple	67	128	195
Total Keseluruhan				552

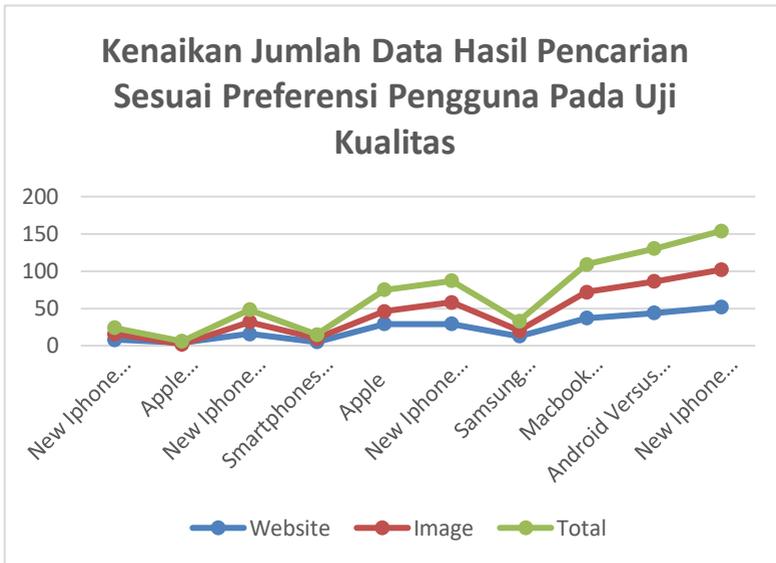
Tabel 5.17 Jumlah Data sesuai keinginan pengguna yang Diterima

Percobaan ke-n	Keyword	Data		
		Website	Image	Total
1	New Iphone Apple	8	16	24
Total Keseluruhan				24
2	Apple Smartphones	4	2	6
3	New Iphone Apple	16	32	48
Total Keseluruhan				54
4	Smartphones Apple	5	10	15
5	Apple	29	46	75
6	New Iphone Apple	29	58	87
Total Keseluruhan				177
7	Samsung Smartphones	13	20	33
8	Macbook Terbaru Apple	37	72	109
9	Android Versus Apple	44	86	130
10	New Iphone Apple	52	102	154
Total Keseluruhan				426



Gambar 5.40. Grafik Kenaikan Jumlah Data Hasil Pencarian

Dari hasil Tabel 5.9 dan 5.10 dapat diberikan analisis pada saat pengguna melakukan pencarian dengan banyak *Keyword* yang saling berkaitan, terlihat bahwa hasil pencarian mengalami kenaikan kualitas sesuai dengan proses personalisasinya. Hal ini terjadi karena sistem secara dinamis akan menyusun *Taxonomy Repository* pengguna guna mengakuratkan hasil pencarian berdasarkan riwayat pencarian pengguna itu sendiri.



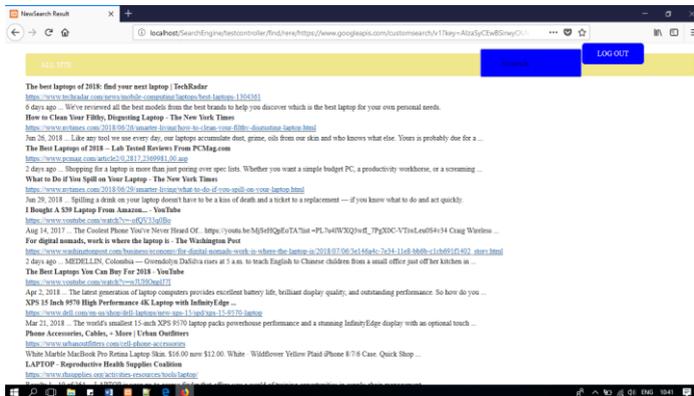
Gambar 5.41. Grafik Kenaikan Jumlah Data Hasil
Sesuai Preferensi Pengguna Pencarian

Data diatas menunjukkan ketika satu *Keyword* dimasukkan secara berulang-ulang, data terkait *Keyword* tersebut juga akan selalu dikembangkan agar hasil pencarian maksimal. *Keyword* yang diuji kualitas hasil pencariannya pada percobaan di atas yaitu “*New Iphone Apple*”. Pada percobaan ke-1, hasil untuk *Keyword* “*New Iphone Apple*” tersebut memiliki keakuratan hasil yang ditampilkan dengan relevansi keinginan yaitu sebanyak 24 data dari 29 jumlah data. Pada percobaan ke-2, ke-4, ke-5, ke-7, ke-8, dan ke-9 pengguna memasukkann *Keyword* yang bersesuaian dengan *Keyword* “*New Iphone Apple*”. Kemudian pada percobaan ke-3, ke-6 dan ke-10 pengguna memasukkan lagi

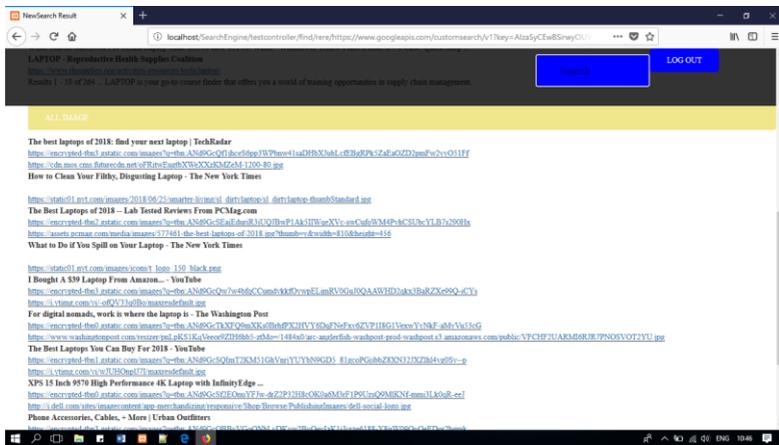
Keyword “New Iphone Apple” untuk dilihat kualitasnya, sehingga disini dapat dilihat akumulasi data *Repository* pengguna menunjang hasil pencarian yang dimunculkan sistem. Pada percobaan ke-3, *Keyword “New Iphone Apple”* memiliki keakuratan hasil yang ditampilkan dengan relevansi keinginan yaitu sebanyak 48 data dari 58 jumlah data. Pada percobaan ke-6, *Keyword “New Iphone Apple”* memiliki keakuratan hasil yang ditampilkan dengan relevansi keinginan yaitu sebanyak 87 data dari 108 jumlah data. Pada percobaan ke-10, *Keyword “New Iphone Apple”* memiliki keakuratan hasil yang ditampilkan dengan relevansi keinginan yaitu sebanyak 154 data dari 195 jumlah data. Peningkatan jumlah data mengindikasikan banyaknya data alternatif. Total keseluruhan jumlah data yang diterima mulai dari percobaan ke-1 sampai percobaan ke-10 juga selalu mengalami peningkatan yaitu 19, 80, 258 dan 552, begitupun jumlah data sesuai keinginan pengguna juga mengalami peningkatan yaitu 24, 54, 117, dan 426. Hal ini menandakan bahwa semakin banyak pencarian terhadap suatu *Keyword* yang saling berkaitan akan mendukung proses personalisasi. Meskipun parameter pengambilan data yang digunakan sudah dipisahkan dengan persyaratan privasi atau privasi khusus pengguna, dari data hasil uji coba terlihat bahwa jumlah data yang sesuai semakin meningkat disetiap pencarian. Sehingga dengan adanya perlindungan privasi pada penelusuran web terpersonalisasi yang dibuat, sistem telah memberikan akurasi yang tinggi pada relevansi keinginan pengguna terhadap hasil pencarian. Oleh karena itu, sistem telah memberikan hasil yang kualitasnya sama seperti mesin pencari web terpersonalisasi yang tidak memiliki perlindungan privasi seperti *Google*..

5.2.5 Perbandingan Aplikasi

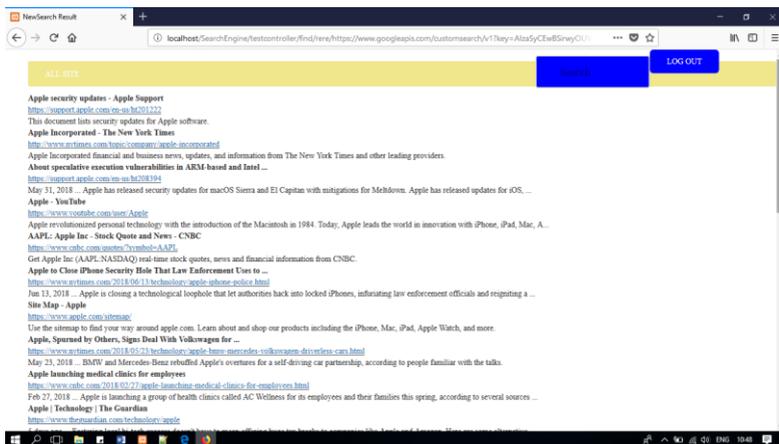
Selanjutnya menguji perbandingan pencarian dari sistem pencari web (*Web Search Engine*) memanfaatkan *Google API Custome Search* yang dibuat pada Tugas Akhir ini dengan mesin pencari *Google*. Uji coba ini dilakukan dengan membandingkan antara hasil pencarian pada aplikasi yang telah dibuat dan hasil pencarian pada halaman pertama dari mesin pencari *Google*. Pengujian dilakukan dengan memasukkan *Keyword* yang terdiri dari satu kata, dua kata dan tiga kata. *Keyword* yang digunakan yaitu “*Laptop*”, “*Apple Smartphones*” dan “*Buku Cerita Anak*”. Hasil pencarian pada aplikasi yang dibuat terdapat pada Gambar 5.42, 5.43, 5.44, 5.45, 5.46, 5.47 dan hasil perlindungan privasi yang memperlihatkan parameter pengambilan data adalah privasi umum saja dapat dilihat pada Gambar 5.48, 5.49, 5.50 serta data user yang masuk dapat dilihat pada Gambar 5.51.



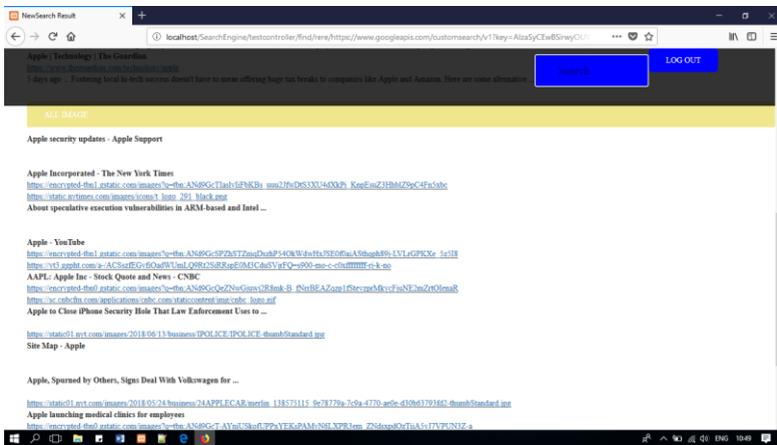
Gambar 5.42. Hasil Pencarian “*Laptop*” pada Kategori *Website* di Aplikasi yang telah dibuat



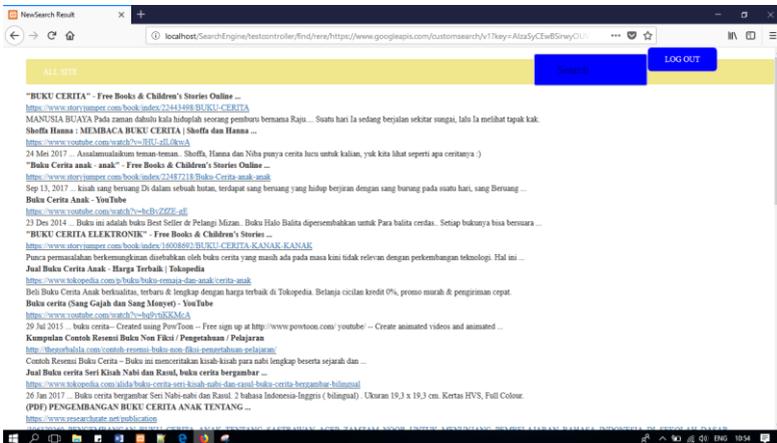
Gambar 5.43. Hasil Pencarian “*Laptop*” pada katagori *Image* di Aplikasi yang telah dibuat



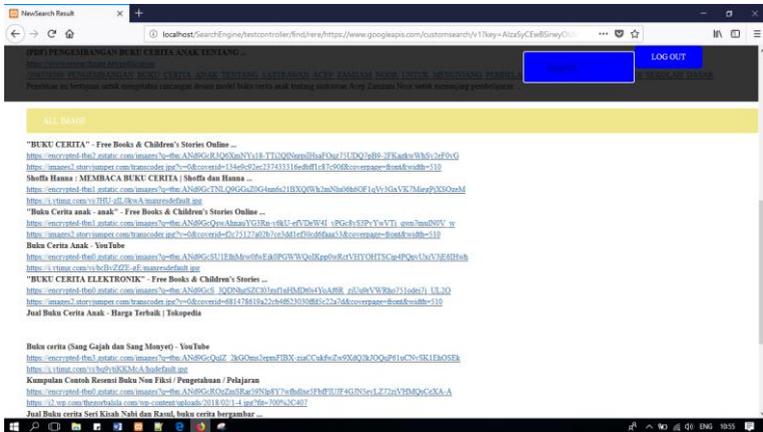
Gambar 5.44. Hasil Pencarian “*Apple Smartphones*” pada katagori *Website* di Aplikasi yang telah dibuat



Gambar 5.45. Hasil Pencarian “Apple Smartphones” pada katagori *Image* di Aplikasi yang telah dibuat



Gambar 5.46. Hasil Pencarian “Buku Cerita Anak” pada katagori *Website* di Aplikasi yang telah dibuat



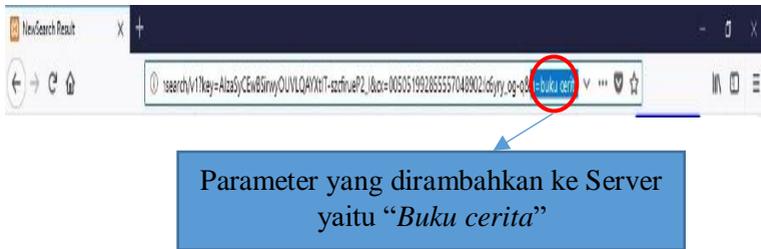
Gambar 5.47. Hasil Pencarian “*Buku Cerita Anak*” pada kategori *Image* di Aplikasi yang telah dibuat



Gambar 5.48. Parameter untuk pencarian “*Laptop*” ke Server pada Aplikasi yang telah dibuat pada



Gambar 5.49. Parameter untuk pencarian “*Apple Smartphone*” ke Server pada Aplikasi yang telah dibuat

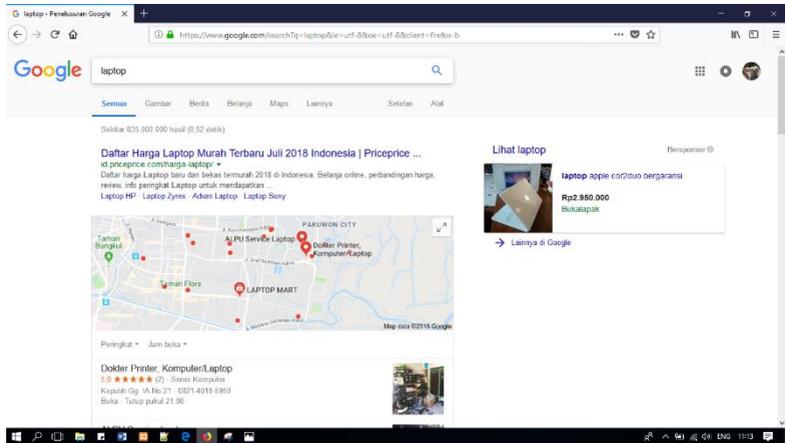


Gambar 5.50. Parameter untuk pencarian “*Buku Cerita Anak*” ke Server pada Aplikasi yang telah dibuat

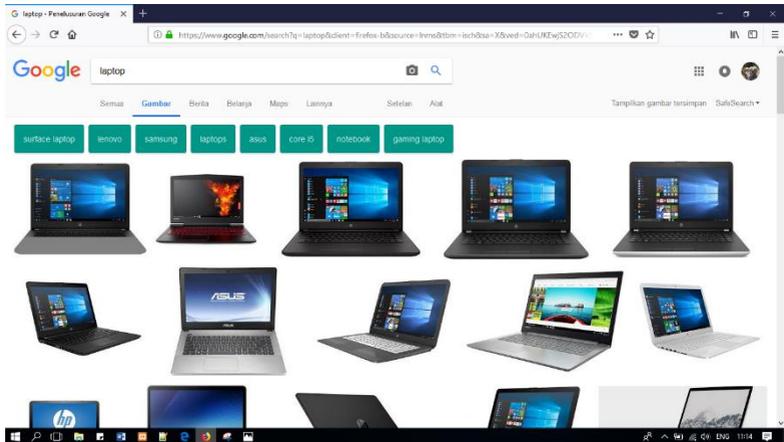


Gambar 5.51. Data Login Pengguna yang Masuk ke Sistem Aplikasi yang telah dibuat

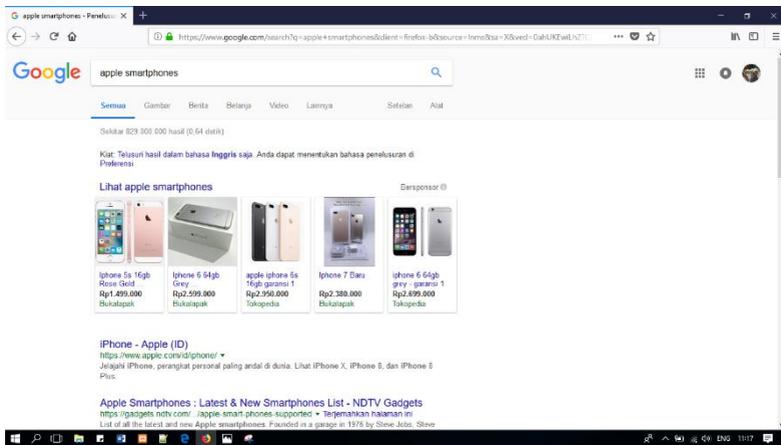
Sedangkan untuk hasil pencarian pada mesin pencari *Google* menghasilkan informasi terdapat pada Gambar 5.52, 5.53, 5.54, 5.55, 5.56, 5.57 dan pelanggaran privasi dimana parameter pengambila data merupakan keseluruhan *Keyword* sehingga persyaratan privasi pengguna dilanggar dapat dilihat pada Gambar 5.58, 5.59 5.60 serta data user yang masuk dapat dilihat pada Gambar 5.61



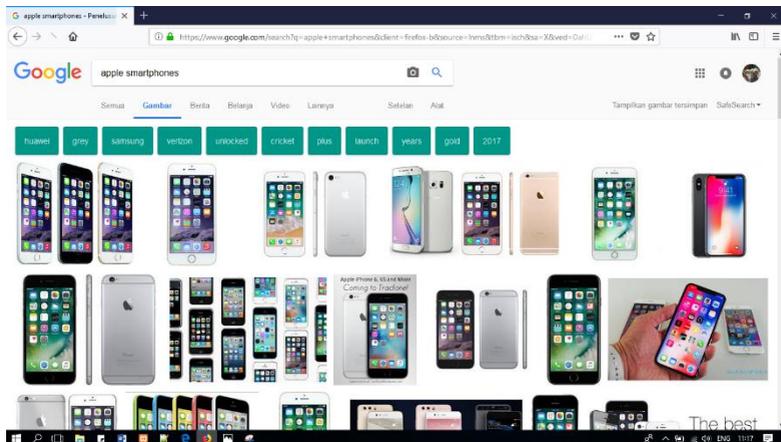
Gambar 5.52. Hasil Pencarian “*Laptop*” pada kategori *Website* di *Google*



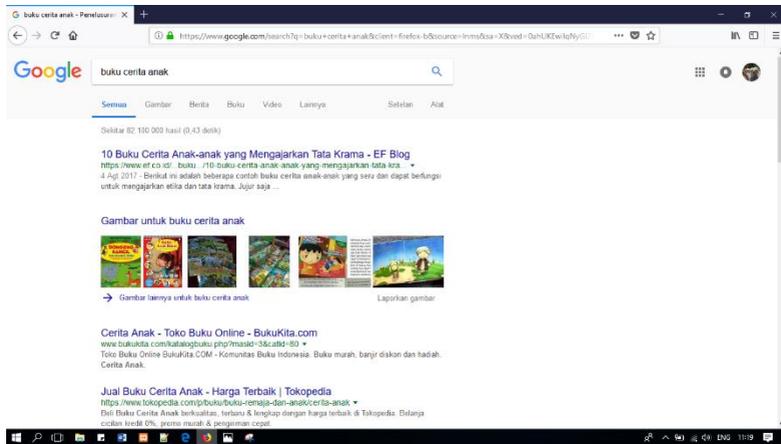
Gambar 5.53. Hasil Pencarian “*Laptop*” pada kategori *Image* di *Google*



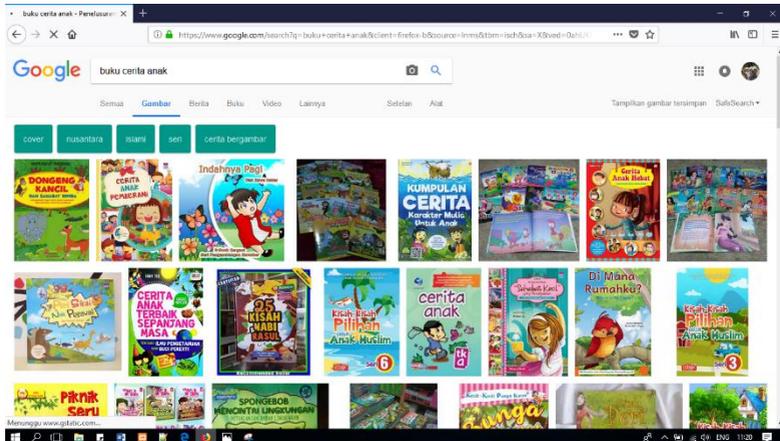
Gambar 5.54 Hasil Pencarian “Apple Smartphones” pada katagori *Website* di Google



Gambar 5.55. Hasil Pencarian “Apple Smartphones” pada katagori *Image* di Google



Gambar 5.56. Hasil Pencarian “*Buku Cerita Anak*” pada katagori *Website* di Google



Gambar 5.57. Hasil Pencarian “*Buku Cerita Anak*” pada katagori *Image* di Google

Dari hasil perbandingan antara mesin pencari *Google* dan aplikasi mesin pencari yang dibuat pada Tugas Akhir ini, terlihat bahwa kedua mesin pencari tersebut sudah dapat melakukan personalisasi untuk pengguna sehingga hasil yang ditampilkan ke pengguna sudah relevan dengan keinginan pengguna. Kualitas pencarian jika hanya membandingkan dari sisi personalisasi data, kedua mesin pencari menghasilkan hasil pencarian yang berkualitas, tetapi jika melihat dari perlindungan privasi kedua sistem memiliki perbedaan. Terlihat bahwa pada mesin pencari *Google*, *Query/Keyword* yang dimasukkan pengguna semuanya akan dijadikan parameter pencarian sehingga terjadi pelanggaran privasi pengguna, sedangkan pada aplikasi sistem pencari web yang dibuat pada Tugas Akhir ini, parameter yang digunakan untuk proses pengambilan data ke *Server* merupakan *Query/Keyword* yang sudah diproses untuk memisahkannya dengan persyaratan privasi pengguna, sehingga privasi pengguna terlindungi tetapi kualitas pencarian tetap meningkat.

5.1.4 Hasil Uji Coba

Setelah melakukan beberapa uji coba terhadap aplikasi yang dibuat untuk meningkatkan kualitas hasil pencarian tanpa melanggar privasi pengguna pada penelusuran web terpersonalisasi dengan menggunakan algoritma *Greedy*, dimulai dari uji coba kesesuaian *Keyword* dengan hasil yang ditampilkan, keakuratan hasil dengan relevansi keinginan pengguna dan perbandingan aplikasi yang dibuat dengan mesin pencari web *Google*. Hasil dari beberapa uji coba tersebut dapat dilihat pada Tabel 5.18.

Tabel 5.18 Hasil uji coba

No	Proses Uji Coba	Hasil yang diharapkan	Keterangan
1.	Kesesuaian <i>Keyword</i> dengan hasil yang ditampilkan	Informasi yang ditampilkan sesuai dengan <i>Keyword</i> yang dimasukkan	Berhasil
2.	Keakuratan hasil pencarian dengan relevansi keinginan pengguna	<ol style="list-style-type: none"> 1. Mengelompokkan data sesuai katagori pencarian yaitu <i>Website</i> dan <i>Image</i> 2. Melakukan perlindungan privasi pengguna dengan algoritma yang digunakan 3. Memberikan hasil pencarian yang sesuai atau relevan dengan keinginan pengguna 	Berhasil

3.	Peningkatan kualitas hasil pencarian tanpa melanggar persyaratan privasi pengguna	<ol style="list-style-type: none"><li data-bbox="499 193 762 344">1. Membangun <i>Taxsonomy Repository</i> pengguna<li data-bbox="499 344 762 496">2. Memberikan hasil pencarian dengan kualitas yang tinggi<li data-bbox="499 496 762 735">3. Hasil pencarian yang sesuai dengan preferensi pengguna selalu meningkat	Berhasil
4.	Perbandingan aplikasi	Aplikasi yang dibuat berbeda dengan mesin pencari yang sudah ada sebelumnya	Berhasil

BAB VI PENUTUP

Pada bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan dan saran yang dapat digunakan jika penelitian ini dikembangkan.

6.1 Kesimpulan

Dari hasil uji coba aplikasi peningkatan kualitas hasil pencarian tanpa melanggar privasi pengguna pada penelusuran web terpersonalisasi menggunakan algoritma *Greedy* dapat disimpulkan sebagai berikut:

1. Aplikasi yang dibuat telah mampu mengimplementasikan perlindungan persyaratan privasi pengguna pada saat pengambilan data ke *Server*. Hal ini dapat dilihat pada Gambar 5.10.
2. Pada uji coba keakuratan hasil dengan relevansi keinginan pengguna, penggunaan *Keyword* Berbahasa Inggris dan Bahasa Indonesia menghasilkan pencarian yang akurat. Hal ini ditunjukkan pada Tabel 5.6.
3. Pada uji coba peningkatan kualitas hasil pencarian tanpa melanggar persyaratan privasi pengguna (privasi khusus), hasil pencarian mengalami kenaikan kualitas. Hal ini ditunjukkan pada grafik Gambar 5.40 dan 5.41. Semakin banyak pencarian terhadap suatu *Keyword* yang saling berkaitan akan mendukung proses personalisasi. Meskipun parameter pengambilan data yang digunakan sudah dipisahkan dengan persyaratan privasi atau privasi khusus pengguna, dari data hasil uji coba terlihat bahwa jumlah data yang sesuai semakin meningkat disetiap pencarian. Sehingga dengan adanya perlindungan privasi pada penelusuran web terpersonalisasi yang dibuat,

sistem telah memberikan akuransi yang tinggi pada relevansi keinginan pengguna terhadap hasil pencarian. Oleh karena itu, sistem telah memberikan hasil yang kualitasnya sama seperti mesin pencari web terpersonalisasi yang tidak memiliki perlindungan privasi seperti *Google*.

6.2 Saran

Berdasarkan hasil uji coba pada aplikasi yang telah dibuat, aplikasi masih memiliki banyak kekurangan. Untuk itu diberikan saran untuk pengembangan aplikasi web diantaranya adalah sebagai berikut:

1. Mendapatkan sumber data melalui proses *Crawling*. Hal ini dikarenakan data yang hanya dapat diakuisisi dari *Server Google API Custome Search* terbatas.
2. Memperbaiki fungsi *Filtering* data sehingga data hasil *Filter* lebih sesuai dengan persyaratan privasi pengguna (privasi khusus).

DAFTAR PUSTAKA

- [1] B. Sekharbabu, P. Lakshmi Prasanna at al (2016). "Profile Based Personalized Web Search Using Greedy Algorithm". **ARPN Journal of Engineering and Applied Sciences VOL. 11 No.9.**
- [2] Kolte, Shilpa at al. (2016). "Client side Privacy Protection Using Personalized Web Search". **Procedia Computer Science Vol. 79**, Hal. 1029-1035.
- [3] Totare, Reshma Y at al. (2014). "Personalized Web Search Protecting Privacy Using Greedy Algorithm". **International Journal of Emerging Technology and Innovative Engineering Volume I, Issue 2.**
- [4] Lidan Shou, He Bai, Ke Chen, and Gang Chen. (2014). "Supporting Privacy Protection, in Personalized WebSearch," **Proc. IEEE transactions on knowledge and data engineering VOL:26 NO:2.**
- [5] Jayant Kiran simply. "Personalize Web search with Enhanced Privacy Protection". **International Journal of Advanced Research in Computer Science and software Engineering.**
- [6] J. Ghilby Varghese, Varghese Charlse M. (2015). "Privacy Protection in Personalized Web Search Using Homomorphic Encryption". **International Journal of Scientific and Research Publications Volume 5, Issue 9.**
- [7] V. Kavitha, T. Uma maheswari. (2015). "Comparison User Customizable Privacy-preserving Search (UPS) with User Customizable Online Privacy-preserving Search with K-anonymity (UCOPSK)". **International Journal of Sciences and Engineering Volume 3, Issue 9.**
- [8] Totare, Reshma Y at al. (2014). "Personalized Web Search Protecting Privacy Using Greedy Algorithm". **International Journal of Emerging Technology and Innovative Engineering Volume I, Issue 2.**

- [9] Wardana, S.hut., m.si,. (2010). **Menjadi Master PHP dengan Framework CodeIgniter**. Jakarta : Elex Media.
- [10] Griffiths, Adam. (2010). **CodeIgniter 1.7 Professional Development**. Brimingham : Packt Publishing.
- [11] <https://www.codeigniter.com> diakses pada tanggal 5 Juli 2018.
- [12] J. Teevan, S.T. Dumais, and E. Horvitz. (2005). "Personalizing Search via Automated Analysis of Interests and Activities," **Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR)**, pp. 449-456.
- [13] Dwi, K. U., Adi, M. Y., Ariel, M. D. P. (2014). **Sinkronisasi Otomatis Status Dosen dengan raspberry Pi Melalui Facebook Graph API**. Skripsi. Program Studi Teknik Informatika, Universitas BINUS.
- [14] Andika, R.(2011). **Penerapan CI (CodeIgniter) Dalam Pengembangan Sistem Informasi Manajemen Surat Dan Pengarsipan**. Skripsi. Program Studi Teknik Informatika, Universitas Islam Negeri Jakarta.
- [15] https://id.wikipedia.org/wiki/Unified_Modeling_Language diakses pada tanggal 5 Juli 2018.
- [16] Dharwiyanti, Sri. (2003). **Pengantar Unified Modeling Language (UML)**. Ilmu Komputer.
- [17] <https://id.wikipedia.org/wiki/XAMPP> diakses pada tanggal 5 Juli 2018
- [18] <https://id.wikipedia.org/wiki/PHP> diakses pada tanggal 5 Juli 2018.
- [19] Nurseitov, N. dkk. (2009). **Comparison of JSON and XML Data Interchange Formats: A Case Study. Department of Computer Science**, Montana State University,USA.
- [20] <https://console.developers.google.com/> diakses pada tanggal 5 Juli 2018