

**PENERAPAN INFORMATION RETRIEVAL DALAM
PENCARIAN SOURCE CODE BAHASA PYTHON DENGAN
PENGENALAN RELASI CLASS**

Skripsi



oleh

YEHESKIEL REZA ANDRAZEIN

71170149

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2021

**PENERAPAN INFORMATION RETRIEVAL DALAM
PENCARIAN SOURCE CODE BAHASA PYTHON DENGAN
PENGENALAN RELASI CLASS**

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar Sarjana Komputer

Disusun oleh

YEHESKIEL REZA ANDRAZEIN

71170149

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2021

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI/TESIS/DISERTASI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Kristen Duta Wacana, saya yang bertanda tangan di bawah ini:

Nama : Yeheskiel Reza Andrazein
NIM : 71170149
Program studi : Informatika
Fakultas : Teknologi Informasi
Jenis Karya : Skripsi (tuliskan salah satu)

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Kristen Duta Wacana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

**“PENERAPAN INFORMATION RETRIEVAL DALAM PENCARIAN
SOURCE CODE BAHASA PYTHON DENGAN PENGENALAN RELASI
CLASS”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Kristen Duta Wacana berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : 1 Juli 2021

Yang menyatakan



(Yeheskiel Reza Andrazein)
NIM.71170149

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

PENERAPAN INFORMATION RETRIEVAL DALAM PENCARIAN SOURCE CODE BAHASA PYTHON DENGAN PENGENALAN RELASI CLASS

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 23 Juni 2021



YEHESKIEL REZA ANDRAZEIN
71170149

HALAMAN PERSETUJUAN

Judul Skripsi : PENERAPAN INFORMATION RETRIEVAL
DALAM PENCARIAN SOURCE CODE BAHASA
PYTHON DENGAN PENGENALAN RELASI
CLASS

Nama Mahasiswa : YEHESKIEL REZA ANDRAZEIN

N I M : 71170149

Matakuliah : Skripsi (Tugas Akhir)

Kode : TIW276

Semester : Genap

Tahun Akademik : 2020/2021

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 20 Mei 2021

Dosen Pembimbing I



Budi Susanto, S.Kom., M.T.

Dosen Pembimbing II



Kristian Adi Nugraha, S.Kom., M.T.

HALAMAN PENGESAHAN

PENERAPAN INFORMATION RETRIEVAL DALAM PENCARIAN SOURCE CODE BAHASA PYTHON DENGAN PENGENALAN RELASI CLASS

Oleh: YEHESKIEL REZA ANDRAZEIN / 71170149

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 11 Juni 2021

Yogyakarta, 23 Juni 2021
Mengesahkan,

Dewan Penguji:

1. Budi Susanto, SKom., M.T.
2. Kristian Adi Nugraha, S.Kom., M.T.
3. Gloria Virginia, S.Kom., MAI, Ph.D.
4. Rosa Delima, S.Kom., M.Kom.

Budi
Susanto

Digitally signed by Budi Susanto,
DN: cn=Budi Susanto, o=Universitas Kristen Duta Wacana, ou=FTSI, email=budisusanto@ukdw.ac.id



Dekan

(Restyananto S.Kom. MSIS. Ph.D.)

Ketua Program Studi

(Gloria Virginia Ph.D.)

UCAPAN TERIMAKASIH

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena Rahmat-Nya penulis dapat melaksanakan penelitian ini dengan baik. Penulis mengucapkan banyak terimakasih kepada kedua orang tua penulis sebagai *support system* penulis selama menjalani perkuliahan. Penulis juga mengucapkan terimakasih kepada dosen pembimbing yang telah membantu dan membimbing penulis dalam penelitian ini. Tak lupa penulis ucapkan terimakasih kepada rekan-rekan penulis yang turut mendukung penulis.

©UKDW

INTISARI

Pencarian referensi pada sekumpulan dokumen *source code* adalah hal yang umum dilakukan dalam pengembangan perangkat lunak. Seiring bertambahnya baris kode dan banyaknya file yang digunakan, mencari kata kunci pada sebuah repositori *source code* menjadi kurang praktis, terutama jika ingin mencari korelasi antara sebuah *term* dengan *term* yang lain. Untuk itu pada penelitian ini penulis mencoba untuk menerapkan konsep penerapan *Information Retrieval* dalam kegiatan pencarian referensi pada sebuah repositori *source code* yang dapat mengenali relasi *term* yang dicari dengan *term* yang lain apabila yang dicari adalah sebuah *class*.

Untuk membangun sebuah sistem IR, diperlukan pembuatan indeks yang dapat membantu proses pencarian agar dapat menemukan hasil lebih cepat. Pembuatan indeks pada penelitian ini berfokus pada nama *class* dan relasinya dengan *class* lain. Definisi fungsi yang ada pada sebuah *class* juga turut dilakukan pengindeksan. Pembuatan indeks ini dimulai dengan mem-*parsing* file *source code* untuk diambil informasi tentang *class* dan fungsi yang ada di dalamnya. Setelah itu informasi telah yang didapat dimodelkan relasinya dalam bentuk graf sebagai indeks pencarian. Proses pencarian kemudian akan mengambil informasi dari indeks pencarian yang telah dibuat.

Penelitian ini menghasilkan sebuah aplikasi penyimpanan repositori berbasis web untuk *source code* berbahasa Python yang dilengkapi *file browser* dan mesin pencarian untuk mencari referensi pada repositori. Hasil pengujian pada penelitian ini menunjukkan mesin pencari yang telah dibuat mampu mengembalikan seluruh dokumen yang diinginkan beserta relasi yang ada pada hasil pencarian tersebut.

Kata kunci – *Information Retrieval*, *indexing*, *parsing*, pencarian *source code*, relasi *class*, repositori Python, graf

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
UCAPAN TERIMAKASIH.....	vi
INTISARI.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	3
1.7 Struktur Penulisan.....	5
BAB 2 TINJAUAN PUSTAKA.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Landasan Teori.....	9
2.2.1 Information Retrieval.....	9
2.2.2 Indexing.....	10
2.2.3 Inverted Index.....	10
2.2.4 Parametric & Zone Indexing.....	11
2.2.4 Graf.....	11

2.2.5 Redis Database.....	12
BAB 3 PERANCANGAN SISTEM	13
3.1 Kebutuhan Sistem.....	13
3.1.1 Kebutuhan Fungsional	13
3.1.2 Kebutuhan Teknis	13
3.2 Blok Diagram Sistem	13
3.3 Arsitektur Sistem.....	17
3.4 Rancangan Struktur Data dan Basis Data.....	18
3.4.1 Rancangan Model Hasil Parsing.....	18
3.4.2 Rancangan Basis Data Sistem.....	22
3.5 Rancangan Antarmuka Aplikasi.....	28
3.5.1 Halaman <i>Welcome</i>	28
3.5.2 Halaman <i>Create Repository</i>	28
3.5.3 Halaman <i>Processing Repository</i>	29
3.5.4 Halaman Utama Aplikasi.....	29
3.6 Rancangan Pengujian Sistem	30
BAB 4 HASIL DAN PEMBAHASAN.....	33
4.1 Hasil Implementasi.....	33
4.1.1 Persiapan	33
4.1.2 Pembuatan Modul Parser Source Code	37
4.1.3 Pembuatan Modul Indexer.....	41
4.1.4 Pembuatan modul pencarian.....	43
4.1.5 Pembuatan Tampilan Antarmuka	48
4.2 Pengujian Sistem	53
4.3 Pembahasan Hasil Pengujian.....	54
BAB 5 KESIMPULAN DAN SARAN	61

5.1 Kesimpulan.....	61
5.2 Saran.....	61
Daftar Pustaka.....	63
Lampiran.....	64
Lampiran 1 – Modul Parser Source Code.....	64
Lampiran 2 – Modul Indexer Source Code.....	65
Lampiran 3 – Endpoint pencarian.....	68
Lampiran 4 – Tabel Hasil Pengujian.....	79
1. Repository buatan penulis.....	79
2. Repository dari GitHub (Slack).....	101
Lampiran 5 – Kartu Konsultasi.....	112

©UKDWN

DAFTAR GAMBAR

Gambar 1.1 Contoh kode program.....	3
Gambar 1.2 Contoh index	4
Gambar 2.1 Fitur Search Everywhere pada IDE PyCharm (https://resources.jetbrains.com/help/img/idea/2020.2/py_search_all_window.png)	7
Gambar 2.2 Pencarian class dengan awalan "ca" (https://resources.jetbrains.com/help/img/idea/2020.2/py_goto_class.png)	8
Gambar 2.3 Fitur Find Usage pada IDE PyCharm (https://www.jetbrains.com/help/img/idea/2020.2/py_find_usages.png)	8
Gambar 2.4 Zone index (An Introduction to Information Retrieval)	11
Gambar 2.5 Contoh penyimpanan key-value pada Redis untuk tipe data String.. (Redis in Action).....	12
Gambar 3.1 Rancangan pembuatan index repository	14
Gambar 3.2 Flowchart pengumpulan file source code.....	14
Gambar 3.3 Flowchart tokenisasi source code.....	15
Gambar 3.4 Flowchart proses indexing pada Redis.....	16
Gambar 3.5 Rancangan sistem IR.....	17
Gambar 3.6 Arsitektur sistem	18
Gambar 3.7 Class Diagram hasil parsing pada sistem.....	22
Gambar 3.8 Contoh Visualisasi Graph	26
Gambar 3.9 Representasi index dan inverted index untuk node Class	26
Gambar 3.10 Representasi index dan inverted index untuk node Function.....	26
Gambar 3.11 Representasi index dan inverted index untuk relasi node Class dengan node Class.....	27
Gambar 3.12 Representasi index dan inverted index untuk relasi node Class dengan node Function	27
Gambar 3.13 Rancangan halaman welcome	28
Gambar 3.14 Rancangan halaman create repository	28
Gambar 3.15 Rancangan Halaman processing repository	29
Gambar 3.16 Rancangan halaman utama aplikasi	29

Gambar 4.1 Konfigurasi modul pada redis.conf	34
Gambar 4.2 Redis berjalan pada port 6379	34
Gambar 4.3 Konfigurasi network pada redis.conf	35
Gambar 4.4 Struktur file project saat pertama kali dibuat	36
Gambar 4.5 Struktur file project setelah penambahan modul	37
Gambar 4.6 Penambahan konfigurasi pada settings.py	37
Gambar 4.7 Pseudocode modul parser	39
Gambar 4.8 Potongan baris kode definisi class RepositoryModel	40
Gambar 4.9 Pseudocode modul indexer	42
Gambar 4.10 Potongan baris kode untuk endpoint auto completer	43
Gambar 4.11 Pseudocode endpoint search	47
Gambar 4.12 Definisi class SearchResultModel dan SearchResultRelationModel	48
Gambar 4.13 Tampilan halaman loader	49
Gambar 4.14 Tampilan halaman Welcome	49
Gambar 4.15 Tampilan form pada halaman Create Repository	50
Gambar 4.16 Tampilan indikator pemrosesan repository	50
Gambar 4.17 Halaman utama aplikasi dengan file source code terbuka setelah memilih file	51
Gambar 4.18 Tampilan halaman utama aplikasi dengan hasil pencarian dan lokasi spesifik kemunculan term	51
Gambar 4.19 Tampilan utama aplikasi dengan hasil pencarian umum yang tidak memiliki term khusus	52
Gambar 4.20 Tampilan auto completer	52
Gambar 4.21 Tampilan hasil pencarian dengan relasi berupa class induk dan class anak	52
Gambar 4.22 Tampilan hasil pencarian dengan relasi berupa class yang memiliki fungsi tertentu	53
Gambar 4.23 Tampilan hasil pencarian yang tidak memiliki relasi	53

DAFTAR TABEL

Tabel 3.1 UML Model <i>class</i> RepositoryModel	18
Tabel 3.2 UML Model <i>class</i> DocumentModel	19
Tabel 3.3 UML Model <i>class</i> ImportModuleModel.....	19
Tabel 3.4 UML Model <i>class</i> ClassModel	20
Tabel 3.5 UML Model <i>class</i> ParentClassModel	21
Tabel 3.6 UML Model <i>class</i> FunctionModel.....	21
Tabel 3.7 Kamus data tabel Repositories	22
Tabel 3.8 Kamus data tabel FileModel	23
Tabel 3.9 Struktur penyimpanan dokumen pada RediSearch	23
Tabel 4.1 Hasil pengujian sistem untuk repositori buatan penulis.....	54
Tabel 4. 2 Hasil pengujian sistem untuk repositori yang diambil dari GitHub.....	57

©UKDVM

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pencarian *term* pada *source code* selama ini hanya berdasarkan *keyword* yang diberikan oleh pengguna. Salah satu contohnya apabila kita ingin mencari di mana saja sebuah *class* dengan nama “A” dipakai, maka kita akan mengetikkan nama *class* tersebut dalam pencarian *source code*. Dari hasil pencarian tersebut maka akan ditampilkan pada *file* mana saja dan pada baris mana saja *class* A dituliskan. Pada sebuah project besar yang terdiri dari banyak *file* dan penggunaan *class* yang dapat berulang kali terjadi, pencarian dengan metode seperti ini agaknya membuat pencarian kurang efisien. Hal tersebut disebabkan karena pencarian hanya berdasar pada bagian mana kemunculan *keyword* berada tanpa mengetahui relasi *keyword* tersebut dengan *file source code* yang lain. Jika kita ingin mencari *class* yang diturunkan dari *class* “A” ini, kita perlu melihat satu per satu pada *file* mana saja yang menuliskan “A” sebagai *parent*-nya. Oleh sebab itu kita tidak bisa menuliskan, sebagai contoh, “*inherited from A*” sebagai kata kunci pencarian. Maka dari itu dirasa akan bermanfaat jika diciptakan sebuah sistem pencarian *source code* dengan pendekatan semantik yang dapat mengenali relasi sebuah *term* (pada kasus ini adalah sebuah *class*) dengan *term* yang lain pada sebuah *project* pemrograman.

Untuk membangun sebuah sistem pencarian dengan pendekatan semantik tersebut, diperlukan sebuah indeks yang mampu mengenali relasi-relasi antar *class* pada sebuah *project* pemrograman. Indeks yang dibangun tersebut selain berisikan nama-nama *class* yang ada pada *source code*, juga memuat relasi antar *class* tersebut seperti *inheritance* dan asosiasi. Selain itu juga diperlukan *inverted index* yang dapat mengarahkan hasil pengindeksan ke dokumen *source code* yang dimaksud. Setelah *index* dan *inverted index* untuk *source code* selesai dibuat, maka terbentuklah sebuah *repository* untuk sebuah *project source code* yang nantinya dapat diambil informasinya dengan *information retrieval system*.

Pada penelitian ini penulis akan membangun sebuah sistem temu balik untuk pencarian *source code* yang tidak hanya mampu mencari berdasarkan *keyword* namun juga mencari relasi antar-*class* menggunakan *query string* khusus. Hal pertama yang akan dilakukan adalah melakukan *preprocessing* diantaranya berupa tokenisasi terhadap *source code*. Informasi dari hasil tokenisasi ini akan disimpan menjadi sebuah indeks yang berisi nama *class* beserta relasinya menggunakan *database* Redis. Dari setiap *term* yang telah dibuat indeksnya, akan dibuat juga *inverted index* untuk mengarahkan hasil *index* ke dokumen *source code* yang dimaksud. Selanjutnya adalah membuat sistem temu balik untuk mengembalikan hasil pencarian yang sesuai dengan *query* pengguna yang diambil dari indeks yang telah dibuat.

1.2 Perumusan Masalah

Bagaimana membuat sistem temu balik untuk pencarian *source code* pada sebuah *repository* terindeks yang dapat mengenali relasi antar-*class*?

1.3 Batasan Masalah

Pada penelitian ini penulis membuat batasan terhadap ruang lingkup penelitian agar penelitian dapat lebih fokus terhadap permasalahan yang dibahas. Adapun batasan masalah yang dibuat penulis sebagai berikut:

1. Pencarian semantik hanya dilakukan apabila yang dicari adalah sebuah *class*, selebihnya hanya pencarian menggunakan *keyword matching*.
2. Relasi *inheritance* digunakan sebagai acuan apakah sebuah *class* memiliki hubungan dengan *class* yang lain.
3. *Term* pada indeks yang akan dibuat relasinya dengan *term* lain adalah nama-nama *class* dan nama-nama fungsi yang ada pada sebuah *class*.
4. Hasil pencarian pada penelitian ini tidak dilakukan pembobotan, sehingga metode evaluasi menggunakan metode evaluasi *information retrieval* untuk dokumen tak berperingkat.

1.4 Tujuan Penelitian

Penelitian ini dilakukan untuk menemukan penerapan sistem temu balik untuk pencarian *source code* yang tidak hanya dapat mencari berdasarkan *keyword*,

tetapi juga dengan pendekatan semantik dapat menampilkan korelasinya dengan *term* lain apabila yang dicari adalah sebuah *class*.

1.5 Manfaat Penelitian

Penelitian ini akan menghasilkan sebuah *repository source code* dengan sistem temu kembali untuk pencarian *source code* yang tidak hanya dapat mencari *term* pada *source code* tersebut namun juga relasinya dengan *term* lain. *Repository* ini diharapkan bisa menangani berbagai *project* yang dikembangkan dengan bahasa *Python*. Pengenalan relasi *class* yang terdapat pada *repository* ini akan berguna untuk menemukan *dependency* apa saja yang digunakan pada sebuah *class*. Sistem temu kembali yang diterapkan pada *repository* ini diharapkan mampu mempermudah pengembang perangkat lunak untuk mencari penggunaan *class* pada *project* yang terdapat pada *repository* ini.

1.6 Metodologi Penelitian

Untuk membangun sebuah sistem temu balik pada pencarian *source code*, akan dilakukan beberapa tahapan sebagai berikut:

1. Melakukan *preprocessing* terhadap *source code*

Preprocessing pada *source code* dilakukan untuk mengambil informasi nama *class* yang ada. Sebagai contoh kita memiliki *file* *Burger.py* dengan isi sebagai berikut

```
class Burger:
    """
    Base class for Burger
    """
    def __init__(self, meat="Beef", spicy=False):
        self.meat = meat
        self.spicy = spicy

    def yourOrder(self):
        print("Your order is:",self.meat,"Burger,",
"spicy" if self.spicy else "not spicy")

class CheeseBurger(Burger):
    """
    CheeseBurger has cheese with beef!
    """
    def __init__(self, spicy=False):
        super().__init__(self,spicy=spicy)

    def yourOrder(self):
        print("Your order is: Cheese Burger,", "spicy" if self.spicy else "not spicy")
```

Gambar 1.1 Contoh kode program

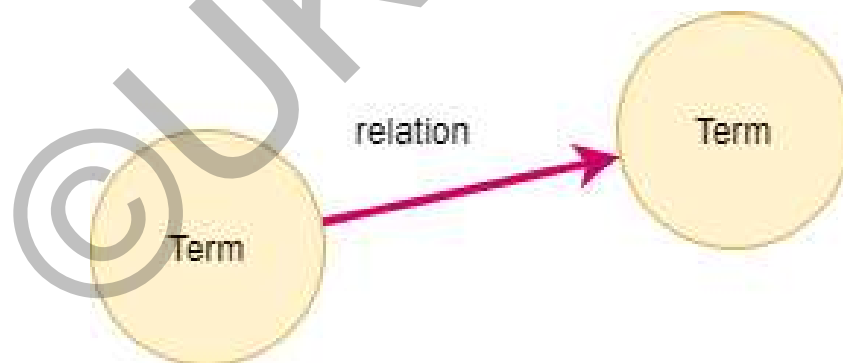
Dari contoh tersebut, diketahui ada 2 *class* dengan nama *Burger* dan *CheeseBurger*. Kedua nama *class* tersebut akan menjadi *term* dalam indeks.

2. Menemukan relasi dari *term* hasil *preprocessing*

Term yang diperoleh dari *preprocessing* akan dicari relasinya dengan *term* lain sehingga tercipta model blok dari *source code* yang digunakan. Mengambil contoh dari *file* *Burger.py* pada poin sebelumnya, diketahui *class* *Burger* dan *CheeseBurger* memiliki relasi *inheritance*. Dari sini akan ada penambahan *term* yang mengindikasikan relasi dari kedua *class* tersebut. Untuk *class* *Burger* karena memiliki relasi berupa *parent* atau *superclass* dari *class* *CheeseBurger*, maka *term* barunya adalah “*parent of CheeseBurger*”. Untuk *class* *CheeseBurger*, karena merupakan turunan dari *class* *Burger*, maka dapat ditambahkan *term* baru berupa “*inherited from Burger*”.

3. Melakukan *indexing* pada *term*

Term hasil *preprocessing* akan dibuat indeksinya menggunakan *RedisGraph*. Pada tahap ini setiap *term* dari hasil *preprocessing* akan dimasukkan ke dalam *RedisGraph* sebagai sebuah *node*. Struktur penyimpanan indeks dalam database *RedisGraph* sendiri akan dibuat seperti contoh berikut:



Gambar 1.2 Contoh index

Berdasarkan contoh di atas, setiap *term* akan menjadi sebuah *node* dan saling berelasi dengan *node* lain yang direpresentasikan dalam bentuk *edge*. Penggunaan indeks berbentuk graf sendiri dipilih karena dapat merepresentasikan relasi suatu *term* dengan *term* yang lain dengan jelas.

4. Membangun sistem temu balik

Setelah *repository* terindeks terbentuk, langkah selanjutnya adalah membuat sistem temu balik agar pengguna dapat melakukan pencarian terhadap *term* yang

ada pada *repository*. Secara sederhana, sistem temu balik untuk pencarian relasi *class* yang akan dibangun ini akan mencocokkan *query* pencarian dengan *key* pada *database*. Jika ditemukan *key* yang sesuai, maka seluruh dokumen yang berada di *set* untuk *key* tersebut akan dikembalikan sebagai hasil.

1.7 Struktur Penulisan

Struktur penulisan pada penelitian ini akan disusun dalam 5 bab. Bab 1 terdiri dari Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Metodologi Penelitian. Bab 2, terdiri dari Tinjauan Pustaka yang terkait dengan topik sistem temu balik pada pencarian source code dan Landasan Teori yang digunakan penulis sebagai dasar dalam melakukan penelitian ini. Bab 3 berisi Metodologi Penelitian yang mendetail dalam penelitian ini. Bab 4 yaitu Hasil dan Pembahasan akan membahas tentang hasil yang telah dicapai dengan adanya pembuatan sistem yang diusulkan beserta pengujian yang dilakukan terhadap sistem. Bab 5 yaitu Kesimpulan dan Saran berisi kesimpulan dari penelitian yang telah dilakukan beserta beberapa evaluasi dari penelitian ini yang dapat digunakan sebagai saran agar penelitian dalam bidang ini dapat lebih dikembangkan dengan baik.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengaplikasian pengenalan relasi *class* untuk sistem pencarian *source code* memungkinkan pengguna untuk mencari abstraksi dan pewarisan dari sebuah *class* tanpa harus satu per satu menelusuri file *source code*. Berdasarkan penelitian yang telah dilakukan, penggunaan model graf membuat struktur pengindeksan menjadi lebih jelas sehingga memudahkan dalam pencarian objek dan relasinya. Pemanfaatan Redisearch sebagai *database indexing* untuk dokumen *source code* turut membantu dalam kecepatan pencarian dikarenakan sistem pencarian tidak perlu melakukan *grepping* satu per satu pada file *source code* melainkan dengan melakukan *query* ke *database index*.

Sistem temu balik yang telah dibuat dinilai dapat mengembalikan dokumen yang dimaksud dengan cukup baik. Hal tersebut dapat dilihat pada nilai rataan harmonik F yang bernilai 0.5 hingga 1. Dokumen *source code* yang diharapkan muncul terbukti dapat diambil seluruhnya oleh sistem pencarian yang telah dibuat meskipun pada beberapa kasus ada dokumen lain yang tidak diinginkan ikut terambil. Untuk lebih mendetailkan kriteria pencarian, pengguna dapat menggunakan *query string* khusus yang disediakan oleh sistem agar mendapatkan dokumen yang dimaksud dengan tepat. Penggunaan *query string* khusus ini mampu memberikan nilai *precision* 1 untuk *test case* yang diberikan.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, penulis memiliki beberapa saran untuk kelanjutan penelitian di bidang ini. Saran juga berguna untuk mengembangkan sistem yang telah dibuat agar dapat mencakup hal-hal yang sekiranya belum terpenuhi pada penelitian ini. Berikut adalah saran yang dapat penulis tuliskan.

1. Penambahan kemampuan untuk mengenali relasi asosiasi pada *class*. Relasi asosiasi merupakan jenis relasi yang banyak diterapkan dalam pemrograman

OOP. Dengan pengenalan relasi asosiasi ini diharapkan akan memudahkan pengguna dalam menemukan *dependency* dari suatu *class*.

2. Pembobotan hasil pencarian agar dapat menghasilkan hasil pencarian yang terurut berdasarkan relevansi dengan *query* pencarian.
3. Pengembangan algoritma pencarian agar dapat dibuat lebih efisien.
4. Penambahan kemampuan pencarian dengan pengenalan bahasa natural.

©UKDW

Daftar Pustaka

- Aldous, J. M., & Wilson, R. J. (2004). *Graphs and Applications: An Introductory Approach*. Great Britain: Springer.
- Butler, S., Wermelinger, M., Yu, Y., & Sharp, H. (2011). Improving the Tokenisation of Identifier Names. *European Conference on Object-Oriented Programming* (hal. 130-154). Springer, Berlin, Heidelberg.
- Carlson, J. L. (2013). *Redis in Action*. Shelter Island: Manning Publications Co.
- Glory, V., & Dominic, S. (2015). Compressing Inverted Index Using Optimal FastPFOR. *Journal of Information Processing*, 185-191.
- Google. (2020, June 22). *Searching code*. Diambil kembali dari Cloud Source Repositories documentation: <https://cloud.google.com/source-repositories/docs/searching-code>
- JetBrains s.r.o. (2020, September 7). *Hierarchy tool window*. Diambil kembali dari JetBrains Product Documentation: PyCharm: <https://www.jetbrains.com/help/pycharm/hierarchy-tool-window.html>
- JetBrains s.r.o. (2020, July 29). *Part 1. Find by name, recent usages, and searching everywhere*. Diambil kembali dari JetBrains Product Documentation: PyCharm: <https://www.jetbrains.com/help/pycharm/part-1-finding-a-file-class-or-symbol-by-name.html>
- JetBrains s.r.o. (2020, August 19). *Searching Everywhere*. Diambil kembali dari JetBrains Product Documentation: PyCharm: <https://www.jetbrains.com/help/pycharm/searching-everywhere.htm>
- Karnalim, O. (2018). Language-Agnostic Source Code Retrieval Using Keyword & Identifier Lexical. *International Journal of Computer Systems & Software Engineering*, 29-47.
- Kaur, H., & Gupta, V. (2016). Indexing process insight and evaluation. *2016 International Conference on Inventive Computation Technologies (ICICT)* (hal. 1-5). Coimbatore: IEEE.
- Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval*. Cambridge: Cambridge University Press.
- Mukherjee, R., Chaudhuri, S., & Jermaine, C. (2020). Searching a Database of Source Codes Using Contextualized Code. *arXiv preprint arXiv:2001.03277*.
- Negi, A., Bhirud, M., Jain, D. S., & Mittal, M. (2012). Index based Information Retrieval System. *International Journal of Modern Engineering Research (IJMER)*, 945-948.