SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN RELASI CLASS

Skripsi

oleh
VALENT CHRISTIAN ADIPUTRA
71170163

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2021

SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN 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

VALENT CHRISTIAN ADIPUTRA
71170163

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	: Valent Christian Adiputra
NIM	: 71170163
Program studi	: Informatika
Fakultas	: Fakultas Teknologi Informasi
Jenis Karya	: Skripsi

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:

"SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN 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 : 6 Juli 2021

Yang menyatakan

(Valent Christian Adiputra)
NIM.71170163

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN 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, 19 Juni 2021

VALENT CHRISTIAN ADIPUTRA
71170163

HALAMAN PERSETUJUAN

Judul Skripsi	:	SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN RELASI CLASS
Nama Mahasiswa	:	VALENT CHRISTIAN ADIPUTRA
N I M	:	71170163
Matakuliah	:	Skripsi (Tugas Akhir)
Kode	:	TIW276
Semester	:	Genap
Tahun Akademik	:	2020/2021

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 8 Mei 2021

Dosen Pembimbing I	Dosen Pembimbing II
Budi Susanto Digitally signed by Budi Susanto Date: 2021.05.17 21:21:11 +07'00'	DocuSigned by: 8F4094970B77466...
Budi Susanto, SKom.,M.T.	Antonius Rachmat C., S.Kom.,M.Cs.

HALAMAN PENGESAHAN

SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN RELASI CLASS

Oleh: VALENT CHRISTIAN ADIPUTRA / 71170163

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, 21 Juni 2021
Mengesahkan,

Dewan Penguji:

1. Budi Susanto, SKom.,M.T.
2. Antonius Rachmat C., S.Kom.,M.Cs.
3. Restyandito, S.Kom.,MSIS, Ph.D
4. Matahari Bhakti Nendya, S.Kom., M.T.

Budi Susanto

Antonius RC sign

Digital Signer:Restyandito
CN:C=ID, E=dito@ti.ukdw.ac.id, O=Univ. Kristen Duta Wacana, OU=Fak.Teknologi Informasi, CN=Restyandito
Date:2021.07.01 12:01:16 +07:00

Digitally signed by Matahari Bhakti Nendya
Date: 2021.07.02 15:58:51 +07'00'

Dekan	Ketua Program Studi
(Restyandito,S.Kom.,MSIS.,Ph.D.)	(Gloria Virginia,Ph.D.)

UCAPAN TERIMAKASIH

Dalam proses penulisan dan pembuatan skripsi ini, sangat banyak pihak yang telah mendukung dengan pemberian bantuan baik secara fisik, non fisik, ilmu, maupun bantuan dalam bentuk lainnya. Pada bagian ini, penulis mengucapkan rasa terimakasih kepada:

1. Tuhan Yesus Kristus atas perlindungan dan rahmatnya yang memampukan penulis dalam segala proses perkuliahan.
2. Kepada seluruh keluarga, terkhususnya ayah dan ibu yang memberikan dukungan pewnuh dalam segala aspek, baik material mauupun moral, serta telah memberikan kesempatan bagi penulis untuk menjalani pendidikan tinggi.
3. Kepada kedua dosen pembimbing penulis, Bapak Budi Susanto, S.Kom., M.T. serta Bapak Antonius Rachmat C., S.Kom., M.Cs., yang tiada hentinya memberikan bimbingan, ilmu serta kesabaran dalam membimbing penulis hingga akhirnya dapat menyelesaikan skripsi ini.
4. Kepada seluruh dosen yang telah mengajar penulis selama proses perkuliahan.
5. Kepada Yeheskiel Reza yang telah membantu dalam proses pengerjaan skripsi melalui berbagai proses diskusi, baik dalam hal teknis maupun non teknis.
6. Kepada seluruh sahabat penulis, Arnan, Nicholas, Alvin, Michael, Dennis, Riko, dan teman-teman lain yang belum dapat penulis sebutkan satu persatu yang telah mendukung dan memotivasi penulis dalam proses pengerjaan skripsi.

Penulis hanya dapat mengucapkan terimakasih atas jasa yang telah diberikan, dan berdoa agar Tuhan membalas kebaikan yang telah diberikan dengan berkat yang berlimpah. Dengan harapan skripsi ini dapat berguna baik penulis maupun pembaca di kemudian hari.

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA

Saya yang bertanda tangan di bawah ini:

NIM : 71170163

Nama : Valent Christian Adiputra

Prodi / Fakultas : Informatika / Fakultas Teknologi Informasi

Judul Tugas Akhir : SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN RELASI CLASS

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan **Hak Bebas Royalti Non Ekslusif (Non-exclusive Royalty-free Right)** serta bersedia Tugas Akhirnya dipublikasikan secara online dan dapat diakses secara lengkap (full access).

Dengan Hak Bebas Royalti Nonekslusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk database, merawat, dan memublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 21 Juni 2021

Yang menyatakan,

(71170163 – Valent Christian Adiputra)

INTISARI

SISTEM TEMU KEMBALI UNTUK TEMPAT PENYIMPANAN KODE PROGRAM JAVA DENGAN PENDEKATAN RELASI CLASS

Penelitian ini bertujuan membangun sebuah sistem temu kembali informasi untuk kode program bahasa Java dalam sebuah penyimpanan kode program (*repository*). Jumlah dokumen kode program yang besar dalam sebuah *repository* menyebabkan informasi menjadi lebih sulit untuk ditemukan. Hal ini terjadi karena pengembangan sebuah aplikasi / *software* terus-menerus dilakukan, dan terjadi penambahan fitur seiring perkembangan waktu.

Penelitian ini akan melakukan proses pengolahan kode program bahasa java dalam repository hingga ke tahap menjadi sebuah indeks yang dapat diakses dengan melakukan pencarian. Kode program java akan melalui tahap *preprocessing* hingga terbentuk dokumen JSON berisi informasi kode program yang akan diindeks. Dokumen JSON ini kemudian akan menjadi input bagi SOLR dan neo4j dan kemudian akan diindeks. Indeks ini akan berisi informasi tentang kode program, secara spesifik juga menggambarkan relasi antar *class*. Indeks yang terbentuk akan diakses melalui proses temu kembali informasi.

Hasil penelitian ini adalah sebuah aplikasi yang mampu melakukan *preprocessing* terhadap dokumen kode program java, melakukan proses *indexing*, serta melakukan temu kembali informasi yang memberikan *output* berupa hasil pencarian dari *query* yang diberikan oleh pengguna kepada sistem.

Kata kunci – temu kembali informasi, tempat penyimpanan kode program, bahasa pemrograman Java, relasi class, struktur data, *inverted index*

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sebuah proyek pembuatan program dapat terdiri dari banyak *source code*, sehingga membuat pengembang aplikasi mengalami kesulitan dalam melacak bagian kode dalam sebuah proyek pengembangan perangkat lunak. Pada berbagai IDE (*Integrated Development Environment*) yang ada saat ini, pengembang aplikasi dapat melakukan pencarian bagian kode dengan menggunakan *keyword search*. Pencarian dengan cara ini dapat dikatakan bergantung pada kata atau token yang ditemukan dalam kumpulan kode program, karena terdapat kemungkinan terdapat *method*, *class*, atau nama variabel yang menggunakan *term* atau penamaan variabel dan *class* yang sama, sehingga hasil pencarian masih sangat luas, dan semakin tidak efektif apabila dokumen kode dalam *project* tersebut sangat banyak.

Proses pencarian berbasis *keyword* yang hanya menampilkan hasil dengan pencocokan *string* saja masih memiliki beberapa kekurangan. Hal ini disebabkan karena berbagai *class* yang saling berelasi satu sama lain belum tentu memiliki *term* atau nama yang serupa. Sebagai contoh, *class* dosen, mahasiswa dan manusia dapat memiliki relasi *parent* dan *child*, dimana manusia sebagai kelas induk dari dosen dan mahasiswa. Dengan metode pencocokan *string*, dengan memasukkan term "dosen", maka kita tidak akan menemukan *class* manusia maupun mahasiswa, yang sejatinya memiliki relasi dengan dosen. Proses pencarian seperti ini dapat memakan waktu lama, terutama bagi *developer* yang baru membaca atau mempelajari suatu kode program, sehingga belum memahami struktur kode program dalam *repository* tersebut.

Penelitian serupa pernah dilakukan (Hsu & Lin, 2011), dimana dilakukan ekstraksi informasi dair kode program seperti *filename*, deklarasi *class*, dan deklarasi *method*, yang kemudian diindeks dan dilakukan pengurutan hasil pencarian dengan menggunakan VSM. Penelitian lain yang menerapkan prinsip *Information Retrieval* (IR) dilakukan oleh Mahmoud dan Niu(Mahmoud & Niu, 2011), yang berfokus pada relevansi untuk memasukkan bagian-bagian lain dalam kode program seperti blok meta data dan *comments* kedalam indeks.

Perbedaan penelitian ini dengan penelitian serupa adalah fokus penelitian, metode dan penggunaan *tools*. Penelitian ini berfokus pada representasi relasi *class* yang diindeks dalam basis data graf dan menggunakan indeks SOLR untuk membantu proses temu kembali dari basis data graf. Proses *indexing* dilakukan dengan metode menyusun sebuah dokumen JSON yang berisi informasi hasil *parsing* dari kode program, dimana dokumen JSON ini menjadi *input* informasi untuk diindeks. *Tools* yang digunakan untuk sistem temu kembali adalah apache SOLR, serta neo4j sebagai *graph database* yang dimanfaatkan untuk menyimpan relasi antar *class*.

Pada penelitian ini, hal yang dilakukan adalah melakukan *indexing* untuk *repository* dokumen *source code* dengan bahasa pemrograman java. Tahapan yang dilakukan adalah dengan melakukan *preprocessing* pada dokumen *source code*. Hasil *preprocessing* ini kemudian dicocokkan dengan model blok kode sehingga dapat dikenali bagian mana yang berisi pendefinisian *class*, penggunaan *class* turunan, penggunaan sebuah *class* sebagai atribut *class* lain dan sebagainya. Blok kode yang telah teridentifikasi tadi selanjutnya di indeks dan disimpan dalam sebuah basis data berupa dokumen XML/JSON. Hasil indeks dari *source code* ini nantinya bermanfaat jika dipadukan dengan sistem *information retrieval* untuk mengakses hasil indeks melalui proses pencarian (*searching*).

1.2 Rumusan Masalah

1. Apa metode *preprocessing* kode program bahasa java yang tepat untuk membuat indeks dalam sistem temu kembali?
2. Apa saja data yang perlu disimpan dalam sebuah dokumen hasil *preprocessing* dengan struktur data yang merepresentasikan relasi antar *class*?
3. Apakah sistem temu kembali yang dibuat mampu menyediakan informasi yang relevan dengan pengukuran kualitas menggunakan *precision, recall* dan *accuracy*?

1.3 Batasan Masalah

1. Model blok kode, indeks, serta sistem temu kembali informasi ini digunakan untuk kode bahasa prmrograman Java.
2. Kode program mengalami *pre-processing* untuk mengekstrak *term* yang disimpan dalam basis data berupa dokumen XML/JSON yang berasal dari hasil kode program.
3. Indeks kode program disusun dari dokumen *term* berupa dokumen XML/JSON yang kemudian diindeks oleh Apache SOLR.
4. Indeks relasi antar *class* disimpan dalam bentuk graf menggunakan basis data neo4j
5. Sistem temu kembali informasi menggunakan *tool* apache SOLR dan Neo4j.
6. Sistem temu kembali informasi yang dibangun berfokus pada relasi *class* dan *interface* berupa relasi *inheritance, association,* dan *realization* pada bahasa pemrograman Java.
7. Sistem hanya menyimpan data *methods* yang dideklarasikan secara eksplisit dalam sebuah *class/interface*.

1.4 Tujuan Penelitian

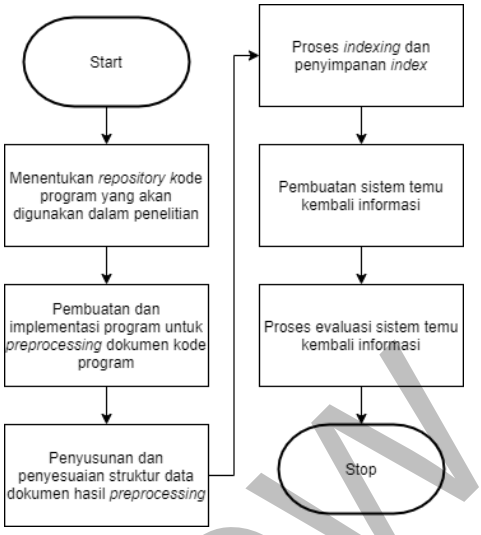
Penenilitan ini bertujuan untuk membangun sebuah sistem yang mampu melakukan *preprocessing* (tokenisasi) terhadap kode program bahasa Java. *Preprocessing* dari kode program Java ini menghasilkan *term* yang disimpan dalam dokumen XML/JSON. Hasil *preprocessing* kode program kemudian diindeks oleh SOLR dan disimpan sebagai indeks berupa *Inverted Index*. Hasil indeks ini kemudian dapat diakses dengan sistem temu kembali informasi berbasis Apache SOLR. Sistem temu kembali informasi dapat menemukan lokasi *term* pada kode program, serta lokasi *term* lain yang memiliki relasi *class* dengan *term* yang dicari.

Penelitian ini juga bertujuan menemukan struktur data indeks yang optimal yang mampu merepresentasikan *class* dan relasi diantaranya. Pengujian keberhasilan penelitian ini dilakukan dengan menyusun skenario *query* serta menentukan terlebih dahulu informasi yang diharapkan dapat didapat dari proses temu kembali. Metode pengukuran yang dilakukan dengan skenario tersebut menggunakan perhitungan *precision* dan *recall*.

1.5 Manfaat Penelitian

1. Menyusun sistem temu kembali kode program Java untuk mempermudah pelacakan dan pemeliharaan kode dalam sistem temu kembali informasi.
2. Memberikan hasil pencarian yang lebih informatif dan menggambarkan struktur kode program, dimana *class* lain dalam indeks yang memiliki relasi dengan *class* sesuai *keyword* yang dicari akan muncul dalam hasil pencarian.

1.6 Metodologi Penelitian



Gambar 1. 1 Alur Proses Penelitian

Gambar 1.1 merupakan gambaran alur dalam penelitian yang akan dilakukan.

1. Langkah pertama yang perlu dilakukan adalah menentukan *repository* kode program yang digunakan dalam penelitian.
2. *Preprocessing* kode program dilakukan terhadap setiap kode program yang terdapat dalam *repository.* Preprocessing yang dilakukan terhadap kode program java menggunakan *tools* javaparser. Javaparser menghasilkan *Abstract Syntax Tree* (AST) dari kode program, yang kemudian dapat diambil bagian-nagian tertentu dari dalamnya dengan menggunakan *method visitor* (Smith et al., 2017).
3. Proses selanjutnya adalah identifikasi blok kode program yang berhubungan dengan *class*, seperti deklarasi *class,* penggunaan *class,* dan turunan *class.* Dari tahap sebelumnya dihasilkan sebuah AST yang berisi informasi tentang kode program. Pada tahap ini, beberapa data hasil parsing kode program tersebut disusun dalam sebuah dokumen JSON dengan bentuk pasangan *key value*. Sebagai contoh, pasangan *key*

value dapat berupa NamaClass: Mobil. Dokumen inilah yang menjadi *input* untuk diindeks oleh SOLR dan neo4j.

4. Dokumen JSON yang dihasilkan pada tahap sebelumnya kemudian disimpan dan diindeks oleh sistem SOLR dan neo4j. Basis data indeks kode program yang telah dibuat kemudian digunakan dalam sistem temu kembali informasi.
5. Pengguna kemudian memberikan *query* pada sistem temu kembali informasi, dan sistem mengembalikan hasil pencarian. Dalam tahap ini dilakukan juga pemrosesan *query,* serta proses penyusunan penampilan hasil pencarian.
6. Proses pengujian dilakukan dengan menyusun skenario berupa *query* tertentu serta hasil-hasil yang diharapkan diberikan oleh sistem. Perhitungan hassil pengujian dilakukan dengan menghitung *precision* dan *recall* dari skenario yang telah disusun.

1.7 Sistematika Penulisan

Laporan ini terdiri dari 5 bab. Bab-bab tersebut yaitu Pendahuluan, Tinjauan Pustaka dan Dasar Teori, Metodologi Penelitian, Hasil dan Pembahasan, serta Kesimpulan dan Saran. Setiap bab berisi informasi mengenai penelitian yang dilakukan.

Pedoman dan penelitian dicantumkan pada bab I hingga bab III, yaitu Pendahuluan, Tinjauan Pustaka dan Dasar Teori, dan Metodologi Penelitian. Bab I (Pendahuluan) berisi latar belakang masalah serta bagaimana masalah tersebut diatasi melalui penelitian yang dilakukan. Bab II (Tinjauan Pustaka dan Dasar Teori) berisi referensi dari berbagai sumber yang digunakan peneliti dalam melakukan penelitiannya. Bab III (Metodologi Penelitian) berisi penjelasan mendetail dari metode yang digunakan oleh peneliti dalam penelitian tersebut.

Hasil yang dicapai dan kesimpulan dari penelitian dituliskan dalam bab IV dan bab V, yaitu Hasil dan Pembahasan, serta Kesimpulan dan Saran. Bab IV (Hasil dan Pembahasan) berisi hasil dari penelitian yang telah dilakukan, yang kemudian dianalisis dan dibahas lebih lanjut oleh peneliti. Bab V (Kesimpulan dan Saran) pernyataan penulis mengenai hasil penelitian dan pembahasannya, serta saran dan perbaikan untuk penelitian agar memperoleh hasil yang lebih baik.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Sesudah melakukan penelitian, dengan melalui tahapan implementasi sistem dan pengujian sistem temu kembali, penulis dapat menyimpulkan beberapa hal terkait dengan penelitian ini. Penelitian ini membuktikan bahwa hasil *preprocessing* dengan menggunakan javaparser yang merepresentasikan relasi antar *class/interface* dalam sebuah basis data graf mampu membantu proses temu kembali informasi untuk menemukan *class/interface* lain yang saling berelasi satu sama lain. Metode *preprocessing* kode program java dengan menggunakan javaparser dan kemudian disusun menjadi dokumen JSON dengan data hasil *preprocessing* tersebut dapat digunakan secara efektif untuk menyusun indeks pada SOLR dan Neo4j.

Terdapat data yang perlu disimpan dari sebuah dokumen kode program java. Dalam penelitian ini pencarian dilakukan dengan menggunakan parameter nama *class/interface*, nama *method*, dan *method return type*. Karena itu, maka ketiga data tersebut, serta lokasi penyimpanan dokumen kode program perlu disimpan dalam dokumen JSON, yang kemudian diindeks dan dapat diakses melalui sistem temu kembali dari proses pencarian oleh pengguna.

Hasil evaluasi dari sistem temu kembali pada penelitian ini menunjukkan hasil nilai *precision* yang baik, yakni 100% terkecuali pada kasus dimana tidak ditemukan hasil pencarian (nilai *precision 0*). Dengan demikian dapat dikatakan bahwa informasi yang menjadi *output* dari sistem temu kembali ini merupakan informasi yang relevan bagi pengguna. Sementara itu, hasil perhitungan *recall* cukup bervariasi, dimana terkadang sistem memiliki nilai *precision* yang sempurna, namun tidak melakukan *retrieval* terhadap seluruh dokumen yang dikategorikan sebagai dokumen yang relevan (*recall* < 100%) untuk menghindari dilakukannya *query* berulang-ulang. Dalam pengujian terdapat pula skenario tertentu dimana

performa sistem dirasa kurang baik, atau bahkan tidak dapat sama sekali menemukan dokumen yang relevan sehingga menghasilkan nilai 0 pada *precision* dan *recall* sehingga menghasilkan 0% *accuracy*.

5.2. Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat bermanfaat bagi keberlanjutan penelitian dan bagi peneliti selanjutnya. Adapun hal yang dapat diperbaiki dari penelitian ini adalah sebagai berikut:

1. pengayaan dari relasi antar *class/interface* yang direpresentasikan dalam basis data graf, seperti relasi asosiasi antar *class*. Hal ini dapat bermanfaat karena dapat memperluas kemampuan pencarian.
2. Pembobotan terhadap hasil pencarian yang dirumuskan dengan kriteria tertentu untuk dapat memunculkan hasil pencarian yang terurut dari proses temu kembali informasi.
3. Pengayaan jenis *query* dan pengolahan *keyword* yang lebih handal dan mampu menghasilkan output pencarian yang lebih baik dan mampu mengatasi kasus tertentu yang menimbulkan kegagalan dalam penelitian ini.

DAFTAR PUSTAKA

Hsu, S.-K., & Lin, S.-J. (2011). *A Block-Structured Model for Source Code Retrieval* (pp. 161–170). https://doi.org/10.1007/978-3-642-20042-7_17

Kendal, S. (2009). *Object oriented programming using Java*. Bookboon.

Kim, K., Kim, D., Bissyandé, T. F., Choi, E., Li, L., Klein, J., & Traon, Y. Le. (2018). F a C o Y. *Proceedings of the 40th International Conference on Software Engineering*, 946–957. https://doi.org/10.1145/3180155.3180187

Kumar, J. (2015). *Apache Solr search patterns*. Packt Publishing Ltd.

Liu, Z. H., Hammerschmidt, B., & McMahon, D. (2014). JSON data management. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 1247–1258. https://doi.org/10.1145/2588555.2595628

Mahmoud, A., & Niu, N. (2011). Source code indexing for automated tracing. *Proceeding of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering - TEFSE '11*, 3. https://doi.org/10.1145/1987856.1987859

Manning, C. D., Raghavan, P., & Schutze, H. (2012). *Introduction to Information Retrieval*. Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

MU, L., ZHANG, Y.-Y., & HAN, Y. (2020). Research and Realization on Big Data of Science and Technology Resources Search Engine Based on SOLR. *DEStech Transactions on Social Science, Education and Human Science*, *icesd*. https://doi.org/10.12783/dtssehs/icesd2020/34507

Needham, M., & Hodler, A. E. (2019). *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media.

Pauzi, Z., & Capiluppi, A. (2020). Text Similarity Between Concepts Extracted from Source Code and Documentation. *International Conference on Intelligent Data Engineering and Automated Learning*, 124–135.

Rafał Kuć. (2013). *Apache Solr 4 Cookbook* (Second Edi). Packt Publishing.

Roshdi, A., & Roohparvar, A. (2015). Review: Information Retrieval Techniques and Applications. *International Journal of Computer Networks and Communications Security*, *3*(9), 373–377.

Smith, N., van Bruggen, D., & Tomassetti, F. (2017). *JavaParser: visited*. Leanpub, oct. de.