TESIS

ANALISIS TINGKAT KESAMAAN SOURCE CODE PADA BAHASA PEMROGRAMAN KOTLIN MENGGUNAKAN ALGORITMA WINNOWING

Disusun oleh:

Nama : **Yustikamasy Astica**
NIM : **21.51.2107**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023

TESIS

ANALISIS TINGKAT KESAMAAN SOURCE CODE PADA BAHASA PEMROGRAMAN KOTLIN MENGGUNAKAN ALGORITMA WINNOWING

ANALYSIS OF THE SIMILARITY LEVEL OF SOURCE CODE IN THE KOTLIN PROGRAMMING LANGUAGE USING WINNOWING ALGORITHM

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Yustikamasy Astica**
NIM	**: 21.51.2107**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

HALAMAN PENGESAHAN

ANALISIS TINGKAT KESAMAAN SOURCE CODE PADA BAHASA PEMROGRAMAN KOTLIN MENGGUNAKAN ALGORITMA WINNOWING

ANALYSIS OF THE SIMILARITY LEVEL OF SOURCE CODE IN THE KOTLIN PROGRAMMING LANGUAGE USING WINNOWING ALGORITHM

Dipersiapkan dan Disusun oleh

Yustikamasy Astica

21.51.2107

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 02 Oktober 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 Oktober 2023
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS TINGKAT KESAMAAN SOURCE CODE PADA BAHASA PEMROGRAMAN KOTLIN MENGGUNAKAN ALGORITMA WINNOWING

ANALYSIS OF THE SIMILARITY LEVEL OF SOURCE CODE IN THE KOTLIN PROGRAMMING LANGUAGE USING WINNOWING ALGORITHM

Dipersiapkan dan Disusun oleh

Yustikamasy Astica

21.51.2107

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 02 Oktober 2023

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Anggota Tim Penguji

Alva Hendi Muhammad, S.T., M.Eng., Ph.D.
NIK. 190302493

Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024

Pembimbing Pendamping

Anggit Dwi Hartanto, M.Kom
NIK. 190302163

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 Oktober 2023
Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Yustikamasy Astica
NIM : 21.51.2107
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut
Analisis Tingkat Kemiripan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom.
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 02 Oktober 2023
Yang Menyatakan,

Yustikamasy Astica

HALAMAN PERSEMBAHAN

Alhamdulillah, Alhamdulillahirobbil Alamin, puji syukur saya panjatkan kehadirat Allah SWT yang telah memberikan nikmat dan berkat yang luar biasa kepada saya, sehingga saya bisa menyelesaikan tesis ini dengan baik. Saya juga berterimakasih kepada orang-orang yang secara langsung maupun tidak langsung telah membantu saya dalam menyelesaikan tesis ini. Tesis ini saya persembahkan kepada:

1. Allah SWT, atas segala nikmat berupa kesehatan, kekuatan, dan masih banyak lagi karena hanya atas izin dan karunia-Nya maka tesis ini dapat dibuat dengan baik dan dapat diselesaikan dengan lancar.
2. Mama dan Papa saya tercinta yang tanpa lelah mendukung semua keputusan dan pilihan hidup saya apapun itu serta selalu mendoakan yang terbaik untuk kesuksesan anaknya.
3. Kakak-kakak saya, Mbak Adin, Mas Nandi, Mas Sutan, Mas Ghani, dan Mbak Cindy yang selalu mendukung saya, memberikan semangat untuk semua keputusan saya, dan memberikan masukan ketika saya sedang kesusahan.
4. Sahabat terbaikku Monica, Ogen, Mas Rafi, Alvinia, Anggita yang selalu ada ketika saya memulai perkuliahan S2 ini. Serta Bayu, Mitha, dan Salsa sebagai teman seperjuangan studi s2.
5. Teman-teman tim product Farmacare, Fri, Komang, Alex, Derby, Riska, Yopi, Jagi, dan teman-teman Farmacare lainnya yang membantu untuk saya bisa kuliah sambil meniti karir di Farmacare. Serta Kak Icha dan Endi yang telah menemani saya dan membantu saya dengan sabar.
6. Ucapan terima kasih kepada semua pihak yang membantu serta mendukung saya yang tidak bisa saya sebutkan satu persatu.
7. *Last but not least, I wanna thank me. I wanna thank me for believing in me. I wanna thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting.*

HALAMAN MOTTO

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا (٥) إِنَّ مَعَ الْعُسْرِ يُسْرًا (٦)

"Karena sesungguhnya beserta kesulitan ada kemudahan, sesungguhnya beserta kesulitan itu ada kemudahan." (QS. Asy-Syarh [94]: 5-6)

Tidak apa-apa jika merasa belum mengerti semuanya, karena semua orang mungkin juga merasakan hal yang sama.

-Yustikamasy Astica-

KATA PENGANTAR

Assalamualaikum Wr.Wb.

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat, taufik, hidayah, serta inayah-Nya, sehingga penulis dapat menyelesaikan tesis ini dengan lancar. Tidak lupa sholawat serta salam penulis haturkan kepada Nabi Muhammad SAW.

Tesis ini disusun dalam rangka memenuhi salah satu syarat kelulusan jenjang Program Magister Teknik Informatika Universitas Amikom Yogyakarta. Proses penyusunan hingga selesainya laporan tesis ini tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak, baik sevcara langsung maupun tidak langsung. Maka dari itu, sebagai rasa hormat penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M, selaku Rektor Universitas AMIKOM Yogyakarta.
2. Ibu Prof. Dr. Kusrini, M.Kom, selaku Direktur Pascasarjana Universitas AMIKOM Yogyakarta
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom, selaku dosen pembimbing 1 penulis yang telah memberikan petunjuk, bimbingan, dan nasihatnya dalam proses penulisan tesis ini.
4. Bapak Anggit Dwi Hartanto, M.Kom, selaku dosen pembimbing 2 penulis yang telah memberikan bimbingan dalam penulisan tesis ini.
5. Mama, Papa, dan Kakak-Kakak saya tersayang yang telah memberikan doa, kasih sayang, dan motivasi kepada penulis. Bapak dan Ibu Dosen dan staff Universitas Amikom Yogyakarta yang telah banyak memberikan ilmunya selama penulis berkuliah.
6. Teman-teman seperjuangan PJJ Angkatan 6 dan MTI-A Angkatan 27-A, tim Farmacare, dan teman TKJ 2, atas segala bantuan, doa, dan dukungan semangatnya.
7. Semua pihak yang telah membantu dalam penyelesaian tesis ini yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap tesis ini dapat menjadi manfaat bagi dunia pendidikan. Kritik dan saran akan sangat membantu perkembangan dan penyempurnaan dalam karya tulis ini. Sekian dari penulis, apabila terdapat kesalahan dan kekurangan mohon maaf sebesar-besarnya.

Wassalamualaikum Wr. Wb.

Yogyakarta, 15 Oktober 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

INTISARI

Plagiarisme merupakan tindakan yang meniru hasil pekerjaan orang lain secara langsung maupun tidak langsung. Di lingkungan akademis, plagiarisme tidak hanya berlaku pada dokumen tekstual tetapi juga dokumen source code. Plagiarisme source code di dunia akademis biasanya terjadi ketika mahasiswa meniru kode milik mahasiswa lain dan melakukan submission seolah-olah seperti pekerjaan milik mahasiswa tersebut. Sehingga diperlukan pemeriksaan plagiarisme secara otomatis, algoritma winnowing akan digunakan untuk membantu mendeteksi kemiripan pada source code yang merupakan sebagai suatu cara untuk mendeteksi adanya suatu tindak plagiariasme.

Algoritma Winnowing yang biasanya digunakan untuk mendeteksi plagiarisme dokumen, penelitian ini mendeteksi pada source code. Hasil yang dihasilkan pada penelitian ini adalah tingkat drajat kesamaan pada dua source code akan menghasilkan nilai similarity yang berbeda-beda jika dataset yang sudah digunakan sudah melalui tahap text preprocessing atau tanpa preprocessing. Jika dataset yang digunakan sudah melalui tahap text preprocessing, nilai similaritynya akan cukup rendah dikarenakan jumlah karakter yang digunakan berkurang cukup banyak. Algoritma Winnowing dan Jaccard Similarity bekerja cukup baik untuk mendeteksi plagiarisme pada source code.

Kata kunci: Plagiarism, Winnowing algorithm, Jaccard similarity

ABSTRACT

Plagiarism is an act of imitating the work of others directly or indirectly. In academia, plagiarism occurs not only in textual documents but also in source code documents. Source code plagiarism in the academic world usually occurs when students copy another student's code and submit it as if it were the student's work. So that an automatic plagiarism check is needed, a screening algorithm will be used to help detect similarities in the source code, which is a way to see a plagiarism violation.

The Winnowing algorithm, which is usually used to detect plagiarism documents, in this study detects source code. The results produced in this study are the degree of similarity in the two source codes will make different similarity values if the dataset used has gone through the text preprocessing stage or without preprocessing. If the dataset has gone through the text preprocessing step, the similarity value will be low because the number of characters used is significantly reduced. The Winnowing Algorithm and Jaccard Similarity work well enough to detect plagiarism in source code.

Keyword: Plagiarism, Winnowing algorithm, Jaccard similarity

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Plagiarisme merupakan tindakan yang meniru hasil pekerjaan orang lain secara langsung maupun tidak langsung. Di lingkungan akademis, plagiarisme yang biasanya sering terjadi adalah pada dokumen tekstual seperti essay, laporan, dan bahkan penelitian. Akan tetapi, plagiarisme tidak hanya berlaku pada dokumen tekstual tetapi juga dokumen source code. Plagiarisme source code di dunia akademis biasanya terjadi ketika mahasiswa meniru kode milik mahasiswa lain dan melakukan submission seolah-olah seperti pekerjaan milik mahasiswa tersebut (Cosma & Joy, 2008).

Menurut Kamus Besar Bahasa Indonesia (KBBI), plagiarisme atau yang lebih dikenal dengan plagiat adalah pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat) sendiri. Di dalam Peraturan Menteri Pendidikan Nasional Republik Indonesia No. 17 Tahun 2010, pada pasal 1 mendefinisikan bahwa plagiat adalah perbuatan secara sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai. Istilah plagiasi sudah ada dalam dunia perkodingan namun plagiarisme pada source code adalah tugas yang mudah dilakukan, namun tidak mudah dideteksi. Biasanya, ketika siswa menyelesaikan masalah yang sama dengan menggunakan bahasa pemrograman yang sama, kemungkinan besar

penyelesaian tugasnya akan kurang lebih sama. Ada beberapa strategi modifikasi kode sumber yang dapat digunakan untuk menutupi plagiarisme kode sumber (Hage et al., 2011)

Didalam ranah ilmu komputer atau informatika, source code merupakan suatu hal yang penting untuk para programmer, karena source code merupakan bagian utama untuk membangun sebuah sistem atau aplikasi. Dalam pembuatan aplikasi, programmer memiliki banyak cara untuk membedakannya dalam penulisan kode programnya, seperti bahasa yang digunakan, library, penggunaan function, dan lain-lain. Permasalahan yang biasanya timbul di lingkungan akademis yang berkaitan dengan ilmu komputer adalah terjadinya plagiarisme di tugas-tugas pemrograman. Apalagi akhir-akhir ini banyak sekolah atau universitas yang menggunakan konsep belajar secara online, maka kesempatan untuk berbuat plagiat semakin luas karena tidak adanya pengawasan secara langsung dari pengajar. Dengan demikian, otomasi untuk mendeteksi plagiarisme pada source code yang memiliki kualitas tinggi sangat dibutuhkan untuk memenuhi kebutuhan karena jumlah mahasiswa yang terus berkembang.

Terdapat banyak penelitian yang dilakukan untuk mengatasi plagiarisme pada source code. Namun dari setiap penelitian untuk pendeteksian plagiarisme memiliki algoritma dan nilai hasil akurasi yang berbeda-beda tergantung bagaimana peneliti membentuk model untuk menyelesaikan permasalahan yang diangkat.

Berdasarkan (Cheers et al., 2021) menyatakan bahwa studi telah menunjukkan antara 50% hingga 79% mahasiswa sarjana akan menjiplak setidaknya satu kali selama karir akademis mereka. Dengan rate setinggi itu, sangat mungkin seorang

akademisi harus menilai kasus dugaan plagiarisme. Dalam mata kuliah ilmu komputer, plagiarisme sering dijumpai sebagai plagiarisme source code. Plagiarisme kode sumber terjadi ketika seorang siswa mengambil kode sumber orang lain dan mulai mengirimkannya sebagai karya mereka sendiri.

Beberapa indikator yang menjadi perhatian setiap peneliti dalam meningkatkan performa dari pendeteksian plagiarisme, menurut (Leman et al., 2019) penggunaan stemming berpengaruh terhadap akurasi nilai similaritas yang dihasilkan dengan menggunakan stemming menghasilkan nilai yang cenderung kurang baik dibandingkan tanpa menggunakan stemming. Sehingga dapat dilakukan penelitian untuk menguji apakah dengan pendeteksian menggunakan preprocessing dan tanpa preprocessing.

Penelitian oleh (Puspaningrum et al., 2020) menjelaskan bahwa algoritma winnowing mampu mendeteksi plagiarisme dalam waktu yang cukup cepat. Secara tekstual, algoritma ini sangat efektif dalam menangani copy paste dan plagiarisme relokasi. Pemilihan nilai k-gram akan mempengaruhi nilai similaritas. Nilai k-gram yang kecil akan memberikan nilai kemiripan yang lebih besar, semakin kecil substring, semakin kecil karakter akan mempengaruhi nilai hash dan dapat memberikan nilai sidik jari.

Nilai paramater k dan w sangat berpengaruh terhadap nilai similarity yang dihasilkan. Penelitian yang dilakukan oleh (Ramli et al., 2021) dengan membandingkan 2 tugas secara acak dan mendapatkan nilai similarity yang beragam. Semakin besar nilai k dan w maka semakin kecil nilai similarity yang dihasilkan dan semakin kecil nilai k dan w maka semakin besar juga nilai similarity

yang dihasilkan. Sehingga digunakan nilai k = 2 untuk meningkatkan peluang dalam menemukan kesamaan antara kedua sampel yang dibandingkan mengingat algoritma winnowing ini bekerja dengan cara melakukan pengecekan di setiap code yang diuji.

Penelitian oleh (Widaningrum et al., 2020) mendapatkan hasil bahwa algoritma Winnowing dapat mendeteksi plagiarisme tingkat tinggi pada dokumen, algoritma Rabin Karp pada tingkat menengah, dan algoritma Knuth Morris Pratt dapat mendeteksi plagiarisme pada tingkat rendah pada dokumen yang memiliki subjek yang sama, dengan susunan gram = 5, window = 7 dan base = 3. Hasil analisis menegaskan bahwa algoritma optimal dari ketiganya adalah algoritma Winnowing, yang menghasilkan kemiripan sebesar 94,3%. Ini juga menunjukkan akurasi tertinggi, dengan perbedaan hanya 1,19% dari pemeriksaan oleh ahli manusia, sebesar 93,11%

Menurut (Roy & Cordy, 2007) dan (Burrows et al., 2004) terdapat tiga kategori utama dalam pendekatan deteksi plagiarisme: berbasis teks, berbasis kode berorientasi atribut, dan berbasis kode berorientasi struktur. Ada dua kelompok metode untuk deteksi plagiarisme berbasis teks: ranking dan fingerprinting. Pemeringkatan didasarkan pada konsep pengambilan informasi, di mana item koleksi yang diindeks diberi peringkat berdasarkan perkiraan kemiripan dengan kueri. Dalam fingerprinting, dokumen diringkas sebagai kumpulan bilangan bulat (sidik jari) kompak yang mewakili aspek-aspek kunci dari sebuah dokumen. Keuntungan dari pendekatan text-based adalah bahwa perhitungan hash dapat

disesuaikan untuk dapat mendeteksi transformasi yang umum digunakan sehingga bisa digunakan untuk kumpulan data yang besar (Tao et al., 2013).

Penelitian (Snsebatik & Samarinda, 2017) mendapatkan hasil bahwa Algoritma winnowing lebih baik digunakan untuk mendeteksi plagiarisme pada dokumen teks dengan mengurangi nilai n-gram dan jauh lebih cepat prosesnya karena winnowing menggunakan rumus rolling hash sehingga tidak perludilakukan perhitungan lagi dari iterasi pertama. Selanjutnya penelitian yang dilakukan oleh (Satia Budhi et al., 2017) menggunakan metode N-gram serta metode jaccard similarity coefficient dan cosine similarity coefficient sebagai metode yang digunakan untuk menghitung similarity untuk source code dalam bahasa pemrograman C++ dan mendapatkan hasil bahwa aplikasi hanya bisa menampilkan bahwa source code tersebut plagiat atau tidak plagiat.

Berdasarkan beberapa penelitian di atas, maka pada penelitian ini akan melakukan eksperimen dalam melakukan mendeteksi plagiarisme. Pada penelitian ini berfokus pada deteksi plagiarisme berbasis teks pada metode fingerprinting sehingga mengabaikan sintaks pengkodean, dan cukup membandingkan kata-kata bahasa Inggris dari kode sumber yang disediakan. Penelitian ini menggunakan Algoritma Winnowing dan melakukan pengujian menggunakan jaccard similarity. Dalam melakukan penelitian ini, algoritma winnowing akan digunakan untuk membantu mendeteksi kemiripan pada source code yang akan digunakan sebagai suatu cara untuk mendeteksi adanya suatu tindak plagiariasme dan jaccard similarity diperlukan sebagai document fingerprint. Penggunaan algoritma Winnowing digunakan untuk menguji bahwa algoritma winnowing tidak hanya bisa

digunakan untuk dokumen teks biasa namun juga bisa digunakan untuk pendeteksian plagiarisme pada source code dan bisa menghitung nilai similaritynya. Jaccard similarity dipilih sebagai algoritma yang digunakan untuk menghitung nilai *similarity* karena jaccard similarity bekerja dengan membandingkan himpunan karena secara langsung mengukur tumpang tindih antara dua himpunan, yang selaras dengan konsep perpotongan kata. Hal ini sesuai dari pernyataan (Rinartha, 2017) jaccard similarity biasanya digunakan untuk membandingkan dokumen dan menghitung nilai kemiripan (similarity) dari dua buah objek atau dokumen.

Pada penelitian ini nantinya akan menerapkan algoritma winnowing dengan membandingkan hasil apabila menggunakan preprocessing grammar Kotlin dan tidak menggunakan preprocessing sama sekali. Dengan demikian, dapat menghasilkan model yang lebih dapat baik terhadap berbagai macam serangan plagiarist.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang ada, maka rumusan masalah pada penelitian ini adalah:

a. Bagaimana kinerja algoritma winnowing dalam melakukan pendeteksian plagiarisme pada *source code*?
b. Berapa nilai *similarity* penggunaan algoritma winnowing terhadap deteksi plagiarisme dengan preprocessing dan tanpa preprocessing?

c. Apakah proses *text preprocessing* mempengaruhi hasil nilai *similarity* dan performa penggunaaan algoritma winnowing dalam mendeteksi tingkat kesamaan *source code*?

1.3. Batasan Masalah

Bagian ini memuat penjelasan tentang:

a. Algoritma yang digunakan untuk pengujian source code menggunakan algoritma Winnowing,
b. Dataset source code yang digunakan menggunakan bahasa pemrograman kotlin dengan tema 'Aplikasi Github User'
c. Data yang diuji memiliki ekstensi *.kt*
d. Nilai k-gram dan window yang digunakan dibatasi antara 2 sampai 10.
e. Nilai basis hash yang digunakan dibatasi antara 2 sampai 30.
f. Nilai k-gram, Basis hash dan window saat pengujian ditentukan oleh user
g. Pada proses Rolling Hash, menggunakan nilai ASCII dari karakter
h. Maksimal karater dalam source code sebanyak 10.000 karakter.

1.4. Tujuan Penelitian

Tujuan penelitian yang diangkat berdasarkan rumusan masalah pada penelitian ini adalah sebagai berikut:

a. Mengetahui apakah algoritma winnowing dapat mendeteksi plagiarisme pada source code
b. Mengetahui seberapa besar hasil similarity perbandingan source code menggunakan algoritma winnowing

1.5. Manfaat Penelitian

Manfaat pada penelitian ini diantaranya sebagai berikut:

a. Dapat menjadi peluang pengembangan model pada pendeteksian plagiarisme pada source code.
b. Mendeteksi plagiarisme source code secara otomatis menggunakan algoritma Winnowing.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian terdahulu yang pernah dilakukan, relevan dan dijadikan studi literatur adalah sebagai berikut:

Penelitian yang dilakukan oleh (Awale et al., 2020) mengusulkan untuk melakukan pendekatan machine learning untuk mendeteksi plagiarisme pada tugas pemrograman. Penelitian ini melakukan perhitungan berdasarkan perbedaan fitur seperti, similarity score, jumlah variabel dan fungsi yang tidak digunakan, dll yang berasal dari source code. Algoritma yang digunakan adalah XGBoost learning algorithm dan membandingkan hasilnya dengan Support Vector Machine (SVM). Untuk dataset yang digunakan adalah menggunakan tugas pemrograman yang diserahkan oleh siswa selama dua kursus pemrograman pengantar di Universitas Sarajeva yang ditemukan online dengan bahasa pemrograma C/C++ yang setelah itu disortir untuk memisahkan pasangan yang melakukan plagiarisme dan tidak melakukan plagiarisme. Dengan model yang diciptakan mendapatkan hasil skor akurasi 94% dan skor rata-rata f1 0,905 pada set pengujian, dan jika dibandingkan dengan SVM, model menggunakan XGBoost lebih baik pada dataset yang digunakan.

Penelitian oleh (Ramli et al., 2021) menggunakan algoritma winnowing untuk mendeteksi plagiarism pada source code pemrograman dengan menggunakan 10 tugas mahasiswa menggunakan algoritma winnowing dan mendapatkan nilai similarity yang beragam dengan membandingkan 2 tugas secara acak. Dan

mendapatkan kesimpulan bahwa nilai parameter k dan w sangat berpengaruh terhadap nilai similarity yang dihasilkan yaitu semakin besar nilai k dan w maka semakin kecil nilai similarity yang dihasilkan dan semakin kecil nilai k dan w maka semakin besar juga nilai similarity yang dihasilkan. Sehingga digunakan nilai k = 2 untuk meningkatkan peluang dalam menemukan kesamaan antara kedua sampel yang dibandingkan mengingat algoritma winnowing ini bekerja dengan cara melakukan pengecekan di setiap code yang diuji..

Penelitian deteksi plagiarisme yang dilakukan oleh (Widjaja & Gunawan, 2020) menggunakan algoritma Mamber dan Winnowing pada teks dokumen abstrak. Dataset yang digunakan adalah dokumen abstrak dari siswa yang telah diunggah ke internet. Dalam implementasinya penelitian ini menggunakan 2 pendekatan, yaitu pendekatan Biword dan Triword ditanamkan pada algoritma winnowing, sedangkan Biword untuk algoritma Manber. Hasil dari melakukan pengujian dengan 10 dokumen, rata-rata similarity algoritma mamber adalah 90,56% sedangkan algoritma winnowing 94%. Dan untuk algoritma winnowing dengan pendekatan triword menghasilkan akurasi 91, 22%. Sehinggakan pendekatan biword dan algoritma winnowing lebih baik daripada algoritma manber dan algoritma winnowing dengan pendekatan triword.

Penelitian lain yang dilakukan oleh (Widjaja & Gunawan, 2020) berlatar belakang karena proses pengecekan plagiarisme source code secara manual merupakan tugas yang repetitif, sulit, dan memerlukan waktu yang lama sehingga diperlukan otomasi untuk deteksi plagiarisme source code yang memiliki kualitas tinggi sangat dibutuhkan. Dataset yang digunakan pada penelitian ini adalah

menggunakan bahasa pemrograman java yang dikumpulkan dari kelas dasar pemrograman Universitas Kristen Petra, Pada penelitian ini menggunakan 3 algoritma utama, yaitu levenshtein distance, greedy string tiling, dan bigram yang akan menghasilkan 12 features dan kumpulan feature statistik. Laly pada langkah akhir, features akan digunakan untuk memproses training maupun inference dengan model XGBoost. Hasil pengujian menunjukkan bahwa menggunakan features yang diajukan beserta preprocessing memiliki performa metrik yang lebih baik dari penelitian sebelumnya, yaitu f1-score sebesar 99%. Penerapan preprocessing juga dapat meningkatkan performa metrik pada features yang diajukan di di penelitian sebelumnya.

Winnowing merupakan algoritma berbasis hashing-approach yang menerapkan hash-function dan pembentukan window untuk memperoleh fingerprints pada saat pencocokan pola, algoritma tersebut digunakan oleh (Ramdhani et al., 2022). Pada penelitian sebelumnya hanya menghitung tingkat kesamaan kata berdasarkan karakter (character-level), sedangkan perhitungan tingkat kesamaan berdasarkan kata (word-level) masih jarang dilakukan, sehingga pada penelitian ini bertujuan untuk mengukur tingkat kesamaan kata menggunakan algoritma Winnowing dan word-level trigrams. Hasil penelitian menunjukkan bahwa algoritma Winnowing yang diterapkan dapat mendeteksi kesamaan pada teks sebesar 76,84%, 52,29%, 37,40%, dan 19,29% dan dapat disimpulkan bahwa metode pencocokan pola dengan algoritma Winnowing dan word-level trigrams dapat digunakan untuk mengukur tingkat kesamaan teks.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing

No	Judul	Peneliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
1	Plagiarism Detection in Programming Assignments using Machine Learning	Awale, N., Pandey, M., Dulal, A., & Timsina, B, Journal of Artificial Intelligence and Capsule Networks 2020	Melakukan pendekatan machine learning untuk mendeteksi plagiarisme pada tugas pemrograman.	Dengan model yang diciptakan mendapatkan hasil skor akurasi 94% dan skor rata-rata f1 0,905 pada set pengujian, dan jika dibandingkan dengan SVM, model menggunakan XGBoost lebih baik pada dataset yang digunakan.	Meningkatkan algoritma kami dengan menggabungkan fitur berbasis kompiler (khusus untuk bahasa pemrograman) dalam model klasifikasi yang sudah dibuat.	Dataset yang digunakan adalah menggunakan tugas pemrograman yang diserahkan oleh siswa selama dua kursus pemrograman pengantar di Universitas Sarajeva yang ditemukan online dengan bahasa pemrograma C/C++ yang setelah itu disortir untuk memisahkan pasangan yang melakukan plagiarisme dan tidak melakukan plagiarisme.

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing (Lanjutan)

No	Judul	Penciliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
2	Uji Plagiarism pada Tugas Mahasiswa Menggunakan Algoritma Winnowing	Ramli, M. S., Cokrowibowo, S., & Rustan, M. F., Journal of Applied Computer Science and Technology, 2021	Mendeteksi plagiarism pada source code pemrograman dengan menggunakan 10 tugas mahasiswa menggunakan algoritma winnowing dan mendapatkan nilai similarity yang beragam dengan membandingkan 2 tugas secara acak	Hasil penelitian yang telah dilakukan terhadap 10 tugas mahasiswa menggunakan algoritma winnowing menghasilkan nilai similarity yang beragam sebagai persentase kesamaan antara 2 tugas yang dibandingkan. Dengan rata-rata nilai similarity keseluruhan dari 10 tugas yaitu 75,12 %	Meningkatkan peluang dalam menemukan kesamaan antara kedua sampel yang dibandingkan	Dataset yang digunakan pada penelitian inii adalah tugas mahasiswa untuk kemudian dilakukan analisis terhadap data tersebut serta hasil dari penelitian yang dilakukan sebelumnya.
3	Plagiarism detection using manber and winnowing algorithm	Faisal, M., Nugroho, F., M El Sulthan, M., Amini, F., Hariyadi, M. A., & Sedayu, A., International Journal of Advanced Science and Technology, 2020	Untuk menggunakan algoritma Mamber dan Winnowing dalam mendeteksi plagiarisme pada teks dokumen abstrak	Hasil dari melakukan pengujian dengan 10 dokumen, rata-rata *similarity* algoritma mamber adalah 90,56% sedangkan algoritma winnowing 94%. Dan untuk algoritma winnowing dengan pendekatan triword menghasilkan akurasi 91, 22%.	Pendekatan biword dan algoritma winnowing lebih baik daripada algoritma manber dan algoritma winnowing dengan pendekatan triword.	Pada penelitian ini dilakukan penelitian menggunakan algoritma Mamber dan Winnowing dalam mendeteksi plagiarisme.

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing (Lanjutan)

No	Judul	Peneliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
4	Deteksi Plagiarisme pada Kode Bahasa Pemrograman Java menggunakan XGBoost.	Widjaja, T., Gunawan, A., & Liliana, L., *Jurnal Infra*, 2022	Berlatar belakang oleh proses pengecekan plagiarisme source code secara manual merupakan tugas yang repetitif, sulit, dan memerlukan waktu yang lama sehingga diperlukan otomasi untuk deteksi plagiarisme source code yang memiliki kualitas tinggi sangat dibutuhkan.	Dengan menggunakan features yang diajukan beserta preprocessing memiliki performa metrik yang lebih baik dari penelitian sebelumnya, yaitu f1-score sebesar 99%. Penerapan preprocessing juga dapat meningkatkan performa metrik pada features yang diajukan di di penelitian sebelumnya	Membuat dataset yang lebih banyak dalam hal kuantitas maupun variasi serangan plagiat sehingga dapat membuat sebuah model yang dapat mengakomodasi lebih banyak jenis variasi serangan plagiat	Algoritma yang digunakan pada penelitian ini untuk menghitung pairwise features dengan menggunakan 3 algoritma utama, yaitu levenshtein distance, greedy string tiling, dan bigram yang akan menghasilkan 12 features dan kumpulan feature statistik. Di langkah akhir, features akan digunakan untuk proses training maupun inference pada model XGBoost.

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing (Lanjutan)

No	Judul	Peneliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
5	Increasing K-means clustering algorithm effectivity for using in source code plagiarism detection.	Hrkút, P., Ďuračík, M., Mikušová, M., Callejas-Cuervo, M., & Zukowska, J., *International Conference on Smart Technologies, Systems and Applications*, 2022	Bertujuan untuk membuat sistem seperti itu untuk menemukan plagiarisme dalam kumpulan data kode sumber yang besar	Selama menjalankan sistem, komponen utama adalah database yang digunakan untuk mencari kecocokan kode sumber dan pengujian panjang rata-rata satu siklus dari algoritma, yang berarti menugaskan vektor ke klaster dan penghitungan ulang pusat serta menetapkan jumlah cluster menjadi 100.	Proses evaluasi plagiarisme yang rumit dan ketidakmampuan untuk menggunakan basis skala besar.	Pada penelitian ini dilakukan penelitian menggunakan Algoritma K-Means Clustering mendeteksi plagiarisme.
6	Detection of text similarity for indication plagiarism using winnowing algorithm based K-gram and jaccard coefficient.	Puspaningrum, E. Y., Nugroho, B., Setiawan, A., & Hariyanti, N., *Journal of Physics: Conference Series*, 2020	Untuk mendeteksi kesamaan teks. Semakin banyak kesamaan kata maka semakin teridikasi dokumen tersebut plagiarisme.	Menunjukkan bahwa penyesuaian nilai k-gram dan window dapat mempengaruhi hasil akhir dari nilai persentase kemiripan. Semakin kecil nilai k-gram maka semakin besar nilai persentasi	Penggunaan algoritma ini untuk mendeteksi plagiarisme memerlukan pengecekan lebih lanjut terhadap file yang akan diproses, karena membutuhkan file teks tanpa kesalahan penulisan	Data yang digunakan adalah abstrak jurnal berbahasa Indonesia. Input sistem ini adalah abstrak di jurnal berbahasa Indonesia. Pengujian data bersifat abstrak dengan membandingkan kumpulan data dalam database

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing (Lanjutan)

No	Judul	Peneliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
7	Rabin karp and Winnowing algorithm for Statistics of text document plagiarism detection.	Leman, D., Rahman, M., Ikorasaki, F., Riza, B. S., & Akbbar, M. B., *International Conference on Cyber and IT Service Management (CITSM)*, 2019	Bertujuan untuk membangun sistem pendeteksi plagiarisme pada dokumen teks menggunakan algoritma Rabin-Karp secara otomatis. Input dalam aplikasi yang dibangun berupa dokumen teks yang berekstensi .txt.	Penggunaan stemming berpengaruh terhadap akurasi nilai similaritas yang dihasilkan. Menggunakan stemming menghasilkan nilai yang cenderung kurang baik dibandingkan tanpa menggunakan stemming. Namun pada kasus tertentu seperti mengubah bentuk kalimat algoritma Rabin-Karp yang disisipkan ke dalam stemming menghasilkan akurasi nilai kemiripan yang lebih baik.	Aplikasi pendeteksi plagiarisme yang dibangun dapat digunakan untuk menghitung perhitungan plagiarisme berdasarkan tingkat kemiripan huruf, angka dan simbol per karakter	Pada penelitian ini dilakukan penelitian gabungan dengan menggunakan algoritma Rabin karp dan winnowing dalam mendeteksi plagiarisme untuk dokumen teks.
8	Code Plagiarism Detection Method Based on Code Similarity and Student Behavior Characteristics.	Huang, Q., Song, X., & Fang, G., *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2020	Mengklasifikasi masalah deteksi plagiarisme sebagai masalah klasifikasi biner dan menggunakan LightGBM untuk mengambil keputusan.	Setelah dilakukan penelitian mode LightGBM lebih baik daripada algoritma lainnya dikarenakan nilai F1 dan akurasinya cukup tinggi. Hasil menunjukkan bahwa akurasi mendekati 99% dan skor F1 mendekati 98%.	Algoritma LightGBM berdasarkan kesamaan kode dan karakteristik perilaku siswa, dapat memperoleh hasil yang lebih baik.	Menggunakan LightGBM untuk pendeteksian plagiarisme source code.

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing (Lanjutan)

No	Judul	Peneliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
9	Source-Code Similarity Measurement: Syntax Tree Fingerprinting for Automated Evaluation.	Verma, A., Udhayanan, P., Shankar, R. M., KN, N., & Chakrabarti, S. K., *The First International Conference on AI-ML-Systems*, 2021	Membangun sistem yang meniru evaluasi manual tugas pemrograman berdasarkan struktur dan bukan outuput dari program menggunakan kesamaan struktural antara program yang diberikan oleh solusi referensi. Menyajikan pendekatan ML untuk memetakan skor prediksi sistem ke skor menggunakan rubrik.	Nilai error yang dihitung untuk setiap pernyataan masalah adalah mean absolute error (MAE) untuk akurasi dan root mean squared error (RMSE) untuk presisi. Penelitian ini mendapatkan nilai mean absolute error secara konsisten berada di bawah 0,06 dan kesalahan yang dilaporkan serendah 12 persen dengan penyimpangan sekitar 3 persen, menunjukkan bahwa skor yang dihasilkan secara otomatis berkorelasi tinggi dengan skor yang diberikan instruktur. Hasil tersebut bisa dianggap memuaskan untuk tujuan penilaian struktural.	Evaluasi masalah pemrograman berukuran kecil dalam kursus pemrograman dasar, sehingga kompleksitasnya tidak mempengaruhi kinerjanya dalam skenario yang dimaksud.	Model yang digunakan adalah SSM. Structural Similarity Measurement, sebagai sistem untuk mengevaluasi tugas pemrograman siswa. Sistem menilai penugasan berdasarkan kesamaan struktural yang dimilikinya dengan solusi referensi yang diberikan

Tabel 2.1. Matriks literatur review dan posisi penelitian
Analisis Tingkat Kesamaan Source Code Pada Bahasa Pemrograman Kotlin Menggunakan Algoritma Winnowing (Lanjutan)

No	Judul	Peneliti, Media Publikasi, Dan Tahun	Tujuan Penelitian	Kesimpulan	Saran Atau Kelemahan	Perbandingan
10	Penerapan Algoritma Winnowing dan Word-Level Trigrams Untuk Mengidentifikasi Kesamaan Kata	Ramdhani, R., Fadlil, A., & Sunardi, S., *JURIKOM (Jurnal Riset Komputer*, 2022	Menemukan kesamaan kata pada teks menggunakan algoritma winnowing. Winnowing merupakan algoritma berbasis hashing-approach yang menerapkan hash-function dan pembentukan window untuk memperolch fingerprints pada saat pencocokan pola.	Hasil penelitian menunjukkan bahwa algoritma Winnowing yang diterapkan menggunakan word-level trigrams dapat mendeteksi kesamaan pada teks sebesar 76,84%, 52,29%, 37,40%, dan 19,29%. Dari hasil penelitian dapat disimpulkan bahwa metode pencocokan pola dengan algoritma Winnowing dan word-level trigrams dapat digunakan untuk mengukur tingkat kesamaan teks.	Dapat dikembangkan lagi untuk mendeteksi kesamaan kata pada dua dokumen karya ilmiah menggunakan algoritma pencocokan pola lainnya seperti algoritma Boyer-Moore Horspool, dll serta penelitian dapat diperluas dengan melakukan penandaan lokasi similaritas pada teks menggunakan algoritma hybrid	Penelitian ini menggunakan perhitungan tingkat kesamaan berdasarkan kata (word-level) yang jarang digunakan.

2.3. Landasan Teori

2.3.1. Source code

Kode program adalah suatu rangkaian pernyataan atau deklarasi yang ditulis dalam bahasa pemrograman komputer yang terbaca manusia. Kode program yang menyusun suatu program biasanya disimpan dalam satu atau lebih berkas teks, dan dapat pula ditampilkan dalam bentuk cuplikan kode (code snippet) yang dicetak pada buku atau media lainnya.

Berdasarkan (Bebbington, 2014) source code dituliskan dalam satu atau lebih bahasa pemrograman. Tujuan dari pemrograman adalah untuk menemukan urutan instruksi yang secara otomatis melakukan tugas tertentu atau memecahkan suatu masalah. Proses pemrograman yang demikian itu seringkali membutuhkan kemampuan tinggi di banyak subjek yang berbeda, termasuk pengetahuan akan domain dari aplikasi, algoritma khusus, dan logika formal.

Kegiatan-kegiatan yang terkait dengan pemrograman adalah pengujian, men-debug, dan mengurus source code, implementasi dalam pembangunan sistem, dan manajemen seperti mesin kode program komputer. Hal ini bisa dianggap sebagai bagian dari proses pemrograman, tetapi seringkali sebutan software development digunakan untuk proses yang lebih besar, sedangkan istilah programming, implementation, atau coding digunakan untuk penulisan source code yang sesungguhnya. Rekayasa perangkat lunak menggabungkan rekayasa teknik dengan pengembangan perangkat lunak.

2.3.2. Plagiarisme

Pengertian Plagiarisme menurut Pasal 1 angka 1 Permendiknas 17/2010 adalah perbuatan baik sengaja maupun tidak sengaja memperoleh atau berusaha memperoleh kredit atau nilai suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah pihak lain dan diakui karya ilmiahnya, tanpa mencantumkan sumbernya. secara tepat dan memadai.

Terdapat beberapa pendekatan yang digunakan untuk mengukur kesamaan dua program dapat diklasifikasikan secara luas berdasarkan aspek program yang dibandingkan. Pendekatan yang dimaksud adalah struktural, semantik, dan perilaku. Pendekatan struktural dengan mengidentifikasi dari struktur umum dari *source code*, pada pendekatan ini biasanya dengan cara mengidentifikasi dengan membandingkan rangkaian token leksikal, yang mewakili struktur kode sumber dalam kaitannya dengan elemen leksikal yang penting (Joy & Luck, 1999).

Selanjutnya terdapat pendekatan semantik adalah mengukur kesamaan melalui makna kode sumber. Hal ini melalui analisis semantik, yaitu menganalisis kode sumber untuk mengekstrak informasi yang tidak diungkapkan melalui tata bahasa suatu bahasa pemrograman. Pendekatan semantik biasanya menganalisis suatu program melalui grafik ketergantungan program (Ferrante et al., 1987).

Lalu terdapat pendekatan yang terakhir yaitu pendekatan perilaku, menganalisis perilaku runtime suatu program. Ada beragam teknik yang diterapkan untuk mengidentifikasi kesamaan perilaku. Hal ini dapat dilakukan dengan menganalisis kesetaraan fungsional suatu program (Li et al., 2016), penggunaan data saat runtime (Jhi et al., 2011), mengidentifikasi interaksi serupa dengan

lingkungan eksekusi (Anjali et al., 2015), atau mengidentifikasi logika program yang serupa (Luo et al., 2017). Pendekatan tersebut didasarkan pada asumsi bahwa perilaku suatu program merupakan fitur pengidentifikasi yang unik, dan bahwa perilaku yang serupa menunjukkan program yang serupa.

(Purwitasari et al., 2011) menyatakan bahwa plagiarisme adalah mengambil karya orang lain dan dipublikasikan sebagai penyebabnya seseorang mengabaikan kehormatan, mengabaikan kejujuran, cenderung menipu dan memandang rendah orang lain.

Berikut beberapa jenis teknik plagiarisme yang dikenal selama ini:

a. *Word-for-word plagiarism*: menyalin setiap kata secara langsung tanpa diubah sedikitpun
b. *Plagiarism of the form of a source*: menyalin dan menulis ulang kode-kode program tanpa mengubah struktur dan jalannya program
c. *Plagiarism of authorship*: mengakui hasil karya orang lain sebagai hasil karya sendiri dengan cara mencantumkan nama sendiri menggantikan nama pengarang sebenarnya.

Ada banyak cara yang dapat diterapkan untuk melakukan transformasi atas suatu kode program. Dua strategi yang paling umum untuk melakukan transformasi ini adalah

a. Transformasi leksikal

Perubahan leksikal adalah perubahan pada kode program (source code). Untuk melakukan transformasi ini, tidak diperlukan pengetahuan akan

bahasa pemrograman yang digunakan dalam kode program (source code) tersebut. Transformasi jenis ini meliputi:

1. Komentar diubah (ditambah, dikurangi, atau diganti)
2. Format penulisan atau identasi program diubah
3. Nama variable atau identifier diubah

b. Transformasi struktural

Perubahan struktural adalah perubahan pada struktur program. Untuk melakukan transformasi ini, diperlukan pengetahuan akan bahasa pemrograman yang digunakan dalam kode program (source code) tersebut. Metode ini sangat bergantung pada keahlian seseorang dalam menggunakan bahasa pemrograman tersebut. Transformasi ini meliputi:

1. pengubahan urutan metode yang tidak mengubah jalannya program
2. prosedur diubah menjadi fungsi dan sebaliknya
3. pemanggilan prosedur/fungsi diganti dengan isi prosedur/fungsi itu sendir
4. pengubahan bentuk pengulangan yang digunakan (repeat..until menjadi while..do)
5. penggunaan switch..case untuk menangani nested if

2.3.3. Algoritma Winnowing

Algoritma Winnowing adalah algoritma untuk menghasilkan suatu deret bilangan unik (fingerprint) yang mewakili suatu dokumen. (Schleimer et al., 2003) dengan fingerprint tersebut kita bisa mengetahui tingkat kemiripan satu dokumen dengan dokumen yang lain. Document fingerprinting merupakan metode yang digunakan untuk mendeteksi keakuratan salinan antar dokumen atau hanya

sebagian teks saja. Prinsip kerja dari metode document fingerprinting ini adalah dengan menggunakan teknik hashing. Teknik hashing adalah sebuah fungsi yang mengkonversi setiap string menjadi bilangan. Proses ini ditujukan agar dapat mengidentifikasi kemiripan, termasuk bagian-bagian kecil yang mirip dalam dokumen yang berjumlah banyak

Secara garis besar, algoritma Winnowing bekerja sebagai berikut:

1. Membuang karakter-karakter yang tidak relevan seperti spasi, tanda baca maupun artikel atau kata yang tidak penting, misalnya kata sambung. Langkah ini sesuai dengan syarat algoritma pendeteksi penjiplakan yaitu whitespace insensitivity dan noise suppression.
2. Membentuk rangkaian k-gram dari teks. Semisal ditentukan n = 6, maka akan dihasilkan sekumpulan gram atau string yang berukuran 6.
3. Melakukan fungsi hash untuk setiap gram. Persamaan (1) adalah perhitungan fungsi hash dari algoritma winnowing

$$H_{(C_1 \ldots C_k)} = C_1 * b^{k-1} + C_2 * b^{k-2} + C_{k-1} * b^k + c_k$$

$$H_{(C_2 \ldots C_{k+1})} = (H_{(C_2 \ldots C_k)} - c_1 * b^{k-1}) * b + c_{k+1} \ldots (1)$$

Keterangan:

H = nilai hash,

c = karakter pada gram,

b = bilangan basis,

k = jumlah karakter gram.

4. Membuat himpunan-himpunan yang disebut window yang terdiri dari i nilai hash. Jika i = 6, maka di dalam satu window terdapat 6 nilai hash.
5. Memilih fingerprint dari hasil hashing dengan pembagian hasil hash berdasarkan satu nilai window w, dan kemudian dipilih nilai hash terkecil dari setiap window tersebut.

Langkah b – e, berkaitan dengan syarat *Position Independence*. Dengan membentuk fingerprint dari sebuah dokumen teks, maka dapat dilakukan komparasi dengan fingerprint dokumen lainnya tanpa menghiraukan posisi dari fingerprint tersebut.

2.3.4. K-Grams

K-Gram adalah rangkaian terms dengan panjang K. Kebanyakan yang digunakan sebagai terms adalah kata. KGram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode K-Gram ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen.

Berikut ini adalah contoh diagram alir dari proses k-gram:

Gambar 2.1 Diagram alir dari proses k-gram

2.3.5. Rolling Hash

Perhitungan-perhitungan nilai-nilai hash dari setiap gram merupakan fungsi yang digunakan untuk menghasilkan nilai hash dari rangkaian gram dalam algoritma Winnowing adalah rolling hash. Rolling Hash adalah suatu cara untuk mentransformasi sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (fixed-length) yang berfungsi sebagai penanda string tersebut. Fungsi untuk menghasilkan nilai ini disebut fungsi hash, sedangkan nilai yang dihasilkan disebut nilai hash.

2.3.6. Jaccard *Similarity*

Penelitian yang pernah dilakukan (Barrón-Cedeno et al., 2009) dengan melakukan perbandingan berbagai metode untuk mengukur kesamaan antar teks, terdapat 7 metode pengukuran kesamaan teks yang meliputi *vector space, fingerprinting, and probabilistic models*. Mendapatkan hasil bahwa Jaccard, Cosine and Machine Translation memiliki kualitas yang bisa dibilang sama dalam hal Recall, HFM dan sep. Karena kesederhanaannya, Jaccard similarity tampaknya menjadi pilihan terbaik dalam mengukur kesamaan antar teks.

Jaccard Similarity memiliki kelebihan dalam mengukur tingkat kesamaan umum berdasarkan term-matching yang baik pada plagiarisme tanpa penulisan ulang dan kebingungan rendah, serta menghasilkan kinerja presisi tinggi (Haddi et al., 2013) Algoritma Winnowing bekerja dengan memecah dokumen menjadi kumpulan data (urutan kata yang berdekatan) sehingga Jaccard similarity adalah pilihan yang tepat untuk membandingkan himpunan karena secara langsung mengukur tumpang tindih antara dua himpunan, yang selaras dengan konsep perpotongan kata.

Untuk menghitung similarity dari source code yang akan dibandingkan maka digunakan Jaccard *Similarity* yang dapat digunakan untuk mengukur kemiripan antara dua set data. Jaccard *similarity* melakukan kalkulasi dengan menghitung seberapa banyak data yang sama antara dua buah set yang dibagi dengan jumlah elemen unik di kedua set (Gomaa & Fahmy, 2013). Jaccard *similarity* memiliki nilai di antara 0 – 1, semakin tinggi maka kedua set tersebut akan semakin mirip.

Persamaan (2) adalah persamaan untuk menghitung jaccard *similarity*. Di persamaan tersebut terdapat 2 buah set, yaitu set A dan set B.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|} \quad (3)$$

Keterangan:

J = Jaccard *Similarity*

A = Source code 1

B = Source code 2

2.3.7. *Text Preprocessing*

Text Preprocessing merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Tujuan dari *text preprocessing* yakni menghasilkan sebuah set *term index* yang bisa mewakili dokumen (Sanjaya, 2008). Teks yang sudah diperoleh akan melakukan tahapan dari text preprocessing terlebih dahulu untuk mendapatkan teks yang sudah siap diolah. Proses ini terdiri dari beberapa tahap pembersihan dokumen berikut ini (Purbo, 2019):

a. *Tokenizing* merupakan proses penguraian deskripsi yang semula berupa kalimat menjadi kata
b. *Filtering* adalah tahap mengambil kata penting dari hasil proses token. Bisa menggunakan algoritma *stoplist* atau *word list*. Filtering dapat juga diartikan sebagai proses mengambil kata – kata penting dari hasil proses token atau penghapusan *stopwords*. *Stopwords* merupakan kosa kata yang bukan merupakan ciri (kata unik) dari suatu dokumen

c. *Stemming* merupakan tahap untuk mencari *root* kata dari hasil *filtering*. Stemming adalah proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi bentuk kata dasarnya *(stem)*

d. *Tagging* merupakan tahap untuk mencari bentuk awal/*root* dari tiap kata lampau atau hasil dari proses stemming Terdapat beberapa kata lampau yang dikembalikan ke bentuk awal, misalkan pada data pesan dengan kata *"won"* diubah ke bentuk awal menjadi *"win"*

e. *Analyzing* merupakan tahap penentuan seberapa jauh keterhubungan antar suatu kata atau term terhadap suatu dokumen atau kalimat dengan menghitung nilai/bobot keterhubungan

Pada penelitian ini akan menggunakan *stemming* dan *tokenizing*. *Stemming* digunakan untuk menghapus bagian- bagian yang cenderung statis dan sama untuk semua code seperti deklarasi *package*, deklarasi *class*, deklarasi *main function*. Hal tersebut dilakukan agar bagian yang umum tidak memengaruhi penilaian apakah suatu source code plagiat atau tidak plagiat. *Tokenizing* berfungsi untuk mengelompokkan tiap token dari source code menjadi beberapa kategori.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini masuk kedalam penelitian eksperimental dengan melakukan pengujian terhadap pendeteksian plagiarisme pada source code dengan membandingkan hasil dengan atau tanpa preprocessing. Untuk mengetahui hal tersebut diperlukan untuk menghitung nilai similarity menggunakan Jaccard Similarity yang dapat digunakan untuk mengukur kemiripan antara dua set data.

Sifat pada penelitian ini adalah deskriptif, karena hasil yang akan dijabarkan dari penelitian ini merupakan hasil pengujian pada dtaaset untuk mengetahui berapa besar nilai similarity yang dihasilkan menggunakan algoritma winnowing.

Penelitian ini menggunakan pendekatan kuantitatif yang nantinya hasil dari penelitian ini berupa angka yaitu nilai similarity source code yang dibandingkan.

3.2. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah *source code* dengan bahasa pemrograman kotlin yang diperoleh dari github dengan tema yang sama lalu dibagi menjadi masing-masing 10 sampel dengan indikasi mirip dan tidak mirip.

3.3. Metode Analisis Data

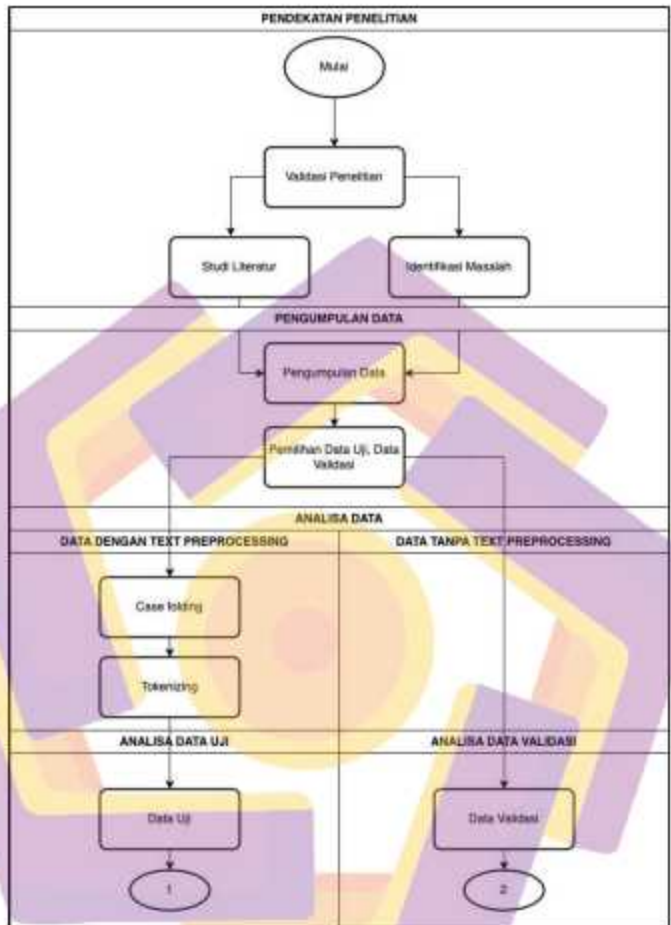
Pada tahapan analisis data yang dilaukan pertamakali yaitu melakukan pengumpulan data. Dataset diperoleh dari Github dengan bahasa pemrograman kotlin dengan tema "Aplikasi GiHub User". Kemudian dipilih secara manual source code yang memiliki kode yang mendekati plagiarisme atau tidak. Kemudian data tersebut akan diproses menggunakan text preprocessing akan menggunakan

stemming dan tokenizing. Stemming digunakan untuk menghapus bagian- bagian yang cenderung statis dan sama untuk semua code seperti deklarasi package, deklarasi class, deklarasi main function. Hal tersebut dilakukan agar bagian yang umum tidak memengaruhi penilaian apakah suatu source code plagiat atau tidak plagiat. Tokenizing berfungsi untuk mengelompokkan tiap token dari source code menjadi beberapa kategori. Kategori-kategori tersebut adalah whitespace, comments, strings, operators, keywords, functions, variables, and numbers. Lalu akan dilakukan pembentukan k-grams, menghitung nilai has menggunakan rolling hash.

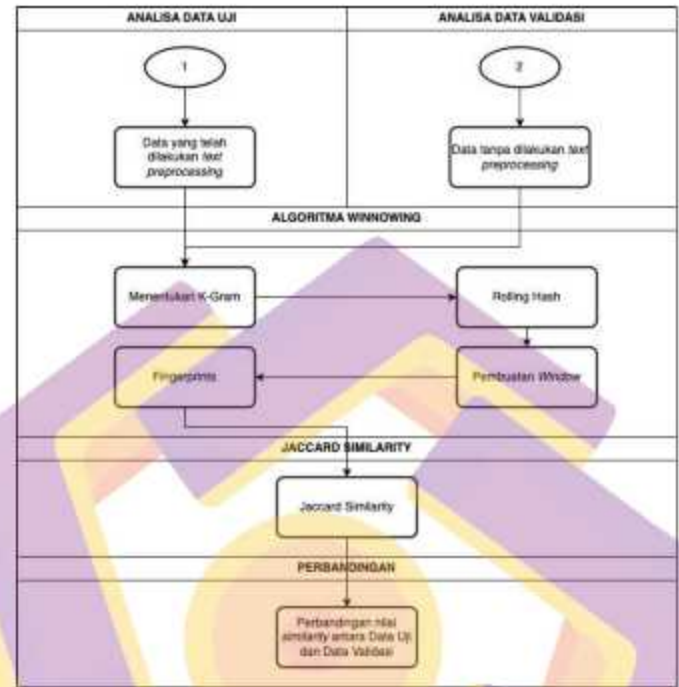
Hasil nilai hash tersebut nantinya akan dibandingkan setelah mendapatkan nilai fingerprint dari masing-masing source code. Setelah itu akan dihitung nilai similarity nya menggunakan jaccard similarity.

3.4. Alur Penelitian

Alur penelitian dibentuk untuk mempermudah proses pada penelitian . Alur penelitian dimulai dari pendekatan penelitian, pengumpulan data, analisa data, algoritma winnowing, jaccard *similarity*, dan perbandingan. Alur dari penelitian ini dapat diperhatikan pada Gambar 3.1 dan 3.2 dibawah ini:



Gambar 3.1 Diagram alur penelitian bagian 1



Gambar 3.2 Diagram alur penelitian bagian 2

Berikut penelitian pada alur penelitian pada Gambar 3.1 dan 3.2 dijelaskan pada beberapa poin menurut dari alur penelitian tersebut dan terdapat beberapa proses alur penelitian yang di jelaskan lebih detail seperti pada proses alur algoritma Winnowing dan Jaccard *Similarity*. Penjelasannya sebagai berikut:

3.4.1. Pendekatan Penelitian (Validasi Penelitian, Studi Literatur, Identifikasi Masalah)

Tahap awal penelitian yaitu melakukan validasi penelitian tentang permasalahan serta studi literatur untuk mengetahui masalah yang akan di analisa serta penggunaan algoritma yang tepat berdasarkan studi literatur sehingga dalam

pemilihan algoritma tersebut didasarkan pada hasil ilmu yang dianalisa oleh penelitian sebelumnya.

3.4.2. Pengumpulan Data (Pemilihan Data Uji, Data Validasi)

Source code bahasa pemrograman kotlin merupakan populasi pada penelitian ini. Mengingat jumlah *source code* bahasa pemrograman kotlin pada Github sangat banyak sehingga hanya mengambil dari tema "Aplikasi Github User" yang merupakan tema dari *submission* untuk sertifikasi kelas pemrograman kotlin. Dat sampel pada penelitian ini terdapat masing-masing 10 sampel pada indikasi mirip dan tidak mirip. Pada 10 sampel tersebut terdapat 2 *source code* yang akan dibandingkan nilai *similarity*nya. Setiap sampel memiliki masing-masing 2 source code yang akan dibandingkan. Pengambilan sampel menggunakan *white box sampling* yang memungkinkan para ilmuwan mengambil sampel titik-titik terputus dalam fungsi keluaran secara selektif dan efisien, dengan mempertimbangkan batasan kesalahan masukan. *White box sampling* bekerja dengan menganalisis eksekusi program dan dapat melakukan menentukan sampel dengan pertimbangan tertentu (Bao et al., 2012). Pengumpulan data dilakukan dengan dua tipe data yaitu data uji dan data validasi, untuk kedua data didapat dengan mengambil *source code* dari Github yang telah dipilih secara manual untuk dibandingkat dengan kategori *source code* mirip dan tidak mirip. Berikut ditampilkan data *source code* yang telah didapatkan pada Tabel 3.1

Tabel 3.1 Contoh data mentah *source code* bahasa pemrograman kotlin

Sampel	Source code 1	Source code 2	Indikasi

Sampel 1	package com.sidratul.finalsubmissionbfaa.adapter
	
	import android.view.LayoutInflater
	import android.view.ViewGroup
	import androidx.recyclerview.widget.RecyclerView
	import com.bumptech.glide.Glide
	import com.sidratul.finalsubmissionbfaa.databinding.ItemRowUsersBinding
	import com.sidratul.finalsubmissionbfaa.db.User
	
	class UserAdapter : RecyclerView.Adapter<UserAdapter.ListViewHolder>() {
	
	private val list = ArrayList<User>()
	private var onItemClickCallback: OnItemClickCallback? = null
	
	override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
	val view = ItemRowUsersBinding.inflate(LayoutInflater.from(parent.context), parent, false)
	return ListViewHolder(view)
	}
	
	override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
	holder.bindItem(list[position])
	}
	
	override fun getItemCount(): Int = list.size
	
	inner class ListViewHolder(private val binding: ItemRowUsersBinding): RecyclerView.ViewHolder(binding.root) {
	
	fun bindItem(user: User){
	
	binding.root.setOnClickListener {
		package com.iman.fundamental_submission3_iman.mainActivity
		
		import android.annotation.SuppressLint
		import android.view.LayoutInflater
		import android.view.ViewGroup
		import androidx.recyclerview.widget.RecyclerView
		import com.bumptech.glide.Glide
		import com.iman.fundamental_submission3_iman.databinding.ItemRowUsersBinding
		import com.iman.fundamental_submission3_iman.model.User
		
		class UserAdapter : RecyclerView.Adapter<UserAdapter.ListViewHolder>() {
		
		private val list = ArrayList<User>()
		private var onItemClickCallback: OnItemClickCallback? = null
		
		override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
		val view = ItemRowUsersBinding.inflate(LayoutInflater.from(parent.context), parent, false)
		return ListViewHolder(view)
		}
		
		override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
		holder.bindItem(list[position])
		}
		
		override fun getItemCount(): Int = list.size
		
		inner class ListViewHolder(private val binding: ItemRowUsersBinding): RecyclerView.ViewHolder(binding.root) {
		
		fun bindItem(user: User){
		
		binding.root.setOnClickListener {
		
		onItemClickCallback?.onItemClicked(user)
			Mirip

Tabel 3.2 Contoh data mentah *source code* bahasa pemrograman kotlin (Lanjutan)

	onItemClickCallback?.onItemClicked(user)
	}
	
	binding.apply {
	
	tvItemUsername.text = user.login
	tvItemUrl.text = user.url
	Glide.with(itemView)
	.load(user.avatar_url)
	.centerCrop()
	.into(imgItemPhoto)
	
	}
	}
	}
	
	fun setList(users: ArrayList<User>) {
	list.clear()
	list.addAll(users)
	notifyDataSetChanged()
	}
	
	fun setOnItemClickCallback(onItemClickCallback: OnItemClickCallback) {
	this.onItemClickCallback = onItemClickCallback
	}
	
	interface OnItemClickCallback{
	fun onItemClicked(user: User)
	}
	
	}
		}
		
		binding.apply {
		
		tvItemUsername.text = user.login
		tvItemUrl.text = user.url
		Glide.with(itemView)
		.load(user.avatar_url)
		.centerCrop()
		.into(imgItemPhoto)
		
		}
		}
		}
		
		fun setList(users: ArrayList<User>) {
		list.clear()
		list.addAll(users)
		notifyDataSetChanged()
		}
		
		fun setOnItemClickCallback(onItemClickCallback: OnItemClickCallback) {
		this.onItemClickCallback = onItemClickCallback
		}
		
		interface OnItemClickCallback{
		fun onItemClicked(user: User)
		}
		
		}	

Sampel 2	package com.example.githubusersubmission
	
	import androidx.appcompat.app.AppCompatActivity
	import android.os.Bundle
	import android.widget.ImageView
	import android.widget.TextView
	
	class DetailActivity : AppCompatActivity() {
	
	companion object {
	const val DETAIL_USER = "detail_user"
	}
		package com.gopal.aplikasigithubuser
		
		import androidx.appcompat.app.AppCompatActivity
		import android.os.Bundle
		import android.view.MenuItem
		import com.bumptech.glide.Glide
		import kotlinx.android.synthetic.main.activity_detail.*
		
		class DetailActivity : AppCompatActivity() {
		
		companion object{
		const val USER = "user"
		}
			Tidak Mirip

Tabel 3.3 Contoh data mentah *source code* bahasa pemrograman kotlin (Lanjutan)

	override fun onCreate(savedInstanceState: Bundle?) {
	
	super.onCreate(savedInstanceState)
	
	setContentView(R.layout.activity_detail)
	
	supportActionBar?.title = "Detail User"
	
	val avatar: ImageView = findViewById(R.id.img_avatar)
	val name: TextView = findViewById(R.id.txt_name)
	val username: TextView = findViewById(R.id.txt_username)
	val follower: TextView = findViewById(R.id.txt_user_follower)
	val following: TextView = findViewById(R.id.txt_user_following)
	val company: TextView = findViewById(R.id.txt_company)
	val location: TextView = findViewById(R.id.txt_location)
	val repository: TextView = findViewById(R.id.txt_repository)
	
	val users = intent.getParcelableExtra<User>(DETAIL_USER) as User
	
	avatar.setImageResource(users.avatar)
	name.text = users.name
	username.text = users.username
	follower.text = users.followers
	following.text = users.following
	company.text = users.company
	location.text = users.location
	repository.text = users.repository
	}
	}
		override fun onCreate(savedInstanceState: Bundle?) {
		
		super.onCreate(savedInstanceState)
		
		setContentView(R.layout.activity_detail)
		
		val userDetail = intent.getParcelableExtra<User>(USER)
		
		tv_detail_name.text = userDetail?.name
		tv_detail_location.text = userDetail?.location
		tv_detail_repo.text = userDetail?.repository
		tv_detail_company.text = userDetail?.company
		tv_detail_followers.text = userDetail?.followers
		tv_detail_following.text = userDetail?.following
		
		Glide.with(this)
		.load(userDetail?.avatar)
		.into(iv_detail_avatar)
		
		val actionBar = supportActionBar
		actionBar!!.title = userDetail?.username
		
		actionBar.setDisplayHomeAsUpEnabled(true)
		}
		
		override fun onOptionsItemSelected(item: MenuItem): Boolean {
		onBackPressed()
		return super.onOptionsItemSelected(item)
		}
		
		}	

Berdasarkan Tabel 3.1 didapatkan hasil berupa *source code* bahasa pemrograman kotlin dengan indikasi awal yang memiliki kategori mirip dan tidak mirip. Sehingga nantinya data mentah dari *source code* tersebut akan diproses kedalam *text preprocessing*.

3.4.3. Pengolahan Data (*Text preprocessing*)

Tahap ini melakukan proses *case folding* dan *tokenizing*. Proses *case folding* berfungsi untuk menghilangkan karakter yang tidak valid dan mengubah huruf menjadi huruf kecil dan menghilangkan karakter yang dianggap tidak valid seperti angka, tanda baca, dan *Uniform Resources Locator* (URL). Proses selanjutnya adalah *tokenization* yang berguna untuk memecah urutan karakter atau string teks menjadi unit yang lebih kecil yang biasa disebut token. Setiap token mewakili unit teks yang bermakna, seperti kata, simbol, atau tanda baca. Berikut ditampilkan hasil dari kedua tahap *text preprocessing* yang dtampilkan pada Tabel 3.2

Tabel 3.4 Hasil *Text preprocessing*

Sampel	Source code 1	Source code 2
Sampel 1	packagecomsidratulfinalsubmissionbfaaadapterimportandroidviewlayoutinflaterimportandroidviewviewgroupimportandroidxrecyclerviewwidgetrecyclerviewimportcombumptechglideglideimportcomsidratulfinalsubmissionbfaadatabindingitemrowusersbindingimportcomsidratulfinalsubmissionbfaadbuserclassuseradapterrecyclerviewadapteruseradapterlistviewholderprivatevallistarraylistuserprivatevaronitemclickcallbackonitemclickcallback?n	packagecomimanfundamentalsubmission3imanmainactivityimportandroidannotationsuppresslintimportandroidviewlayoutinflaterimportandroidviewviewgroupimportandroidxrecyclerviewwidgetrecyclerviewimportcombumptechglideglideimportcomimanfundamentalsubmission3imandatabindingitemrowusersbindingimportcomimanfundamentalsubmission3imanmodeluserclassusersadapterrecyclerviewadapteruseradapterlistviewholderprivatevallistarraylistuserprivatevaronitemclickcallbackonitemclickcallback?n

Tabel 3.5 Hasil *Text preprocessing* (Lanjutan)

	ulloverridefunoncreateviewhold erparentviewgroupviewtypeintli stviewholdervalviewitemrowuse rsbindinginflatelayoutinflaterfro mparentcontextparentfalseretur nlistviewholderviewoverridefuno nbindviewholderholderlistviewh olderpositionintholderbinditemli stpositionoverridefungetitemco untintlistsizeinnerclasslistviewh olderprivatevalbindingitemrowu sersbindingrecyclerviewviewhol derbindingrootfunbinditemuser userbindingrootsetonclicklisten eronitemclickcallback?o nitemclickeduserbindingapplytvi temusernametextuserlogintvite murltextuserurlglidewithitemvie wloaduseravatarurlcentercropin toimgitemphotofunsetlistusersa rraylistuserlistclearlistaddalluser snotifydatasetchangedfunsetoni temclickcallbackonitemclickcall backonitemclickcallbackthisonit emclickcallbackonitemclickcallb ackinterfaceonitemclickcallback funonitemclickeduseruser	ulloverridefunoncreateviewholderp arentviewgroupviewtypeintlistview holdervalviewitemrowusersbindingi nflatelayoutinflaterfromparentcont extparentfalsereturnlistviewholderv iewoverridefunonbindviewholderho lderlistviewholderpositionintholder binditemlistpositionoverridefungeti temcountintlistsizeinnerclasslistvie wholderprivatevalbindingitemrowu sersbindingrecyclerviewviewholder bindingrootfunbinditemuseruserbin dingrootsetonclicklisteneronitemcli ckcallback?o nitemclickeduserbindingapplytvite musernametextuserlogintvitemurlt extuserurlglidewithitemviewloadus eravatarurlcentercropintoimgitemp hotofunsetlistusersarraylistuserlistc learlistaddallusersnotifydatasetcha ngedfunsetonitemclickcallbackonit emclickcallbackonitemclickcallback thisonitemclickcallbackonitemclick callbackinterfaceonitemclickcallbac kfunonitemclickeduseruser

Tabel 3.6 Hasil *Text preprocessing* (Lanjutan)

Sampel 2	packagecomexamplegithubuser submissionimportandroidxappc ompatappappcompatactivityim portandroidosbundleimportandr oidwidgetimageviewimportandr oidwidgettextviewclassdetailacti vityappcompatactivitycompanio nobjectconstvaldetailuserdetail useroverridefunoncreatesavedin stancestatebundle?s uperoncreatesavedinstancestate setcontentviewrlayoutactivityde tailsupportactionbar?t itledetailuservalavatarimagevie wfindviewbyidridimgavatarvalna metextviewfindviewbyidridtxtna mevalusernametextviewfindvie wbyidridtxtusernamevalfollower textviewfindviewbyidridtxtuserf ollowervalfollowingtextviewfind viewbyidridtxtuserfollowingvalc ompanytextviewfindviewbyidrid txtcompanyvallocationtextviewfi ndviewbyidridtxtlocationvalrepo sitorytextviewfindviewbyidridtxt repositoryvalusersintentgetparc elableextrauserdetailuserasuser avatarsetimageresourceusersav atarnametextusersnameuserna metextusersusernamefollowerte xtusersfollowersfollowingtextus ersfollowingcompanytextusersc ompanylocationtextuserslocatio nrepositorytextusersrepository	packagecomgopalaplikasigithubuse rimportandroidxappcompatappapp compatactivityimportandroidosbun dleimportandroidviewmenuitemim portcombumptechglideglideimport kotlinxandroidsyntheticmainactivit ydetail * classdetailactivityappcompatactivit ycompanionobjectconstvaluseruser overridefunoncreatesavedinstances tatebundle?s uperoncreatesavedinstancestateset contentviewrlayoutactivitydetailval userdetailintentgetparcelableextrau serusertvdetailnametextuserdetail? n ametvdetaillocationtextuserdetail?l ocationtvdetailrepotextuserdetail?r epositorytvdetailcompanytextuserd etail?c ompanytvdetailfollowerstextuserde tail?f ollowerstvdetailfollowingtextuserde tail?f ollowingglidewiththisloaduserdetail ?a vatarintolvdetailavatarvalactionbar supportactionbaractionbar !! titleuserdetail?u sernameactionbarsetdisplayhomea supenabledtrueoverridefunonoptio nsitemselecteditemmenuitemboole anonbackpressedreturnsuperonopt ionsitemselecteditem

Berdasarkan Tabel 3.2 telah ditampilkan hasil dari pemrosesan *text preprocessing* sehingga proses pada *text preprocessing* telah selesai, selanjutnya data hasil *text preprocessing* akan dilanjutkan pada pemrosesan pada tahapan algoritma Winnowing.

3.4.4. Analisa Data Uji dan Data Validasi

Data uji dan data validasi selanjutnya masing-masing akan memiliki dua varian dari tahap *text preprocessing*, data validasi ini adalah data yang sebelumnya telah dipilih secara manual untuk diidentifikasi dengan kategori *source* terindikasi mirip dan tidak mirip. Data uji dan data validasi akan diproses ke dalam algoritma Winnowing untuk mengetahui tingkat *similarity* yang akan dinilai akurasinya berdasarkan kategori yang ditentukan.

3.4.5. Algoritma Winnowing

Berikut detail dari alur penelitian pada tahap algoritma Winnowing:

Gambar 3.2 menjelaskan bahwa tahap pada langkah ini menentukan variasi *k-gram* terlebih dahulu selanjutnya data uji dan data validasi akan dihitung nilai *hash*-nya menggunakan *Rolling Hash* yang nantinya akan dibuat beberapa *window* dan setiap *window* akan dicari nilai terkecilnya atau *fingerprint*

3.4.5.1. Menentukan K-Gram

Berikut ditampilkan hasil dari pemrosesan data dari *text preprocessing* dengan jumlah *k* dengan bilangan prima yaitu 2, 3, 5, dan 7 sehingga hasil dari perpotongan kata berdasarkan *k-gram* dengan inputan data *text preprocessing* ditampilkan pada tabel 3.3

Tabel 3.7 Hasil Perpotongan K-Gram dengan data yang sudah dilakukan *Text Preprocessing*

Sample	Source code 1				Source code 2			
	$k=2$	$k=3$	$k=5$	$k=7$	$k=2$	$k=3$	$k=5$	$k=7$
Sample 1	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'ec', 'co', 'om', 'ms',	['pac', 'ack', 'cka', 'kag', 'age', 'gec', 'eco', 'com', 'oms', 'msi',	['packa', 'ackag', 'ckage', 'kagec', 'ageco',	['package', 'ackagec', 'ckageco', 'kagecom', 'agecoms',	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'ec', 'co', 'om', 'mi',	['pac', 'ack', 'cka', 'kag', 'age', 'gec', 'eco', 'com', 'omi', 'mim',	['packa', 'ackag', 'ckage', 'kagec', 'ageco',	['package', 'ackagec', 'ckageco', 'kagecom', 'agecomi',

Tabel 3.8 Hasil Perpotongan K-Gram dengan data yang sudah dilakukan *Text Preprocessing* (Lanjutan)

	'si', 'id', 'dr', 'ra', 'at', 'tu', 'ul', 'lf', 'fi', 'in', 'na', 'al', .., 'eo', 'on', 'ni', 'it', 'te', 'em', 'mc', 'cl', 'li', 'ic', 'ck', 'kc', 'ca', 'al', 'll', 'lb', 'ba', 'ac', 'ck', 'kf', 'fu', 'un', 'no', 'on', 'ni', 'it', 'te', 'em', 'mc', 'cl', 'li', 'ic', 'ck', 'ke', 'ed', 'du', 'us', 'se', 'er', 'ru', 'us', 'se', 'er', 'r ']	'sid', 'idr', 'dra', 'rat', 'atu', 'tul', 'ulf', 'lfi', 'fin', .., 'mcl', 'cli', 'lic', 'ick', 'ckc', 'kca', 'cal', 'all', 'llb', 'lba', 'bac', 'ack', 'ckf', 'kfu', 'fun', 'uno', 'non', 'oni', 'nit', 'ite', 'tem', 'emc', 'mcl', 'cli', 'lic', 'ick', 'cke', 'ked', 'edu', 'dus', 'use', 'ser', 'eru', 'rus', 'use', 'ser', 'er ', 'r \n']	'gecom', 'ecoms', 'comsi', 'omsid', 'msidr', .., 'itemc', 'temcl', 'emcli', 'mclic', 'click', 'licke', 'icked', 'ckedu', 'kedus', 'eduse', 'duser', 'useru', 'serus', 'eruse', 'ruser', 'user ', 'ser \n']	'gecomsi', 'ecomsid', 'comsidr', 'omsidra', 'msidrat', 'sidratu', 'idratul', 'dratulf', 'ratulfi', 'atulfin', .., 'mclicke', 'clicked', 'lickedu', 'ickedus', 'ckeduse', 'keduser', 'eduseru', 'duserus', 'useruse', 'seruser', 'eruser ', 'ruser \n']	'im', 'ma', 'an', 'nf', 'fu', 'un', 'nd', 'da', 'am', 'me', 'en', 'nt', 'ta', 'al', 'ls', .., 'ca', 'al', 'ac', 'ck', 'ki', 'in', 'nt', 'te', 'er', 'rf', 'fa', 'ac', 'ce', 'eo', 'on', 'ni', 'it', 'te', 'em', 'mc', 'cl', 'li', 'ic', 'ck', 'kc', 'ca', 'al', 'll', 'lb', 'ba', 'ac', 'ck', 'kf', 'fu', 'un', 'no', 'on', 'ni', 'it', 'te', 'em', 'mc', 'cl', 'li', 'ic', 'ck', 'ke', 'ed', 'du', 'us', 'se', 'er', 'ru', 'us', 'se', 'er', 'r ']	'ima', 'man', 'anf', 'nfu', 'fun', 'und', 'nda', 'dam', 'ame', 'men', 'ent', 'nta', 'tal', 'als', .., 'ite', 'tem', 'emc', 'mcl', 'cli', 'lic', 'ick', 'ckc', 'kca', 'cal', 'all', 'llb', 'lba', 'bac', 'ack', 'ckf', 'kfu', 'fun', 'uno', 'non', 'oni', 'nit', 'ite', 'tem', 'emc', 'mcl', 'cli', 'lic', 'ick', 'cke', 'ked', 'edu', 'dus', 'use', 'ser', 'eru', 'rus', 'use', 'ser', 'er ', 'r \n']	'gecom', 'ecomi', 'comim', 'omima', 'miman', 'imanf', 'manfu', 'anfun', 'nfund', .., 'itemc', 'temcl', 'emcli', 'mclic', 'click', 'licke', 'icked', 'ckedu', 'kedus', 'eduse', 'duser', 'useru', 'serus', 'eruse', 'ruser', 'user ', 'ser \n']	'gecomim', 'ecomima', 'comiman', 'omimanf', 'mimanfu', 'imanfun', 'manfund', 'anfunda', 'nfundam', .., 'temclic', 'emclick', 'mclicke', 'clicked', 'lickedu', 'ickedus', 'ckeduse', 'keduser', 'eduseru', 'duserus', 'useruse', 'seruser', 'eruser ', 'ruser \n']

Tabel 3.9 Hasil Perpotongan K-Gram dengan data yang sudah dilakukan *Text Preprocessing* (Lanjutan)

Sample 2	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'ec', 'co', 'om', 'mg', 'go', 'op', 'pa', 'al', 'la', 'ap', 'pl', 'li', 'ik', 'ka', 'as', 'si', 'ig', 'gi', 'it', 'th', 'hu', 'ub', 'bu', 'us', .., 'on', 'no', 'op', 'pt', 'ti', 'io', 'on', 'ns', 'si', 'it', 'te', 'em', 'ms', 'se', 'el', 'le', 'ec', 'ct', 'te', 'ed', 'di', 'it', 'te', 'em', 'm ']	['pac', 'ack', 'cka', 'kag', 'age', 'gec', 'eco', 'com', 'omg', 'mgo', 'gop', 'opa', 'pal', 'ala', 'lap', 'apl', 'pli', 'lik', 'ika', 'kas', 'asi', 'sig', 'igi', 'git', 'ith', 'thu', 'hub', 'ubu', 'bus', 'use', .., 'ono', 'nop', 'opt', 'pti', 'tio', 'ion', 'ons', 'nsi', 'sit', 'ite', 'tem', 'ems', 'mse', 'sel', 'ele', 'lec', 'ect', 'cte', 'ted', 'edi', 'dit', 'ite', 'tem', 'em ', 'm \n']	['packa', 'ackag', 'ckage', 'kagec', 'ageco', 'gecom', 'ecomg', 'comgo', 'omgop', 'mgopa', 'gopal', 'opala', 'palap', 'alapl', 'lapli', 'aplik', 'plika', 'likas', 'ikasi', .., 'onsit', 'nsite', 'sitem', 'items', 'temse', 'emsel', 'msele', 'selec', 'elect', 'lecte', 'ected', 'ctedi', 'tedit', 'edite', 'ditem',	['package', 'ackagec', 'ckageco', 'kagecom', 'agecomg', 'gecomgo', 'ecomgop', 'comgopa', 'omgopal', 'mgopala', 'gopalap', 'opalapl', 'palapli', 'alaplik', 'laplika', 'aplikas', 'plikasi', 'likasig', 'ikasigi', 'kasigit', 'asigith', .., 'ptionsi', 'tionsit', 'ionsite', 'onsitem', 'nsitems', 'sitemse', 'itemsel', 'temsele', 'emselec', 'mselect', 'selecte', 'elected', 'lectedi', 'ectedit',	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'ec', 'co', 'om', 'me', 'ex', 'xa', 'am', 'mp', 'pl', 'le', 'eg', 'gi', 'it', 'th', 'hu', 'ub', 'bu', 'us', 'se', 'er', 'rs', 'su', 'ub', 'bm', .., 'si', 'it', 'to', 'or', .., 'ry', 'yt', 'te', 'ex', 'xt', 'tu', 'us', 'se', 'er', 'rs', 'sr', 're', 'ep', 'po', 'os', 'si', 'it', 'to', 'or', 'ry', 'y ']	['pac', 'ack', 'cka', 'kag', 'age', 'gec', 'eco', 'com', 'ome', 'mex', 'exa', 'xam', 'amp', 'mpl', 'ple', 'leg', 'egi', 'git', 'ith', 'thu', 'hub', 'ubu', 'bus', 'use', 'ser', 'ers', 'rsr', 'sre', 'rep', 'epo', 'pos', 'osi', 'sit', 'ito', 'tor', 'ory', 'ry ', 'y \n']	['packa', 'ackag', 'ckage', 'kagec', 'ageco', 'gecom', 'ecome', 'comex', 'omexa', 'mexam', 'examp', 'xampl', 'ample', 'mpleg', 'plegi', 'legit', 'egith', 'githu', 'ithub', 'thubu', .., 'sitor', 'itory', 'toryt', 'oryte', 'rytex', 'ytext', 'textu', 'extus', 'xtuse', 'tuser', 'users', 'sersr', 'ersre', 'rsrep', 'srepo',	['package', 'ackagec', 'ckageco', 'kagecom', 'agecome', 'gecomex', 'ecomexa', 'comexam', 'omexamp', 'mexampl', 'example', 'xampleg', 'amplegi', 'mplegit', 'plegith', 'legithu', 'github', 'githubu', 'ithubus', 'thubuse', 'hubuser', .., 'extuser', 'xtusers', 'tusersr', 'usersre', 'sersrep', 'ersrepo', 'rsrepos', 'sreposi', 'reposit', 'eposito', 'positor', 'ository', 'sitory ', 'itory \n']

Tabel 3.10 Hasil Perpotongan K-Gram dengan data yang sudah dilakukan *Text Preprocessing* (Lanjutan)

			item ', 'tem \n']	'ctedite', 'teditem', 'editem ', 'ditem \n']			'repos', 'eposi', 'posit', 'osito', 'sitor', 'itory', 'tory ', 'ory \n']	

Selain dari pemrosesan pemotongan karakter berdasarkan jumlah nilai *k* pada K-Gram yang telah dilakukan *text preprocessing* selanjutnya dilakukan pemotongan karakter pada data tanpa dilakukan *text preprocessing*. Berikut ditampilkan hasil dari pemotongan karakter pada hasil tanpa dilakukan proses *text preprocessing* ditampilkan pada Tabel 3.4

Tabel 3.11 Hasil Perpotongan K-Gram dengan data tanpa dilakukan *Text Preprocessing*

Sample	Source code 1				Source code 2			
	k = 2	*k* = 3	*k* = 5	*k* = 7	*k* = 2	*k* = 3	*k* = 5	*k* = 7
Sample 1	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'e ', 'co', 'om', 'm.', '.s', 'si', 'id', 'dr', 'ra', 'at', 'tu', 'ul', 'l.', '.f', 'fi', 'in', 'na', 'al', 'ls', 'su', 'ub', 'bm', 'mi', '..', '(u',	['pac', 'ack', 'cka', 'kag', 'age', 'ge ', 'e c', 'com', 'om.', 'm.s', '.si', 'sid', 'idr', 'dra', 'rat', 'atu', 'tul', 'ul.', 'l.f', '.fi', 'fin', 'ina', 'nal', 'als', 'lsu', 'sub', 'ubm', 'bmi', 'mis',	['packa', 'ackag', 'ckage', 'kage ', 'age c', 'ge co', 'e com', 'com.s', 'om.si', 'm.sid', '.sidr', 'sidra', 'idrat', 'dratu', 'ratul',	['package', 'ackage ', 'ckage c', 'kage co', 'age com', 'ge com.', 'e com.s', 'com.sid', 'om.sidr', 'm.sidra', '.sidrat', 'sidratu', 'idratul', 'dratul.', 'ratul.f',	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'e ', 'co', 'om', 'm.', '.i', 'im', 'ma', 'an', 'n.', '.f', 'fu', 'un', 'nd', 'da', 'am', 'me', 'en', 'nt', 'ta', 'al', 'l_', '_s', 'su', .., 'd(', '(u', 'us', 'se',	['pac', 'ack', 'cka', 'kag', 'age', 'ge ', 'e c', 'com', 'om.', 'm.i', '.im', 'ima', 'man', 'an.', 'n.f', '.fu', 'fun', 'und', 'nda', 'dam', 'ame', 'men', 'ent', 'nta', 'tal', 'al_', 'l_s', '_su', 'sub', .., 'ed(',	'packa', 'ackag', 'ckage', 'kage ', 'age c', 'ge co', 'e com', 'com.i', 'om.im', 'm.ima', '.iman', 'iman.', 'man.f', 'an.fu', 'n.fun',	['package', 'ackage ', 'ckage c', 'kage co', 'age com', 'ge com.', 'e com.i', 'com.ima', 'om.iman', 'm.iman.', '.iman.f', 'iman.fu', 'man.fun', 'an.fund', 'n.funda',

Tabel 3.12 Hasil Perpotongan K-Gram dengan data tanpa dilakukan *Text Preprocessing* (Lanjutan)

	'us', 'se', 'er', 'r:', ': ', 'Us', 'se', 'er', 'r)', ')\n', '\n ', ']\n', '\n\n', '\n)']	.., 'd(u', '(us', 'use', 'ser', 'er:', 'r: ', ': U', 'Use', 'ser', 'er)', 'r)\n', ')\n ', '\n ', ']\n\n', '\n\n)']	'atul.', 'tul.f', 'ul.fi', 'l.fin', '.fina', .., 'licke', 'icked', 'cked(', 'ked(u', 'ed(us', 'd(use', '(user', 'user:', 'ser: ', 'er: U', 'r: Us', ': Use', 'User)', 'ser)\n', 'er)\n ', 'r)\n ', ')\n ', '\n ']	'atul.fi', 'tul.fin', 'ul.fina', 'l.final', .., 'Clicked', 'licked(', 'icked(u', 'cked(us', 'ked(use', 'ed(user', 'd(user:', '(user: ', 'user: U', 'ser: Us', 'er: Use', 'r: User', ': User)', 'User)\n ', 'ser)\n ', 'er)\n ', 'r)\n ', ')\n]', '\n]\n']	'er', 'r:', ': ', 'Us', 'se', 'er', 'r)', ')\n', '\n ', ']\n', '\n\n', '\n)']	'd(u', '(us', 'use', 'ser', 'er:', 'r: ', ': U', 'Use', 'ser', 'er)', 'r)\n', ')\n ', '\n ', ']\n\n', '\n\n)']	'.fund', 'funda', 'undam', 'ndame', 'damen', .., 'icked', 'cked(', 'ked(u', 'ed(us', 'd(use', '(user', 'user:', 'ser: ', 'er: U', 'r: Us', ': Use', 'User)', 'ser)\n', 'er)\n ', 'r)\n ', ')\n ', '\n ']	'.fundam', 'fundame', 'undamen', 'ndament, .., 'Clicked', 'licked(', 'icked(u', 'cked(us', 'ked(use', 'ed(user', 'd(user:', '(user: ', 'user: U', 'ser: Us', 'er: Use', 'r: User', ': User)', 'User)\n ', 'ser)\n ', 'er)\n ', 'r)\n ', ')\n]', '\n]\n']
Sample 2	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'e ', 'co', 'om', 'm.', 'g', 'go', 'op', 'pa', 'al', 'l.', '.a', 'ap', 'pl', 'li', 'ik', 'ka', 'as', 'si', 'ig', 'gi',	['pac', 'ack', 'cka', 'kag', 'age', 'ge ', 'e c', 'com', 'om.', 'm.g', '.go', 'gop', 'opa', 'pal', 'al.', 'l.a', '.ap', 'apl', 'pli', 'lik', 'ika', 'kas', 'asi', 'sig', 'igi', 'git',	['packa', 'ackag', 'ckage', 'kage ', 'age c', 'ge co', 'e com', 'com.g', 'om.go', 'm.gop', '.gopa', 'gopal', 'opal.', 'pal.a', 'al.ap', 'l.apl', '.apli', 'aplik',	['package', 'ackage ', 'ckage c', 'kage co', 'age com', 'ge com.', 'e com.g', 'com.gop', 'om.gopa', 'm.gopal', '.gopal.', 'gopal.a', 'opal.ap',	['pa', 'ac', 'ck', 'ka', 'ag', 'ge', 'e ', 'co', 'om', 'm.', '.e', 'ex', 'xa', 'am', 'mp', 'pl', 'le', 'e.', '.g', 'gi', 'it', 'th', 'hu', 'ub', 'bu', 'us', 'se', 'er', 'rs', ..,	['pac', 'ack', 'cka', 'kag', 'age', 'ge ', 'e c', 'com', 'om.', 'm.e', '.ex', 'exa', 'xam', 'amp', 'mpl', 'ple', 'le.', 'e.g', '.gi', 'git', 'ith', 'thu', 'hub', 'ubu', 'bus', 'use', 'ser',		['package', 'ackage ', 'ckage c', 'kage co', 'age com', 'ge com.', 'e com.e', 'com.exa', 'om.exam', 'm.examp', '.exampl', 'example', 'xample.',

Tabel 3.13 Hasil Perpotongan K-Gram dengan data tanpa dilakukan *Text Preprocessing* (Lanjutan)

	'it', 'th', 'hu', .., 'ns', 'sl', 'lt', 'te', 'em', 'mS', 'Se', 'el', 'le', 'ec', 'ct', 'te', 'ed', 'd(', '(i', 'it', 'te', 'em', 'm)', '}\n', '\n ', '}\n', '\n\n', '\n)']	'ith', 'thu', 'hub', .., 'nsl', 'slt', 'ite', 'tem', 'emS', 'mSe', 'Sel', 'ele', 'lec', 'ect', 'cte', 'ted', 'ed(', 'd(i', 'it', 'te', 'tem', 'em)', 'm)\n', ')\n ', '\n ', '}\n\n', '\n\n)']	'plika', 'likas', 'ikasi', 'kasig', 'asigi', 'sigit', 'igith', 'githu', 'ithub', 'thubu', .., 'ption', 'tions', 'ionsl', 'onslt', 'nslte', 'sltem', 'itemS', 'temSe', 'emSel', 'mSele', 'Selec', 'elect', 'lecte', 'ected', 'cted(', 'ted(i', 'ed(it', 'd(ite', '(item', 'item)', 'tem)\n', 'em)\n ', 'm)\n ', '}\n ', '\n ']	'pal.apl', 'al.apli', .., 'mSelect', 'Selecte', 'elected', 'lected(', 'ected(i', 'cted(it', 'ted(ite', 'ed(item', 'd(item)', '(item)\n', 'item)\n ', 'tem)\n ', 'em)\n ', 'm)\n ', '}\n ', '\n ']	'te', 'ex', 'xt', 't ', '= ', 'us', 'se', 'er', 'rs', 's.', '.r', 're', 'ep', 'po', 'os', 'si', 'it', 'to', 'or', 'ry', 'y\n', '\n ', '}\n', '\n}', '}\n']	'ers', 'rsu', .., '.te', 'tex', 'ext', 'xt ', 't =', '= u', 'use', 'ser', 'ers', 'rs.', 's.r', '.re', 'rep', 'epo', 'pos', 'osi', 'sit', 'ito', 'tor', 'ory', 'ry\n', 'y\n ', '\n ', '}\n}', '\n}\n']	'packa', 'ackag', 'ckage', 'kage ', 'age c', 'ge co', 'e com', 'com.e', 'om.ex', 'm.exa', '.exam', 'examp', 'xampl', 'ample', 'mple.', 'ple.g', 'le.g)', .., 'users', 'sers.', 'ers.r', 'rs.re', 's.rep', '.repo', 'repos', 'eposi', 'posit', 'osito', 'sitor', 'itory', 'tory\n', 'ory\n ', 'ry\n ', 'y\n ', '\n ']	'ample.g', 'mple.gi', .., 'rs.repo', 's.repos', '.reposi', 'reposit', 'eposito', 'positor', 'ository', 'sitory\n', 'itory\n ', 'tory\n ', 'ory\n ', 'ry\n ', 'y\n)', '\n)\n']

3.4.5.2. Perhitungan fungsi Hash per k-gram (Rolling Hash)

Setelah mendapatkan hasil dari potongan karakter berdasarkan K-Gram selanjutnya menghitung nilai *hash* menggunakan rumus Rolling Hash, berikut contoh perhitungan Rolling Hash pada salah satu porongan karakter pada sampel 1 dengan data yang sudah dilakukan *text preprocessing* dengan K-gram = 2 pada potongan karakter "ac" dengan nilai basis = 10, sebagai berikut:

$$H_{(ac)} = ascii(a)x\ 10^{(1)} + ascii(c)x\ 11^{(0)}$$
$$= 97\ x\ 10 + 99\ x\ 1$$
$$= 1069$$

Sehingga mendapatkan hasil perubahan dari potongan karakter menjadi *hash* berdasarkan perhitungan Rolling Hash, berikut ditampilan hasil darii perhitungan Rolling hash dengan data yang telah dilakukan *text preprocessing* berdasarjan potongan K-Gram ditampilkan pada Lampiran 1.

Selain itu juga dilakukan perhitungan pada data tanpa dilakukan *text preprocessing*, berikut hasil dari perhitungan *hash* menggunakan Rolling hasil pada data tanpa dilakukan text preprocessing berdasarkan K-Gram yang terbentuk ditampilkan pada Lampiran 2.

3.4.5.3. Pembuatan nilai *Window*

Setelah diketahui *hash* pada sebaran K-Gram langkah selanjutnya yaitu pembuatan *window* dengan jumlah w=3. Sehingga *hash* yang ada akan dikelompokkan kedalam window yang terbentuk. Berikut hasil dari *window* pada dengan data yang sudah dilakukan *text preprocessing* berdasarkan K-Gram yang terbentuk pada Lampiran 3

Berikut ditampilkan hasil dari pembentukan *window* terhadap *hash* yang terbentuk pada proses sebelumnya dengan data tanpa dilakukan *text preprocessing* ditampilkan pada Lampiran 4.

3.4.5.4. Pemilihan *Fingerprint* dari setiap *Window*

Setelah terbentuk *window* pada *hash* dengan data yang telah dilakukan *text preprocessing* dan tanpa dilakukan *text preprocessing* selanjutnya pada setiap *window* akan dipilih *fingerpirnt* yaitu nilai terkecil pada setiap *window* dan jika bertemu dengan nilai yang sama akan diambil paling kanan dan ketika ada nilai yang sama akan dipilih salah satu. Sehingga hasil dari *fingeprint* didapatkan dengan nilai terkecil, berikut penentuan *fingerprint* pada *window* K-Gram pada beberapa data dengan data yang sudah dilakukan *text preprocessing:*

[1217, **1069**, 1097], [**1069**, 1097, 1167], [1097, 1167, **1073**], [1167, **1073**, 1131]

Sehingga *fingerprint* ditemukan sebagai berikut: [1069,1073]

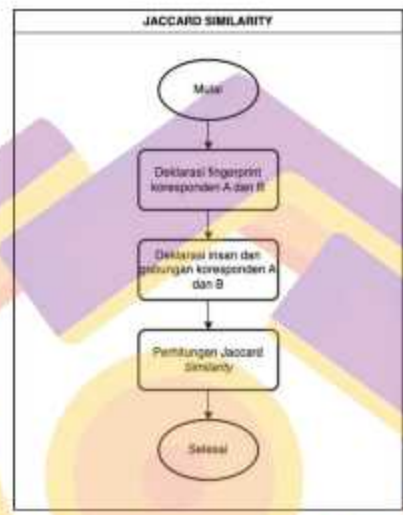
Berikut hasil *fingerprint* pada *window* yang terbentuk dengan data yang sudah dilakukan *Text Preprocessing*, ditampilkan pada Lampiran 5.

Berikut hasil dari *fingerprint* berdasarkan *hash window* yang berasal dari data tanpa dilakukan *text preprocessing*, ditampilkan pada Lampiran 6

Berdasarkan Tabel pada Lampiran 5 dan Lampiran 6 telah didapatkan hasil *fingerprint* untuk setiap sampel *source code* dengan data yang telah dilakukan *text preprocessing* dan tanpa dilakukan *text preprocessing*, sehingga setiap *source code* memiliki *fingeprint* masing-masing.

3.4.6. Jaccard *Similarity*

Berikut detail dari alur penelitian pada tahap menghitung nilai *similarity* menggunakan Jaccard *Similarity*.



Gambar 3.3 Alur Jaccard *Similarity*

Gambar 3.3 menjelaskan bahwa setelah pemrosesan algoritma Winnowing, hasil dari *fingeprint* tersebut dihitung tingkat presentase *similarity*-nya menggunakan koefisien Jaccard *Similarity* untuk membandingkan nilai *fingerprint* satu dokumen dengan dokumen yang lain berdasarkan *fingerprint* yang sama. Dengan urutan dari deklarasi koresponden *fingerprint string* A dan B selanjutnya akan diketahui irisan dan gabungan antara koresponden *fingerprint string* A dan B dan terakhir perhitungan Jaccard *Similarity* dapat di lakukan.

3.4.6.1. Deklarasi *Fingerprint* Koresponden A dan B

Berikut ditampilkan perhitungan untuk mengetahui hasil dari tingkat *similarity* berdasarkan hasil *fingeprint* pada sampel 1 yang telah dihasilkan pada

proses algoritma Winnowing, dengan menggunakan data yang telah dilakukan proses *text preprocessing* dan k-gram dengan nilai $k = 2$.

Total *Fingerprint Source Code* 1: 873

[90000000000000000000000000000000001285, ..,

9000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000000000000005001185]

Total *Fingeprint Source Code* 2: 883

[90000000000000000000000000000000001285,

9000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000000000000005001185]

3.4.6.2. Deklarasi Irisan dan Gabungan Koresponden A dan B

Berdasarkan total *fingerprint* yang ada pada data koresponden A dan B sebelumnya maka diketahui total dari *fingerprint* pada setiap koresponden

selanjutnya maka dari itu akan diketahui *fingerprint* yang sama atau irisan antara koresponden A dan B mendapatkan hasil 871. Sedangkan untuk gabungan koresponden A dan B mendapatkan hasil 885.

3.4.6.3. Perhitungan Jaccard *Similarity*

Setelah mengetahui *fingerprint* disetiap koresponden data dan mengetahui nilai irisan dan gabungan antara kedua koresponden, selanjutnya melakukan perhitungan Jaccard *Similarity* dapat dilakukan sebagai berikut:

$$J(A,B) = \frac{871}{885} x\ 100 = 98{,}41\%$$

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Gambaran Umum Penelitian

Dalam penelitian ini data yang akan diteliti adalah source code yang diambil dari Github dengan tema projek yang sama yaitu 'Pembuatan Aplikasi Github'. Data yang diambil saling berpasang-pasangan untuk nantinya dapat dibandingkan dan memiliki indikasi awal antara duplikat dan tidak dublikat. Setelah dilakukan pengambilan dan selanjutnya data tersebut diproses pada tahap *text preprocessing*. Setelah dilakukan tahap *text preprocessing* selanjutnya diproses pada tahapan Algoritma Winnowing dengan pemilih K-Gram dengan bilangan prima 2, 3, 5, dan 7 dengan nilai window 3. Berdasarkan algoritma Winnowing akan dilakukan tahapan pemrosesan yaitu menentukan nilai K-Gram, Rolling Hash, dan pembuatan *window* dan *fingerprint* untuk proses ini terdapat dua data yang diproses, yaitu data yang sudah melalui tahap *text preprocessing* dan. Setelah terbentuk *fingerprint* pada setiap data maka akan diproses pada perhitung Jaccard *Similarity* untuk mengetahui hasil *similarity* pada data yang diproses menggunakan algoritma Winnowing.

Berdarkan hasil perhitungan tingkat kesamaan kedua data yang dibanding dengan menggunakan algoritma Winnowing, maka akan dibandingkan performa yang dilakukan dengan perhitungan kecepatan waktu dan membandingkan hasil *similarity* pada algoritma Winnowing.

4.2. Pengumpulan Data

Pada penelitian ini menggunakan data yang diambil dari Github dengan tema "Pembuatan Aplikasi Github User" yang mendapatkan 2 aplikasi yang

masing-masing memiliki 5 source code dengan ekstensi ".kt" yang bisa dibandingkan. Data yang akan dibandingkan akan dilabeli secara manual dengan 2 kategori, yaitu kode yang terindikasi mirip dan tidak mirip.

4.3. Text Preprocessing

Proses selanjutnya setelah melakukan pengumpulan data, data yang telah dipersiapkan sebelumnya akan diolah pada proses *text preprocessing*.

4.3.1. Case folding

Langkah pertama pada *text preprocessing*, yaitu case folding untuk menghilangkan karakter yang tidak valid dan mengubah huruf menjadi huruf kecil. Berikut source code yang digunakan untuk pemrosesan *case folding* ditampilkan pada gambar 4.1

Gambar 4.1 *Source code* untuk *Case folding*

Berdasarkan Gambar 4.1 pada tahap *case folding* terdapat beberapa fungsi yang diterapkan, yaitu:

a. *Fungsi* cleaned_text() yang berisikan berbagai tanda baca yang akan dihapus pada *source code*

b. Fungsi re.sub() digunakan dengan ekspresi reguler (r"http\S+") untuk menghapus URL dari teks. Pola regex ini cocok dengan substring apa pun yang dimulai dengan "http" dan diikuti oleh satu atau beberapa karakter bukan spasi.

c. Fungsi lower() dipanggil pada cleaned_text() untuk mengubah semua huruf/ karakter menjadi huruf kecil.

d. Fungsi replace() digunakan untuk menghapus karakter tertentu dari teks, termasuk spasi, karakter baris baru (\n), karakter tab (\t), dan titik (.)

4.3.2. Tokenizing

Tahap berikutnya adalah tokenizing yang berfungsi untuk memecah urutan karakter atau string teks menjadi unit yang lebih kecil yang biasa disebut token. Setiap token mewakili unit teks yang bermakna, seperti kata, simbol, atau tanda baca. Tokenizing merupakan langkah mendasar dalam pemrosesan bahasa alami dan pemrosesan bahasa pemrograman. Source code dari tokenizing ditampilkan pada gambar 4.2

Gambar 4.2 *Source Code* untuk *Tokenizing*

Kode pada Gambar 4.2 digunakan untuk membuat aturan untuk menandai kode Kotlin dan menetapkan jenis token yang sesuai untuk penyorotan sintaks atau tujuan pemrosesan kode lainnya. Kamus token yang digunakan mendefinisikan berbagai kategori token dan ekspresi regulernya yang sesuai.

4.4. Data Uji dan Data Validasi

Data akan memiliki dua varian dari tahap *text preprocessing* yaitu *source code* tanpa proses *text preprocessing* dan *source code* yang sudah dilakukan *text preprocessing*. data validasi ini adalah data yang sebelumnya dipilih secara manual dengan kategori yaitu kode yang sama persis, kode yang mendekati sama, dan kode yang memiliki perbedaan yang jauh. Data uji dan data validasi akan diproses ke dalam algoritma *similarity* untuk mengetahui tingkat *similarity* dan akan dinilai akurasinya berdasarkan kategori yang ditentukan

4.5. Algoritma Winnowing

Tahapan proses algoritma Winnowing yaitu menentukan nilai KGram, Rolling Hash, pembuatan window serta terakhir menentukan *fingerprint*.

4.5.1. Menentukan k-gram

Pada penelitian ini digunakan nilai k yang pada penelitian ini akan menggunakan bilangan prima yaitu 2, 3, 5, dan 7. Nilai k tersebut selanjutnya digunakan untuk memecah rangkaian karakter yang berasal dari data yang disajikan. Pemilihan bilangan tersebut dikarenakan *source code* memiliki teks yang pendek, sehingga nilai k yang kecil akan menangkap pola dengan tepat.

Berikut merupakan *source code* yang digunakan untuk memotong karakter sejumlah k yang ditampilkan pada gambar 4.3

```
text = str(text)
k_grams = []
for start in range(len(text) - self.k + 1):
    # only perform the work necessary if the current character is not a whitespace
    if text[start] != " ":
        end = self.k
        k_gram = text[start:end]
        # k_gram = self.clean_text(k_gram)
        # after cleaning up our k-gram, we can end up with a much smaller string that is below the denoted k-gram length
        # if this is the case, then we will continually add the missing length to the string we are slicing
        while(len(k_gram) < self.k):
            end = end + (self.k - len(k_gram)) # we will need to add as many letters as are missing from the k-gram
            k_gram = text[start: end]
            # k_gram = self.clean_text(k_gram)
        hashed_k_gram=self.custom_hash(k_gram)
        k_grams.append(hashed_k_gram)
        # print("K-gram:", k_gram)
        # print("Hash:", hashed_k_gram)
return k_grams
```

Gambar 4.3 *Source code* untuk Menentukan K-Gram

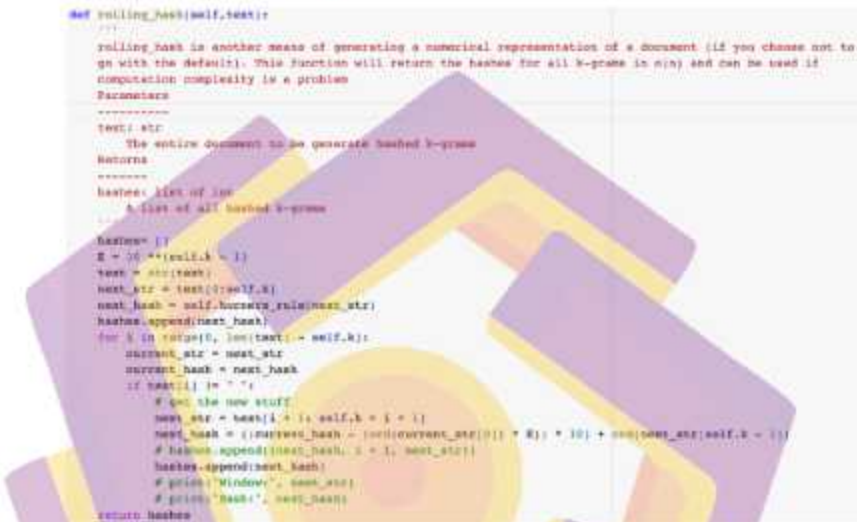
Berdasarkan Gambar 4.3 dilakukan pemotongan karakter berdasarkan masukan nilai *k*. Pada tahap ini membutuhkan sepotong teks dan bilangan bulat k yang mewakili berapa lama setiap K-gram. Hal tersebut akan menampilkan berbagai k-gram serta lokasinya di teks asli.

Contoh masukan: "helloworld", 5

Contoh Keluaran: [("hello",0) , ("ellow", 1), ("llowo",2), ("lowor",3), ("oworl",4), ("world",5)]

4.5.2. Rolling hash

Setelah dilakukan pemotongan karakter selanjutnya dilakukan proses *hashing* menggunakan Rolling Hash.



Gambar 4.4 *Source code hashing* menggunakan Rolling hash

Dalam Gambar 4.4, Nilai hash mengacu pada representasi numerik atau kode hash yang dihasilkan untuk setiap k-gram dalam dokumen menggunakan algoritma rolling hash. Nilai hash adalah representasi numerik ukuran tetap yang dihasilkan dari input data, dalam hal ini, k-gram yang diekstraksi dari dokumen. Tujuan hashing adalah untuk mengubah data arbitrer, seperti string atau objek lain, menjadi nilai numerik yang dapat digunakan secara efisien untuk berbagai operasi, seperti pengindeksan, perbandingan, atau identifikasi.

4.5.3. Pembuatan window

Setelah diketahui nilai *hash* pada setiap gram yang terbentuk selanjutkan memasukkan nilai hash tersebut ke dalam sebuah *window* sehingga akan membentuk kelompok membentuk *window* masing-masing sesuai urutannya. Pada langkah ini juga menentukan nilai terkecil pada setiap *window*. Tahap ini ditampilkan pada Gambar 4.5

```
def select_fingerprints(self, hash_list, w):
    '''
    this helper function takes in a  set of
    Parameters
    ----------
    hash_list : list of (int, int)

    w : int
        w represents the window size for which we select a rightmost minimum

    Returns
    -------
    fingerprints : list of int
    '''
    fingerprints = []
    min_index = -1
    prev_min_index = -1
    # traverse over the hash_list
    for hash_index in range(len(hash_list) - w +1):
        min_value =  float("inf")
        #traverse over each window
        for window_index in range(hash_index, hash_index + w ):
            if hash_list[window_index] <= min_value:
                min_index = window_index
                min_value = hash_list[window_index];
        # If the minimum value of the previous window is no longer the minimum value
        if min_index != prev_min_index:
          prev_min_index = min_index
          fingerprints.append(hash_list[min_index])
```

Gambar 4.5 *Source code* untuk pembuatan *window*

Pada tahap ini hash list digunakan menggunakan hash index pada setiap jendela berukuran w, nilai minimum paling kanan dipilih. Minimum paling kanan ditentukan dengan mengulang elemen di dalam *window* dan menemukan nilai terkecil. Indeks nilai minimum disimpan pada variabel min_index, dan jika nilai minimum berubah dibandingkan dengan jendela sebelumnya, ini dianggap sebagai

fingerprint baru. *Fingerprint* yang sesuai dengan nilai minimum pada *window*, dilampirkan ke daftar *fingerprint*.

4.5.4. Fingerprint

Pada langkah ini digunakan untuk menghasilkan *fingerprint* dari *source code* yang diberikan. Langkah ini bisa dilihat pada Gambar 4.6

```
def generate_fingerprints(doc_path):
    '''
    Given the path of a text file, this function generates fingerprints for the contents of the file.
    Parameters:
    -----------
    doc_path : str
        path to the text file
    Returns:
    --------
    fingerprints : list of str
        list of fingerprints for the given document
    '''
    with open(doc_path) as document:
        contents = document.read()
        # print(contents)
        # create a new instance of the WinnowedDoc class with the updated constructor
        winnowed_doc = WinnowedDoc(contents, k_gram_size, window_size, True)


        # call the fingerprints method on the new instance
        fingerprints = winnowed_doc.fingerprints
    return fingerprints
```

Gambar 4.6 *Source code* untuk tahap *Fingerprint*

Pada tahap ini akan memproses semua proses yang ada pada fungsi winnowed_doc sehingga membentuk *fingerprint* yang nantinya akan dibandingkan menggunakan Jaccard *Similarity*.

4.6. Jaccard Similarity

Setelah menemukan nilai *fingerprint* dari sebaran nilai *k* yang berbeda selanjutnya akan dilakukan pencocokan nilai *fingeprint* yang sama untuk mendapatkan nilai *similarity*.

4.6.1. Deklarasi fingerprint A dan B

Langkah pertama pada proses Jaccard *Similarity* yaitu mendeklarasikan 2 parameter yang akan diproses, yaitu kedua *fingeprint* dari kode yang akan dibandingkan. Berikut merupakan *source code* yang digunakan untuk melakukan proses Jaccard *Similarity*

```
def jaccard_coefficient(doc1_fp, doc2_fp):
    '''
    Helper function that computes how similar 2 arrays of fingerprints are.
    Parameters
    ----------
    doc1_fp : list of str
        list of fingerprints for 1 document
    doc2_fp : list of str
        list of fingerprints for the second document
    Returns
    -------
    similarity : float
        Similarity measure of the 2 documents
    '''
    doc1_fp = set(doc1_fp)
    doc2_fp = set(doc2_fp)
```

Gambar 4.7 *Source code* untuk proses Jaccard *Similarity*

Pada Gambar 4.7 terdapat fungsi set() yang berfungsi untuk mengubah *fingeprint* menjadi set yang akan menghilangkan *fingeprint* yang terduplikat dan memungkinkan operasi set yang efisien.

4.6.2. Perhitungan Jaccard Similarity

Langkah terakhir adalah menghitung *similarity* dengan rumus Jaccard *Similarity*. *Source code* perhitungan tersebut ditampilkan pada Gambar 4.8

```
def jaccard_coefficient(doc1_fp, doc2_fp):
    '''
    Helper function that computes how similar 2 arrays of fingerprints are.
    Parameters
    ----------
    doc1_fp : list of str
        list of fingerprints for 1 document
    doc2_fp : list of str
        list of fingerprints for the second document
    Returns
    -------
    similarity : float
        Similarity measure of the 2 documents
    '''
    doc1_fp = set(doc1_fp)
    doc2_fp = set(doc2_fp)

    intersection = doc1_fp.intersection(doc2_fp)
    union = doc1_fp.union(doc2_fp)
    return (len(intersection) / len(union)) * 100
```

Gambar 4.8 *Source code* untuk Perhitungan Jaccard *Similarity*

Variabel *intersection* digunakan untuk memberikan hasil dari operasi *intersection* antara doc1_fp dan doc2_fp, operasi tersebut mengembalikan satu set yang berisi elemen umum antara dua set. Variabel union ditetapkan sebagai hasil dari operasi gabungan antara doc1_fp dan doc2_fp, operasi ini mengembalikan satu set yang berisi semua elemen unik dari kedua set. Jaccard *Similarity* dihitung dengan membagi panjang himpunan irisan dengan panjang himpunan gabungan yang mewakili rasio sidik jari umum dengan total *fingerprint* yang unik.

4.7. Uji Coba dan Evaluasi

4.7.1. Lingkungan Uji Coba

Lingkungan uji coba pada penelitian ini berupa perangkat keras dan perangkat lunak yang digunakan untuk melakukan uji coba penerapan algoritma winnowing untuk mendeteksi plagiarisme pada *source code*. Lingkungan uji coba ini berupa laptop dengan spesifikasi perangkat keras yang digunakan sebagai berikut:

Processor : 1,4 GHz Quad-Core Intel Core i5
Memory (RAM) : 8 GB
Graphics : Intel Iris Plus Graphics 645 1536 MB

Perangkat lunak yang digunakan sebagai berikut:

Sistem Operasi : MacOS Monterey Version 12.3.1
Pengembang : Google Colaboratory
Bahasa Pemrograman : Phyton

4.7.2. Skenario Uji Coba

Terdapat beberapa skenario uji coba yang dilakukan diantaranya:

a. Perhitungan tingkat *similarity* dan waktu pemrosesan berdasarkan algoritma Winnowing dengan data berupa hasil dari *text preprocessing*. Data masukkan nilai *k* pada K-Gram berupa nilai bilangan prima 2, 3, 5, dan 7. Menggunakan nilai window bilangan prima 2 serta menggunakan nilai basis = 10. Data yang di uji menggunakan data hasil dari *text preprocessing*.

b. Perhitungan tingkat *similarity* dan waktu pemrosesan berdasarkan algoritma Winnowing dengan data tanpa dilakukan *text preprocessing*. Data

masukkan nilai *k* pada K-Gram berupa nilai bilangan prima 2, 3, 5, dan 7. Menggunakan nilai window bilangan prima 2 serta menggunakan nilai basis = 10. Data yang di uji menggunakan data hasil tanpa dilakukan *text preprocessing*.

Hasil dari kedua skenario selanjutnya akan dilakukan perbandingan hasil tingkat *similarity* dan waktu pemrosesan berdasarkan algoritma Winnowing dengan data yang sudah dilakukan *text preprocessing* dan tanpa dilakukan *text preprocessing*.

4.7.3. Pembagian Data

Pada penelitian ini menggunakan data yang sudah dikategorikan secara manual, untuk kategori yang digunakan adalah kategori, yaitu kode yang terindikasi mirip dan tidak mirip. Masing-masing kategori tersebut akan melalui skenario uji coba yang akan dilakukan

4.7.4. Skenario 1 Perhitungan Tingkat *Similarity* dan Waktu Pemrosesan Algoritma Winnowing dengan Data yang sudah dilakukan *Text Preprocessing*

Pada skenario pengujian ini dilakukan perhitungan tingkat *similarity* dan waktu pemrosesan dari berbagai macam perbandingan yaitu menggunakan sample kode yang memiliki kategori mirip dengan tidak mirip dengan masing-masing 10 sample. Pemakaian hasil berdasarkan dari penggunaan *text preprocessing* dan tanpa *text preprocessing*. Serta menggunakan nilai *k* pada K-Gram yang bervariasi yaitu 2, 3, 5, dan 7.

Berikut hasil dari rata-rata perhitungan *similarity* dan waktu pemrosesan menggunakan algoritma Winnowing dari data dengan yang sudah melalui *text preprocessing* ditampilkan pada Tabel 4.1

Tabel 4.1 Hasil *Similarity* dan Waktu Pemrosesan Algoritma Winnowing dengan Data yang sudah dilakukan *Text Preprocessing* pada indikasi Tidak Mirip

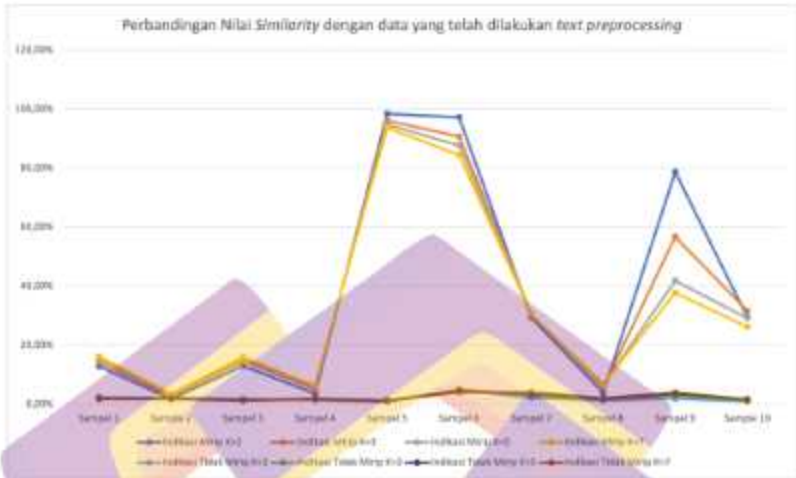
Sample	Persentase Similarity				Waktu pemrosesan			
	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7
Sample 1	2,33%	2,05%	1,86%	1,77%	9s	13s	11s	11s
Sample 2	2,09%	1,97%	1,88%	1,80%	7s	12s	13s	12s
Sample 3	1,77%	1,41%	1,26%	1,25%	9s	10s	11s	10s
Sample 4	1,44%	1,69%	1,76%	1,85%	11s	12s	8s	21s
Sample 5	0,77%	1,08%	1,15%	1,26%	11s	16s	8s	10s
Sample 6	5,02%	4,97%	4,25%	4,11%	9s	10s	9s	8s
Sample 7	2,14%	3,13%	3,64%	3,90%	12s	10s	11s	21s
Sample 8	1,27%	1,70%	1,80%	2,03%	11s	14s	10s	12s
Sample 9	1,82%	2,93%	3,51%	3,89%	7s	11s	10s	14s
Sample 10	0,93%	1,42%	1,57%	1,63%	8s	10s	11s	9s

Berdasarkan Tabel 4.1 dengan indikasi tidak mirip maka nilai hasil *similarity* terkecil merupakan nilai paling baik, berdasarkan penggunaan data dari 10 sample yang telah melalui proses *text preprocessing* diketahui bahwa nilai $k = 2$ pada source code memiliki nilai rata-rata *similarity* 19,58%. Sedangkan untuk pemrosesan waktu dengan pemrosesan tercepat didapatkan pada pemrosesan dengan nilai $k = 2$ dan 5 dan memiliki rata-rata waktu pemrosesan tercepat yang berdekatan yaitu 9,4 detik dan 10,2 detik. Berikut hasil dari rata-rata perhitungan *similarity* dan waktu pemrosesan menggunakan algoritma Winnowing dari data dengan indikasi mirip pada Tabel 4.2

Tabel 4.2 Hasil Similarity dan Waktu Pemrosesan Algoritma Winnowing dengan Data yang sudah dilakukan Text Preprocessing pada indikasi Mirip

Sample	Persentase *Similarity*				Waktu pemrosesan			
	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7
Sample 1	12,81%	14,25%	15,51%	16,21%	12s	10s	11s	30s
Sample 2	1,75%	2,66%	3,10%	3,51%	11s	9s	11s	11s
Sample 3	13,01%	14,72%	15,46%	15,91%	9s	10s	9s	8s
Sample 4	3,52%	4,98%	5,81%	6,25%	11s	9s	7s	9s
Sample 5	98,41%	96,06%	94,23%	93,64%	8s	9s	12s	13s
Sample 6	97,17%	90,70%	87,67%	84,44%	19s	9s	10s	11s
Sample 7	29,42%	30,48%	30,36%	30,50%	12s	11s	11s	10s
Sample 8	3,81%	5,69%	6,48%	6,81%	7s	10s	8s	9s
Sample 9	78,70%	56,73%	41,83%	37,73%	11s	11s	9s	14s
Sample 10	29,34%	31,59 %	29,09%	26,16%	11s	10s	10s	11s

Berdasarkan Tabel 4.2 dengan indikasi tidak mirip maka nilai hasil *similarity* terbesar merupakan nilai paling baik, berdasarkan penggunaan data dari 10 sample yang telah melalui proses *text preprocessing* diketahui bahwa nilai k = 7 pada source code memiliki nilai rata-rata *similarity* 367,94%. Sedangkan untuk pemrosesan waktu dengan pemrosesan tercepat didapatkan pada pemrosesan dengan nilai k=3 dan 5 dan memiliki rata-rata waktu pemrosesan tercepat yang sama yaitu 9,8 detik. Berikut grafik perbandingan hasil *similarity* menggunakan data yang sudah dilakukan *text preprocessing* dengan indikasi mirip dan tidak mirip ditampilkan pada Gambar 4.9



Gambar 4.9 Grafik Perbandingan Nilai *Similarity* dengan data yang telah dilakukan *text preprocessing*

Grafik pada Gambar 4.9 menampilkan bahwa perbandingan nilai *similarity* antara tabel 4.1 dan 4.2 pada pemrosesan dengan data yang telah dilakukan *text preprocessing* mendapatkan hasil bahwa untuk rata-rata nilai *similarity* dengan indikasi mirip memiliki nilai *similarity* diatas 3% sedangkan nilai *similarity* dengan indikasi tidak mirip memiliki nilai *similarity* dibawah 1%.

4.7.5. Skenario 2 Perhitungan Tingkat *Similarity* dan Waktu Pemrosesan Algoritma Winnowing dengan Data tanpa *Text Preprocessing*

Pada skenario pengujian ini dilakukan perhitungan tingkat *similarity* dan waktu pemrosesan dari berbagai macam perbandingan yaitu menggunakan sample kode yang memiliki kategori mirip dengan tidak mirip dengan masing-masing 10 sample. Pemakaian hasil berdasarkan dari penggunaan tanpa *text preprocessing*. Serta menggunakan nilai *k* pada K-Gram yang bervariasi yaitu 2, 3, 5, dan 7.

Tabel 4.3 Hasil Similarity dan Waktu Pemrosesan Algoritma Winnowing dengan Data tanpa Text Preprocessing pada indikasi Tidak Mirip

Sample	Persentase *Similarity*				Waktu pemrosesan			
	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7
Sample 1	0,54%	0,43%	0,43%	0,22%	6s	9s	12s	13s
Sample 2	0,48%	0,38%	0,39%	0,9%	20s	8s	8s	7s
Sample 3	0,65%	0,52%	0,52%	0,26%	9s	10s	8s	8s
Sample 4	0,24%	0,20%	0,2%	0,10%	19s	14s	12s	17s
Sample 5	0,2%	0,11%	0,11%	0,06%	17s	10s	17s	7s
Sample 6	0,67%	0,50%	0,34%	0,17%	8s	11s	12s	10s
Sample 7	0,50%	0,46%	0,38%	0,31%	10s	11s	9s	7s
Sample 8	0,50%	0,46%	0,38%	0,31%	10s	7s	8s	11s
Sample 9	2,08%	1,91%	0,17%	0,08%	6s	6s	9s	9s
Sample 10	0,04%	1,91%	0,08%	0,04%	10s	11s	10s	10s

Berdasarkan Tabel 4.3 dengan indikasi tidak mirip maka nilai hasil *similarity* terkecil merupakan nilai paling baik, berdasarkan penggunaan data dari 10 sample yang telah melalui proses *text preprocessing* diketahui bahwa nilai $k = 7$ pada source code memiliki nilai rata-rata *similarity* 2,45%. Sedangkan untuk pemrosesan waktu dengan pemrosesan tercepat didapatkan pada pemrosesan dengan nilai $k = 5$ dan 7 dan memiliki rata-rata waktu pemrosesan tercepat yang berdekatan yaitu 9,7 dan 9,9 detik. Berikut hasil dari rata-rata perhitungan *similarity* dan waktu pemrosesan menggunakan algoritma Winnowing dari data dengan indikasi mirip pada Tabel 4.4

Tabel 4.4 Hasil Similarity dan Waktu Pemrosesan Algoritma Winnowing dengan Data tanpa Text Preprocessing pada indikasi Mirip

Sample	Persentase *Similarity*				Waktu pemrosesan			
	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7
Sample 1	0,69%	0,64%	0,53%	0,43%	9s	10s	20s	10s
Sample 2	0,15%	0,12%	0,12%	0.06%	11s	7s	6s	12s
Sample 3	0,82%	0,76%	0,63%	0,50%	10s	15s	8s	10s
Sample 4	0,28%	0,22%	0,22%	0,11%	7s	7s	8s	8s
Sample 5	0,35%	0,28%	0,28%	0,14%	14s	6s	7s	7s
Sample 6	0,21%	0,41%	0,41%	0,21%	8s	6s	8s	8s
Sample 7	0,25%	0,20%	0,20%	0,10%	8s	8s	9s	8s
Sample 8	0,27%	0,22%	0,22%	0,11%	10s	12s	26s	7s
Sample 9	0,54%	0,44%	0,44%	0,22%	8s	9s	7s	8s
Sample 10	0,45%	0,36%	0,36%	0,18%	7s	7s	8s	8s

Berdasarkan Tabel 4.3 dengan indikasi mirip maka nilai hasil *similarity* terbesar merupakan nilai paling baik, berdasarkan penggunaan data dari 10 sample yang telah melalui proses *text preprocessing* diketahui bahwa nilai $k = 2$ pada source code memiliki nilai rata-rata *similarity* 4,01%. Sedangkan untuk pemrosesan waktu dengan pemrosesan tercepat didapatkan pada pemrosesan dengan nilai $k = 7$ dan memiliki rata-rata waktu pemrosesan tercepat yaitu 8,6 detik. Berikut grafik perbandingan hasil *similarity* menggunakan data tanpa *text preprocessing* dengan indikasi mirip dan tidak mirip ditampilkan pada Gambar 4.10

Gambar 4.10 Grafik Perbandingan Nilai *Similarity* dengan data tanpa *text preprocessing*

Grafik pada Gambar 4.10 menampilkan bahwa perbandingan nilai *similarity* antara tabel 4.3 dan 4.4 pada pemrosesan dengan data tanpa dilakukan *text preprocessing* mendapatkan hasil bahwa untuk rata-rata nilai *similarity* dengan indikasi mirip dan tidak mirip memiliki nilai dibawah 1%.

4.7.6. Hasil Perbandingan Hasil Tingkat *Similarity* dan Waktu Pemrosesan berdasarkan Data dengan *Text Preprocessing* dan Tanpa *Text Preprocessing*

Pada skenario pengujian ini dilakukan perbandingan dari hasil tingkat *similarity* dan waktu pemrosesan dari berbagai macam perbandingan seperti pemakaian variasi data source code dengan 10 sample dengan emakaian hasil *text preprocessing* dan tanpa *text preprocessing* menggunakan algoritma Winnowing.

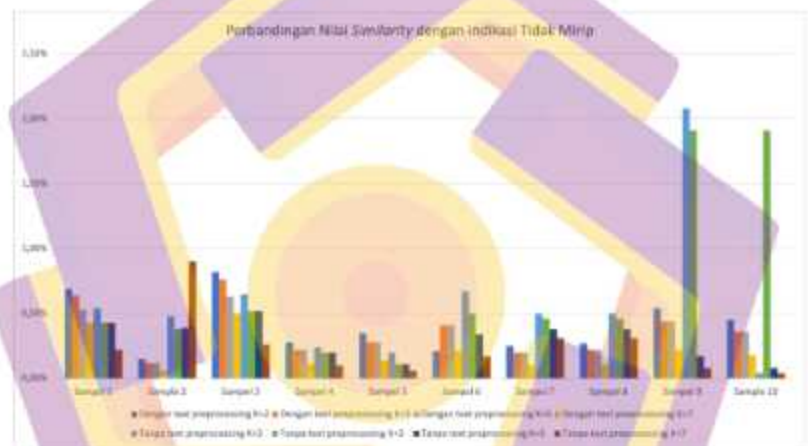
Berikut perbandingan hasil 10 sample dari perhitungan *similarity dan* waktu pemrosesan menggunakan algoritma Winnowing dari data dengan indikasi tidak mirip pada *text preprocessing* dan tanpa *text preprocessing* ditampilkan pada tabel 4.5

Tabel 4.5 Hasil *Similarity* dan Waktu Pemrosesan berdasarkan Algoritma Winnowing Berdasarkan *Text Preprocessing* dan tanpa *Text Preprocessing* pada indikasi Tidak Mirip

Sample	Persentase Similarity dengan text preprocessing				Waktu pemrosesan dengan text preprocessing				Persentase Similarity tanpa text preprocessing				Waktu pemrosesan tanpa text preprocessing			
	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7
Sample 1	2,33%	2,05%	1,86%	1,77%	9s	13s	11s	11s	0,54%	0,43%	0,43%	0,22%	6s	9s	12s	13s
Sample 2	2,09%	1,97%	1,88%	1,80%	7s	12s	13s	12s	0,48%	0,38%	0,39%	0,9%	20s	8s	8s	7s
Sample 3	1,77%	1,41%	1,26%	1,25%	9s	10s	11s	10s	0,65%	0,52%	0,52%	0,26%	9s	10s	8s	8s
Sample 4	1,44%	1,69%	1,76%	1,85%	11s	12s	8s	21s	0,24%	0,20%	0,2%	0,10%	19s	14s	12s	17s
Sample 5	0,77%	1,08%	1,15%	1,26%	11s	16s	8s	10s	0,2%	0,11%	0,11%	0,06%	17s	10s	17s	7s
Sample 6	5,02%	4,97%	4,25%	4,11%	9s	10s	9s	8s	0,67%	0,50%	0,34%	0,17%	8s	11s	12s	10s
Sample 7	2,14%	3,13%	3,64%	3,90%	12s	10s	11s	21s	0,50%	0,46%	0,38%	0,31%	10s	11s	9s	7s
Sample 8	1,27%	1,70%	1,80%	2,03%	11s	14s	10s	12s	0,50%	0,46%	0,38%	0,31%	10s	7s	8s	11s
Sample 9	1,82%	2,93%	3,51%	3,89%	7s	11s	10s	14s	2,08%	1,91%	0,17%	0,08%	6s	6s	9s	9s
Sample 10	0,93%	1,42%	1,57%	1,63%	8s	10s	11s	9s	0,04%	1,91%	0,08%	0,04%	10s	11s	10s	10s

Terdapat hasil perbandingan dari hasil nilai *similarity* terkecil pada data indikasi tidak mirip dengan data yang berdasarkan pemrosesan tanpa *text preprocessing* yaitu didapatkan oleh pemrosesan dari algoritma Winnowing dengan nilai k = 7 dengan nilai *similarity* yaitu 2,45%. Serta mendapatkan nilai *similarity* terkecil pada data indikasi tidak mirip dengan data yang berdasarkan pemrosesan

dengan *text* preprocessing yaitu didapatkan oleh pemrosesan dari algoritma Winnowing dengan nilai k = 2 dengan nilai *similarity* yaitu 19,58%. Sedangkan untuk waktu pemrosesan tercepat didapatkan pada algoritma Winnowing dengan *text preprocessing* dengan nilai *k* = 2 dengan waktu pemrosesan 9,4 detik. Berikut grafik perbandingan hasil *similarity* menggunakan kedua data dengan indikasi tidak mirip ditampilkan pada Gambar 4.11



Gambar 4.11 Grafik Perbandingan Nilai *Similarity* dengan Indikasi Tidak Mirip

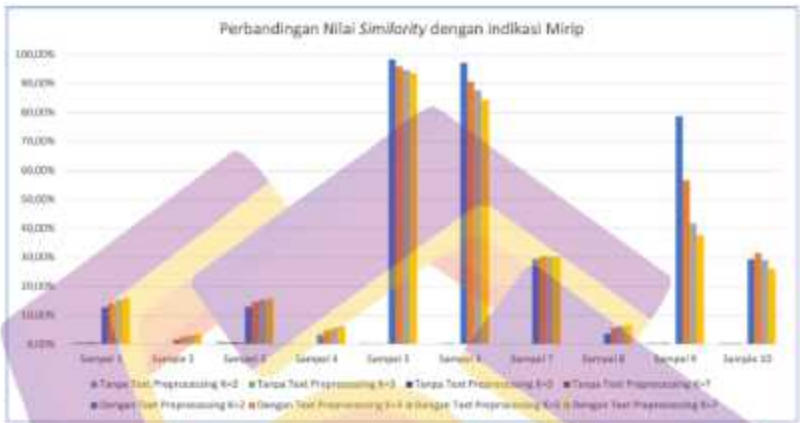
Grafik pada Gambar 4.11 menampilkan bahwa perbandingan nilai *similarity* antara tabel 4.5 dengan kedua data, yaitu data yang dilakukan *text preprocessing* dan tanpa *text preprocessing* dengan indikasi tidak mirip mendapatkan hasil bahwa untuk rata-rata nilai *similarity* adalah dibawah 2%. Berikut merupakan hasil nilai *similarity* dan waktu pemrosesan menggunakan algoritma Winnowing dengan *text preprocessing* dan tanpa *text preprocessing* dengan indikasi mirip ditampilkan pada Tabel 4.6.

Tabel 4.6 Hasil Similarity dan Waktu Pemrosesan berdasarkan Algoritma Winnowing Berdasarkan Text Preprocessing dan tanpa Text Preprocessing pada indikasi Mirip

Sample	Persentase *Similarity* dengan *text preprocessing*				Waktu pemrosesan dengan *text preprocessing*				Persentase *Similarity* tanpa *text preprocessing*				Waktu pemrosesan tanpa *text preprocessing*			
	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7	k=2	k=3	k=5	k=7
Sample 1	12,81 %	14,25 %	15,51 %	16,21 %	12s	10s	11s	30s	0,69%	0,64%	0,53%	0,43%	9s	10s	20s	10s
Sample 2	1,75%	2,66%	3,10%	3,51%	11s	9s	11s	11s	0,15%	0,12%	0,12%	0,06%	11s	7s	6s	12s
Sample 3	13,01 %	14,72 %	15,46 %	15,91 %	9s	10s	9s	8s	0,82%	0,76%	0,63%	0,50%	10s	15s	8s	10s
Sample 4	3,52%	4,98%	5,81%	6,25%	11s	9s	7s	9s	0,28%	0,22%	0,22%	0,11%	7s	7s	8s	8s
Sample 5	98,41 %	96,06 %	94,73 %	93,64 %	8s	9s	12s	13s	0,35%	0,28%	0,28%	0,14%	14s	6s	7s	7s
Sample 6	97,17 %	90,70 %	87,67 %	84,44 %	19s	9s	10s	11s	0,21%	0,41%	0,41%	0,21%	8s	6s	8s	8s
Sample 7	29,42 %	30,48 %	30,36 %	30,50 %	12s	11s	11s	10s	0,25%	0,20%	0,20%	0,10%	8s	8s	9s	8s
Sample 8	3,81%	5,69%	6,48%	6,81%	7s	10s	8s	9s	0,27%	0,22%	0,22%	0,11%	10s	12s	26s	7s
Sample 9	78,70 %	56,73 %	41,83 %	37,73 %	11s	11s	9s	14s	0,54%	0,44%	0,44%	0,22%	8s	9s	7s	8s
Sample 10	29,34 %	31,59 %	29,09 %	26,16 %	11s	10s	10s	11s	0,45%	0,36%	0,36%	0,18%	7s	7s	8s	8s

Terdapat hasil perbandingan dari hasil nilai *similarity* terbesar pada data indikasi mirip dengan data yang berdasarkan pemrosesan tanpa *text preprocessing* yaitu didapatkan oleh pemrosesan dari algoritma Winnowing dengan nilai k = 2 dengan nilai *similarity* yaitu 4,01%. Serta mendapatkan nilai *similarity* terbesar pada data indikasi mirip dengan data yang berdasarkan pemrosesan dengan *text* preprocessing yaitu didapatkan oleh pemrosesan dari algoritma Winnowing dengan nilai k = 2 dengan nilai *similarity* yaitu 367,94. Sedangkan untuk waktu pemrosesan tercepat didapatkan pada algoritma Winnowing tanpa *text preprocessing* nilai *k* = 7

dengan waktu pemrosesan 8,6 detik. Berikut grafik perbandingan hasil *similarity* menggunakan kedua data dengan indikasi mirip ditampilkan pada Gambar 4.12



Gambar 4.12 Grafik Perbandingan Nilai *Similarity* dengan Indikasi Mirip

Grafik pada Gambar 4.12 menampilkan bahwa perbandingan nilai *similarity* antara tabel 4.6 dengan kedua data, yaitu data yang dilakukan *text preprocessing* dan tanpa *text preprocessing* dengan indikasi tidak mirip mendapatkan hasil untuk data yang sudah dilakukan *text preprocessing* memiliki hasil nilai *similarity* yang cukup tinggi dengan rata-rata diatas 3% sedangkan untuk nilai *similarity* tanpa dilakukan *text preprocessing* memiliki rata-rata dibawah 3%.

4.7.7. Evaluasi Skenario 1

Berdasarkan skenario 1 yaitu menggunakan algoritma Winnowing sebagai pendeteksi *similarity* yaitu dengan menggunakan 10 sample yang berdasarkan dengan data dengan pemrosesan *text preprocessing* dengan 2 indikasi, yaitu indikasi tidak mirip dan mirip.

Didapatkan hasil pada rata-rata berdasarkan data dengan pemrosesan *text preprocessing* pada indikasi tidak mirip pada Tabel 4.1 bahwa penerapan nilai *k* yang bervariasi dari nilai *k* = 2, 3, 5 dan 7. Rata-rata waktu pemrosesan pada skenario 1 memiliki waktu yang cukup baik dan tercepat pada nilai *k*=2 dan 3 dan menghasilkan pengaruh nilai *k* terhadap perbedaan hasil rata-rata *similarity* yaitu jika nilai *k* semakin kecil maka nilai *similarity* semakin kecil, hal itu dibuktikan pada hasil rata-rata dari nilai *k*, *k*=2 memiliki rata-rata nilai similarity terendah yaitu 19,58%. Untuk penggunaan data dengan indikasi tidak mirip, semakin kecil nilainya akan semakin baik sehingga sesuai dengan indikasi awal. Nilai similarity yang didapatkan pada 10 sample yang ada termasuk nilai similarity yang cukup rendah karena nilai *similiarity* tidak mencapai 50% sehingga dapat disimpulkan bahwa lebih dari 90% code yang dibandingkan tidak memiliki indikasi mirip.

Selanjutnya hasil pada rata-rata berdasarkan data dengan pemrosesan *text preprocessing* pada indikasi mirip pada Tabel 4.2 bahwa penerapan nilai k yang bervariasi dari nilai *k* = 2, 3, 5, dan 7 menghasilkan bahwa pengaruh terhadap pengaruh nilai *k* terhadap perbedaan hasil rata-rata *similarity* yaitu semakin tinggi nilai *k* maka nilai *similarity* yang dihasilkan semakin tinggi, hal itu dibuktikan pada nilai *k* = 7 mendapatkan rata-rata nilai *similarity* 367,94%. Untuk penggunaaan data indikasi mirip, semakin besar nilai *similarity* nya akan semakin baik sehingga sesuai dengan indikasi awal. Pada 10 sample yang diujikan terdapat beberapa nilai *similarity* yang cukup tinggi yaitu diatas 70%. Hal itu membuktikan bahwa Algoritma Winnowing dengan data yang sudah melalui *text preprocessing* dapat melakukan pendeteksian plagiarisme pada source code. Penerapan *text*

preprocessing untuk mendeteksi kesamaan dalam lebih cocok bila diterapkan proses *stopword removal* karena kombinasi mereka dapat membuat kata-kata unik yang dihasilkan dari *text preprocessing* untuk membantu proses pencocokan ketika melibatkan perubahan posisi (Budiman & Widjaja, 2020).

Terdapat juga beberapa sampel yang memiliki nilai *similarity* rendah pada sampel yang terindikasi mirip hal itu dikarenakan algoritma Winnowing kurang bisa mendeteksi source code yang mirip namun memiliki variabel yang namanya berbeda sehingga algoritma Winnowing bisa melakukan pendeteksian source code dengan baik apabila kode yang dibandingkan memiliki fungsi atau nama yang sama. Algoritma winnowing bisa menemukan apabila terdapat kata yang hanya dirubah posisinya karena dapat menemukan nilai fingerprint yang sama dalam teks dokumen meskipun telah dilakukan perubahan posisi kata dalam teks, namun nilai *similarity* yang diberikan mengalami penurunan disebabkan karena beberapa terdapat beberapa perubahan nama variabel sehingga fingerprint yang ditemukan bukan fingerprint dengan nilai terkecil dalam suatu window (Ridho, 2013)

4.7.8. Evaluasi Skenario 2

Berdasarkan skenario 2 yaitu menggunakan algoritma Winnowing sebagai pendeteksi *similarity* yaitu dengan menggunakan 10 sample yang berdasarkan dengan data tanpa *text preprocessing* dengan 2 indikasi, yaitu indikasi tidak mirip dan mirip.

Didapatkan hasil pada rata-rata berdasarkan data tanpa *text preprocessing* pada indikasi tidak mirip pada Tabel 4.3 bahwa penerapan nilai k yang bervariasi dari nilai k = 2, 3, 5 dan 7 menghasilkan bahwa rata-rata waktu pemrosesan pada

skenario 2 memiliki waktu yang cukup baik dan tercepat pada nilai k=7 dan pengaruh nilai k terhadap perbedaan hasil rata-rata *similarity* yaitu penggunaan nilai k pada kgram dengan nilai kecil akan menaikkan nilai tingkat similarity dan semakin besar nilai k pada kgram akan memperkecil nilai similarity serta didapatkan (Hidayat et al., 2020), hal itu dibuktikan pada 10 sampel yang disediakan, 8 diantaranya apabila nilai k nya cukup tinggi maka semakin rendah nilai *similarity*. Hal itu bisa terjadi karena ketika pengelompokan k-gram terdapat banyak angka yang harus dibandingkan, sehingga ketika dibandingkan kesamaan nilai k nya akan sulit ditemukan.. Untuk penggunaan data dengan indikasi tidak mirip, semakin kecil nilainya akan semakin baik sehingga sesuai dengan indikasi awal. Nilai similarity yang didapatkan pada 10 sample yang ada termasuk nilai similarity yang cukup rendah karena nilai *similiarity* tidak mencapai 50% sehingga dapat disimpulkan bahwa lebih dari 90% code yang dibandingkan tidak memiliki indikasi mirip dan penyebab lainnya adalah karena data yang digunakan tanpa dilakukan *text preprocessing* sehingga spasi dan karakter mempengaruhi cukup banyak sehingga nilai *similarity* yang dihasilkan cukup kecil oleh karena itu diperlukan proses *stopword removal* dengan menerapkan *stop word removal* dapat mengurangi waktu untuk proses selanjutnya dan dapat meningkatkan akurasinya (Gunawan et al., 2018).

Selanjutnya hasil pada rata-rata berdasarkan data tanpa *text preprocessing* pada indikasi mirip pada Tabel 4.4 bahwa penerapan nilai k yang bervariasi dari nilai k = 2, 3, 5, dan 7 menghasilkan bahwa pengaruh terhadap pengaruh nilai k terhadap perbedaan hasil rata-rata *similarity* yaitu semakin kecil nilai k maka nilai

similarity yang dihasilkan semakin tinggi, hal itu dibuktikan pada keseluruhan sampel yang ada ketika nilai k=2 mendapatkan hasil similarity yang paling tinggi diantara nilai k yang lain. Untuk penggunaaan data indikasi mirip, semakin besar nilai *similarity* nya akan semakin baik sehingga sesuai dengan indikasi awal. Pada 10 sample yang diujikan terdapat beberapa nilai *similarity* yang dihasilkan cukup rendah karena tidak mencapai 1%, hal itu bisa terjadi karena tidak ada proses *text preprocessing* yang terjadi.

4.7.9. Evaluasi Perbandingan dari 2 Skenario

Berdasarkan hasil dari 2 skenario yang telah dilakukan yaitu menggunakan algoritma Winnowing sebagai pendeteksi *similarity* yaitu dengan menggunakan 10 sample yang berdasarkan dengan data dengan *text preprocessing* dan data tanpa *text preprocessing* dengan 2 indikasi, yaitu indikasi tidak mirip dan mirip.

Didapatkan hasil pada rata-rata berdasarkan data dengan *text preprocessing* dan tanpa *text preprocessing* pada indikasi tidak mirip pada Tabel 4.5 bahwa penerapan nilai k yang bervariasi dari nilai k = 2, 3, 5 dan 7. Rata-rata waktu pemrosesan pada skenario 3 memiliki waktu yang cukup baik dan tercepat pada nilai k=7. Serta menghasilkan *similarity* terendah pada data tanpa *text preprocessing*, hal tersebut bisa terjadi karena data yang dibandingkan merupakan data asli sehingga spasi, karakter, dan jumlah line yang berbeda dapat membuat data yang dibandingkan memiliki nilai *similarity* yang kecil. Nilai *similarity* yang kecil pada indikasi tidak mirip bisa dikatakan baik, namun belum tentu akurat. Sehingga pada data dengan *text preprocessing* memiliki nilai *similarity* yang lebih tinggi daripada data tanpa *text preprocessing* namun keakuratannya terjamin karena

dapat disimpulkan bahwa proses *text preprocessing* dapat menyebabkan penurunan nilai kemiripan, dibandingkan dengan nilai yang akan diperoleh jika kita menghitung kemiripan teks tanpa preprocessing. Dengan kata lain, perbedaan antara teks terlihat jelas ketika kesamaan teks dihitung setelah teks preprocessing (Petrović & Stanković, 2019)

Selanjutnya hasil pada rata-rata berdasarkan data dengan *text preprocessing* dan tanpa *text preprocessing* pada indikasi mirip pada Tabel 4.6 bahwa penerapan nilai k yang bervariasi dari nilai k = 2, 3, 5, dan 7 menghasilkan bahwa pengaruh pada data tanpa *text preprocessing* terhadap pengaruh nilai k dengan nilai *similarity* adalah semakin kecil nilai k, maka nilai similarity yang dihasilkan semakin tinggi. Hal tersebut bisa terjadi dikarena apabila nilai k nya kecil sehingga jumlah *k-grams* yang dibandingkan semakin kecil dan detail. Selain itu juga mendapatkan hasil bahawa pengaruh pada data dengan *text preprocessing* terhadap pengaruh nilai k dengan nilai *similarity* adalah semakin besar nilai k, maka nilai *similarity* yang dihasilkan semakin tinggi. Hal tersebut bisa terjadi karena apabila *source code* yang dibandingkan memiliki fungsi pada line yang sama maka akan menghasilkan nilai *similarity* yang tinggi.

BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan yang dihasilkan berdasarkan skenario pengujian dan evaluasi diantaranya adalah:

a. Algoritma winnowing bisa digunakan untuk melakukan pendeteksian *source code*, algoritma winnowing pada umumnya digunakan untuk melakukan pendeteksian pada teks dokumen. Namun memiliki beberapa kekurangan apabila membandingkan *source code* yang memiliki jumlah *line code* yang berbeda, sehingga akan memperkecil nilai *similarity*-nya dan apabila *source code* yang isinya sama namun hanya memindahkan posisinya saja.

b. Hasil nilai *similarity* dengan data tanpa *text preprocessing* dari 10 sample pada indikasi mirip dan 10 sampel pada indikasi tidak mirip mendapatkan hasil *similarity* yang kecil yang rata-rata dibawah 1%, sedangkan penerapan algoritma winnowing dengan data yang sudah melalui *text preprocessing* lebih akurat dibandingankan dengan data tanpa melalui *text preprocessing* karena pada hasil *similarity* pada data yang memiliki indikasi mirip tidak ada yang mencapai 50% sedangkan pada data dengan melalui *text preprocessing* terdapat hasil *similarity* lebih dari 70% dengan menggunakan nilai k = 2, 3, 5, dan 7 serta nilai *window* = 3.

c. *Text preprocessing* sangat mempengaruhi hasil *similarity* pada pendeteksian *source code* menggunakan algoritma winnowing. Hal itu disebabkan apabila *source code* yang dibandingkan tidak dilakukan proses *text preprocessing* akan

membuat jumlah karakter dan spasi yang dibandingkan cukup banyak dan membuat waktu pemrosesan cukup lama.

5.2. Saran

Terdapat saran untuk penelitian selanjutnya, diantaranya:

a. Menambah mekanisme atau algoritma lainnya untuk mendeteksi plagiarisme apabila kode hanya merubah nama variabelnya saja.
b. Disarankan menambahkan pengecekan apabila hanya merubah posisi *line code*.
c. Dapat menggunakan algoritma yang lebih optimal dalam membandingkan plagiarisme pada *source code* dengan pendekatan berbasis kode berorientasi atribut atau berbasis kode berorientasi struktur.

DAFTAR PUSTAKA

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Anjali, V., Swapna, T. R., & Jayaraman, B. (2015). Plagiarism Detection for Java Programs without Source Codes. *Procedia Computer Science*, *46*, 749–758. https://doi.org/10.1016/j.procs.2015.02.143

Awale, N., Pandey, M., Dulal, A., & Timsina, B. (2020). Plagiarism Detection in Programming Assignments using Machine Learning. *Journal of Artificial Intelligence and Capsule Networks*, *2*(3), 177–184. https://doi.org/10.36548/jaicn.2020.3.005

Bao, T., Zheng, Y., & Zhang, X. (2012). White box sampling in uncertain data processing enabled by program analysis. *ACM SIGPLAN Notices*, *47*(10), 897–914. https://doi.org/10.1145/2398857.2384681

Barrón-Cedeno, A., Eiselt, A., & Rosso, P. (2009). Monolingual Text Similarity Measures: A Comparison of Models over Wikipedia Articles Revisions. *ICON*, 29–38.

Bebbington, Shaun. (2014). What is coding. . *Learn Programming-What Is Programming*.

Budiman, A. E., & Widjaja, A. (2020). Analisis Pengaruh Teks Preprocessing Terhadap Deteksi Plagiarisme Pada Dokumen Tugas Akhir. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(3). https://doi.org/10.28932/jutisi.v6i3.2892

Burrows, S., Tahaghoghi, S. M., & Zobel, J. (2004). *Efficient and effective plagiarism detection for large code repositories*. 8–15.

Cheers, H., Lin, Y., & Smith, S. P. (2021). Academic Source Code Plagiarism Detection by Measuring Program Behavioral Similarity. *IEEE Access*, *9*, 50391–50412. https://doi.org/10.1109/ACCESS.2021.3069367

Cosma, G., & Joy, M. (2008). Towards a definition of source-code plagiarism. *IEEE Transactions on Education*, *51*(2), 195–200. https://doi.org/10.1109/TE.2007.906776

Ferrante, J., Ottenstein, K. J., & Warren, J. D. (1987). The program dependence graph and its use in optimization. *ACM Transactions on Programming Languages and Systems*, *9*(3), 319–349. https://doi.org/10.1145/24039.24041

Gomaa, W. H., & Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. In *International Journal of Computer Applications* (Vol. 68, Issue 13).

Gunawan, D., Sembiring, C. A., & Budiman, M. A. (2018). The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *Journal of Physics: Conference Series*, *978*(1). https://doi.org/10.1088/1742-6596/978/1/012120

Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, *17*, 26–32. https://doi.org/10.1016/j.procs.2013.05.005

Hage, J., Rademaker, P., & van Vugt, N. (2011). *Plagiarism detection for Java: a tool comparison*. 33–46.

Hidayat, W., Utami, E., & Hartanto, A. D. (2020). Pemilihan Parameter Terbaik pada Algoritma Winnowing dalam Mendeteksi Tingkat Kesamaan Dokumen Bahasa Indonesia. *Citec Journal*, *7*(2).

Jhi, Y.-C., Wang, X., Jia, X., Zhu, S., Liu, P., & Wu, D. (2011). Value-based program characterization and its application to software plagiarism detection. *Proceedings of the 33rd International Conference on Software Engineering*, 756–765. https://doi.org/10.1145/1985793.1985899

Joy, M., & Luck, M. (1999). Plagiarism in programming assignments. *IEEE Transactions on Education*, *42*(2), 129–133. https://doi.org/10.1109/13.762946

Leman, D., Riza, B. S., & Akbbar, M. B. (2019). *RABIN KARP AND WINNOWING ALGORITHM FOR STATISTICS OF TEXT DOCUMENT PLAGIARISM DETECTION*.

Li, S., Xiao, X., Bassett, B., Xie, T., & Tillmann, N. (2016). Measuring code behavioral similarity for programming and software engineering education. *Proceedings of the 38th International Conference on Software Engineering Companion*, 501–510. https://doi.org/10.1145/2889160.2889204

Luo, L., Ming, J., Wu, D., Liu, P., & Zhu, S. (2017). Semantics-Based Obfuscation-Resilient Binary Code Similarity Comparison with Applications to Software and Algorithm Plagiarism Detection. *IEEE Transactions on Software Engineering*, *43*(12), 1157–1177. https://doi.org/10.1109/TSE.2017.2655046

Petrović, Đ., & Stanković, M. (2019). THE INFLUENCE OF TEXT PREPROCESSING METHODS AND TOOLS ON CALCULATING TEXT SIMILARITY. *Facta Universitatis, Series: Mathematics and Informatics*, 973. https://doi.org/10.22190/fumi1905973d

Purbo, O. W. (2019). *Text Mining: Analisis Medsos, Kekuatan Brand, Dan Intelijen Di Internet*. ANDI.

Purwitasari, D., Kusmawan, P. Y., & Yuhana, U. L. (2011). DETEKSI KEBERADAAN KALIMAT SAMA SEBAGAI INDIKASI PENJIPLAKAN DENGAN ALGORITMA HASHINGBERBASISN-GRAM. *Jurnal Ilmiah KURSOR*.

Puspaningrum, E. Y., Nugroho, B., Setiawan, A., & Hariyanti, N. (2020). Detection of Text Similarity for Indication Plagiarism Using Winnowing Algorithm Based K-gram and Jaccard Coefficient. *Journal of Physics: Conference Series*, *1569*(2). https://doi.org/10.1088/1742-6596/1569/2/022044

Ramdhani, R., Fadlil, A., & Sunardi, S. (2022). Penerapan Algoritma Winnowing dan Word-Level Trigrams Untuk Mengidentifikasi Kesamaan Kata. *JURIKOM (Jurnal Riset Komputer)*, *9*(2), 427. https://doi.org/10.30865/jurikom.v9i2.4060

Ramli, M. S., Cokrowibowo, S., & Rustan, M. F. (2021). Uji Plagiarism pada Tugas Mahasiswa Menggunakan Algoritma Winnowing. *Journal of Applied Computer Science and Technology*, *2*(2), 108–112. https://doi.org/10.52158/jacost.v2i2.177

Ridho, M. (2013). Rancang Bangun Aplikasi Pendeteksi Penjiplakan Dokumen Menggunakan Algoritma Biword Winnowing. *(Doctoral Dissertation, UNIVERSITAS ISLAM NEGERI SULTAN SYARIEF KASIM RIAU)*.

Rinartha, K. (2017). SIMPLE QUERY SUGGESTION UNTUK PENCARIAN ARTIKEL MENGGUNAKAN JACCARD SIMILARITY. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, *3*(1).

Roy, C. K., & Cordy, J. R. (2007). A survey on software clone detection research. *Queen's School of Computing TR*, 64–68.

Sanjaya, W. (2008). *Perencanaan dan desain sistem pembelajaran*. Kencana.

Satia Budhi, G., Wibisono, A., & Tanojo, R. (2017). *PENGECEKAN PLAGIARISME PADA CODE DALAM BAHASA C++*.

Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003). *Winnowing: Local Algorithms for Document Fingerprinting*. 702.

Snsebatik, P., & Samarinda, ; Stmik Widya Cipta Dharma. (2017). PERANCANGAN APLIKASI DETEKSI PLAGIARISME KARYA ILMIAH MENGGUNAKAN ALGORITMA WINNOWING. In *Seminar Nasional Serba Informatika* (Vol. 1, Issue 1).

Tao, G., Guowei, D., Hu, Q., & Baojiang, C. (2013). Improved Plagiarism Detection Algorithm Based on Abstract Syntax Tree. *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, 714–719. https://doi.org/10.1109/EIDWT.2013.129

Widaningrum, I., Mustikasari, D., Arifin, R., & Pratiwi, H. A. (2020). Evaluation of the accuracy of winnowing, rabin karp and knuth morris pratt algorithms in plagiarism detection applications. *Journal of Physics: Conference Series*, *1517*(1). https://doi.org/10.1088/1742-6596/1517/1/012093

Widjaja, T., & Gunawan, A. (2020). *Deteksi Plagiarisme pada Kode Bahasa Pemrograman Java menggunakan XGBoost*.

LAMPIRAN

LAMPIRAN 1. Tabel Hasil Rolling Hash berdasarkan K-Gram dengan data yang sudah dilakukan *Text Preprocessing*

Sample	Source code 1				Source code 2			
	$k = 2$	$k = 3$	$k = 5$	$k = 7$	$k = 2$	$k = 3$	$k = 5$	$k = 7$
Sample 1	[1069, 1097, 1167, 1073, 1131, 1109, 1101, 1219, ..., 00000000000000000	10797, 11067, 11773, 0831, 11409, 11201, 1119, 12303, ..., 00000000000000000000 00000000039000000000 00000000000000000000 00000000000000000000	[1080773, 1107831, 1178409, 1084201, ..., 89630000000000000000 00000000000404200000 00000000000000000000 00000000000000000000	[108078409, 110784201, 117942119, 108421305, ... 00000000000000000000 00000000000000000000 000000000000000040429 10000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000126962730]	[1069, 1097, 1167, 1073, 1131, 1109, 1101, 1219, ..., 00000000000000000000 00000000000500000000 00000000000000000000	[10797, 11067, 11773, 10831, 11409, 11201, 11119, 12295, ..., 00000000000000000000 00000000000000000000 00000000000000000000 00000000003900000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000011730]	[1080773, 1107831, 1178409, 1084201, ..., 0000000000000000 0000000000000000 0000000000000000 0000000000000004 0420000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000	[108078409, 110784201, 117942119, 108421295, ..., 000000000000000000 000000000000000000 000000000000000404 291000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000

	00000000000000000 00000000000000000 00000000050000000 0000000000000330]	00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 000000000000011730]	00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 0000000001262730]		00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 000000000000000330]		000000000000000 000000000000000 000000000000000 000000000000000 0000000000000126 2730]	00000000000001269 62730]
Sample 2	[1069, 1097, 1167, 1073, 1131, 1109, 1101, 1219, ..., 91000000000000000 00000000000000000 00000000000000000 00000000000000000 00000000000000000	[10797, 11067, 11773, 10831, 11409, 11201, 11119, 12293, ..., 25100000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000	[1080773, 1107831, 1178409, 1084201, ..., 24130000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000005972999	[108078409, 110784201, 117842119, 106421293, ..., 24113400000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000	[1069, 1097, 1167, 1073, 1131, 1109, 1101, 1219, ..., 49999999999999999999 99999999999999999999 99999999999999999999 99999999999999999999 99830000000000000000	[10797, 11067, 11773, 10831, 11409, 11201, 11119, 12291, ..., 60999999999999999999 99999999999999999999 99999999999999999999 99999999999999999999 99898000000000000000	[1080773, 1107831, 1178409, 1084201, ..., 597259999999999 999999999999999 999999999999999 999999999999999	[108078409, 110784201, ..., 59732099999999999 99999999999999999 99999999999999999 99999999999999999 99999999999999838 70750000000000000

	00000000000000000	0000000000006299999	99999999999999999999	0000000000005973209	00000000000000000000	00000000000000000000	999999999999999	00000000000000000
	0000000000499999	9999999999999999999	99999999999999999999	9999999999999999999	00000000000000000000	00000000000000000000	999999988871000	00000000000000000
	99999999999999999	9999999999999999999	99999999999999999999	9999999999999999999	00000000000000000000	00000000000000000000	000000000000000	00000000000000000
	99999999999999999	9999999999999999999	99999999999999999999	9999999999999999999	00000000000000000000	00000000000000000000	000000000000000	00000000000000000
	99999999999999999	9999999999999999999	99999999999999999999	9999999999999999999	00000000000000000000	00000000000000000000	000000000000000	00000000000000000

9999999999999999	9999999999999995998	9999999999999989999	9999999999999999999	0000000000000000000	0000000000000000000	00000000000000	000000000000000
9999999999999999	9999999999999999999	9999999999864800000	9999999999999999999	0000000000000000000	0000000000000000000	00000000000000	000000000000000
9999999999999999	9999999999980000000	0000000000000000000	9999999998964979000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
9999999999999999	0000000000000000000	0000012020999999999	0000000000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
9999999999988000	0000121999999999999	9999999999999999999	0000012020409999999	0000000000000000000	0000000000000000000	00000000000000	000000000000000
0000000000000000	9999999999999999999	8907000000000000000	9999999999999999999	0000000000000000000	0000000000000000000	00000000000000	000000000000000
0000000000001139	8910000000000000000	0000000000000000000	8906855000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
9999999999999999	0000000000000000000	0183700000000000000	0000000000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
9999999999999890	0170000000000000000	0000000000000000000	0183555000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
0000000000000000	0000000000000000000	2962000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
0000000000000000	3000000000000000000	0000000000000000000	2963670000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000
0000200000000000	0000000000000000000	0129620000000000000	0000000000000000000	0000000000000000000	0000000000000000000	00000000000000	000000000000000

	00000000000000000000000003000000000000000000000000000000000000030000000000000000000000000000000002100000000000000000000000000000000000000000000000000000000110000000000000014000000000000000000000000000000000000000000000000000	23000000000000000000000000000000000151000000000000000000000000000000000000000000000000000001990000000000000127000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011230]	00000000000000001527289999999999999999999999999999999999999999999999999999999937087000000000001275300000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001272230]	02963350000000000000000000000000000152738499999999999999999999999999999999999999999999999999999369583420000000001275188000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111772230]	00330]	0012430]	00	00

	00000000000000000 00000000000000000						000000000000000 000000000000000	00000000000000000 00000000000000000

	00000000000000000 00000000000000000 00000000000000000 000330]						000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 00001236430	00000000000000000 00000000000000000 00000000001178364 30]

LAMPIRAN 2. Hasil Rolling Hash berdasarkan K-Gram dengan data tanpa dilakukan *Text Preprocessing*

Sample	Source code 1				Source code 2			
	$k = 2$	$k = 3$	$k = 5$	$k = 7$	$k = 2$	$k = 3$	$k = 5$	$k = 7$
Sample 1	[1069, 1097, 1167, 1073, 1131, 1042, 419, 1099, -64, -11425, ..., -	[10797, 11067, 11773, 10831, 11342, 10519, 4301, 11056, -323, 112145, ..., -	[1080773, 1107831, 1178342, 1083519, 1135301, ..., -	[108078342, 110783519, 117835301, ..., -	[1069, 1097, 1167, 1073, 1131, 1042, 419, 1099, -64, -11435, ..., -	[10797, 11067, 11773, 10831, 11342, 10519, 4301, 11056, -335, -112241, ..., -	[1080773, 1107831, 1178342, 1083519, 1135301, ..., -	[108078342, 110783519, 117835301, ..., -
	1200000000000000	1866000000376029999	0000000000000000	0000000121363500000	0000000000000000	1180000000000000000	0000000000000000	0000000000000000
	0000000000000000	9999999988925865979	0000000121370000000	0000000000000000002	0000000000000029999	0000000000000000000	1122200000000000	0000000000000
	0000000000000000	6799999999999999999	0000000000000000000	0012136350000000000	9999997589999999999	0000000000000028599	0000000000000	0000000000000
	0000000130000000	9999999999999999999	0012137000000000000	0000000000012136350	9999999999999999999	9999998087999999999	0112290000000	0000001213635
	0000000000000000	9999999999999999999	0000000000121370000	0000000000000000000	9999999999999998870	9999099999999999999	0000000000000	0000000000000
	0000000000130000	9998699999961000774	0000000000000000000	0000000000000000001	8199999789999739000	9999999999999988870	0000000000000	0000000000000
	0000000000000000	0002999999999999999	0000000000000000001	1223620000000000000	0000000000008199999	8829999809999792000	0000000000000	0000012136350
	0000000000130000	9999997589290000142	1222000000000000000	0000000000112379500	7899900000000000000	0000000000008929999	0000000000000	0000000000000
	0000000000000000	9460000000000000000	0000000000112390000	0000000000000000000	0003099999999999999	8869530000000000000	0011229000000	0000000000000
	0000001300000000	0000299999999999999	0000000000000000000	0000000000000000000	9997170000860000014	0962729999999999999	0000000000000	1213635000000
	0000000000000000	9400000037589006599	0000000000000000000	0000000000000000000	9990000000000000000	9997541000929000014	0000000000000	0000000000000
	0000000000000000	9999999999999999999	0000000000000000000	0112379500000000000	0000000189999999870	2947000000000000000	0000000000286	0001213635000
	0001200000000000	9999997589290000142	0112390000000000000	0000000000000000000	0000003700028999999	0000001859999999866	8099999981087779	0000000000000
	0000000000000000	9950000000000037378	0000000000000000000	0000000028684639998	9999867645863579399	0000001760029999999	9999999999999	0000000000000
	1200000000000000	4000000008580007900	0000000028680999998	1088862999999999999	9999999999999999999	9999889258659796799	9999999999999	0000000000001
	0000000000000000	2999999999999999999	1087799999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999988	1223820000000
	0000000000000000	9980107500000415000	9999999999999999999	9999999988731014515	9999999999999999998	9999999999999999999	731014499794139	0000000000000
	0000000000000000	0000000000000187999	9999999988731014499	7941372657735190000	4999999500000690002	9999999999999999998	239675400000000	0000112292800
	0000000120000000	9999999999999999999	7941392396754000000	0000009014515795424	8999999999999999998	6999998610000774000 2	0000090144997 99	0000000000000
	0000000000000000	9999999999887092807 5	0000009014499789547 1	5380000000000002771	9978086000000149990 0	9999999999999999999	5471500000000000	0000000000000
	0000000000000000	0000000000286000008	5000000000000802771	5379999999999997946	0000000000000000000	3975892900001429460	0000027714999999	0000000000000
	0000000000029999	0799999999999999999	4999999999999897946	0251372169013593626 7	2899999999999999000 0	0000000000000000000	9999999999974 61	0000000000000
	9999957589999999	9999999998900209999 9	1209372300013594157 0	9000000000000000000	0000369887789999999	2999999999999999400	2093723000135 94	0000000000000
	9999999999999999	9999999999999999999	0000000000000000000	1873626999867486100 3	9999999999999999999	0000375890065999999	1570000000000	0011229280000

	999999999999999	999999999999975880	187319999986745000	6865139818599999889	9978086000000149980	9999999999999999999	0000000018731	0000000000000	
	999998870819999	999999998870900999	6867297999999999883	9268068362227389299	0000000000369707000	9975892900001429950	999986745000036	0000000000000	
	8969975900000000	999999999991791600000	8267263101240999999	9999999999999999999	0000007900009002899	0000000000375784000	867299799999999	0000000000000	
	0000000081999978	0009009999999999731	9999999999999999999	9999999999999999999	9999999999999999979	0000008580007900299	9889926726910l2	0000028684469	
	9990000000000000	6000000833000000000	9999999999999999999	9999999999998657887	0070000004300000000	9999999999999999980	409999999999999	9998108986299	
	0000309999999999	0000000083900000000	9999999999998657899	6149697833982998185	0000000000183999999	1075000004150000000	999999999999	9999999999999	
	9999999717000860	0000000000073900000	6156007831202937899	9999999999999975800	9999999999999999999	0000000000187999999	999999999999	9999999999999	
	0000014999000000	0000080799999999999	9999999999999975800	1416901359345360000	9999998870850070000	9999999999999999999	999999999999	9999999999999	
	0000000000000000	9999997777758927875O	7230001359399500000	0000000000002998185	0000000300000072999	9999998870928075000O	3865789996156O0	9887310145157	
	1899999998700000	0000000004899999999	0000000000000299799	9999999999309960036S	9999999999999999999	0000000286000008079	7833202997999999	9413726577351	
	0037000289999999	9999989001300000000	9999999999311000036S	3797735733999999999	9999988777099999999	9999999999999999999	999999999999997	9000000000009	
	9998876498697939	0813000000000000000	3000734839999999999	9999999999999975826	9999999999999999999	9999989002099999999	580072300013593	0145157995424	
	9999999999999999	8739999999999999999	9999999999999975800	1416901359006158000O	9999999987807299999	9999999999999999999	995000000000000	5380000000000	
	9999999999999999	9997588278750000000O	7230001359011300000O	0000367652340800000S	9999987085999999999	9999999975880799999	000000029979999	0002771537999	
	9999999999999999	0000000000000000000	0000367673740000000S	6491170724398185999	9999995708000000008	9999887090099999999	999999993111000	9999999997794	
	9999999998499999	0000000000027299999	6499080802997999999	9999999999794141554	5999999999907080000	9999999791600000009	036830007348999	6025137216901	
	9500000690000299	9999999998888992999	9999999999974134050	3041555980000000000	0007600000000000000	0099999999897916000	999999999999999	3593626790000	
	9999999999999999	9999999999999999978	0041553000000000000	0001898882999999999	0000076000000000000	9008330000000000000	999999998758007	0000000000000	
	9978086000001499	4299999999999999997	0001898979999999999	9999999999999999998	0000000670000000000	0000839000000000000	220001359611300	0018736269998	
	9000000000000000	5886400000027300000	9999999999999999998	8731288145715000000O	0729999999999999999	0000000739000000000	000000003676737	6748610036865	
	0000002899999999	0000000000092795300	8731287337300000000O	2866407981561599999	9977777888439800000O	0807999999999999999	400000008649908	1398185999999	
	9999990000003638	0000000000000283999	2865700081552999999	9999999999999999989	0000004899999999999	3877777389278750000O	080299799999998	8899268068362	
	8778999999999999	6355l	9999999999999999989	0030170499999999999	9887730000000000730	0000048999999999999	999999999979413	2273699999999	
	9999999999999999		0030089999999999999	9999999999999999999	0000000000000007899	9890013000000000813	405000415530000	9999999999999	
	7808600000149980		9999999999999999999	9975881561509999887	9999999999999999957	0000000000000008739	000000000000189	9999999999999	
	0000000000036970		9975881552999999887	3110172299999999999	8084986000000000000	9999999999999999997	379999999999999	9999999999999	
	7000000007800009		3110889999999999999	7913344300009101722O	0000000000000000000	5892787500000000000	999999999999999	9999999998865	
	0002899999999999		7913220000009100899	9999997913352700840S	0000000030999999999	0000000000000000000	999887312873373	7887961496978	
	9999999979007000O		9999997913220008404	1270000000000000846	9999999887719999999	0000000027299999999	000000000286370	3398299818599	
	0043000000000000		9000000000000000846	1965000000000000000	9999999999999978995	9999999899529999999	008155299999999	9999999999999	
	0000001899999999		1800000000000000000	7477172000000817205	9999999999999978078	9999999999999784299	999999999999999	9758031416901	
	9999999999999999		7476400000000817185	9999999999999977777	0000000310000000000	9999999999999975886	998900300899999	3593453860000	
	9999999887085007		9999999999999977777	5895218590010000000O		4000000273000000000	999999999999999	0000000000000	

	000000000000030000		5799800129000000004	884978999999999890021	000000008499900000000	000000009279530000000	999999999999999	2998185999999
	00072999999999999		8853999999999999890021	6146000000821611800	000000000029999975751	000000000285999635551	975881552999999	9999309960036
	99999999999999998		5400000000821540000	000000000882092599999			588731100899999	8379773573399
	877799999999999999		000000000881999999999	99899999999975802218			999999997913220	9999999999999
	99999999999999999		99999999999975799800	617730000000000000000			0000009100699999	99999999993758
	99999999978072999		161000000000000000000	000000000000000000000			999997913220000	03141690135900
	99999998870859999		000000000000000000000	027715379999999999999			840490000000000	06158000000000
	99999999999997080		027714999999999999999	89000152679999999999			0000000846180000	36765234080000
	000000008599999999		8300014399999999999999	99999997842868799999			0000000000000007	00864911707340
	99999708000000760		99999997842449999999	999999997579651801000			4764000000000081	33818599999999
	000000000000000000		99999999757940990000	277153800000000000000			7189999999999999	99999999794140
	007600000000000000		277150000000000000000	092882925380000000000			9999777775795800	15543041555980
	000000006700000000		092874715000000000000	0002865376435390]			0129000000000048	00000000000000
	000007299999999999		00028656644355]				8339999999999983	01898982999999
	99999999977777808						0021540000000080	99999999999999
	49860000000000004						2154000000000000	99999999999880
	83999999999999887						0008819999999999	73128814571500
	73000000000007300						9999999999975799	00000028664070
	000000000000000079						9800161000000000	08156150999999
	99999999999999999						0000000000000000	99999999999999
	99997806986000000						0000000000000000	99999890030170
	000000000000000000						2771499999999999	04999999999990
	000000000000000000						9998900014339999	99999999999990
	003099999999999999						9999999999999999	99999999997580
	99998877199999999						9784244999999999	81561509999980
	99999999999997080						9999999757940990	87311017229990
	99999999999999999						0000277150000000	99999999979130
	97807800000031000						0000000000928740	34430000910170
	000000000000000000						7150000000000000	22999999979130
	84999000000000000						0028656644355]	35270084051270
	000002999997575]							00000000000000
								08461365000000
								00000000000740

								7717200000008 1720529999999 9999999777775 8952185900100 0000048849789 9999999890021 6146000000821 6118000000000 0008820925999 9999999999999 7580221861773 0000000000000 0000000000000 0000000000277 1537999999999 9998900015267 9999999999999 9999784286879 9999999999997 5796518010027 7153800000000 0000009288292 5380000000000 0028653764353 90]
Sample 2	[1069, 1097, 1167, 1073, 1131, 1042, 419, 1099, ..., - 12000000000000000 00000000000000000 00130000000000000 00000000000000000 00000000000000013	[10797, 11067, 11773, 10831, 11342, 10519, 4301, 11056, ..., - 11800000000000000000 00000000000000001200 00000000000000000000 00000000000000000000 00012000000000000000	[1080773, 1107831, 1178342, 1083519, ..., - 11227000000000000000 00000000000000001213 70000000000000000000 00000000000000000000 00012137000000000000 00000001213370000000	[108078342, 110783519, ..., - 11227810000000000000 00000000000000001213 63500000000000000000 00000000000000000000 00012136350000000000 00000001213635000000	[1069, 1097, 1167, 1073, 1131, 1042, 419, 1099, ..., - 12000000000000000000 00000000000000000000 13000000000000000000 00000000000000000000 00000001300000000000	[10797, 11067, 11773, 10831, 11342, 10519, 4301, 11056, ..., - 11800000000000000000 00000000000000000000 12000000000000000000 00000000000000000000 00000001200000000000	1080773, 1107831, ..., - 11225000000000 00000000000000 00000000012137 00000000000000 00000000000000 00000000000121	[108078342, 110783519,..., - 11226670000000 00000000000000 00000000000000 01213635000000 00000000000000 00000000000000

0000000000000000	0000000120000000000	0000000000000011222	0000000000000011223	0000000000013000000	0000000000120000000	3700000000000	0000000001213
0000013000000000	0000000000000011800	0000000000000000000	8200000000000000000	0000000000000000000	0000000000000000000	0000001213700	6350000000000
0000000000000001	0000000000000000000	0000000441700000000	0000000441780000000	0013000000000000000	0012000000000000000	0000000000000	0000000121363
2000000000000000	0000000450000000000	0000000000000000000	0000000000000000000	0000000000032999999	0000000000034499999	0000000121370000	5000000000000
0000000000000400	0000000000000000000	0000000000000000000	0000000000000000000	9999995869999999999	9999998199999999999	0000000000000	0000000000001
0000000000000000	0000000000000000000	0343179999999982429	0343178699999982425	9999988707899999986	9999988708663999987	0000000343179999	2136350000000
0000000000000000	0344999999999981999	3999999999999987307	3229999999999987307	9999877899979000999	7998900679980102499	9999998242939999	0000000000000
0000000000000000	9999999999999987086	4879999877480067713	4878599877497087714	9999977599999999999	9999977809999999999	9999999998730	0003431786999
3299999999999786	6999999878000008739	7938967802952997580	6538934063932057580	7980600000143990000	7589290000142947000	7497999987747 89	9999824253229
9999999999999998	8008659794799997852	7220001359415700000	1416901359362679000	0000000000000000000	0000000000000000000	0077137937881179	9999999999887
8707899999987000	9000001429470000000	0000000000000000502	0000000000000000502	0059999988785000000	0050299998901300000	9999978484099999	3074978599877
0007899978979979	0000000000000000502	0799890137720000000	0691890137741900000	0000000000000000000	0000000000000000000	9999975800723000	4875007714653
2999997808600000	9999898130000000000	0000000000000000000	0000000000000000000	0000000690000000000	0000007740000000000	0135941570000	7856848999978
1499900000000000	0000000000000000000	0077829000000000000	0077837640000000000	0000000000000000000	0000000000000000000	0000000000000	4854929999999
0000000000000509	0077400000000000000	0000000000000000000	0000000000000000000	0000000005499999999	0000000092099999999	005020799890137	7580314169013
9999887850000000	0000000000000000000	0000726816839999789	0000726906774499789	9999999999974400000	9999999999974930000	720000000000000	5936267900000
0000000000000000	0000725965999999788	0190000000000000000	0048000000000000000	3000000650210000099	2860000726189000347	0000000000000	0000000000000
0000000690000000	5000000050000000000	0000000000000000930	0000000000000000930	9999997130000000000	9999997835000000000	0077829000000	0005020691890
0000000000000000	0000000000000000522	4899999999999997913	5804999999999999913	0000000000000000649	0000000000000000725	0000000000000	1377419000000
0000000000000006	9999999999999997916	4400000000000000489	5585000000000000580	8700016999999730000	8820018899999835000	0000000000009	0000000000000
4997999999971500	0000000000000922999	9999999999999999791	4999999999999999791	0000000000000000000	0000000000000000000	3878999999999	0000000007783
0000000000000000	9999999999999999791	3440000000000000009	3558500000000000009	0064998000000163999	0072596600000188999	9999997531610 0	7640000000000
0000000000000000	6000000000000000009	3048999999999999979	3058049999999999979	9971300000000000000	9978350000000000000	2865000726840 71	0000000000000
0859999999999999	2299999999999999978	1344000000000000000	1355850000000000000	0000000000000065009	0000000000000072608	0034857999978 37	0000000000009
9997080000000000	1600000000000000000	0093048999999999999	0093058049999999999	0000000169999971300	7000000188999997835 0	3600000000000	3888039999999
0000859999999999	0092299999999999999	9999791344000000000	9999791355850000000	0000000000000000000	0000000000000000000	0000000000726	9999999997531
9999999997080000	9999791600000000000	0000093048999999999	0000093058049999999999	0000000000000650099 0	0000000000007260870	6607718829999 78	7493286476072
0000000000000085	0000092299999999999	9999999997913344000 0	9999999997913555850 0	0000001699999971300 0	0000018899999783500	3736000000000	6933221434856
9999999999999999	9999999997916000000	0000000000003048999	0000000000009355849	0000000000000000000	0000000000000000000	0000000000726	6899783726460
7080000000000000	0000000000092299999	9999999999999997998	9999999999999997998	0000000000006501200	0000000000007261180	8172600001882 99	0000000000000
0000000859999999	9999999999999997914	3700000000000000000	2348000000000000000	0000169999971300000	0000188999997835000	3978373600000 00	0000000007267
9999999999970800	0000000000000000000	8404900000000000846	8405127000000000846	0000000000000000000	0000000000000000000	0000000000000	3345108318597
0000000000000008	8330000000000000839	1800000000000000000	1965000000000000000	0000650030000001699	0000726018000000 1889	0000072678703 00	8372646000000

	5999999999999999	0000000000000000000	0008171900000000000	0008172053000000000	9999713000000000000	9999835000000000000	0018829999783 73	0000000000000
	9999997080000000	0008080000000000000	0000000726685959999	0000000726761354299	0000000000000000649	0000000000000000725	6000000000000	0072690714820
	0000000000085999	0000000726036999999	7946490000000000076	7946482200000000076	8700000000169999997	8810000000188999997	0000000000000	0188318597837
	9999999999999999	7945000000000000075	2089999999999979134	2171399999999979135	1300000000000000000	8350000000000000000	0072678703000	2646000000000
	9579300000000000	3999999999999979160	4000000000000000762	5850000000000000762	0000000000000064997	0000000000000072596	0188299997837 36	0000000000000
	0000000076000000	0000000000000000753	0899999999999999999	1713999999999999999	9997150000000000000	5997886000000000000	0000000000000	0007268736362
	0000000076000000	9999999999999999999	9999999999999999997	9999999999999999997	0000000000000000000	0000000000000000000	0000000000000	0018831859783
	0000000000000000	9999999999999999997	5800728001359523400	5803141690135948229 6	0000000018000000600	0000000099860009250	0072678815000 1	7246000000000
	0730000000000000	5892900000142959000 0	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	8829999783736 00	0000000000000
	0000000065001999	0000000000000000000	2482600000044962989000	2493015000449627190 0	8000000000000649999	0000000000000726999	0000000000000	0000000007268
	9999712000000000	2490000000449999899 00	2117000000000078333	2124980000000007833 9	9997080000000082999	9997916000000089899	0000000726777	7363660001883
	0000006599999999	1400000000000077400	0014029999999999999	8013968499999999999	9999999978080000000	9999999791600000000	9900001882999 97	1859783726460
	9999970800000000	0014999999999999999	9999999999999644355	9999999999999644353	0000078999999999708	0000865999999999791	8373600000000 00	0000000000000
	0000000006699999	9999999999999963553		90	0000000000000078999	6000000000000865999	0000000000000	0000000000000
	9999999999999999				9999999978080000000	9999999791600000000	0072670094000 00	0000726874864
	9999999999999999				0000078999999999708	0000865999999999791	0188299997837 36	1018831859783
	9780860000014999				0000000000007899999	6000000000865699999	0000000000000	7246000000000
	0000000000000000				9999970800000000000	9999979160000000000	0000000000000	0000000000000
	0000000240000000				0689999999999997079	6789999999999997915	7268173078901 90	0007268636796
	0449999887780000 0				9999999999999963540	9999999999999964700	0000000000000	0018831859783
	0000000069000001						0000000000000	7246000000000
	9999999999999999						0000000061054636	0000000000000
	9999999999999975						0934050000000 00	0000726777931
	75]						0000000000000	1000018831859
							0000000007340 39	7837264600000
							9997913910000 00	0000000000000
							9079799999999979	0000000000726
							1391000000000008	9071277900480
							7438999999999791	0000000000000
							3910000000000008	0000000000000
							7438999999999979	0000000000010
							1391000000000000	5449931341334
							8756499999999791	0000000000000

							391000000000864 339999999979139 100000000007787 299999999997913 909999999999964 4355]	0000000000000 0000073413089 9791397840000 9080526999999 7913979100000 0008744838999 9997913977600 0000000874484 2999999979139 7760000000008 7573689999979 1397730000000 8644208999999 7913978200000 0007788113999 9999979139787 9999999964435275]

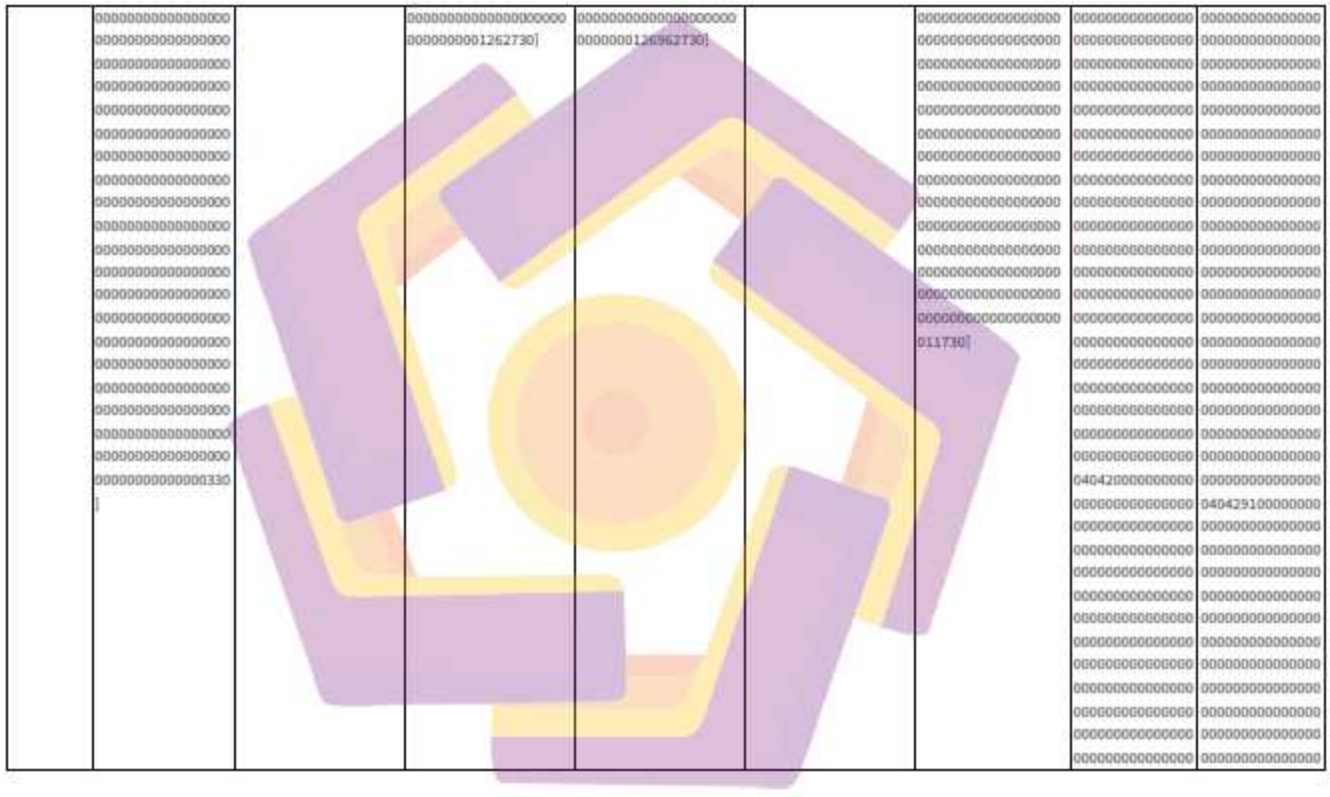
LAMPIRAN 3. Hasil Hasil Sebaran Hash pada Window berdasarkan K-Gram pada data yang sudah dilakukan Text Preprocessing

Sample	Source code 1				Source code 2			
	$k = 2$	$k = 3$	$k = 5$	$k = 7$	$k = 2$	$k = 3$	$k = 5$	$k = 7$
Sample 1	[1217, 1069, 1097] [1069, 1097, 1167], ..., [900000000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..., [900000000000000000	[1228067, 1080773, 1107831] [1080773, 1107831, 1178409] , ..., [896300000000000000	[122807831, 108078409, 110784201] [108078409, 110784201, 117842119] , ..., [896457000000000000	[1217, 1069, 1097] [1069, 1097, 1167], ..., [900000000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..., [900000000000000000	[1228067, 1080773, 1107831] [1080773, 1107831, 1178409] , ..., [896300000000	[122807831, 108078409, 110784201] [108078409, 110784201, 117842119] , ..., [896457000000000

	000000000000000000	000000000000000000000	000000000000404200000	000000000000404291000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0390000000000000000	0000000000000000	0000000000000000
	000000000005000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0040420000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0040429100000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	00000000012624,	000000000000000000000	000000000000000000000	00000000001124,	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	900000000000000000000	000000000000000000000	000000000000000000000	900000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	00000001269624,	00000126369624,	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	896300000000000000000	896457000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000001	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	2624,	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	9000000000000000000	0000000000000000	0000000000000000
	000000000001124,	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	900000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000012	0000000000000000

	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	69624,	0000000000001263
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	8963000000000000	69624,
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	8964570000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000003900000000	000000000000000000000	000000000000000000000	000000000005000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000004042000000	000000000000404291000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0390000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000500000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000011272,	000000000000000000000	000000000000000000000	000000000001172,	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	300000000000000000000	000000000000000000000	000000000000000000000	900000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000001296272,	000000113696272,	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	896300000000000000000	896457000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0404200000000000	0000000000000000
	000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000000	0000000000000000000	0000000000000000	0404291000000000

	00000000000000000 00000000000000000 00000000000000000 00000000000000000 00000000000000000 00000000000001172, 90000000000000000 00000000050000000	00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000390000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000011730|	00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000404200000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000	00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000404291000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000	00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000250000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000330|	000000000000000000 11272, 900000000000000000 039000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000	000000000000000 00000000000012 96272, 896300000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000	000000000000000 000000000001136 96272, 896457000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000

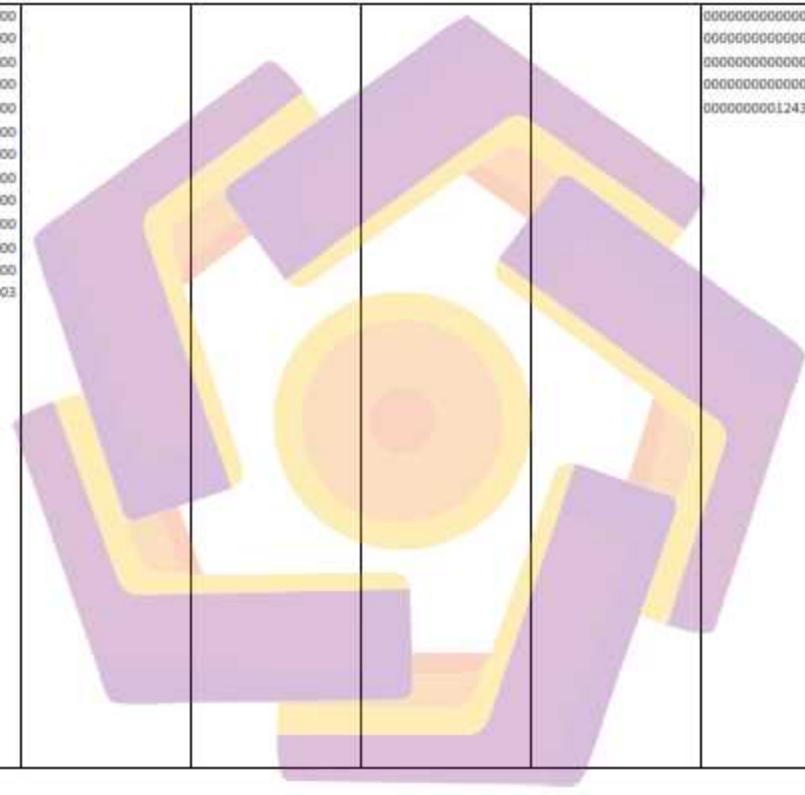


							000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000001 262730]	000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000126 962730]
Sample 2	[1217, 1069, 1097] [1069, 1097, 1167], ..., [4999999999999999 9999999999999999 9999999999999999 9999999999999999 9999999999999989 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..., [2510000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000609999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999888000000 0000000000000000000 0000012199999999999 9999999999999999999 8910000000000000000	[1228067, 1080773, 1107831] [1080773, 1107831, 1178409], ..., [2413000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000005972999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999886480000 0000000000000000000 0000012020999999999	[122807831, 108078409, 110784201] [108078409, 110784201, 117642119], ..., [2411349800000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000005973209 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999886497000 0000000000000000000 0000012020409999999	[1217, 1069, 1097] [1069, 1097, 1167], ..., [9100000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 0000000000000499999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999999999999 9999999999880000000 0000000000000000000 0000011399999999999 9999999999999999999 8900000000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..., [6099999999999999999 999999999999999999 999999999999999999 999999999999999999 999999999989800000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000	[1228067, 1080773, 1107831] [1080773, 1107831, 1178409], ..., [5972999999999 999999999999999 999999999999999 999999999999999 999999999999999 999999998987100 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000	[122807831, 108078409, 110784201] [108078409, 110784201, 117642119], ..., [59732099999999 999999999999999 999999999999999 999999999999999 999999999999999 999999998987075 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000

	0000000000000000	0000000000000000000	9999999999999999999	9999999999999999999	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0170000000000000000	8907000000000000000	8906895000000000000	0200000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	3000000000000000000	0183700000000000000	0183595000000000000	3000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0300000000000000000	2962000000000000000	2963670000000000000	0300000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000191	0000000000000000000	0000000000000000000	0000000000000000210	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0296200000000000000	0296335000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000192	0000000000000000192	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000001	7299999999999999999	7384999999999999999	0000000000000000001	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	9900000000000001270	9999999999999999999	9999999999999999999	1000000000000001400	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	9999999999999999993	9999999999999999991	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	7087000000000001275	6983420000000001275	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	3000000000000000000	1880000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000001271	0000000000000000000	0000000000000000000	0000000000000000111	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	9,	0000000000000000000	0000000000000000000	9,	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	2510000000000000000	0000000000000111771	0000000000012721771	9100000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	9,	9,	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	7413000000000000000	2411349000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000012361,	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000004999999	6099999999999999	0000000000000000	0000000000000000
	0000000000000126	9999999999999999999	0000000000000000000	0000000000000000000	9999999999999999999	9999999999999999	0000000000000000	0000000000000000
	1,	9999999999999999999	0000000000005972999	0000000000005973209	9999999999999999999	9999999999999999	0000000000000000	0000000000000000
	4999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999	0000000000000000	0000000000000000
	9999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999800000	0000000000000000	0000000000000000
	9999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	0000000000000000	0000000000000000	0000000000000000
	9999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	0000000000000000	0000000000000000	0000000000000000
	9999999999999890	9999999999888000000	9999999999999999999	9999999999999999999	9999999998800000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000000	9999999999999999999	9999999999999999999	0000000000000000000	0000000000000000	0000000000000000	0000000000000000

	0000000000000000	0000012199999999999	9999999999886480000	9999999998564979000	0000011999999999999	0000000000000000	0000000000000	0000000000000
	0000000000000000	9999999999999999999	0000000000000000000	0000000000000000000	9999999999999999999	0000000000000000	0001178361,	0123678361,
	0000000000000000	8910000000000000000	0000012020999999999	0000012020480999999	8900000000000000000	0000000000000000	597299999999999	597320999999999
	0000000000000000	0000000000000000000	9999999999999999999	9999999999999999999	0000000000000000000	0000000000000000	999999999999999	999999999999999
	0000000000000000	0170000000000000000	8307000000000000000	8906895000000000000	0200000000000000000	0000000000000000	999999999999999	999999999999999
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	999999999999999	999999999999999
	0000000000000000	3000000000000000000	0183700000000000000	0183595000000000000	3000000000000000000	0000000000000000	999999999999999	999999999999999
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	999999989871000	999999989870750
	0000000000000000	0300000000000000000	2962000000000000000	2963670000000000000	0300000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000191	0000000000000000000	0000000000000000000	0000000000000000210	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0296200000000000000	0296335000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0000000000000000192	0000000000000000191	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000001	7239999999999999999	7384999999999999999	0000000000000000001	0000000000000000	0000000000000	0000000000000
	0000000000000000	9900000000000001270	9999999999999999999	9999999999999999999	1000000000000001400	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	9999999999999999993	9999999999999999993	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	7087000000000001275	6983420000000001275	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	1000000000000000000	1880000000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000112	0000000000000000000	0000000000000000000	0000000000000000011	0000000000000000	0000000000000	0000000000000
	0000000000000000	22,	0000000000000000000	0000000000000000000	22,	0000000000000000	0000000000000	0000000000000
	0000000000000000	2510000000000000000	0000000000000011772	0000000000001121772	9100000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	22,	22,	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	2413000000000000000	2411349000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000	0000000000000
	0000000000000000	0000000000006299999	0000000000000000000	0000000000000000000	0000000000004999999	0000000000000000	0000000000000	0000000000000
	0000000000000000	9999999999999999999	0000000000000000000	0000000000000000000	9999999999999999999	0000000000000000	0000000000000	0000000000000
	0000000000000000	9999999999999999999	0000000000005872999	0000000000005973229	9999999999999999999	0000000012642,	0000000000000	0000000000000
	0000000000000000	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	0000000000000000	0000000000000	0000000000000
	0000000000000000	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	6099999999999999	0000000000000	0000000000000
	0000000000000000	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999999	0000000000000	0000000000000

	000000000000000000 000000000000000000 0000000000000000124 2, 4999999999999999 9999999999999999 9999999999999999 9999999999999999 99999999999999890 000000000000000000	99999999999999999999 99999999999888000000 00000000000000000000 00000121999999999999 99999999999999999999 89100000000000000000 00000000000000000000 01700000000000000000 00000000000000000000 30000000000000000000 00000000000000000000 03000000000000000000 00000000000000000131 00000000000000000000 00000000000000000000 00000000000000000001 99000000000000001270 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000011 230]	99999999999999999999 99999999999999999999 99999999999999999999 99999999998864800000 00000000000000000000 00000120209999999999 99999999999999999999 89070000000000000000 00000000000000000000 01837000000000000000 00000000000000000000 29620000000000000000 00000000000000000000 02962000000000000000 00000000000000000192 72999999999999999999 99999999999999999999 99999999999999999993 70870000000000001275 30000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000001272 230]	99999999999999999999 99999999999999999999 99999999999999999999 99999999998864979000 00000000000000000000 00000120204099999999 99999999999999999999 89069950000000000000 00000000000000000000 01835950000000000000 00000000000000000000 29636700000000000000 00000000000000000000 02963350000000000000 00000000000000000192 73849999999999999999 99999999999999999999 99999999999999999993 69834200000000001275 18800000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000111772 230]	99999999999999999999 99999999999880000000 00000000000000000000 00000119999999999999 99999999999999999999 89000000000000000000 00000000000000000000 02000000000000000000 00000000000000000000 30000000000000000000 00000000000000000000 03000000000000000000 00000000000000000210 00000000000000000000 00000000000000000000 00000000000000000001 10000000000000001400 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000 330]	999999999999999999 999999999999999999 999999999989800000 000000000000000000	000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 0001283642, 597299999999999 999999999999999 999999999999999 999999999999999 999999999999999 999999989871000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000	000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 0126783642, 597320999999999 999999999999999 999999999999999 999999999999999 999999999999999 999999989870750 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000 000000000000000



LAMPIRAN 4. Tabel Hasil Sebaran Hash pada Window berdasarkan K-Gram pada data tanpa dilakukan Text Preprocessing

Sample	Source code 1				Source code 2			
	$k = 2$	$k = 3$	$k = 5$	$k = 7$	$k = 2$	$k = 3$	$k = 5$	$k = 7$
Sample 1	[1217, 1069, 1097] [1069, 1097, 1167], ..,[-1200000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..,[- 11800000000000000	[1228067, 1080773, 1107831] [1080773, 1107831, 1178342], ..,[- 11229000000000000	[122807831, 108078342, 110783519] [108078342, 110783519, 117835301] ,..,[- 11237950000000000	[1217, 1069, 1097] [1069, 1097, 1167] [1097, 1167, 1073], ..,[- 12000000000000000	[12269, 10797, 11067] [10797, 11067, 11773] [11067, 11773, 10831], ..,[- 11800000000000000	[1107831, 1178342, 1083519] [1178342, 1083519, 1135301] ,..,[- 11229000000000000	[122807831, 108078342, 110783519] [108078342, 110783519, 117835301] , ..,[- 11225280000000000
	0000000000000000	0000000000000000	0000000000000012	0000000000000000	0000000000130000	0000000000120000	0000000000000000	0000000000000000
	0000000000000130	0012000000000000	1370000000000000	0000000012136350000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000120000	0000000000000001213	0000000000000000000	0000000013000000	0000000012000000	0000121370000000	0000000000000000
	0000000013000000	0000000000000000	7000000000000000	0012136350000000000	0000000000000000	0000000000000000	0000000000000000	0000121363500000
	0000000000000000	0000000000000001	0000000001213700000	0000000000121363500	0001300000000000	0001200000000000	0000012137000000	0000000000000000
	0000000000000000	1800000000000000	0000000000000001	0000000000000000000	0000000000000000	0000000000000000	0000000000000000	0000001213635000
	0000001200000000	0000000001180000000	2137000000000000	0000000000000000001	0000000000000130	0000000000000118	0000121370000000	0000000000000000
	0000000000000000	0000000000000000	0000000000000000	1223820000000000000	0000000000000000	0000000000000000	0000000000000121	0000121363500000
	0001200000000000	0000000000000000	0000000112220000000	0000000001123795000	0000000001200000	0000000001180000	3700000000000000	0000000000000121
	0000000000000000	0000000000000000	0000000000000000011	0000000000000000000	0000000000000000	0000000000000000	0000000000000000	3635000000000000
	0000000000000000	0118000000000000000	2290000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000011222000	0000000000000000
	0000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000011223820000
	0000001200000000	0000000028999999999	0000000000000000000	0112379500000000000	0000000001200000	0000000001180000000	0000112290000000	0000000000000000
	0000000000000000	0879999999999999999	0000000000000112290	0000000000000000000	0000000000000000	0000000000000000	0000000000000000	0000112292800000000
	0000000000000000	9999999999999999999	0000000000000000000	0000000028684699996	0000000000000000	0000000000000000	0000000000000000	0000000000000000
	0029999999997589	9999999998870829999	0000000000000000000	1089882999999999999	0029999999997589	0028599999998879	0000000000000000	0000000000000000
	9999999999999999	8003799732000000000	0000000002868999999	9999999999999999999	9999999999999999	9999999999999999	0000000011229000	0000000000000000
	9999999999999999	0000008929998806330	8108779999999999999	9999999998873101451 5	9999999999999999	9999999999999999	0000000000000000	0000000011229280 0000
	9999999999988708	0000000000000002729	9999999999999999999	7941372657735190000 0	9999999998870819 99	9999999988708329 9	0000000000000000	0000000000000000
	1999997899699759 0	9999999999999997941	9999999999887310144 9	0000009014515795542 4	9978899397590000 0	9980097997920000 0	0000000028680999 9	0000000000000000
	0000000000000081	0009290000142947000	9794139239675400000 0	5380000000000002771	0000000000819999 7	0000000008929999 8	9361087799999999 9	0000000028684699
	9999978999000000	0000000000000000000	0000000901448979954 7	5379999999999997946	8999000000000000 0	0695300000000000 0	9999999999999999	9981089862999999 99

	000000000000309999	1859999999866000000	150000000000000277	025137216901359362	000030999999999999	000027299999999999	999999999999999988	999999999999999999
	999999999999971	760029999999999988	149999999999999994	900000000000000000	999999717000086000	999999794100092900	731014499794139239	999999999999999988
	000086000000014999	92586597967999999999	6120937230001359415	187362699886748610	000149990000000000	000142947000000000	67540000000000009	731014515794137265
	000000000000000000	999999999999999999999	000000000000000000000	6865139818599999889	000000000000001899	000000000000001859	0144997995471500000	773519000000000009
	000000018399999999	999999999999999999999	016731999998674500000	926806836222736999999	999999870000000370	999999866000000376	000000000002771499	014515799542453800
	870000000370002899	999999999999866999999	3686729979999999988	999999999999999999999	002899999999999887	002999999999999889	999999999999979461	000000000002771537
	999999999988764998	610000774000299999999	992672691012408999999	999999999999999999999	649867979389999999	925865979679999999	209372300013594157	999999999999979460
	6997939999999999	999999999999999975832	999999999999999999999	999999999999865788799	999999999999999999	999999999999999999	000000000000000000	251372169013593626
	999999999999999999	900000142946000000000	999999999999999999999	614896783398299818599	999999999999999999	999999999999999999	000187319999986745	790000000000000000
	999999999999999999	000000000000029999999	999999999999986657899	999999999999995758019	999999999999999999	999999999999999999	000036867299799999	000187362699986748
	999999999999999999	999999999940000000375	961560078332029879999	141690135934538600000	849999995000000900	866999996100007740	999988892672691012	610036865139818599
	849999995000000690	899865999999999999999	999999999999999997580	000000000000029981859	002899999999999999	002999999999999999	409999999878829?,-	999988892680683622
	000289999999999999	999999999999999975832	072300013593995000000	999999999993099600368	899999978886000000	999999975892899854	111290000000000000	273639999999999999
	999999997886000	900000142995000000000	000000000000029979999	379773573399999999999	149999000000000000	78,-	000000000000000000	999999999999999999
	000149990000000000	000003757840000000008	999999999931100000369	999999999999995758003	000000000289999999	118000000000000000	000000000000000000	999999999999999999
	000000000002899	580007990029999999999	830007348999999999999	141690135900615800000	999999990000090369	000000000000000000	000012137000000000	999999986578868709
	999999999999000	999999999998010750000	999999999999999997580	000036765234080000008	887789999999999999	000000000000120000	000000000000000000	4458,-
	000369887789999999	004150009000000000000	072300013590113000000	649117072439881859999	999999999999999997	000000000000000000	000001213700000000	112292800000000000
	999999999999999999	000187999999999999999	000003676737400000000	999999999997794141554	808600000014998000	000000000120000000	000000000000000000	000000000000000000
	999999978086000000	999999999999999999998	864990808029973999999	304155598000000000000	000000000369707000	000000000000000000	000012137000000000	000000000000000000
	014998000000000000	870528877500000000000	999999999999719413405	000189988299999999999	000000007800000090002	000120000000000000	000000000000000121	000012136350000000
	003697070000000000	286000008079999999999	000041553000000000000	999999999999999999998	899999999999999998	000000000000000000	370000000000000000	000000000000000000
	7800009000289999999	999999999999999999985	000018997999999999999	873128814571500000000	996712,-	000000000000000118	000000000000000000	000000121363500000
	999999999999999979	002099999999999999998	999999999999999999999	286640708156159999999	120000000000000000	000000000000000000	000000000112220000	000000000000000000
	007000004300000000	999999999999999999881	887312873373000000000	999999999999999999999	000000000000000000	000000000011800000	000000000000000000	000012136350000000
	000000000000001899	80,-	028657000881552999999	003017049999999999999	000000000000130000	000000000000000000	000011229000000000	000000000000000121
	999999999999999999	118000000000000000000	999999999999999999998	999999999999999999999	000000000000000000	000000000000000000	000000000000000000	363500000000000000
	999999999999999988	000000000000000000000	900300899999999999999	997588156150999999887	000000000013000000	000000000000000000	000000000000000000	000000000000000000
	708500700000000000	000000000120000000000	999999999999999999999	311017229999999999999	000000000000000000	000000000118000000	000000000000000000	000000000112238200
	003000000729999	000000000000000000000	999758815523999999988	791334430000910172299	000130000000000000	000000000000000000	000000001122900000	000000000000000000
	999999999999999999	001200000000000000000	731100899999999999999	999999791335270084405	000000000000000000	000000000000000000	000000000000000000	000011229280000000
	999999887779999999	000000000000120000000	979132200000091000999	127000000000000000846	000000000000000120	002859999999980879	000000000000000000	000000000000000000
	999999999999999999	000000000000000000000	999999979132200006840	136499999999999999996	000000000000000000	999999999999999999	000000000286809999	000000000000000000

	999999999999999997	000000000000000000001	490000000000000000084	4444448, -	000000000012000000	999999999999999999	3981087799999999999	000000000000000000
	80729999999999887	180000000000000000000	618000000000000000000	112379500000000000000	000000000000000000	999999999887089299	999999999999999999	000000001122928000
	08599999999999999	000000000011800000000	0747640000000081718	000000000000000000000	000000000000000000	998009799782000000	999999999999999988	000000000000000000
	99970798809, -	000000000000000000000	99999999999999997777	000000001213635000000	000000000000000000	000000000089299998	731014499794139239	000000000000000000
	12000000000000000	000000000000000000000	7375980012900000000000	000000000000000000000	000000000120000000	069530000000000000	675400000000000009	000000000286844639
	000000000000000000	000000000000000000000	488539999999993983002	001213635000000000000	000000000000000000	000027299999999999	014499799547150000	998108986299999999
	000000000000000130	011800000000000000000	154000000008215400000	000000000001213635000	000000000000000000	999999794100092900	000000000002771499	999999999999999999
	000000000000000000	000000000000000000000	000000000008819999999	000000000000000000000	002999999799975899	000142947000000000	999999999999979461	999999999999999988
	000000000000000130	000000002859999999981	999999999999975739800	000000000000000000001	999999999999999999	000000000000001859	209372300013594157	731014515794137265
	000000000000000000	0879999999999999999999	016100000000000000000	122382000000000000000	999999999999999999	599999866000000376	000000000000000000	773519000000000009
	000000000013000000	999999999999999999999	000000000000000000000	000000000011237950000	999999999887081993	002999999999999889	000187319999986745	014515799542453800
	000000000000000000	999999999887089299999	002771499999999999999	000000000000000000000	998788963975900000	925865979679999999	000036867299799999	000000000002771537
	000000000000000000	800979979200000000000	9890001439999999999	000000000000000000000	000000000081999997	999999999999999999	999988892672691012	999999999999979460
	000000120000000000	000000892999806353000	999999997984244999999	000000000000000000000	899300000000000000	999999999999999999	409999999986882924,	251372169013593626
	000000000000000000	000000000000000002728	999999999757940993000	011237950000000000000	000030999988999999	999999999999999999	-	790000000000000000
	000120000000000000	999999999999999997941	027715000000000000000	000000000000000000000	999999717000086000	866999996100007740	112290000000000000	000187362699986748
	000000000000000000	000092900000142947000	009287471500000000000	000000000286844099986	000149990000000000	002999999999999999	000000000000000000	610036865139818599
	000000000000000000	000000000000000000000	0000286146448, -, -	108986299999999999999	000000000000001899	999999975892899995	000000000000000000	999988892680683622
	000000000000000000	185999999986600000003	112290000000000000000	999999999999999999999	999999870000000170	663, -	000012137000000000	273699999999999999
	000000120000000000	760029999999999999885	000000000000000000000	999999999887310145151	002899999999999887	118000000000000000	000000000000000000	999999999999999999
	000000000000000000	925865579679999999999	000000000000000000012	794137265773513900000	649869979999999999	000000000000000000	000000121370000000	999999999999999999
	000000000000000000	999999999999999999999	137000000000000000000	000000901451579954240	999999999999999999	000000000000120000	000000000000000000	999999986378784059
	002999999999997589	999999999999999999999	000000000000000001213	538000000000000002771	999999999999999999	000000000000000000	000012137000000000	44548, -
	999999999999999999	999999999999986699999	700000000000000000000	537999999999999997946	999999999999999999	000000000120000000	000000000000000121	111292800000000000
	999999999999999999	610000774000029999999	000000000012137000000	025137216901359362670	049999995000006000	000000000000000000	370000000000000000	000000000000000000
	999999999999988708	999999999999993173692	000000000000000000001	900000000000000000000	002899999999999999	000120000000000000	000000000000000000	000000000000000000
	139999789969975900	900000142946000000000	213700000000000000000	187362699867486100030	999999978886000000	000000000000000000	000000000112220000	000012136350000000
	000000000000000081	000000000000029999995	000000000000000000000	686513981859999999899	149990000000000000	000000000000000118	000000000000000000	000000000000000000
	999997899900000000	999999999984000000375	000000001122200000000	926806836222273899999	000000000289999999	000000000000000000	000011229000000000	000000121363500000
	000000000000309999	890065999999999999999	000000000000000000011	999999999999999999999	999999990000000369	000000000011800000	000000000000000000	000000000000000000
	999999999999998717	999999999999999975852	229000000000000000000	999999999999999999999	887788999999999999	000000000000000000	000000000000000000	000012136350000000
	000086000000014999	900000014299500000000	000000000000000000000	999999999999863578879	999999999999999997	000000000000000000	000000000000000000	000000000000000121
	000000000000000000	000037578400000000008	000000000000000000000	614969783398299818599	888600000014998000	000000000000000000	000000001122900000	363500000000000000

	0000000018999999999	5800079002999999999	0000000000000000112290	9999999999999975801	0000000000036970700	0000000001180000000	0000000000000000000	0000000000000000000
	8700000003700028 9	9999999999981075000	0000000000000000000000	1416901339345386000	0000000078000090002	0000000000000000000	0000000000000000000	0000000000112238200
	9999999998876498	0041500000000000000	0000000000000000000000	0000000000000029881 85	8999999999999999999	0000000000000000000	0000000001286809999	0000000000000000000
	6997939999999999	0001879999999999999	0000000002868699999	9999999993899600368	9996788,-	0028599999998980879	9981087799999999999	0000112292800000000
	9999999999999999	9999999999999999998	0108779999999999999	3797735733999999999	1200000000000000000	9999999999999999999	9999999999999999999	0000000000000000000
	9999999999999999	8709280750000000000	9999999999999999999	9999999999999975803	0000000000000000000	9999999999999999999	9999999999999999988	0000000000000000000
	9999999999999999	2860000080799999999	9999999998873101449	1416901359006150000	0000000000001380000	9999999998870892 39	7310144997941392 39	0000000000000000000
	6499999950000069 0	9999999999999999983	9734139239675400000	0000367652340800000 8	0009000000000000000	9980097997920000000	6754000000000000009	0000000001122928000
	0002899999999999	0020999999999999999	0000000901449979754 7	6491170724398185999 9	0000000001300000000	0000000008929999 8	0144997995471500000	0000000000000000000
	9999999978086000	9999999999999999995	1500000000000000277	9999999999794141554	8000000000000000000	0695300000000000000	0000000002771499	0000000000000000000
	0001499000000000	768,-	1499999999999999734	3041555980000000000	0001300000000000000	0000272999999999999	9999999999957946 1	0000000002868446 39
	0000000000002899	1180000000000000000	6120937230001359415 7	0001898829999999998	0000000000000000000	9999997941000929000	2093723000135941 57	9981089862999999999
	9999999999990000	0000000000000000000	0000000000000000000	9999999999999999998	0000000000000001200	0001423470000000000	0000000000000000000	9999999999999999999
	0003698778999999	0000000012000000000	0187319999986745000	8731288145715000000	0000000000000000000	0000000000000001859	0000368672997999999	9999999999999999988
	9999999999999999	0000000000000000000	3686729979999999958	3866407081561509999	0000000000120000000	9999998660000003760	9998889267269101240	7310145157941372657
	9999999780860000 0	0012000000000000000	3926726910124099999	9999999999999999989	0000000000000000000	0029999999999989889	9999999999999999999	7735150000000000099
	0149800000000000	0000000000012000000	9999999999999999999	0030170499999999999	0000000000000000000	0256597967999999999	4099999999988291359	0145157995424538000
	0036970700000000	0000000000000000000	9999999999999999999	9999999999999999999	0000000000000000000	9999999999999999999	[	0000000000027715370
	7800009000289999 9	0000000000000000001	9999999999998657899	9975881561509999887	0000000000120000000	9999999999999999999		9999999999999979460
	9999999999999979	1800000000000000000	9615600783320289799 9	3110172299999999999	0000000000000000000	9999999999999999999		2513721690135936267
	0079000004300000 0	0000000000118000000	9999999999999997580	7913344300009101722 9	0000000000000000000	8669999961000077400		7900000000000000000
	0000000000001899	0000000000000000000	0723000135939950000	9999997913352706640 5	0029999999999758999	0029999999999999999		0001873626899867480
	9999999999999999	0000000000000000000	0000000000000029979 9	1270000000000000846	9999999999999999999	9999999758928999980		6100368651396185599
	9999999999999988	0000000000000000000	9999999999931100003 6	1964999999999999999	9999999999999999999	8598]		9998889268068362227
	7085007000000000 0	0118000000000000000	8300073489999999999	64444413,-	9999999998870819999			2736399999999999999
	0030000007299999 9	0000000000000000000	9999999999999997580	1122795000000000000	9978999697590000000			9999999999999999999
	9999999999999999	0000000028599999998	0723000135901136000	0000000000000000000	0000000000819999997			9999999999999999999
	9999998877999999 9	0879999999999999999	0000036767374000000	0000000012136350000 0	8999000000000000000			9999999865788794894
	9999999999999999	9999999999999999999	8649908080299789999	0000000000000000000	0000309999999999999			45448]
	9999999999999997	9999999998870892999 9	9999999999997941340 5	0012136350000000000	9999997170000860000			
	8072999999999887	8009799792000000000	0004155300000000000	0000000000012136350 0	0001499900000000000			
	0859999999999999	0000000692999869530	0000189979999999999	0000000000000000000	0000000000000001899			
	999707998787,-	0000000000000000272 9	9999999999999999999	0000000000000000001	9999998700000003700			

	120000000000000000	99999999999999997941	887312873373000000000	122382000000000000000	00289999999999687			
	00000000000000000	0009290000142947000	028657000815529999999	00000000001123795000	649863979399999999			
	00000000000000130	00000000000000000000	99999999999999999998	00000000000000000000	999999999999999999			
	00000000000000000	18599999998660000003	900300899999999999999	00000000000000000000	999999999999999999			
	00000000000000130	76002999999999999889	999999999999999999999	00000000000000000000	999999999999999999			
	00000000000000000	9258659796799999999	999758815529999999988	01123795000000000000	849999995000006900			
	00000000013000000	99999999999999999999	731100899999999999999	00000000000000000000	00289999999999999			
	00000000000000000	99999999999999999999	979112200000910089	00000000286844099996	999999978096000000			
	00000000000000000	99999999999986699999	999999979132200008840	10898629999999999999	149990000000000000			
	00000012000000000	61000677400029999999	490000000000000000084	99999999999999999999	000000000289999999			
	00000000000000000	99999999999999975892	618000000000000000000	99999999988731014515	999999990000000363			
	00012000000000000	90000014294600000000	8747640000000081718	79413726577351900000	88778999999999999			
	00000000000000000	00000000000029999999	99999999999999997777	00000090145157995424	899999999999999997			
	00000000000000000	99999999940000000315	757998001290000000000	53800000000000002771	80960000014958000			
	00000000000000000	89006599999999999999	488539999999999983002	53799999999999997946	000000000036970700			
	00000012000000000	99999999999999975892	154000000008215400000	02513721690135936267	000000078000090002			
	00000000000000000	90000014299500000000	00000000008819999999	90000000000000000000	899999999999999999			
	00000000000000000	00003757840000000008	99999999999997579980	18736269998674861003	99978582]			
	00299999999997589	58000780029999999999	01610000000000800000	6865139818599999889				
	99999999999999999	99999999998601075000	00000000000000800000	92680683622273899999				
	99999999999999999	00415000000000000000	00277149999999999999	99999999999999999999				
	99999999999988708	00018799999999999999	98900014399993999999	99999999999999999999				
	19999578996997590	99999999999999999998	99999999784244999999	99999999999986578879				
	00000000000000081	87092807500000000000	99999999975794099000	61496978339829981859				
	99999789990000000	28600000807999999999	02771500000000000000	99999999999999975803				
	00000000000309999	99999999999999999989	00928747150000000000	14169013593453860000				
	99999999999997717	00209999999999999999	00002865644355]	00000000000002998185				
	00008600000014999	99999999999999999999		99999999993099600168				
	00000000000000000	8648]		37977157339999999999				
	00000001899999999			99999999999999975803				
	87000000037000289			14169013590061580000				
	99999999998876498			00003676523408000008				
	69973999999999999			64911707243981859999				

	99999999999999999 99999999999999999 99999999999999999 94999999500000600 00028999999999999 99999999978086000 00014999000000000 00000000000002899 99999999999990000 00036988778999999 99999999999999999 99999997808600000 01499800000000000 00369707000000000 78000090002899999 99999999999999979 00700000043000000 00000000000001899 99999999999999999 99999999999999988 70850070000000000 00300000007299999 99999999999999999 99999988777999999 99999999999999999 99999999999999997 80729999999999887 08599999999999999 9997079998865]			99999999999794141554 30415559800000000000 00018998629999999999 99999999999999999998 87312881457150000000 28664070815615899999 99999999999999999989 00301704999999999999 99999999999999999999 99758815615099998887 31101722999999999999 79133443000091017229 99999979133527008405 12700000000000000846 19650000000000000000 634444016]				
Sample 2	[1217, 1069, 1097] [1069, 1097, 1167], ..., [- 12000000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..., [- 11800000000000000000	[1228067, 1080773, 1107831] [1080773, 1107831, 1178342], ..., [-	[122807831, 108078342, 110783519], ..., [- 11229280000000000000 00000000000000000000	[1217, 1069, 1097] [1069, 1097, 1167], ..., [- 12000000000000000	[12269, 10797, 11067] [10797, 11067, 11773], ..., [- 11800000000000000	[1228067, 1080773, 1107831], ..., [-112290 00000000000000000 00000000000000000	[122807831, 108078342, 110783519], ..., [- 11226670000000000

0000000000000000	0000000000000001200	1122900000000000000	0000000000000000012	0000000000000000000	0000000000000000000	0000000000000000121	0000000000000000000
0013000000000000	0000000000000000000	0000000000000000000	1363500000000000000	0000000000000000000	0000000000000000000	3700000000000000000	0000121363500000000
0000000000000000	0000000000000000000	0000000000000000012	0000000000000001213	0000130000000000000	0000120000000000000	0000000000000000001	0000000000000000000
0000000000000013	0001200000000000000	1370000000000000000	6350000000000000000	0000000000000000000	0000000000000000000	2137000000000000000	0000000000000000121
0000000000000000	0000000120000000000	0000000000000001213	0000000000001213635000	0000001300000000000	0000001200000000000	0000000000000000121	3635000000000000000
0000013000000000	0000000000000011800	7000000000000000000	0000000000000000001	0000000000000000000	0000000000000000000	3700000000000000000	0001213635000000000
0000000000000001	0000000000000000000	0000000001213700000	2136350000000000000	0000130000000000000	0000120000000000000	0000000001213700000	0000000000000000012
2000000000000000	0000000450000000000	0000000000000000001	0000000000000000000	0000000000000000130	0000000000000000120	0000000000000000000	1363500000000000000
0000000000000400	0000000000000000000	2137000000000000000	0000000112238200000	0000000000000000000	0000000000000000000	0000000000000000000	0000000003431786 9
0000000000000000	0000000000000000000	0000000000000000000	0000000000000000011	0000000000000000000	0000000000000000000	0011222000000000000	9999998242532299 99
0000000000000000	0344999999999981999	0000000112222000000	2292800000000000000	0000000001200000000	0000000001180000000	0000000000000000112	9999999887307497 85
0000000000000000	9999999999999867036	0000000000000000011	0000000000000000000	0000000000000000000	0000000000000000000	2900000000000000000	9987748730077146 53
3299999999999786	6999999878000086799	2290000000000000000	0000000000000000000	0000120000000000000	0000118000000000000	0000000000000000000	7856848999978485 49
9999999999999998	8005658794799957882	0000000000000000000	0000000000000112292	0000120000000000000	0000000000000000000	0000000000000000000	2999999975803141 69
8707899999987000	9000001429470000000	0000000000000000000	9200000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	0135936267900000 00
0007899978997979	0000000000000000502	0000000000000112290	0000000000000000000	0000000000000000000	0000000000000000000	0112290000000000000	0000000000000005020
2999997808600000	9999890130000000000	0000000000000000000	0000000002868469999	0000000012000000000	0000000011800000000	0000000000000000000	6918901377419000 00
1499900000000000	0000000000000000000	0000000000000000000	8108986299999999999	0000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000
0000000000000509	0077400000000000000	0000000002868469999	9999999999999999999	0000000000000000000	0000000000000000000	0028580999999810887	0000077837640000 00
9999887850000000	0000000000000000000	8108779999999999999	9999999999887310145 1	0000000002999999999	0000000002859999999	7999999999999999999	0000000000000000000
0000000000000000	0000725965999999768	9999999999999999999	5794137265773519000 0	9975899999999999999	9980879999999999999	9999999999999999999	0000000000093888 03
0000000690000000	6000000000000000000	9999999999887310144 9	0000000901451579954 2	9999999999999999999	9999999999999999999	9999999988731014 49	9999999999999999975
0000000000000000	0000000000000000922	9794139239675400000 0	4538000000000000277	9999999999999999988	9999999999999999988	9979413923967540 00	3174932864760726 53
0000000000000006	9999999999999997916	0000000901449979954 7	1537999999999999794	7081999987899699975	7088299998009799 79	0000000000901449 97	3221434856689978 37
4997999999971500	0000000000000022999	1500000000000000277	6025137216901399362 5	0000000000000000008	2000000000000000008	9954715000000000000	2646000000000000000
0000000000000000	9999999999999999791	1499999999999999794	7900000000000000000	1999997899900000000	0299998069530000000	0000277149999999999	0000000007267334 51
0000000000000000	6000000000000000009	6120937230001359415 7	0187362699986748610 0	0000000000309999 8	0000000002729999 9	9999997946120937 23	6831859783726460 00
0859999999999999	2299999999999999979	0000000000000000000	3686513981859999998 8	9999999999971700 0	9999999999979410 0	0001359415700000000	0000000000000000000
9997080000000000	1600000000000000000	0187319999867450000 0	9926806836222736999 9	0086000000149990000	0092900000142947000	0000000000000001873	7269071482001883 18
0000859999999999	0092299999999999999	3686728979999999988	9999999999999999999	0000000000000000000	0000000000000000000	1999998674500003685	5978372646000000 00
9999999997080000	9999791600000000000	9926726910124089999 9	9999999999999999999	0001899999998700 0	0001859999998660 0	6729979999999988 9	0000000000000000007
0000000000000085	0000092299999999999	9999999999999999999	9999999999999657887	0000370002899999995	0000376002999999999	9267269101240899999	2687363620018831 85
9999999999999999	9999999997916000000	9999999999999999999	9614968783398298185	9999887649869979199	9999889258659796 7	9999999999999999999	9783726460000000000

	708000000000000000	000000000000092299999	999999999999986578999	999999999999999997580	999999999999999999	999999999999999999	999999999999999999	000000000000000000
	000000085999999999	999999999999999997914	961560078332029979999	314169013393453866000	999999999999999999	999999999999999999	999999999999999999	000726873636600018
	999999999997080	000000000000000000000	999999999999999997580	000000000000002998180	999999999999999999	999999999988684, -	865789996156007833	831859783726460000
	000000000000008	833000000000000000839	072300013593985000000	589999999993099600360	999999984999999500	118000000000000000	202997999999999999	000000000000000000
	599999999999999	000000000000000000000	000000000000029599999	837977357333999999999	000690000289999999	000000000000000000	999999975800723000	000000000726874864
	999999708000000	000808000000000000000	999999999931100003600	999999999999997580000	999999999999957808	000000000000000000	135939950000000000	101883185978372646	
	000000000085999	000000007260369999999	830007348999999999999	314169013598061580000	600000014999000000	000012000000000000	000000000299799999	000000000000000000	
	999999999999999	794500000000000000075	999999999999999997580	000003676523409000000	000900000000000028	000000000000000000	999999993110000368	000000726863679600	
	997930000000000	399999999999999979160	072300013593011300000	864911707243981859999	999999999999990000	000000120000000000	300073489999999999	188318597837264600	
	000000000760000	000000000000000000753	000003676737400000000	999999999999794141550	000036388877899999	000000000000000000	999999999999999957	000000000000000000	
	000000000760000	999999999999999999999	864990808029979899999	430415559800000000000	999999999999999963	000012000000000000	580072300013590113	000007267779311000	
	000000000000000	999999999999999999997	999999999999794134050	000001899882999999999	999999788866000001	000000000000000120	000000000036767374	018831859783726460	
	073000000000000	589290000001429590000	000041553000000000000	999999999999999999999	499800000000000003	000000000000000000	000000086499080802	000000000000000000	
	000000006500199	000000000000000000000	000018997999999999999	687312861457150000000	657070000000007800	000000000000000000	397999999999999999	000000000726907127	
	999971200000000	249000000004499998900	999999999999999999999	028664070815615099999	009000289999999999	000000000118000000	999794134050004155	790048000000000000	
	000006699999999	140000000000000077400	887312873373000000000	999999999999999999998	999999999979097000	000000000000000000	300000000000000001	000000000000000000	
	999997080000000	001499999999999999999	028657000815529999999	900301794999999999999	000430000000000000	000011800000000000	899799999999999999	000000001054499313	
	000000000006699999	999999999999986481, -	999999999999999999998	999999999999999999999	000000018999999999	000000000000000000	999999999999999998	413340000000000000	
	999999999999999999	118000000000000000000	900300899999999999999	999758815615099999888	999999999999999999	000000000000000000	873128733730000000	000000000000000000	
	999999999999999999	000000000000000001200	899999999999999999999	731101722999999999999	999999887085007000	000000000000000000	002865700081552999	734130899791397840	
	978086000000014999	000000000000000000000	999758815529999999988	979133443000091017220	000000000300000007	000000001180000000	999999999999999999	000908052699999979	
	000000000000000	000000000000000000000	731100899999999999999	999999979133527000840	28840, -	000000000000000000	999980030088999999	139791000000068744	
	000000000240000000	000120000000000000000	979132200000091008999	512700000000000000084	120000000000000000	000000000000000000	999999999999999999	838999999791397760	
	044999988778000000	000000001200000000000	999999979132290008400	619650000000000000000	000000000000000000	000000000285999999	999999999999758815	000000008744842999	
	000000006900000001	000000000000000011800	490000000000000000084	074771720000008172000	000000000000000000	998087999999999999	529999999887311008	999979139776000000	
	999999999999999	000000000000000000000	618000000000000000000	529999999872478648, -	000013000000000000	999999999999999999	999999999999997913	000875736899999791	
	999999999999964800	000000045000000000000	073536693, -	112292800000000000000	000000000000000000	999999999999999988	220000009100899999	397730000000864420	
	, -	000000000000000000000	112290000000000000000	000000000000000000000	000000130000000000	708929999800979979	999979132200008404	899999979139782000	
	120000000000000000	000000000000000000000	000000000000000000000	000000000000000000012	000000000000000000	200000000000000008	900000000000000008	000007788113999995	
	000000000000000	034499999999999841999	000000000000000000012	136150000000000000000	000013000000000000	929999806953000000	461800000000000000	997913978799999987	
	001300000000000	999999999999998870860	137000000000000000000	000000000000000001213	000000000000000130	000000000002729999	000074764000000000	7644355, -	
	000000000000000	699999987800000086799	000000000000000001213	635000000000000000000	000000000000000000	999999999999794100	817189999999999999	112266700000000000	
	000000000000013	800965979479999758892	700000000000000000000	000000000001213635000	000000000000000000	009290000014294700	999777775799800129	000000000000000000	

0000000000000000	9000001429470000000	0000000000121370000	0000000000000000001	0000000001200000000	0000000000000000000	0000000004885399999	0000121363500000000
0000013000000000	0000000000000000000502	0000000000000000001	2136350000000000000	0000000000000000000	0001859999999866000	9999998900215400000	0000000000000000000
0000000000000001	9999890130000000000	2137000000000000000	0000000000000000000	0000012000000000000	0000376002999999999	0000821540000000000	0000000000000000121
2000000000000000	0000000000000000000000	0000000000000000000	0000000112238200000	0000000000000000000	9999889258659796?67	0000088199999999999	3635000000000000000
0000000000000400	0077400000000000000000	0000000112220000000	0000000000000000011	0000000000000000000	9999999999999999999	9999999993757998000	0000121363500000000
0000000000000000	0000000000000000000000	0000000000000000011	2292800000000000000	0000000000000000000	9999999999999999999	1610000000000000000	0000000000000000012
0000000000000000	0000725965999999788	2290000000000000000	0000000000000000000	0000000012000000000	9999999999987794,-	0000000000000000000	1363500000000000000
0000000000000000	6000000000000000000	0000000000000000000	0000000000000000000	0000000000000000000	1180000000000000000	0000027714999999999	0000000000034317869
3299999999999786	0000000000000000922	0000000000000000000	0000000000000112292	0000000000000000000	0000000000000000000	9999998900014399999	9999999824253229999
9999999999999998	9999999999999997916	0000000000000112290	8000000000000000000	0000000029999999999	0000000000000000000	9999999999999997784	9999999887307497?5
8707899999987000	0000000000000922999	0000000000000000000	0000000000000000000	9875899999999999993	0000120000000000000	2449999999999999999	9987748750077146?3
0007899978979979	9999999999999999791	0000000000000000000	0000000002868446999?	9999999999999999999	0000000000000000000	7579409900002771?0	7856848999978485?9
2999997806000000	6000000000000000000	0000000002868099999	8108986299999999999	9999999999999999988	0000001200000000000	0000000000000000928	2999999975803141?9
1499900000000000	2299999999999999973	8108779999999999999	9999999999999999999	7081799997899369?5	0000000000000000000	7471500000000000000	0135936267900000?0
0000000000000509	1600000000000000000	9999999999999999999	9999999999887101451	9000000000000000008	0000120000000000000	0286146448,-	0000000000000005020
9999887850000000	0092299999999999999	9999999999887101449	5794137265773515000?0	1999997899900?0000	0000000000000000120	1122900000000000000	6918961377419000000
0000000000000000	9999791600000000000	9794139239675400000?0	0000000901451579954?0	0000000000030999999	0000000000000000000	0000000000000000000	0000000000000000000
0000000069000000	0000092299999999999	0000000901449979954?0	4538000000000000277	9999999999999971?00	0000000000000000000	0000000000000000000	0000077837640000000
0000000000000000	9999999997816000000	1500000000000000277	1537999999999999794	0086000000149990000	0000000001100000000	0000121370000000000	0000000000000000000
0000000000000006	0000000000092299999	1499999999999999734	6025137216901359362?	0000000000000000000	0000000000000000000	0000000000000000000	0000000000938880?
4997999999971500	9999999999999997914	6120937230001359415?0	7900000000000000000	0001999999997?7000	0000110000000000000	0000012137000000000	9999999999999997?
0000000000000000	0000000000000000000	0000000000000000000	0187362699987486100	0000370002899999??	0000000000000000000	0000000000000000000	3174932864760726?3
0000000000000000	8330000000000000839	0187319999986745000?0	3686513581859999988	9999867649869579??	0000000000000000000	0000121370000000000	3221434856689978?7
0859999999999999	0000000000000000000	3686729979999999988	9926806836222736999?	9999999999999999?99	0000000000000000000	0000000000000000121	2646000000000000000
9997080000000000	0008080000000000000	9826726910124099999?	9999999999999999999?	9999999999999999?99	0000000011800000000	3700000000000000000	0000000007263345??
0000859999999999	0000000072003000999	9999999999999999999	9999999999999999999	9999999999999999?99	0000000000000000000	0000000000000000000	0831859783726460?0
9999999997080000	7945000000000000075	9999999999999999999	9999999999999865?887	9999999849999?9?00	0000000000000000000	0000000001122200000	0000000000000000000
0000000000000085	2999999999999979160	9999999999999865?899	9614969783398299815?5	0008000002899999999	0000000002859999999	0000000000000000000	7269071482001883?8
9999999999999999	0000000000000000753	9815600783202897999	9999999999999997?80	9999999999997?808	9980879999999999999	0000112290000000000	5978372646000000?0
7080000000000000	9999999999999999999	9999999999999997?80	3141690135934538600?0	4000000149990000000	9999999999999999999	0000000000000000000	0000000000000000007
0000000859999999	9999999999999999997	0723000135939950000?0	0000000000000029981??	0000000000000000028	9999999999999999988	0000000000000000000	2687363620018831?5
9999999999970800	5832900000142959000?0	0000000000000029979??	5999999999939960036	9999999959999990000	7089299998009799?9	0000000000000000000	9783726460000000?0
0000000000000008	0000000000000000000000	9999999999311000036	8379773573399999999?	0000038887789999999	2000000000000000008	0000000011229000000	0000000000000000000

	599999999999999999	249000000044999989000	830007348999999999999	999999999999999997580	999999999999999999	929999806953000000	000000000000000000	000726873636600018
	999999708000000000	140000000000000077400	999999999999999997580	3141690139006156000	999999780860000001	000000000002729999	000000000000000000	831859783726460000
	000000000000853999	001499999999999999999	072300013590113000000	000003676523408000000	439980000000000003	999999999999979410	000000000286809999	000000000000000000
	999999999999999999	999999999999996448, -	000003676737400000000	864911707243981859999	637070000000007800	009290000014294700	000000000000000000	000000000726874864
	957930000000000000	118000000000000000000	864990808029979999999	999999999999979414155	009000283999999999	000000000000000000	999999999999999999	101883185978372646
	000000000760000000	000000000000000001200	999999999999979413405	430415559800000000000	999999999979007000	000185999999866000	999999999999999988	000000000000000000
	000000000760000000	000000000000000000000	000041553000000000000	000001899882999999999	000430000000000000	000037600299999999	731014499794139239	000000726863679600
	000000000000000000	000000000000000000000	000001899799999999999	999999999999999999999	000900189979999999	999988892586597967	675400000000000009	188318597837264600
	073000000000000000	000120000000000000000	999999999999999999999	883312881457150000000	899999999999999999	999999999999999999	014499799547150000	000000000000000000
	000000000650019999	000000012000000000000	883731287337300000000	028664070815615099999	999999887885007000	599999999999999999	000000000002771499	000007267779311000
	999971200000000000	000000000000000011800	028857000815529999999	999999999999999999988	000000000300000007	999999999999987841	999999999999579461	018831859783726460
	000006599999999999	000000000000000000000	999999999999999999998	900301704999999999998	298737, -		209372300013584157	000000000000000000
	999997080000000000	000000045000000000000	900308089999999999999	999999999999999999999	120000000000000000		000000000000000000	000000007269071 27
	000000000669999999	000000000000000000000	999999999999999999999	999758815615099999988	000000000000000000		000187319999986745	790048000000000000
	999999999999999999	000000000000000000000	999758815529999999988	731101722999999999999	000000000000000000		000036867299799999	000000000000000000
	999999999999999999	034499999999999981999	731100899999999999999	979133443000091017722	000013000000000000		999988992672691012	000000001054499313
	978086000001499 9	999999999999998870860	979132200000091000999	999999979133527000440	000000000000000000		409999999999999999	413340000000000000
	000000000000000	539999987800000867999	999999979132200000840	512700000000000000084	000000130000000000		999999999999999999	000000000000000000
	000000002400000000	800986587947899973892	490000000000000000084	619650000000000000000	000000000000000000		999999999999999999	734130899791397840
	044999988778000000	900000142947000000000	618000000000000000000	074771720000000817200	000013000000000000		999999865778999615	000908052699999979
	000000000690000101	000000000000000000502	0746366816, -	529999999988478644 8, -	000000000000000130		600783320299799999	139791000000008744
	999999999999999999	999989013000000000000	112290000000000000000	112291800000000000000	000000000000000000		999999999999997580	838999999791397760
	999999999999999967	000000000000000000000	000000000000000000000	000000000000000000000	000000000000000000		072300013593995000	000000008744842999
	0, -	007740000000000000000	000000000000000000012	000000000000000000012	000000001200000000		000000000000000029	999979139776000000
	120000000000000000	000000000000000000000	137000000000000000000	136350000000000000000	000000000000000000		979999999999999311	000875736899999791
	000000000000000000	000072986599999999788	000000000000000001213	000000000000000001213	000012000000000000		000036830007348999	197730000000864420
	001300000000000000	600000000000000000000	700000000000000000000	635000000000000000000	000000000000000000		999999999999999999	899999979139782000
	000000000000000000	000000000000000000922	000000000012137000000	000000000012136350000	000000000000000000		999999758007230001	000007788113999999
	000000000000000013	999999999999999997916	000000000000000000001	000000000000000000001	000000000000000000		359011300000000003	997913978799999995
	000000000000000000	000000000000000922999	213700000000000000000	213635000000000000000	000000001200000000		676737400000008649	86443540, -
	000001300000000000	999999999999999999791	000000000000000000000	000000000000000000000	000000000000000000		908080299799999999	112266700000000000
	000000000000000001	600000000000000000009	000000011212000000000	000000011223820000000	000000000000000000		999999999979413405	000000000000000000
	200000000000000000	229999999999999999979	000000000000000000011	000000000000000000011	000000000299999999		000415530000000000	000012136350000000

	0000000000000400	1600000000000000000	2290000000000000000	2292800000000000000	9975899999999999999		0000018979999999	0000000000000000000
	0000000000000000	0092299999999999999	0000000000000000000	0000000000000000000	9999999999999999999		9999999999999999	0000000000000000121
	0000000000000000	9999791600000000000	0000000000000000000	0000000000000000000	9999999999999999988		9999988731287373	3635000000000000000
	0000000000000000	0000092299999999999	0000000000000112290	0000000000000112292	7081999937896997 5		0000000028657000 8	0001213635000000000
	3299999999999786	9999999997916000000	0000000000000000000	8800000000000000000	9000000000000000008		1552999999999999999	0000000000000000012
	9999999999999998	0000000000092299999	0000000000000000000	0000000000000000000	1999997899900000000		9999999999898900300	1363500000000000000
	8707899999987000	9999999999999997914	0000000028680899999	0000000028654499999	0000000000003099999		8999999999999999999	0000000003431786 9
	0007899789979979	0000000000000000000	8108729999999999999	8108986299999999999	9999999999999171700		9999999999999999999	9999998242532299 99
	2999997808600000 0	8330000000000000839	9999999999999999999	9999999999999999999	0286000001439900 0		8758815529999999 88	9999999887307497 85
	1499900000000000	0000000000000000000	9999999999887310144 9	9999999999887310145 1	8000000000000000000		7311008999999999 9	9987748730077146 53
	0000000000000509	0008888000000000000	9734139239675400000 0	5794137265773519000 0	0001899999999870000		9997913220000000 910	7856848999978485 49
	9999887850000000	0000000072603699999	0000000901449979547	0000000901451579547 1	0000370002899999 95		8899999999791322 0	2999999975803141 69
	0000000000000000	7945000000000000073	1500000000000000277	4538000000000000277	8999867649869979 39		0008040900000000 0	0135936279000000 0
	0000000690000000	3999999999999979160	1499999999999999734	1537999999999999794	9999799999999999999		0000084618000000 0	0000000000000005020
	0000000000000000	0000000000000000753	8120937230001359415 7	6025137216901359362 6	9999999999999999999		0000000007476400 0	6918901377419000 00
	0000000000000006	9999999999999999999	0000000000000000000	7900000000000000000	9999999999999999999		0000000817189999 99	0000000000000000000
	4997999999971500	9999999999999999997	0187319999986745000 0	0187362699986746610 0	9999999849999995 00		9999999997777757 9	0000077837640000 00
	0000000000000000	5882900050142890000	3686729979999999988	3686513981859999988	0006900002899999 99		9800129000000000 48	0000000000000000000
	0000000000000000	0000000000000000000	9926726910124099999 9	9926806836222736999 9	9999999999999578 08		8539999999999983002	0000000000938880 3
	0859999999999999	2480000000448998900	9999999999999999999	9999999999999999999	6000000149990000 00		1540000000815400 0	9999999999999975
	3997080000000000	1400000000000077400	9999999999999999999	9999999999999999999	0000000000000228		0000000000681999 3	3174932864760726 93
	0000859999999999	0014999999999999999	9999999999999865789 9	9999999999999865788 7	9999999999999999000		9999999999999999997	3221434856689978 37
	9999999957080000	9999999999999936355	9615600783202997999	9614569783358299818 5	0000365887789999 99		5799800161000000 00	2646000000000000 0
	0000000000000085		9999999999999997580	9999999999999997580	9999999999999999 99		0000000000000000 0	0000000007267334 51
	9999999999999999		0723000135939950000 0	3141690135934538600 0	9999997808600000 01		0000000000277149 0	0831859783726460 00
	7080000000000000		0000000000000299799	0000000000000299818	4998000000000001		9999999999989000 0	0000000000000000 0
	0000000859999999		9999999999311000036	5899999999309960036	4870700000000078 00		1439999999999999 9	7269071482001883 18
	9999999999970800		8300073489999999999	8179773573399999999 8	0090002899999999 99		9999784249999999 9	5978372646000000 0
	0000000000000008		9999999999999997580	9999999999999997580	9999999999975007000		9999999757949990 00	0000000000000000 7
	5999999999999999		0723000135901130000 0	3141690135900615600 0	0004300000000000 0		0277150000000000 0	2687363620018831 85
	9999997080000000		0000036767374000000 0	0000036765234080000 0	0000001899999999 99		0000928747150000 00	9783726460000000 0
	0000000000085999		8649908080299730999 9	8649117072439818599 9	9999999959999999 99		0000000028655644 48	0000000000000000 0
	9999999999999999		9999999999979413405	9999999999979414135	9999998870850070000		. -	0007268736366000 18

	99793000000000000 00000000076000000 00000000075000000 00000000000000000 07300000000000000 00000000065001999 99997120000000000 00000669999999999 99999708000000000 00000000006699999 99999999999999999 99999999999999999 97808600000014999 00000000000000000 00000000240000000 04499998877800000 00000000069000001 99999999999999999 99999999999999975 75]		00041553000000000000 00001899799999999999 99999999999999999999 88731287337300000000 02865700081552999999 99999999999999999998 90030089399999999999 99999999999999999999 99975881552999999988 73110089999999999999 97913220000009108899 99999997913220000640 49000000000000000084 61800000000000000000 07475268049]	43041555980000000000 00001899829999999999 99999999999999999999 88731288145715000000 02866407081561509999 99999999999999999998 90030170439999999999 99999999999999999999 99975881561509999988 73110172299999999999 97913344300009101722 99999997913352700840 51270000000000000084 61965000000000000000 07477172000000081720 52999999999995786444 8]	000000000300000007 2998924]		112290000000000000 000000000000000000 000000000000000000 000012137000000000 000000000000000000 000000121370000000 000000000000000000 000012137000000000 000000000000000121 370000000000000000 000000000000000000 000000000112220000 000000000000000000 000011229000000000 000000000000000000 000000000000000000 000000000000000000 000000001122900000 000000000000000000 000000000000000000 000000000256809999 998108779999999999 999999999999999999 999999999999999988 731014499794139233 675400000000000009 014499799547150000 000000000002771499 999999999999979461 209372300013594157 000000000000000000 000187319999986745 000036867299799999	831859783726460000 000000000000000000 000000000726874864 101883185978372646 000000000000000000 000000726863679600 188318597837264600 000000000000000000 000007267779311000 018831859783726460 000000000000000000 000000000726907127 790048000000000000 000000000000000000 000000001054499313 413340000000000000 000000000000000000 734130899791397840 000908052699999979 139791000000008744 838999999791397760 000000008744842999 999979139776000000 000873736899999791 397730000000864420 899999979139782000 000007788113999999 997913978799999999 964435275]

							9999889926726910 12 4099999999999999 9 9999999999999999 9 9999999999999999 9 9999999865789961 5 600783320299799999 9999999999999758 0 0723000135939950 00 0000000000000002 9 3799999999999931 1 0000368300073489 99 9999999999999999 9 9999997580072300 01 3590113000000000 03 6767374000000086 49 9080802997999999 99 9999999997941340 5 0041553000000000 0 0000018957999999 9 9999999999999999 9 9999986873128737 3 0000000028657000 8 1552999999999999 9 9999999998900300 8 3999999999999999 9 9999999999999999 9 9758815529999999 88 7311008999999999 99 9997913220000009 10 0899999999791322 00 0084049000000000 0 0000084618000000 0 0000000000747640 0	

							0000000817189999999 9999999997777757 980012900000000048 853999999999890021 540000000008215400 000000000000881999 999999999999999975 7799800161000000000 000000000000000000 000000000000277143 999999999999890000 143999999999999999 999978424499999999 999999575794099000 027715000000000000 000092874715000000 000000002865664435 5[	

LAMPIRAN 5. Tabel Hasil Fingerprint berdasarkan Window pada data yang telah dilakukan Text Preprocessing

Sample	Source code 1				Source code 2			
	$k = 2$	$k = 3$	$k = 5$	$k = 7$	$k = 2$	$k = 3$	$k = 5$	$k = 7$
Sample 1	[1069, 1073, 1101, 1205, ..., 900000000000000000 000000000000000000	[10797, 10831, 11119, 12155, ..., 900000000000000000000 000000000039000000000 000000000000000000000 000000000000000000000 000000000000000000000	[1080773, 1084201, ..., 89630000000000000000 00000000000404200000 00000000000000000000 00000000000000000000 00000000000000000000 00000000000000000000	[108078409, 108421395, ..., 89645700000000000000 00000000000040423100 00000000000000000000 00000000000000000000 00000000000000000000	[1069, 1073, 1101, 1158, ..., 000000000000000000 050000000000000000	[10797, 10831, 11119, 11687, ..., 900000000000000000 039000000000000000	[1080773, 1084201, ..., 8963000000000000 0000000000000000	[108078409, 108421295, ..., 89645700000000000 00000000000000000

	0000000005001124]	0012624]	0001269624]	00126369624]	0001124]	0012624]	00000000000000000000000000000000000000000000000000000000000004042001269624]	00000000040429100126369624]
Sample 2	[1069, 1073, 1101, 1141, ..., 9100000000000000000000000000000000	[10797, 10831, 11119, 11522, ..., 251000000000000000000000000000000000000000	[1080773, 1084201, ..., 241300000000000000000000000000000000000000000000000000000000	[108078409, 108421293, ..., 2411349000000000000000000000000000000000	[1069, 1073, 1101, 1130, ..., 49999999999999999999999999999999999999	[10797, 10831, 11119, 11397, ..., 60999999999999999999999999999999999999	[1080773, 1084201, ..., 597299999999999999999999999999	[108078409, 108421291, ..., 5973209999999999999999999999999999

	000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000499999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999988000000000000000000000000000000000001199999999999999999999999999999999999890000000000000000000000000000000000000000020000000000000000000000000000000000000000000030000000000000000000000000000000000000000000000000003000000000000000000000000000000000000021000000000000000000000000000000000000000000000000000000000000000000000000011000000000000000014000000000000000000000000000000000000	00000000000000000000000000000000000000000000000000000000000000000000000000000006099999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999998880000000000000000000000000000000000000121999999999999999999999999999999999999989100000000000000000000000000000000000000001700000000000000000000000000000000000000030000000000000000000000000000000000000000030000000000000000000000000000000000000191000000000000000000000000000000000000000000000000000000000199000000000000001270000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000	000000000000000000000000000000000000000000000000000000000000059729999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999998864800000000000000000000000000000000001202099999999999999999999999999999999999999890700000000000000000000000000000000000000018370000000000000000000000000000000000000000000000000000000002962000000000000000000000000000000000000000002962000000000000000000000000000000000000000192729999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999370870000000000001275000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001117719]	0000000000000000000000000000000000000000000000000000000000000000000000000000000005973209999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999988649790000000000000000000000000000000000012020409999999999999999999999999999999999999998906895000000000000000000000000000000000000000001835950000000000000000000000000000000000000000000000000000000029636700000000000000000000000000000000000000000296335000000000000000000000000000000000000000019273849999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999369834200000000001275188000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000	99999999999999999999999999999999999999999999999999998900	999999999999999999999999999999999999999999999999999989800	999999999999999999999999999999999999999999999999999999989871000	9999999999999999999999999999999999999999999999998987075000

	000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 1119]	00000000000000001271 9]		000000000000012721771 9]	0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 000000001261]	0000000000000000000 0000000000000000000 0000000000000000000 0000000000000000000 000000012361]	0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 0000000000000000 001178361]	000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000000000000 000000000123678361]

LAMPIRAN 6. Tabel Hasil Fingerprint berdasarkan Window pada data tanpa dilakukan Text Preprocessing

Sample	Source code 1				Source code 2			
	$k = 2$	$k = 3$	$k = 5$	$k = 7$	$k = 2$	$k = 3$	$k = 5$	$k = 7$
Sample 1	[1069, 1073, 1042, 419, ..., '	[10797, 10831, 10519, 4301, ..., '	[1080773, 1083519, ...,	[108078342, 108353119, ..., '	[1069, 1073, 1042, 419, ..., '	[10797, 10831, 10519, 4301, ..., '	[1080773, 1083519, ..., '	[108078342, 108353119, ..., '
	12000000000000000	11800000000000000	11229000000000000	11237950000000000	12000000000000000	11800000000000000	11229000000000000	11229280000000000
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000
	00000000000000130	00000000000012000	00000000000012137	00000000000000121	00000000000000013	00000000000000000	00000000000000000	00000000000000000
	00000000000000000	00000000000000000	00000000000000000	36350000000000000	00000000000000000	00001200000000000	00000000000012	00001213635000000
	00000000000000130	00000000001200000	00000000001213700	00000000000000121	00000000000000130	00000000000000000	13700000000000	00000000000000000
	00000000000000000	00000000000000000	00000000000000000	36350000000000000	00000000000000000	00000012000000000	00000000000000	00000012136350000
	00000000013000000	00012000000000000	00012137000000000	00000000012136350	00000000013000000	00000000000000000	00000012137000	00000000000000000
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000001	00001200000000000	00000000000000	00001213635000000
	00000000000000000	00000000000000118	00000000000000112	00000000000000000	38000000000000000	00000000000000120	00000000012137	00000000000000121
	00000012000000000	00000000000000000	22000000000000000	00000011223820000	00000000000000000	00000000000000000	00000000000000	36350000000000000
	00000000000000000	00000000011800000	00000000011239000	00000000000000000	00000012000000000	00000000000000000	00000000121370	00000000000000000
	00012000000000000	00000000000000000	00000000000000000	00011237950000000	00000000000000012	00000000011800000	00000000000000	00000000011223820
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000	00000000000000000
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00001180000000000	00000000000112	00001122928000000
	00000000000000000	00000000011800000	00000000011239000	00000000000000000	00000000000000000	00000000000000000	22000000000000	00000000000000000
	00000012000000000	00000000000000000	00000000000000000	00000011237950000	00000000000000120	00000000000000000	00000000000011	00000000000000000
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	22900000000000	00000000000000000
	00000000000000000	00285999999903879	00286809999981087	00000000000000000	00000000000000000	00000001180000000	00000000000000	00000000112292800
	00299999999997589	99999999999999999	79999999999999999	00286844699998108	00000000299999999	00000000000000000	00000000000000	00000000000000000
	99999999999999999	99999999999999999	99999999999999999	98629999999999999	75899999999999999	00000000000000000	00000000000000	00000000000000000
	99999999999999999	99999999987089299	99999999988710144	99999999999999999	99999999999999999	00000000285999999	00000000000001	00000000286844699
	99999999999988708	99800979979200000	99794139239675400	99999999999988731	99999999988687019999	99808799999999999	12290000000000	99810858629999999
	19999978996997590	00000000089299998	00000000090144997	01451579413726577	97899399739000000000	99999999999999999	00000000000000	99999999999999999
	00000000000000081	06853000000000000	99547150000000000	35190000000000090	00000008199999789990	99999999999999988	00000000000000	99999999999999988
	99999789990000000	00002729999999999	00002771499999999	14515799542453800	00000000000000305	70892999980097997	00000000286809	73101451579413726
	00000000000309999	99999979410092900	99999979461293723	00000000000277158	99999999999999717	20000000000000008	99999810877999	77351900000000009

	999999999999999717	0001429470000000000	0001359415700000000	7999999999999999794	0000860000014999000	9299998069530000000	9999999999999999	0145157995424538000
	0000860000014999	0000000000000001859	0000000000000001873	6025137216901359300	0000000000000000000	0000000000002729999	9999999999999999	0000000000027715370
	0000000000000000	9999998860000000376	1999998674500000368	6267900000000000000	0189999999987000000	9999999999999979410	9999999999988731	9999999999999979460
	0000000189999999	0029999999999998889	6729979999999998889	0000000187362699999	3700003899999999988	0092900000142947000	0144997941392390	2513721690115936260
	8700000003700028999	9258659796799999999	9267265101249999999	8674861003686513999	7649669979399999999999	0000000000000000000	6754000000000000	7900000000000000000
	9999999998876498	9999999999999999999	9999999999999999999	8185999999888926899	9999999999999999999	0001859999999866000	0090144997995470	0001873626399867480
	6997939999999999	9999999999999999999	9999999999999999999	0683622273699999999	9999999999999999999	0000376002999999999	1500000000000000	6100368651398185990
	9999999999999999	9999999999999999999	9999999999999999999	9999999999999999999	9999999999999819999	9999889215865979670	0027714999999999	9999889326806836220
	9999999999999999	8669999961000077400	8657899961568078330	9999999999999999999	9500000639000028999999	9999999999999999999	9999999974612090	2736999999999999990
	9999999999999999	0029999999999999999	2029979999999999999	9999999999999999999	9999999999999997808	9999999999999999999	3723000013594157	9999999999999999990
	8499999950000069000	9999999758920000000	9999999758807230000	8657888796149697830	6000000149950000000000	9999999999999999999	0000000000000000	9999999999999999990
	0002899999999999	1429460000000000000	1359399500000000000	3982998185999999999	0000000000000002899999	9999999986399999610	0000001873199999	9999999865788796140
	9999999997808600000	0000000002999999999	0000000002997999999	9999999997580314199	9999999999000000000036	0007740002999999999	9867453000036867	9678733962998185990
	0001499000000000	9999999940000003750	9999999931100003680	6901359345386000000	9887789999999999999	9999999999997589	299799999999988	9999999999999997580
	0000000000002899	8900659999999999999	8900734899999999999	0000000000000029980	9999999999999937808	2900000142946000000	992672691012409	3141690139534538600
	9999999999990000	9999999999999999997	9999999999999993997	1859999999999993099	6000000149980000000000	0000000000000000029	9999999999999	0000000000000000029
	0003698877899999	5892900000142995000	5800723000135901130	6003683797773573390	0000016970700000000000	9999999999999999400	9999999999999999	9818599999999993090
	9999999999999999	0000600000037578400	0000000000367673740	9999999999999999999	7800009000289999999999	0000375890065999999	9999999999999999	9600368379773573390
	9999999780860000000	0000000854800179002	0000000864900800802	9999999758031416999	9999999999997900700000	9999999999999999999	9999999999999986	9999999999999999999
	01499800000000000	9999999999999999999	9979999999999999999	0135900615800000000	0004300000000000000000	9999997582290000001	5789956156007830	9999997580314169010
	0036970700000000000	9998811075000004150	9997941340500041550	0036765234080000000	0000189999999999999999	4299500000000000003	320299799999999	3590061580000000030
	7800090002899999	0000000000000000001	3000000000000000001	8648117072439818599	9999999999999999999995	7378400000000008580	9999999999975800	6765234080000086490
	9999999999999979	8799999999999999999	8997999999999999999	9999999999999999979	6870850070000000000000	0079002999999999999	0723000135935995	1170724398185999990
	0070000043000000	9999999999999999998	9999999999999999998	4141555404155598000	0300000007229999999999	9999999999980107500	0000000000000000	9999999999794141550
	0000000000001899	8709260750000000000	8731287337300000000	0000000000000001899	9999999999999999998	0004150000000000000	0000299799999999	4304155598000000000
	9999999999999999	0028600000807990999	0028637000815529999	8829999999999999999	8777999999999999999	0000001879999999999	999999311000036	0000001899882999990
	9999999999999988	9999999999999999999	9999999999999999999	9999999999999999968	9999999999999999999	9999999999999999999	830007348999999	9999999999999999990
	7085007000000000000	9999899002099999999	9999899003008999999	7312881457150000000	9997807299999999968	9999998870928075000	9999999999999999	9999998873128814570
	0030000007299999	9999999999999999999	9999999999999999999	0028664070815615000	7052999999999999999	0000000002860000080	9999997580072300	1500000002866407080
	9999999999999999	9999999999997588607	9999999999997598815	9999999999999999999	9708000000000459999	0799999999999999999	0013590113000000	1561309999999999990
	9999998877999999	9999999998870900999	5299999999887311008	9999998900301704999	9999999708000000007600	9999999999985000209	0000036767374000	9999999999998900301
	9999999999999999	9999999999999997916	9999999999999997911	9999999999999999999	0000000000000000076	9999999999999999999	0000086499080800	7049999999999999999
	9999999999999997	0000000009009999999	2200000091008999999	9999999999999999997	0000000000000000000000	9999999999999999999	299799999999999	9999999999999999999

	80729999999999887 08599999999999999 99970800000000085 99999999999970800 00000760000000000 00000000076000000 00000000000000067 00000000000072999 99999999999999997 77778084986000000 00000048999999999 99998877300000000 00073000000000000 00000789999999999 99999999999780849 86000000000000000 00000000000000000 00000000030999999 99999999999887719 99999999999999999 99997089999999999 99999999780780000 00031000000000000 00000008499900000 00000000000029999 97575]	999979160000008330 000000000000000008 390000000000000000 000073900000000000 807999999999999999 999777775892787500 000000000489999999 999999890013000000 000081300000000000 000008739999999999 999999999975892787 500000000000000000 000000000000000000 000002729999999999 999999889992999999 999999999999999784 299999999999999999 758864000000273000 000000000000000927 953000000000000000 02859996355]	999979132200008404 900000000000000008 461800000000000000 000074764000000000 817189999999999999 999777775799800129 000000000488539999 999999900021540000 000082154000000000 000008813999999999 999999999975799800 161000000000000000 000000000000000000 000002771499999999 999999890001439999 999999999999999784 244999999999999999 757940990000277150 000000000000000928 747150000000000000 028656644355]	58815615099999887 31101722999999999 99979133443000091 01722999999979133 52700840512700000 00000000084619650 00000000000000074 77172000000081720 52999999999999997 77775895218590010 00000488497899999 99998900216146000 00082161180000000 00000882092599999 99999999999758022 18617730000000000 00000000000000000 00000000027715379 99999999999890001 52679999999999999 99997842868799999 99999999757965180 10027715380000000 00000009288292538 00000000000028653 76435390]	0670000000000000072999 99999999999999997777 78084986000000000000 48999999999999988773 00000000000730000000 00000000007899999999 99999999999997808498 60000000000000000000 00000000000000000000 00309999999999999999 98877199999999999999 99999999708999999999 99999999976078000000 03100000000000000000 00849990000000000000 00002999997575]	97588079999999999988 70900999999999999999 99979160000000000900 99999999999997916000 00083300000000000000 00000083900000000000 00000000000007390000 00000000807999999999 99999999997777777589 27875000000000000048 99999999999999989001 30000000000008130000 00000000000000873999 99999999999999999997 58927875000000000000 00000000000000000000 00000000000000272999 99999999999999988999 29999999999999999999 99997842999999999999 99999999758864000000 02730000000000000000 00009279530000000000 00000000028599996355]	999999979413405 000415530000000 000000000189975 999999999999999 999999999999999 887312873373000 000000286570008 155299999999999 999999999999998 900300899999999 999999999999999 999999999999975 881552999999988 731100899999999 999999791322000 000910089999999 997913220000840 490000000000000 000846180000000 000000000007476 400000000081718 999999999999999 977777579980012 900000000048853 999999999989002 154000000008215 400000000000000 881999999999999 999999997579980 016100000000000 000000000000000 000000000000277 149999999999999	975881561509999988 731101722999999999 999791334430000910 172299999997913352 700840512700000000 000000846196500000 000000000007477172 000000081720529999 999999999977777589 521859001000000048 849789999999989002 161460000008216118 000000000000882092 599999999999999997 580221861773000000 000000000000000000 000000000000277153 799999999999989000 152679999999999999 999978428687999999 999999757965180010 027715380000000000 000092882925380000 000000002865376435 390]

							9890001439999999999999999999978424499999999999999975794099000027715000000000000000928747150000000000000028656644355]	
Sample 2	[1069, 1073, 1042, 419, ..., -	[10797, 10831, 10519, 4301, ..., -	[1080773, 1083519, ..., -	[108078342, 108353119, ..., -	[1069, 1073, 1042, 419, ..., -	[10797, 10831, 10519, 4301, ..., -	[1080773, 1083519, ..., -	[108078342, 108353119, ..., -
	12000000000000000	11800000000000000	11227000000000000	11227810000000000	12000000000000000	11800000000000000	11225000000000	11226670000000000
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000	00000000000000000
	00130000000000000	12000000000000000	12137000000000000	00121363500000000	13000000000000000	00001200000000000	00000000012137	00001213635000000
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000000	00000000000000	00000000000000000
	00000000000000013	00000000001200000	00000000001213700	00000000000000012	00000001300000000	00000000000000120	00000000000000	00000000000000121
	00000000000000000	00000000000000001	00000000000000001	13635000000000000	00000000001300000	00000000000000000	00000000000121	36350000000000000
	00000130000000000	20000000000000000	21370000000000000	00000121363500000	00000000000000000	00012000000000000	37000000000000	00012136350000000
	00000000000000001	00000000118000000	00000000112220000	00000000000000001	00130000000000000	00000000000000012	00000012137000	00000000000000012
	20000000000000000	00000000000000000	00000000000000000	12223820000000000	00000000000032999999	00000000000000000	00000000000000	13635000000000000
	00000000000000400	00000450000000000	00000441700000000	00000000000000441	99999978699999999999	00000000034499999	00000012137000	00000000034317869
	00000000000000000	00000000000000000	00000000000000000	76000000000000000	99999887078299999966	99999981999999999	00000000000000	99999982425322999
	00000000000000000	00000000000000000	00000000000000000	00000000000000000	99998778999796000999	99999998708669999	00000034317999	99999998873074978
	00000000000000000	00000344999999999	00000343179999999	00000000000000000	99999775999999999999	99877983006799801	99999824293999	99877487500771463
	32999999999999786	98199999999999999	98242939999999999	34317869999999824	78086000000143990000	02499999997780999	99999999988730	78568489997848549
	99999999999999998	99887086699999978	99887307497999877	25322999999999998	00000000000000000000	99999997589290000	74979999877478	29999999758031416 9
	87078999999987000	00000867998009527	48000877127938678	87307497859987749	00509999988785000000	01429470000000000	00771379378317 9	01359362679000000
	00078999789979979	54799997583290000	02952997580072300	77087714653893406	00000000000000000000	00000000000005029	99999784840999 9	00000000000005020
	29999978086000000	14294700000000000	13594157000000000	29320575803141690	00000006900000000000	99989013000000000	99999758007230 0	69189013774190000 0
	14999000000000000	00000000000050299	00000000000050207	13593626790000000	00000000000000000000	00000000000000000	01359415700000 0	00000000000000000
	00000000000000509	99890130000000000	99890137720000000	00000000000000502	00000000084999999995	00000774000000000	00000000000000	00000778376400000 0
	99998875000000000	00000000000000000	00000000000000000	06318901377419000	99999999999974400000	00000000000000000	00502079989013 7	00000000000000000
	00000000000000000	00007740000000000	00007782900000000	00000000000000000	30000006502100000159	00000000009299999	72000000000000	00000000009388803
	00000000690000000	00000000000000000	00000000000000000	00000000778376400	99999971300000000000	99999999999999974	00000000000000	99999999999999975

	0000000000000000	0000000000725965997	0000000000726816837	0000000000000000	0000000000000000649	9300002860000726 18	00778290000000	3174932864760726 93
	0000000000000006	9999978860000000	9999978901900000	0000000000000007	870001699999971300	9000034799999978 35	00000000000000	3221434856689978 37
	4997999999971500	0000000000000000	0000000000000000	2590677449978300	0000000000000000	0000000000000000	00000000000009	2646000000000000
	0000000000000000	0000000009229999 99	0000000009304899 99	4800000000000000	0064998000000163999	0000000007258820 01	387899999999999	0000000007267334 51
	0000000000000000	9999999999791634 00	9999999999791344 00	0000000000000000	997130000000000000	8899999978350000 00	9999999753116100	0831859783726460 00
	0859999999999999	0000000000892299 99	0000000009304899 99	0930580499999999	0000000000000065 0099	0000000000000000	2865000726840 71	0000000000000000
	9997080000000000	9999999999999791	9999999999999791	9997913558500000 00	0000001639999971300	7259660000001889 99	0034857999978 37	7269071482001883 18
	0000859999999999	6000000000000000	3440000000000000	0000930580499999	0000000000000000000	9978350000000000	36000000000000	5978372646000000 00
	9999999997080000	0922999999999999 99	0930489999999999 99	9999999997913558	0000000000006500900	0000000000000007	00000000000726	0000000000000007
	0000000000000085	9978160000000000 00	9979134400000000 00	5800000000000093	0000001639999971300 6	2600700000001889999	6607718829999978	2687363620018831 85
	9999999999999999	0000000092299999 99	0000000093048899 99	0580499999999999	0000000000000000000	9781500000000000	373600000000000	9783726460000000 00
	7080000000000000	9999999999979160 00	9999999999979134 4	7913558500000000	0000000000006501200	0000000000000000	00000000000726	0000000000000000
	0000000859999999	0000000000000921	0000000000000930	0000000930580499 9	0000016399999713000 0	0007260870000000018	817260000188299	0007268736366000 18
	9999999999570800	9999999999999999 99	4899999999999999 99	9999999999979135	0000000000000000000	8999997835000000 00	997837360000000	8316597837264600 00
	0000000000000008	9997916000000000 00	9997913440000000 00	5850000000000009	0000650030000001699	0000000000000000	00000000000000	0000000000000000
	5999999999999999	0000000092299999 99	0000000093048899 99	3058049999999999 99	9999713000000000000	0000000007261180 00	0000072678703 00	0000000007268748 64
	9999997080000000	9999999999999791 4	9999999999997799 6	9999997913558500 00	0000000000000000649	0018899999783500 00	0018829999783 73	1018831859783726 46
	0000000000085999	0000000090000000	3700000000000000	0000000000093058	6700000000169999997	0000000000000000	6000000000000	0000000000000000
	9999999999999999	8883300000000000	8004849000000000	0499999999999999	1300000000000000000	0000007260180000 00	00000000000000	0000007268636796 00
	9979300000000000	0839000000000000	0846180000000000	9979982348000000	0000000000000064997	1889999978350000 00	00072678703000 0	1883185978372646 00
	0000000076000000	0000000008880000	0000000008117190000	0000000008405127 0	9997150000000000000	0000000000000000	0188299997837 36	0000000000000000
	0000000076000000	0000000000000000 72	0000000000000000 72	0000000008461965 0	0000000000000000000	0000072388100000 00	00000000000000	0000072677731100 0
	0000000000000000	5036999999794500 00	6685959999794649 00	0000000000000000	0000000016300000600	0188999997835000 00	00000000000000	0188318597837264 60
	0730000000000000	0000000000075399999	0000000000762089 99	0817205300000000	0000000000000000000	0000000000000000	00072678815000 1	0000000000000000
	0000000065001999	9999999997916000 00	9999999997913440 00	0000000007267613 5	0000000000000064999 8	0000000072596599 7	882999978373600	0000000007269071 27
	9999712000000000	0000000000000075299	0000000000000076208	4599794649220000 0	9997080000000082999	8360000000000000	00000000000000	7900480000000000
	0000066999999999	9999999999999999 99	9999999999999999 99	0000076217139999 9	9999999970800000000	0000000000000000	00000000726777	0000000000000000
	9999970800000000	9999999999999999 99	9999999999999999 99	9999979135585000 00	0000789999999999708	0000000009986000 92	990000188299997	0000000010544993 13
	0000000006699999	9758829000001425 59	8758007230001355 52	0000000000762171 3	0000000000000789999	5000000000000000	837360000000000	4133400000000000 00
	9999999999999999	0000000000000000	3400000000000000	9999999999999999	9999999970800000000	0000000000000000	00000000000000	0000000000000000
	9999999999999999	0000002490000000 44	0000002492600000 44	9999999999999999	0000078999999999708	7269999997916000 00	007267009400000	7341308997913978 40
	9780860000014999	9999890014000000	9629890021170000 00	9758031416901355 4	0000000000078999999	0008389999999979	0188299997837 36	0009080526999999 79
	0000000000000000	0000007740000149999	0000078332001402 99	8229600000000000	9999970800000000000	1600000000000008659	00000000000000	1397910000000008744

	0000000002400000000449999887780000000000000069000001999999999999999999999999999999997575]	9999999999999999999999999996355]	99999999999999999999999999644355]	000000000243301500044962719002124980000000007833980139684899999999999999999999999964435390]	0689999999999957079999999999999996540]	999999999791600000000000000865999999999979160000000000000866999999999979160000000000085699999999979160000000000007699999999999999791599999999999996470]	0000000000000007268173078901900000000000000000000000000000000000000001054626093405000000000000000000000000000000000000734039999791391000000907979999999999791391000000000000874389999999999791391000000000000874389999999999791391000000000000087564999999999791391000000000000964339999999999791391000000000007787299999999999791390999999999999644355]	838999999979139776000000000087448429999999791397760000000000675736899999979139773000000000864420899999997913978200000000778811399999999791397879999999964435275]