

BAB II

TINJAUAN PUSTAKA

1.1 Penelitian Terdahulu

Penelitian terdahulu merupakan sebuah kumpulan penelitian yang sudah ada dengan tema/genre yang hampir sama dengan program yang akan dibuat. Adapun penelitian terdahulu yang hampir sama dengan skripsi ini tercantum dalam tabel 2.1.

Tabel 2.1 Tabel Penelitian Terdahulu

| No | Judul | Latar Belakang | Metode Penelitian | Hasil Penelitian | Penulis |
|----|---|---|--|--|---|
| 1. | Lanjutan Paralel Algoritma Rabin-Karp untuk Pencocokan String | <ol style="list-style-type: none"> Ada banyak pemrosesan paralel Application Programming Interfaces (API) yang tersedia, seperti OpenMP, MPI, CUDA MapReduce, dll. Perbandingan hasil implementasi serial dan paralel akan memberi kita wawasan tentang bagaimana kinerja dan efisiensi dicapai melalui berbagai teknik paralelisme | <ol style="list-style-type: none"> Metode Algoritma Rabin-Karp dalam OpenMP dan CUDA untuk mencapai paralelisme | <p>Hasil eksperimental juga memberi kami wawasan tentang kinerja metodologi implementasi yang berbeda dan menyimpulkan bahwa eksekusi sekuensial adalah varian yang lebih cepat untuk ukuran data kurang dari 100 KB sedangkan OpenMP memberikan hasil terbaik ketika ukuran data berada dalam kisaran 100 KB - 1 MB. Tetapi ketika ukuran data meningkat melebihi 1 MB yang merupakan kasus biasa di sebagian besar aplikasi real-time maka pemrograman CUDA dengan Rabin-Karp adalah solusi yang paling efisien.</p> | <ol style="list-style-type: none"> Omkar Sunil Joshi. Bhargavi R. Upadhyay. Supriya M. (Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru Amrita Vishwa Vidyapeetham Amrita University,2018) |
| 2. | Aplikasi Deteksi Dini Plagiarisme Judul Tugas Akhir | <ol style="list-style-type: none"> Praktek plagiarisme yang sangat sering dilakukan di kalangan | <ol style="list-style-type: none"> Metode yang digunakan dalam aplikasi | <ol style="list-style-type: none"> Aplikasi deteksi dini plagiarisme | <ol style="list-style-type: none"> Drajad Uji Cahyono |

| No | Judul | Latar Belakang | Metode Penelitian | Hasil Penelitian | Penulis |
|----|--|--|--|---|---|
| | Mahasiswa Sekolah Tinggi Ilmu Kesehatan Yayasan Rs. Islam Surabaya Dengan Algoritma Rabin-Karp | <p>akademisi baik di tingkat sekolah maupun di perguruan tinggi.</p> <p>2. Tindakan plagiarisme ini dapat berdampak pada penurunan kreatifitas siswa maupun mahasiswa.</p> <p>3. Berbagai cara persuasif oleh perguruan tinggi diupayakan bagi mahasiswa untuk mencegah kemungkinan dilakukannya kegiatan plagiarisme.</p> | <p>ini yaitu dengan membandingkan kemiripan judul tugas akhir yang telah ada di database dengan judul tugas akhir yang tengah diajukan oleh mahasiswa. Dengan menggunakan algoritma Rabin-Karp</p> | <p>tugas akhir sudah cukup mudah untuk digunakan.</p> <p>2. Performa hasil persentase similaritas dari algoritma Rabin-Karp memiliki ketergantungan dengan nilai K-gram yang diberikan.</p> | (Universitas Nahdlatul Ulama Surabaya, 2018) |
| 3. | Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing | <p>1. Fenomena plagiarisme dalam dunia akademik bukanlah hal yang baru di Indonesia.</p> <p>2. Budaya <i>Copy-Paste</i> dalam penyelesaian tugas</p> | <p>metode n-gram, Jaccard similarity dan algoritma winnowing.</p> | <p>Dari hasil penelitian di atas dapat dipahami n-gram sangat mempengaruhi hasil dari similarity, penggunaan n-gram yang tepat sangat diperlukan. Tingkat drajat kesamaan pada dua dokumen/sample akan menghasilkan nilai similarity yang berbeda-beda jika nilai n-gram dan w-gram yang dimasukan itu berbeda beda nilainya.</p> | <p>1. Sunardi, 2. Anton Yudhana, 3. Iif Alfiatul Mukaromah, (Program Studi Teknik Elektro, Universitas Ahmad Dahlan, 2018),</p> |
| 4. | Implementasi Algoritma Rabin-Karp untuk Membantu Pendeteksian Plagiat pada Karya Ilmiah | <p>1. Adanya orang-orang yang tidak bertanggung jawab menyalahgunakan karya ilmiah dengan mengutip sebagian atau seluruh karya dan</p> | <p>Metode Algoritma Rabin-Karp</p> | <p>Berdasarkan hasil pengujian dan pengukuran skala hasil dari kuesioner dengan Likert's Summated</p> | <p>1. Doddi Aria Putra, 2. Herry Sujaini, 3. Helen Sasty Pratiwi.</p> |

| No | Judul | Latar Belakang | Metode Penelitian | Hasil Penelitian | Penulis |
|----|--|--|-----------------------------------|--|---|
| | | karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai | | Rating menunjukkan bahwa aplikasi sangat positif dan dinilai berhasil | (Program Studi Teknik Informatika Fakultas Teknik Universitas Tanjungpura, 2015) |
| 5. | Sistem Pengecekan Plagiasi Judul Tugas Akhir Menggunakan Algoritma Winnowing di Fakultas Sains dan Teknologi Universitas PGRI Yogyakarta | <ol style="list-style-type: none"> Sistem pengecekan plagiasi tugas akhir sangat penting guna mengidentifikasi judul-judul dengan cara yang mudah. Judul yang telah masuk akan diidentifikasi secara cepat sehingga akan diketahui kesamaan dengan judul yang sudah ada. | <p>Metode Algoritma Winnowing</p> | <p>hasil penelitian dapat diketahui bahwa nilai n-gram berpengaruh terhadap hasil dari similarity.</p> | <ol style="list-style-type: none"> Rafi Purnamasari. Muhammad Fairuzabadi. Ahmad Riyadi. <p>(Program Studi Informatika, Universitas PGRI Yogyakarta, 2021)</p> |

2.2 Plagiarisme

Plagiarisme ialah perbuatan salah yang sungguh-sungguh karena mengambil karya orang lain serta mengakuinya selaku karya sendiri. Aksi plagiarisme merendahkan moral serta harkat pelakon dan berakibat pada disintegritas sivitas akademik sebab aksi melansir tanpa izin wajib dicegah serta apabila telah terjalin wajib diatasi. Sikap menjiplak karya orang lain tanpa mencantumkan sumber serta nama pengarang asli berdampak kurang baik kepada pengembangan ilmu karena dengan melansir tidak hendak timbul pemikiran baru. Pada tahun 2010, Direktorat Jenderal Pembelajaran Besar sudah menghasilkan peraturan tentang metode penangkalan serta penanggulangan plagiarisme tercantum sanksi buat dosen, mahasiswa, serta calon guru besar sekalipun (Pratama, Faisal, and Hanani 2019; Wibowo 2012).

Plagiat bisa dilihat sebagai pelaku tindak pidana sebab mencuri hak cipta orang lain. Semacam aksi seseorang mahasiswa yang mengkopi sebagian bagian penyusunan penelitian tanpa ijin yang legal dari penulis aslinya. Di dunia pembelajaran, pelaku dalam plagiarisme bisa menemukan hukuman

berat dari sekolah/ universitas. Pelaku plagiat disebut selaku plagiator (Pratama, Faisal, and Hanani 2019).

Dalam penelitian sebelumnya bentuk-bentuk dalam praktek plagiarisme yang sering terjadi adalah seperti yang di bawah ini (Gipp and Meuschke 2011; Salmuasih and Sunyoto 2013):

1. *Copy-Paste plagiarism*, mengkopi atau menyalin setiap kata maupun kalimat tanpa ada perubahan.
2. *Disguised plagiarism, plagiarisme* yang sering dilakukan dalam bidang akademis yang tergolong ke dalam praktek untuk menutupi bagian yang disalin, diidentifikasi ke dalam empat teknik, yaitu :
 - a) *Shake-paste*, dengan menyalin dan menjadikan satu kalimat atau paragraf yang berasal dari berbagai sumber atau referensi dengan sedikit penyesuaian kata yang dibutuhkan untuk membuat teks yang koheren.
 - b) *Expensive plagiarism*, menyisipkan kata tambahan.
 - c) *Contractive plagiarism*, menggambarkan sebuah ringkasan dengan melakukan pemangkasan bahan yang akan disalin.
 - d) *Mosaic plagiarism*, melakukan penggabungan segmen kata dari sumber atau referensi yang berbeda dengan cara mengubah urutan kata, menggantikan kata-kata dengan sinonim atau menambah ataupun juga melakukan penghapusan kata-kata.
3. *Technical Disguise*, Merupakan metode meringkas buat menyembunyikan konten plagiarisme dari deteksi otomatis dengan menggunakan kelemahan dari tata cara analisis bacaan dasar, misal dengan mengubah huruf dengan simbol huruf asing.
4. *Undue Paraphrasing*, terencana menuliskan ulang pemikiran asing dengan pemilihan kata serta style plagiator dengan menyembunyikan sumber asli.

5. *Translated Plagiarism*, merupakan Langkah mengkonversikan atau merubah konten dari satu Bahasa diubah ke Bahasa lain.
6. *Idea Plagiarism*, memakai ulang suatu pemikiran atau gagasan asli dari sebuah data sumber teks tanpa bergantung bentuk dari teks sumber.
7. *Self Plagiarism*, menggunakan Sebagian tulisan ataupun keseluruhan dari tulisan pribadi yang juga tidak dibenarkan secara ilmiah.

2.3 Metode Pendeteksi Plagiarisme

Terdapat 3 (tiga) macam dalam Metode Pendeteksi plagiarism, diantaranya *Fingerprint*, Metode *SstringMatching*, dan Metode *Trre-Matching*. Gambar 2.1 menunjukkan Jenis-jenis dalam metode pendeteksi plagiarisme (Muzgovoy, 2007 dalam Sunardi, A. Yudhana 2017)

1. Metode Fingerprinting

Metode Fingerprinting ialah tata cara yang dipergunakan untuk mengetahui sebagian atribut serta struktur dari dokumen. Atribut tersebut diantara lain adalah jumlah kata disetiap baris, jumlah kata yang unik, serta jumlah kutipan pendek.

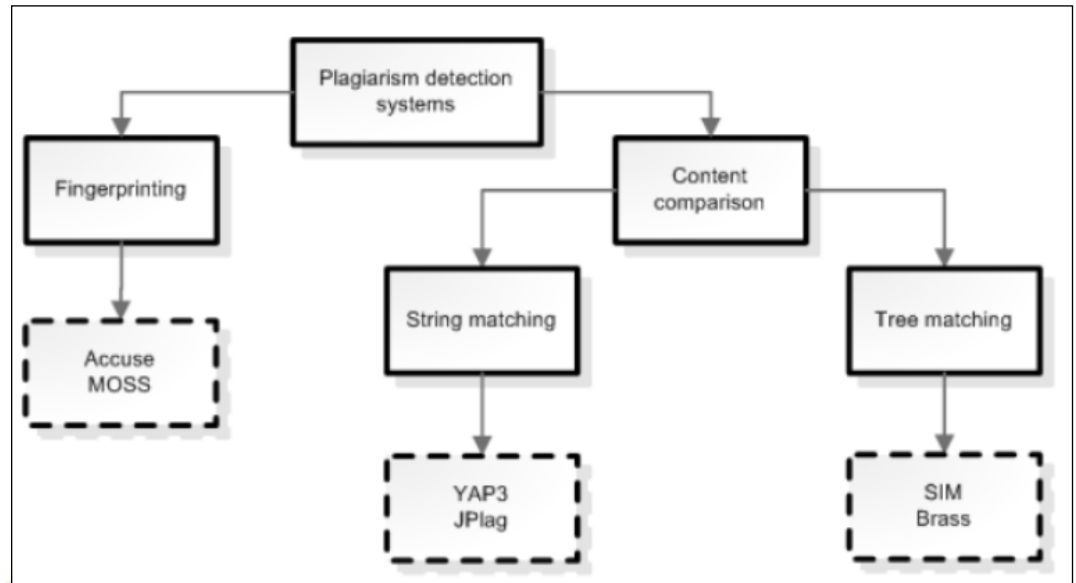
2. Metode String-Matching

Metode string-matching atau pencocokkan string merupakan tahapan membandingkan antara string dokumen satu dengan string dokumen yang lain serta dihitung kesamaannya. Contoh dalam sistem yang mempergunakan metode pencocokkan string yaitu YAP3 dan Jplag.

3. Metode Tree-Matching.

Metode Tree-Matching merupakan metode pembanding dengan syarat dokumen yang akan dibandingkan wajib mempunyai peraturan struktur

yang sama. Contoh sistem yang menerapkan *metode TreeMatching* antara lain SIM dan Brass.



Gambar 2.1 Skema dari Metode Pendeteski Plagiarisme

2.4 Metode Pencocokan String dengan Rabin-Karp

Pencocokkan string ataupun string matching ialah sesuatu tata cara yang digunakan buat menciptakan sesuatu keakuratan/ hasil dari satu ataupun sebagian pola bacaan yang diberikan. Pencocokkan string ialah pokok bahasan yang berarti dalam ilmu komputer sebab teks ialah wujud utama dari pertukaran data antar manusia, misalnya pada literatur, karya ilmiah, serta halaman website (Ginting and Utomo 2019).

Algoritma Rabin-Karp merupakan Algoritma acak yang sederhana dan cenderung dapat berjalan di waktu linear. Algoritma Rabin-Karp merupakan algoritma pencocokkan string yang diciptakan oleh Michael O.Robin dan Richard M. Karp pada tahun 1987. algoritma Rabin-Karp digunakan untuk mencocokkan kata dengan *Single Patern* (pencarian pola tunggal), tetapi lebih efektif bila digunakan untuk pencocokan pola *multi pattern serach* (pola multi pencarian). Algoritme Rabinkarp mempercepat pemeriksaan Gunakan fungsi

hash kata demi kata. Fungsi hash adalah fungsi Ini mengubah kata menjadi nilai yang disebut nilai hash. Fungsi hash yang digunakan oleh algoritma Rabin-Karp adalah Rolling hash.

Pada intinya, Algoritma *Rabin-Karp* akan mengukur perbandingan antara nilai dari hash yang merupakan *pattern* (string masukkan) dan *substring* pada data teks. Apabila memiliki kesamaan, maka proses perbandingan terhadap karakter-karakternya akan dilakukan sekali lagi. Dan bila tidak ada sama, selanjutnya *subtring* akan dipindahkan ke kanan. Kunci yang utama pada performa algoritma Rabin-Karp ini ialah efisien dalam perhitungan terhadap nilai hash substring yang dilakukan pada waktu penggeseran.

2.5 Rolling Hash

Hashing adalah cara untuk mengonversi string menjadi Nilai unik dengan panjang tertentu (panjang tetap), digunakan sebagai penanda (Tanda tangan) string. Panjang string sesuai dengan nilai k-gram yang ditentukan. Fungsi hash atau fungsi hash adalah metode membuat sidik jari dari berbagai file Memasukan data. Fungsi hash akan menggantikan atau mengubah data Buat sidik jari, biasanya disebut hash .(Dedi Leman, Gunadi Widi Nurcahyo 2015)

Nilai dari hash yang akan dilakukan pencarian yang menggunakan fungsi hash didalam algoritma Rabin-Karp ialah representasi dari sebuah nilai ASCII yang memiliki kepanjangan kata dari *American Standar Code for Information Interchange* dimana ASCII tersebut menempatkan angka numerik di setiap karakter, tanda baca, angka, dan karakter yang lainnya. ASCII menyuplai 256 kode yang dapat dibagi kepada dua himpunan standart dan meluas yang dimasing-masing terdiri dari 128 karakter. Himpunan ini merepresentasikan total kombinasi dari 7 atau 8 bit, yang kemudian menjadi angka dari bit dalam 1 byte. ASCII standar menggunakan 7 bit untuk tiap kode dan menghasilkan 128 kode karakter dari 0 sampai 127 (heksadesimal 00H sampai 7FH).

Himpunan ASCII yang diperluas menggunakan 8 bit untuk tiap kode dan menghasilkan 128 kode tambahan dari 128 sampai 255 (heksadesimal 80H sampai FFH)

Rolling Hash ialah fungsi hash yang inputnya dikelompokkan ke dalam blok, dan blok bergerak melalui input secara keseluruhan. Beberapa fungsi hash memungkinkan perhitungan dinamis dari rolling hash. Nilai hash baru dapat dengan cepat dihitung dari nilai hash lama dengan menghapus nilai lama dari grup hash dan menambahkan nilai baru ke grup (Dedi Leman, Gunadi Widi Nurcahyo 2015).

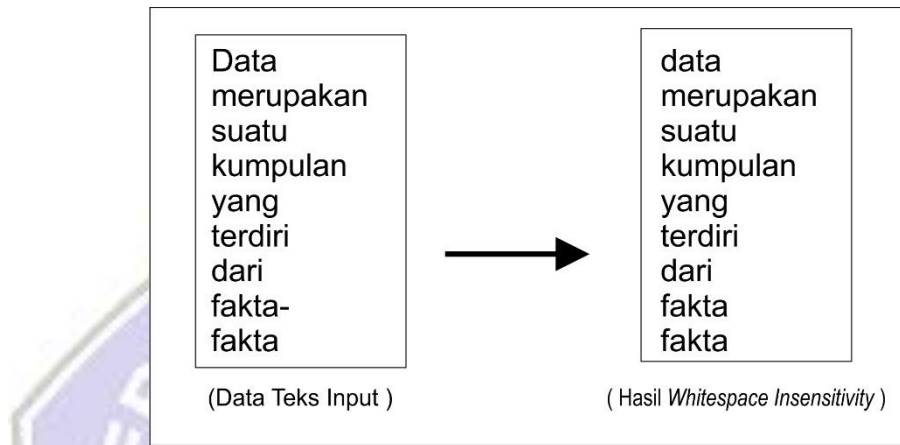
Kunci kinerja algoritma Rabin-Karp adalah perhitungan efektif dari nilai hash dari substring berturut-turut dalam teks. Algoritma Rabin-Karp melakukan perhitungan nilai hash dengan memperlakukan setiap substring sebagai angka dengan basis tertentu. Perhitungan nilai hash dengan rolling hash dapat dilihat pada persamaan (1) Kelebihan dari rolling hash adalah untuk nilai hash selanjutnya dapat dilakukan dengan persamaan (2) sehingga tidak perlu melakukan iterasi dari indeks pertama sampai akhir. Ini tentu saja akan menyimpan nilai hash dari string.

2.6 Tahap Text Preprocessing

Pada biasanya informasi berbentuk bacaan mempunyai format yang besar, ada noise pada informasi, serta bentuk perkataan yang tidak bagus. Buat itu dicoba jalan keluar akta jadi perkataan, tutur alhasil bisa dicoba pemrosesan informasi dengan lebih bagus. Sesungguhnya ini ialah bagian dari text preprocessing. Dengan cara biasa dalam langkah– langkah preprocessing ialah

1. *Whitespace insensitivity*

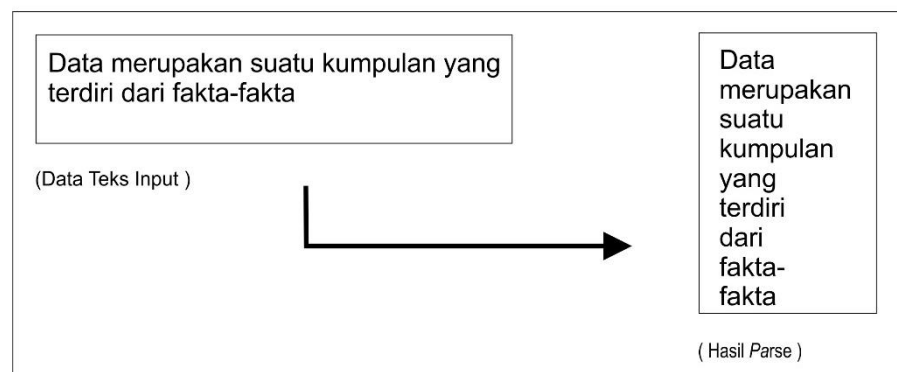
Whitespace insensitivity ialah jenjang yang melenyapkan karakterkarakter khusus semacam ciri baca serta mengganti seluruh token ke wujud graf kecil(lower case). Ilustrasi cara *Whitespace insensitivity* bisa diamati pada Gambar 2.2.



Gambar 2.2 Contoh dari Proses *Whitespace insensitivity*

2. *Parsing*

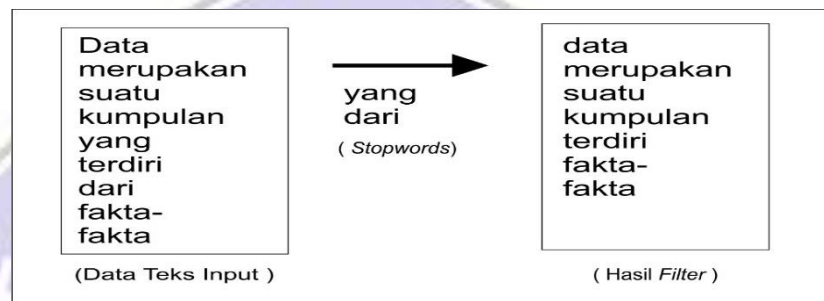
Parsing ialah jalan keluar perkataan jadi tutur. Ilustrasi cara *Parsing* ada pada Gambar 2.3



Gambar 2.3 Contoh dari Proses

3. Filtering

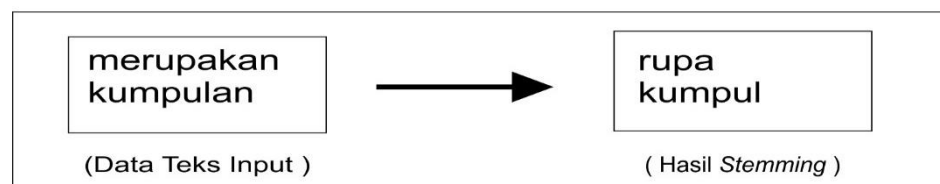
Filtering ialah tahap dalam pengambilan data teks kata yang penting dari hasil token. Pengambilan data kata ini bisa dilakukan dengan menghilangkan kata yang kurang penting (*stopwords*) atau menyimpan kata-kata penting (*wordlist*). Stoplist/stopwords adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bags-of-words*. Contoh stopwords adalah “yang”, “dan”, “dari” dan sebagainya. Contoh proses filtering terdapat dalam Gambar 2.4



Gambar 2.4 Contoh dari Proses

4. Stemming

Stemming ialah langkah mencari root data teks dari masing- masing data teks hasil filtering. Pada langkah ini dicoba cara pengembalian pada pembuatan data teks ke dalam representasi yang serupa. Pada data teks dengan bahasa indonesia, stemming susah diaplikasikan sebab pada bahasa indonesia tidak mempunyai metode wujud dasar yang permanen. Ilustrasi cara stemming ada dalam Gambar 2.5



Gambar 2.5 Contoh dari Proses

2.7 Parsing K-Gram

Penguraian yang digunakan adalah rangkaian panjang Ditetentukan oleh K Sebagian besar istilah yang digunakan adalah kata-kata. Mengurai K-gram Adalah metode yang digunakan untuk menghasilkan kata atau karakter. Metode k-gram ini digunakan untuk mengambil blok karakter huruf K kata secara berurutan dibaca dari teks sumber Akhir dari dokumen. Pada tahap ini digunakan sebagai Masukkan string untuk fase hash (Salmuasih and Sunyoto 2013; Stephanie, Naga, and Mawardi 2021).

Dalam model Markov, nilai K-Gram yang sering digunakan adalah 2-gram (bigram), 3-gram (trigram), 4-gram, 5-gram, dll.). bahasa alami Memproses dan menggunakan K-Gram lebih sering, proses resolusi tag (tokenisasi) Gunakan 3 gram dan 4 gram, sedangkan 2 gram digunakan untuk parsing Kalimat, seperti dalam part of speech (POS). Gunakan 2 gram dalam tokenisasi Akan menyebabkan tingkat kontras antar karakter akan semakin besar. Misalnya, kata "duduk" dan "saya" adalah dua kata yang sangat berbeda. Dengan menggunakan metode bigram untuk mencari kesamaan, hasilnya bigram adalah kata "duduk" akan menghasilkan bigram du, ud, du, uk and Kata "saya" menghasilkan bigram sa, ay, ya. Dengan cara ini, akan ada Kesamaan multi-kata dalam pemrosesan nilai kesamaan. Akan tetapi, bila mempergunakan 3-Gram ("duduk" = dud, udu, duk dan "saya" = say, aya) atau 4-Gram ("duduk" = dudu, uduk dan "saya" = saya) akan lebih mengecilkan pada kemungkinan sampai terjadi kesamaan terhadap kata yang dimana memiliki struktur yang berbeda.

Gambar 2.4 merupakan contoh penerapan nilai 5-Gram, dimana semakin kecil kemungkinan terjadinya kesamaan kata karena K-Gram semakin besar maka semakin kecil adanya kesamaan pada kata.

Teknik Informatika Universitas Muhammadiyah Ponorogo



```
{tekni}{eknik}{kniki}{nikin}{ikinfn}{kinfo}{infor}{nform}
{forma}{ormat}{rmati}{matik}{atika}{tikau}{ikaun}{kauni}
{auniv}{unive}{niver}{ivers}{versi}{ersit}{rsita}{sitas}
{itasm}{tasmu}{asmuh}{smuha}{muham}{uhamm}{hamma}
{ammad}{mmadi}{madiy}{adiya}{diyah}{iyahp}{yahpo}{ahpon}
{hpono}{ponor}{onoro}{norog}{orogo}
```

Gambar 2.6 Parsing dengan 5-Gram

2.8 Persentase kemiripan

Untuk mengukur hasil dari tingkat kemiripan, Rentang persentase nilai kemiripan yang dipergunakan adalah (Alamsyah 2017)

1. 100% : Hasil dari pengujian 100% menunjukkan bahwa data teks mempunyai kemiripan yang sama dikarenakan mulai awal hingga akhir terdapat isi yang sama persis
2. >50% : Hasil dari pengujian 50% menunjukkan bahwa data teks mempunyai kemiripan yang mendekati dokumen aslinya
3. 15-50% : Hasil dari pengujian 15-50% menunjukkan bahwa data teks mempunyai kemiripan dengan tingkat menengah
4. <15% : Hasil dari pengujian 15% menunjukkan bahwa data teks mempunyai kemiripan yang sedikit
5. 0% : Hasil dari pengujian 0% menunjukkan bahwa data teks tidak mempunyai kemiripan atau berbeda.

2.9 CodeIgniter

CodeIgniter ialah aplikasi sumber terbuka yang berbentuk kerangka kerja PHP dengan model MVC(Model, View, Controller) buat membangun web website dinamis dengan memakai PHP. CodeIgniter mempermudah pengembang website buat membuat aplikasi website dengan kilat serta gampang dibanding dengan buatnya dari dini. CodeIgniter dirilis awal kali pada 28 Februari 2006 (Enterprise 2015).

Kerangka kerja secara simpel bisa dimaksud kumpulan dari fungsi- fungsi/ prosedur- prosedur serta kelas- kelas buat tujuan tertentu yang telah siap digunakan sehingga dapat lebih memudahkan serta memesatkan pekerjaan seseorang pemrogram, tanpa wajib membuat guna ataupun kelas dari dini.

1. Memesatkan serta memudahkan pembangunan suatu aplikasi website.
2. Relatif mempermudah dalam proses pemeliharaan sebab telah terdapat pola tertentu dalam suatu kerangka kerja(dengan ketentuan pemrogram menjajaki pola standar yang terdapat).
3. Biasanya kerangka kerja sediakan fasilitas- fasilitas yang universal dipakai sehingga kita tidak butuh membangun dari dini(misalnya validasi, ORM, paginasi, pangkalan informasi ganda, scaffolding, pengaturan tahap, penindakan pengecualian, serta lain- lain.
4. Lebih leluasa dalam pengembangan bila dibanding CMS.
5. Menunjang pembuatan CRUD otomatis(Create, Read, Pembaharuan, Delete) dari pihak ketiga.

Model View Controller ialah sesuatu konsep yang lumayan terkenal dalam pembangunan aplikasi website, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi bersumber pada komponen utama yang membangun suatu aplikasi semacam manipulasi informasi, antarmuka pengguna, serta bagian yang jadi kontrol aplikasi. Ada 3 tipe komponen yang membangun sesuatu pola MVC dalam sesuatu aplikasi ialah

1. View, ialah bagian yang menanggulangi logika presentasi. Pada sesuatu aplikasi website bagian ini umumnya berbentuk berkas

- templat HTML, yang diatur oleh controller. View berperan buat menerima serta merepresentasikan informasi kepada pengguna. Bagian ini tidak mempunyai akses langsung terhadap bagian model.
2. Model, umumnya berhubungan langsung dengan pangkalan informasi buat memanipulasi informasi(insert, pembaharuan, delete, search), menanggulangi validasi dari bagian controller, namun tidak bisa berhubungan langsung dengan bagian view.
 3. Controller, ialah bagian yang mengendalikan ikatan antara bagian model serta bagian view, controller berperan buat menerima permintaan serta informasi dari pengguna setelah itu memastikan apa yang hendak diproses oleh aplikasi.

Dengan memakai prinsip MVC sesuatu aplikasi bisa dibesarkan cocok dengan keahlian pengembangnya, ialah pemrogram yang menanggulangi bagian model serta controller, sebaliknya desainer yang menanggulangi bagian view, sehingga pemakaian arsitektur MVC bisa tingkatkan pemeliharaan serta pengorganisasian kode. Meski demikian diperlukan komunikasi yang baik antara pemrogram serta desainer dalam menanggulangi variabel- variabel yang hendak ditampilkan.

Kelebihan yang diberikan oleh framework CodeIgniter yang lebih unggul dengan Framework PHP antara lain :

1. Performa yang sangat cepat: Salah satu alasan untuk tidak menggunakan framework adalah karena eksekusinya lebih lambat dari PHP, tetapi CodeIgniter sangat cepat sehingga Anda bahkan dapat mengatakan bahwa CodeIgniter adalah framework tercepat dibandingkan dengan framework lainnya.
2. Sangat sedikit konfigurasi (hampir nol konfigurasi): Tentu saja, untuk mencocokkan fleksibilitas database dan perutean, masih diperbolehkan untuk mengkonfigurasi dengan mengubah beberapa file konfigurasi (seperti database.php atau autoload.php), tetapi untuk menggunakan pengaturan default CodeIgniter, Anda hanya perlu mengubah file di folder konfigurasi.

3. Banyak komunitas: Dengan banyaknya komunitas CI, kita bisa lebih mudah berinteraksi dengan orang lain, baik itu bertanya atau teknologi terkini.
4. Dokumentasi yang sangat lengkap: Setiap paket instalasi CodeIgniter dilengkapi dengan panduan pengguna yang sangat baik dan lengkap untuk membantu Anda memulai, dan bahasanya mudah dimengerti.

2.10 Bootstrap

Bootstrap adalah framework CSS gratis dan open source untuk mendesain situs web dan aplikasi web. Kerangka kerja berisi templat desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi, dan komponen antarmuka lainnya, serta ekstensi JavaScript opsional. Tidak seperti kebanyakan kerangka kerja web lainnya, kerangka kerja ini hanya berfokus pada pengembangan front-end (Rozi and Comunity 2015).

Bootstrap awalnya disebut Twitter Blueprint dan dikembangkan oleh Mark Otto dan Jacob Thornton dari Twitter sebagai kerangka kerja untuk mendorong konsistensi antara perangkat internal yang sesuai. Sebelum Bootstrap, berbagai library telah banyak digunakan dalam pengembangan antarmuka, yang mengakibatkan inkonsistensi dan beban pemeliharaan yang berat.

Keuntungan dalam menggunakan framework bootstrap antara lain :

1. Responsif
2. Penggunaan mudah
3. Pengembangan yang cepat
4. Dukungan dari komunitas yang besar
5. Konsisten
6. Open Source