# BAB II

# LANDASAN TEORI

## 2.1. Data *Mining*

Data *mining* merupakan serangkaian proses untuk menggali informasi dengan melakukan analisa data untuk menemukan suatu pola dari kumpulan data tersebut. Data mining mampu menganalisa data yang besar menjadi ekstraksi berupa pola yang mempunyai arti bagi pendukung keputusan (Gunadi & Sensuse, 2012). Data mining juga bisa disebut knowledge discovery adalah proses pengambilan pola pada data yang akan diproses lalu output tersebut berupa informasi yang sangat penting. Proses yang dilakukan untuk mengekstrak pengetahuan dalam data mining adalah pengenalan pola, clustering, asosiasi, prediksi dan klasifikasi (Fitri, Nurjanah, & Astuti, 2018). Data *mining* memiliki variasi untuk menemukan pola dari ekstraksi sebuah kumpulan sekumpulan data tekstual yang disebut dengan *text mining*. *Text mining* memiliki fokus pada pengolahan data berupa kata atau teks.

## 2.2. *Text Mining*

Menurut Hearst *text mining* diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, dengan secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda (Februariyanti, 2012). Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber . Sedangkan menurut Harlian *text mining* memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

*Text mining* merupakan bagian dari *data mining*, yang mana digunakan untuk mendapatkan informasi dari sebuah data atau dokumen berupa sekumpulan teks yang memiliki format yang terstruktur maupun tidak terstruktur dengan jumlah yang besar. Dalam *text mining* memiliki tugas khusus yaitu klasifikasi dan klasterisasi. Sedangkan dalam penerapannya, *text mining* berfungsi untuk mencari pola dalam teks, menganalisis teks agar bisa menghasilkan keluaran berupa informasi yang bermanfaat pada tujuan tertentu. Dikarenakan data yang diproses pada *text mining* merupakan sebuah teks yang tidak terstruktur, maka diperlukan pemilihan teks sebelum dilakukan proses selanjutnya, pada tahap ini dikenal dengan prapemrosesan (*preprocessing*).

## 2.3. Plagiarisme

Plagiarisme atau plagiat adalah penjiplakan atau pengambilan karangan, pendapat orang lain dan menjadikannya seolah-olah karangan sendiri (Sunyoto & Informatika, 2013). Pendekatan deteksi plagiat terbagi menjadi *intrinsic* dan *external*. Pendekatan *external* terbagi lagi menjadi tiga, yaitu perbandingan teks lengkap, kesamaan kata kunci dan *fingerprinting*. Perbandingan teks lengkap diterapkan untuk membandingkan semua isi dokumen, kesamaan kata kunci bekerja dengan cara mengekstrak dan membandingkan kata kunci antara dokumen, dan *fingerprinting* untuk mendeteksi kemiripan antar dokumen dengan prinsip *hashing.*

Berdasarkan penelitian ini mengkategorikan praktek plagiat berdasarkan cara yang digunakan, diantaranya :

* *Copy Paste plagiarism*, menyalin setiap kata tanpa perubahan.
* *Disguised plagiarism*, tergolong ke dalam praktek menutupi bagian yang disalin, teridentifikasi ke dalam empat teknik, yaitu *shake paste, expansive plagiarism, contractive plagiarism,* dan *mosaic plagiarism.*
* *Technical disguise*, teknik meringkas untuk menyembunyikan konten plagiat dari deteksi otomatis dengan memanfaatkan kelemahan dari metode analisis teks dasar, misal dengan mengganti huruf dengan simbol huruf asing.
* *Undue paraphrasing*, sengaja menuliskan ulang pemikiran asing dengan pemilihan kata dan gaya plagiator dengan menyembunyikan sumber asli.
* *Translated plagiarism*, mengkonversi konten dari satu bahasa ke bahasa lain.
* *Idea plagiarism*, menggunakan ide asing tanpa menyatakan sumber.

*Self plagiarism*, penggunaan sebagian atau keseluruhan tulisan pribadi yang tidak dibenarkan secara ilmiah.

## 2.4. *Preprocessing*

Tahapan *preprocessing* atau praproses merupakan bagian yang sangat penting dalam menyiapkan data, hal ini dikarenakan struktur data yang dihasilkan pada tahap pengumpulan tidak beraturan, sehingga menyebabkan proses menjadi tidak berjalan dengan baik.

Merujuk pada penelitian sebelumnya yang dilakukan oleh Himalatha dalam jurnal (Filcha & Hayaty, 2019) maka pada penelitian ini akan dibahas beberapa tahapan *preprocessing* teks antara lain, *case folding,* menghapus karakter selain a-z, menghapus teks dengan 1 karakter, mengganti *slang word,* menghapus *stop word.*

### 2.4.1. *Case folding*

Pada proses ini bertujuan untuk mengubah semua karakter huruf menjadi huruf kecil (*lowercase*), hal ini dilakukan untuk menyamakan arti dari suatu kata yang sama, apabila penulisan besar kecilnya huruf tidak sama.

### 2.4.2. Menghapus karakter kecuali a sampai z

Pada proses ini dilakukan penghapusan untuk seluruh karakter berupa simbol dan angka, atau menyisakan hanya karakter angka, termasuk menghapus hashtag (#) dan mention (@), hal ini dilakukan karena simbol dan angka dianggap tidak terlalu penting, tetapi jika ini diperlukan, maka proses ini dihilangkan.

### 2.4.3. Mengganti *slang word*

Teks yang tidak terstruktur membuat sebuah teks terkadang tidak sesuai dengan ejaan bahasa Indonesia yang baku (EYD) pada konteks ini, kata yang tidak baku disebut dengan *slang word,* untuk mendapatkan informasi dari teks agar maksimal, kata-kata tidak baku, baik kata gaul, singkatan atau yang lain sebanyak mungkin ditampung ke dalam kamus *slang word,* untuk kemudian dilakukan *replace* supaya menjadi kata dengan bahasa Indonesia yang baku sesuai EYD.

### 2.4.4. Menghapus *stop word*

*Stop word* merupakan salah satu kata yang diabaikan dalam pemrosesan, Ringkasnya *stop word* adalah kata hubung atau kata sambung dalam sebuah kalimat, seperti “di”, “pada”, “karena”, “sebuah”, “oleh”, dll. Sebelum melakukan proses penghapusan *stop word*, kumpulkan daftar atau kamus *stop word* yang diberi nama *stoplist*. Kemudian lakukan perbandinngan antara sebuah teks dengan *stoplist*. Jika terdapat kata-kata yang terdapat dalam *stoplist*, maka kata tersebut dihilangkan. Untuk *stoplist* dalam bahasa Indonesia, datanya bersumber dari (Tala, 2003).

## 2.5. *N-Gram*

Dalam jurnal (Sunardi et al., 2018)menjelaskan bahwa, *n-gram* merupakan metode yang yang dilakukan dengan mengambil rangkaian *substring* dari *string* sejumlah n (rangkaian token sepanjang n). Metode *n-gram* sering digunakan pada teknik analisis statistik dan juga bahasa. *N-gram* paling banyak digunakan dalam teks mining (pengolahan kata) dan pengolahan bahasa. Dalam mendeteksi plagiarisme, *n-gram* digunakan untuk mengambil potongan-potongan karakter huruf atau pemisahan *string* sepanjang n dari sebuah kata atau dokumen secara berkelanjutan (kontinuitas) hingga bergeser sesuai dengan *offsite* yang diberikan atau akhir dari suatu kata atau dokumen.

*N-gram* dalam deteksi plagiarisme sangat mempengaruhi tingkat akurasi atau tingkat *similarity* dari sebuah dokumen yang dibandingkan, maka teknik *n-gram* dipadukan dengan pendekatan statistika untuk memperoleh *similarity* dari antar dokumen seperti *Simple Matching, Cosine Similarity, Jaccard Similarity* dan *Dice Coefficient*. Ada 2 (dua) teknik *n-gram* yaitu membagi *string* menjadi suatu set *substring* dengan panjang n (*overlapping n-gram*) dan mengecek untuk membentuk substring yang memiliki struktur yang sama.

## 2.6. *Jaccard Similarity*

Dalam jurnal (Sunardi et al., 2018)menjelaskan bahwa, *Jaccard Similarit*y dan *Jaccard Coefficient* merupakan algoritme yang fungsinya untuk membandingkan dua sampel yaitu dokumen yang satu dengan yang lainnya berdasarkan kata yang dimilikinya. *Jaccard similarity* biasanya digunakan untuk membandingkan dokumen dan menghitung nilai kemiripan (*similarity*) dari dua buah objek atau dokumen. *Jaccard similarity* dapat dirumuskan sebagai berikut:

*Similarity* (X,Y)=…(2.1)

Dimana:

X = *Dataset*

Y = Data tes

Rumus 1 merupakan dari rumus *Jaccard Similarity* atau *Jaccard Coefficient* yang digunakan untuk mencari persamaan dan perbedaan pada dua sampel. sebagai contoh diketahui X “Magister Teknik Informatika Yogyakarta”, dan Y “Magister Teknik Sipil Yogyakarta”. Maka akan menghasilkan nilainya:

*Similarity* (X,Y)= …..(2.2)

*Similarity* (X,Y)== 0,6 x 100% = 60%

Dari dua contoh sampel di atas setelah dihitung kesamaannya menggunakan *Jaccard similarity*, bahwa kedua sampel tersebut memiliki kesamaan atau kemiripan sebesar 60%.

Ukuran tingkat kesamaan:

* a. 0% : dua dokumen tidak memiliki kesamaan
* b. <15% : memiliki sedikit kesamaan
* c. 15-50% : termasuk dalam kategori plagiarisme sedang
* d. 50% : mendeteksi adanya plagiarisme
* e. 100% : dokumen tersebut plagiarism

## 2.7. Algoritme *Winnowing*

Dalam jurnal (Sunardi et al., 2018)menjelaskan bahwa, Algoritme *winnowing* merupakan salah satu algoritme yang berfungsi sebagai dokumen *fingerprint* atau algoritme yang digunakan untuk mendeteksi tindakan plagiarisme dengan menggunakan teknik *hashing*. Input dari algoritme *winnowing* berupa dokumen teks, dan akan menghasilkan keluaran berupa kumpulan nilai hash yang terbentuk dari perhitungan ASCII pada setiap karakter. Dan nilai-nilai *hash* yang akan digunakan sebagai *fingerprint* untuk mendeteksi adanya suatu tindakan plagiarisme. Gambar 2.1 merupakan konsep algoritme *winnowing*.

*Preprocessing*

Membentuk *n-gram*

Penghitung *substring* menjadi nilai *hash*

*Window*

*Fingerprint*

Gambar 1 2.1 Konsep Algoritme *Winnowing*(Sunardi et al., 2018)

Konsep dari algoritme *winnowing* yang ditujukan pada gambar 2.1 yaitu menghapus karakter yang tidak relevan, membentuk *n-gram* dengan panjang n, menghitung nilai *hash*, membentuk nilai *window*, dan memilih nilai *hash* sebagai dokumen *fingerprint*. Nilai *hash* didapatkan dari rumus *Rolling hash*. *Rolling hash* adalah metode *hashing* yang digunakan untuk mencari nilai *hash* dari rangkaian *grams* yang telah terbentuk dan memberikan kemampuan untuk menghitung nilai tanpa mengulangi seluruh *string*. Nilai *hash* merupakan nilai numerik yang dibentuk dari kode ASCII[15]. Berikut rumus *rolling hash*

…(2.3)

….(2.4)

Dimana:

H (C1..Cl) = nilai hash

Cl = nilai ASCII karakter ke -1 pada string

l = panjang string

b = nilai basis hash

Perhitungan awal pada rangkaian *n-gram* paling awal dihitung menggunakan rumus nomor 2, dan rangkaian *n-gram* berikutnya sampai rangkaian *gram* terakhir dihitung menggunakan rumus nomor 3, sehingga proses akan jauh lebih cepat karena tidak menghitung lagi dari awal. Algoritme *winnowing* merupakan ekstensi dari algoritme *Rabin-karp*, proses yang dilakukan hampir sama hanya pada algoritme *winnowing* ditambahkan dengan konsep *window*.

## 2.8. Studi Literatur

Berdasarkan landasan teori yang telah dijelaskan , terdapat penelitian yang sudah ada sebelumnya, yang dirangkum dalam Tabel 2.4 berikut :

Tabel 1 2.4 Studi Literatur

| **No** | **Judul** | **Tujuan Penelitian** | **Metode** | **Hasil Penelitian** |
| --- | --- | --- | --- | --- |
| 1 | Rancang Bangun Aplikasi Pengecekan Kemiripan Judul Skripsi Dengan Metode *Cosine Similarity* (Abdullah & Aribowo, 2018) | Membantu mahasiswa atau dosen untuk melakukan pengecekan judul skripsi terhadap skripsi – skripsi sebelumnya, saat pengajuan judul skripsi, guna menghindari kesamaan skripsi | Metode Cosine Similarity | Sistem diuji dengan 2 metode yaitu *Blackbox Test* dan *Usability Test.* Pengujian dengan metode *Blackbox Test* dihasilkan bahwa 100% sistem sudah berjalan dengan baik, layak digunakan dan sesuai kebutuhan. Sedangkan pengujian dengan metode *Usability Test* 0 % responden menyatakan aplikasi tidak diterima, 20 % responden memberikan penilaian marginal dan 80 % responden menyatakan sistem aplikasi bisa diterima. |
| 2 | Rancang bangun aplikasi pendeteksian kesamaan pada dokumen teks menggunakan algoritma *Enhaced confix stripping* dan algoritma *winnowing*(Khidfi & Sari, 2018) | Membuat sistem untuk menentukan nilai similaritas antara 2 dokumen teks sehingga dosen dapat melihat tingkat similaritas antara tugas mahasiswa dan dapat mengurangi tingkat plagiarisme yang terjadi antara mahasiswa. | Algoritma Enhanced Confix Stripping Stemmer dan Algoritma Winnowing | Untuk mendapatkan hasil pada penelitian ini, peneliti melakukan pengujian dengan menentukan nilai *gram* dan *window* pada perhitungan algoritma *Winnowing,* tujuannya memudahkan user dalam menentukan nilai *similarity* yang akurat. Dari hasil pengujian 5 pasang bab 1 tugas akhir mahasiswa yang berkategori sama menghasilkan nilai *similarity* sekitar 45-20% . |
| 3 | Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode *Cosine Similarity*(Ariantini, Lumenta, & Jacobus, 2016) | Membuat sistem pengecekan kemiripan dokumen teks berbahasa Indonesia | Cosine similarity | Untuk mendapatkan hasil pada penelitian ini dilakukan pengujian menggunakan data dummy. Data dummy adalah data yang digunakan untuk membadingkan dokumen 1 dengan dokumen 2. Nilai actual dari dokumen satu menjadi acuan nilai actual diperoleh dari memeriksa tugas secara manual kemudian dibandingkan dengan dokumen 2 prediksi dari sistem dengan nilai 100% karena didalam dokumen tersebut memang sama persis. Kemudian dokumen satu dengan dokumen tiga nilai actualnya 66% dan nilai prediksi sebagian isi dokumen diambil dari dokumen satu sehingga nilai yang diperoleh adalah 75%. Nilai kesalahan yang diperoleh rata-rata yaitu 7% sehigga masih banyak kesamaan kata atau kesalahan, dengan demikian sistem yang dibangun sudah bisa digunakan untuk mendeteksi kemiripan dokumen. |
| 4 | Penerapan Metode *Cosine Similarity* Untuk Pengecekan Kemiripan Jawaban Ujian Siswa (Fataruba, 2018) | Untuk membantu pengajar memberikan nilai yang objektif dengan mengunakan metode cosine similarity pada sistem agar dapat melakukan penilaian jawaban essay dengan membandingkan kunci jawaban pengajar dengan jawaban peserta didik. Dan juga dapat membandingkan tingkat kemiripan antara siswa satu dengan siswa yang lain. | Cosine similarity | similarity telah berjalan dengan baik untuk ujian essay Biologi Hasil uji coba menunjukkan kesesuaian nilai sistem dengan nilai yang diberikan oleh pengajar tingkat akurasinya 80% mengunakan Confusion Matrix. |
| 5 | Pendeteksian tingkat similaritas dokumen berbasis web menggunakan algoritma *winnowing*(Ulfa, Mustikasari, & Bastian, 2016) | untuk membangun sebuah aplikasi berbasis website menggunakan algoritma Winnowing untuk mencari kesamaan pada dua dokumen teks yang diuji. | Algoritma Winnowing | Pada hasil eksperimen pengukuran kemiripan ini disimpulkan bahwa semakin kecil tingkat persentase kesamaan dokumen teks yang diuji, maka tingkat kemiripan dokumen kecil dan tidak termasuk plagiat, tetapi jika hasil dari pengujian pada dua dokumen semakin besar, maka dokumen tersebut mempunyai tingkat kemiripan yang tinggi dan tindakan tersebut dianggap plagiat. Penelitian ini juga menambahkan perbandingan nilai k-gram, basis (bilangan prima), nilai window, keterangan persentase, dan kategori plagiarisme. |
| 6 | Pendeteksian Plagiarisme Menggunakan Algoritma *Rabin-Karp* dengan Metode *Rolling Hash*(Priambodo, 2018) | Untuk mengatasi plagiat yang biasanya dilakukan terhadap konten digital adalah melakukan copy- paste, quote, dan revisi terhadap dokumen asli. | Algoritma Rabin-Karp dengan metode Rolling Hash. | Hasil pengujian 30 dokumen teks menghasilkan tingkat akurasi yang terbesar yaitu 47.58%. Hasil persentase tersebut termasuk dalam kategori tingkat plagiat 15-50%, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Sedangkan tingkat akurasi yang terkecil yaitu 19.28%, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Berdasarkan analisis proses pendeteksian tingkat plagiarisme menggunakan algoritma raibin-karp dengan metode rolling hash bisa membaca karakter berupa huruf, simbol seperti titik (.), koma (,), dan lain-lain. |
| 7 | Pendeteksian Dokumen Plagiarisme dengan Menggunakan Metode *Weight Tree* (Nurdin, Rizal, & Rizwan, 2019) | Untuk membangun sebuah sistem pendeteksian kemiripan dari dua dokumen teks yang berbeda untuk jenis dokumen teks berbahasa indonesia dengan format file dokumen yaitu: doc, docx, pdf, rtf. Tahapan | Weight Tree | Hasil pengujian sistem dapat dikategorikan sebagai sistem pendeteksian atau pengetesan kemiripan dokumen. Nilai rata-rata persentase kemiripan dalam pengujian sistem ini adalah 71,60%. Sistem yang di bangun ini berhasil dengan tingkat keakuratan mencapai 90%. Algoritma *Weight Tree* yang diterapkan pada sistem ini terbukti mampu mengidentifikasi dengan baik kemiripan dokumen plagiarisme. |
| 8 | Pemodelan Penilaian *Essay* Otomatis Secara *Realtime* Menggunakan Kombinasi *Text Stemming* Dan *Cosine Similarity* (Rinartha, 2017) | Membangun sebuah sistem untuk melakukan penilaian essay otomatis secara realtime. | Cosine similarity | Pengajar akan memasukkan data soal dan jawaban beserta kuncinya ke dalam sistem pada waktu tertentu. Kemudian data diproses di sisi client untuk memperoleh kata-kata kunci dalam jawaban yang kemudian tersimpan di dalam database. Ketika peserta didik menjawab pertanyaan yang ada pada sistem, sistem akan memroses jawaban hingga menemukan kata-kata kunci yang digunakan dalam jawaban tersebut, kemudian dibandingkan dengan jawaban yang sudah ada didalam sistem. Hasil ujian akan ditampilkan pada sisi pengajar untuk mengetahui nilai yang didapatkan oleh peserta didik. |
| 9 | Implementasi Deteksi Plagiarisme Menggunakan Metode *N-Gram* Dan *Jaccard Similarity* Terhadap Algoritma *Winnowing*(Sunardi et al., 2018) | Membangun sebuah sistem untuk mengecek nilai kemiripan antara dokumen x dengan dokumen y | N-gram dan Jaccard similarity dengan Algoritme winnowing | Hasil penelitian yang dilakukan dapat dipahami bahwa n-gram sangat mempengaruhi hasil dari similarity, penggunaan n-gram yang tepat sangat diperlukan. Tingkat kemiripan pada tahap pengujian sebuah dokumen mencapai 100%, sehingga metode Jaccard Similarity memiliki prospek untuk digunakan dalam deteksi plagiat. |
| 10 | Implementasi Algoritma *Rabin-Karp* untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa (Filcha & Hayaty, 2019) | Membangun sebuah sistem untuk melakukan pemeriksaan plagiarisme pada dokumen tugas antar mahasiswa dengan cepat dan tepat. | Rabin-Karp | Hasil dari metode ini dihitung menggunakan *dice coefficient*. Perhitungan akurasi dengan melakukan 20 perbandingan antara sistem pendeteksi plagiarisme dan *software* Plagiarisme Checker X menggunakan *confusion matrix* menghasilkan tingkat keakuratan sebesar 90%. |
| 11 | Deteksi plagiarisme dokumen bahasa indonesia dengan algoritma *jaro-winkler Distance*(Anggara, 2016) | Untuk membandingkan kesamaan antar dokumen teks berbahasa Indonesia, sehingga dapat ditentukan sebuah dokumen tersebut plagiat atau tidak. | Jaro-Winkler distance. | Pengujian terhadap aplikasi menggunakan data abstrak jurnal skripsi. Dari hasil analisis dokumen uji 1 memiliki kesamaan tertinggi dengan dokumen nomer 55 dengan nilai 86,267%, dokumen 2 memiliki kesamaan tertinggi dengan dokumen 66 dengan nilai 95,922% dan dokumen 3 memiliki kesamaan tertinggi dengan dokumen 23 dengan nilai 98.361 %. |
| 12 | Aplikasi pendeteksi plagiat terhadap karya tulis berbasis web menggunakan natural language processing dan algoritma knuth-morris-pratt (Alamanda, Suhery, & Brianorman, 2016) | membuat aplikasi berbasis web yang dapat mendeteksi plagiat terhadap karya tulis menggunakan Natural Language Processing | Algoritma Knuth–Morris-Pratt. | Dari hasil pengujian yang telah dilakukan, rata-rata persentase kemiripan yang dihasilkan dari pendeteksian tanpa menggunakan proses *TF-IDF* sebesar 45,98%, nilai tersebut lebih rendah dibandingkan dengan pendeteksian menggunakan proses *TF-IDF*, *Tokenizing, Filtering* dan *Stemming* yang menghasilkan persentase kemiripan 41,09%. Sedangkan tanpa menggunakan proses *Stemming* yaitu 40,58% serta tanpa menggunakan proses *TF-IDF* dan *Stemming* sebesar 40,54% |
| 13 | *Mining disinformation and fake news: Concepts, methods, and recent advancements*(Shu, Wang, Lee, & Liu, 2020) | Membahas metode pengecekan similarity dokumen | *String Based* | *String Based* ini di bagi menjadi 2, yaitu *Character Based* dan  *Term Based.*  Dari 2 pembagian ini ada beberapa metode yang bisa di gunakan, sesuai dengan kebutuhan. Untuk *Character Based*  pembagianya ada beberapa metode di antaranya, *Longest Common SubString (LCS), Longest Common SubString (LCS), Jaro, Jaro–Winkler, Needleman-Wunsch, Smith-Waterman, N-gram.* Untuk *Term Based* pembagianya ada, *Block Distance, Cosine similarity, Dice’s coefficient, Euclidean distance, Jaccard similarity, Matching Coefficient, Overlap coefficient.* |
| 14 | *Measurement of Text Similarity: A Survey*(Wang & Dong, 2020) | *Untuk menganalisis keuntungan dan kerugian dari suatu metode yang digunakan dalam pendeteksian similarity* |  | Memberikan referensi bagi penelitian dan aplikasi terkait, metode pengukuran kemiripan teks dijelaskan melalui dua aspek yaitu jarak teks dan representasi teks. Jarak teks dapat dibagi menjadi jarak panjang, jarak distribusi, dan jarak semantik; Representasi teks dibagi menjadi teks berbasis string, berbasis korpus, teks semantik tunggal, teks multi-semantik, dan representasi berbasis struktur grafik. |