*Laporan Mata Kuliah Teks dan Web Mining*

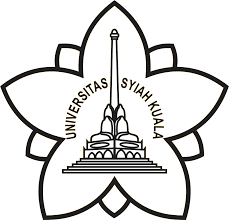
TUGAS 5: MEMBANGUN KAMUS N-GRAM UNTUK KATEGORI BERITA DAN SEPAKBOLA PADA PORTAL DETIK

disusun untuk memenuhi tugas Teks dan Web Mining

Oleh:

**SITI NURRAHMASITA**

**(2108107010015)**



# PROGRAM STUDI INFORMATIKA

# FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

**UNIVERSITAS SYIAH KUALA**

**DARUSSALAM, BANDA ACEH**

# 2024

1. **Deskripsi Tugas**

Menggunakan dataset yang telah didownload untuk Tugas 5 sebanyak 6000 halaman web yang terdiri dari 2 kategori, bangun kamus 1-gram, 2-gram dan 3-gram untuk kedua kategori tersebut. Hilangkan duplikasi kata yang sama pada kedua kamus dengan melakukan observasi eliminasi rasio untuk threshold 50% dan 55%. Rasio yang dimaksud adalah rasio antara jumlah normalisasi frekuensi dari kata yang yang ditemukan di kamus kelas (+) dan jumlah normalisasi frekuensi dari kata yang sama di kelas (-). Penyebut dari perhitungan rasio ini selalu diambil dari nilai normalisasi frekuensi yang terbesar dari kata yang sama. Hasil akhir dalam kamus adalah kata-kata yang disjoint (tidak duplikat) dan memiliki rasio < dari threshold (kemunculan timpang pada kedua kamus). Gunakan script one-gram.pl, two-grams.pl dan three-grams.pl untuk menyelesaikan tugas ini.

Buat laporan yang menjelaskan hasil yang diperoleh. File laporan tugas dalam format PDF dikumpulkan paling telat tanggal 6 Mei 2024, pukul 23.55 WIB via sistem e-learning. Contoh format file tugas yang dikumpulkan adalah NAMA\_NIM.PDF atau NAMA\_NIM.zip atau NAMA\_NIM.tar.gz. Contoh: Asumsi threshold adalah 50%

**Scenario 1**

Jika kata k ditemukan di kamus A dengan nilai frekuensi setelah dinormalisasi adalah 0.6 dan kata yang sama juga ditemukan di dalam kamus B dengan nilai frekuensi setelah dinormalisasi adalah 0.8, maka yang harus anda lakukan adalah melakukan perhitungan sbb: v = 0.6/0.8 = 0.75 (nilai normalisasi yang lebih besar selalu sebagai dijadikan penyebut) Karena nilai v dari kata k lebih besar dari threshold (0.50) maka dapat disimpulkan bahwa kata k tsb 'common' pada kedua kamus, untuk itu kata k harus dieliminasi (dibuang) dari kedua kamus A dan B.

**Scenario 2**

Jika kata k ditemukan di kamus A dengan nilai frekuensi setelah dinormalisasi adalah 0.6 dan kata yang sama juga ditemukan di dalam kamus B dengan nilai frekuensi setelah dinormalisasi adalah 0.2, maka perhitungan v adalah: v = 0.2/0.6 = 0.33 (nilai normalisasi yang lebih besar selalu sebagai dijadikan penyebut) Karena nilai v dari kata k TIDAK LEBIH BESAR dari threshold (0.50) maka dapat disimpulkan bahwa kata k tersebut lebih sering muncul di kamus A (0.6) bila dibanding dengan kamus B (0.2). Oleh karena itu, kata k yang ada di kamus B dieliminasi (dibuang) sementara kata k pada kamus A tetap dipertahankan. Begitu seterusnya untuk semua kata yang muncul di kedua kamus. Scenario tersebut dicek. Akhir dari proses pemeriksaan ini akan diperoleh 2 kamus A dan B dimana kata-katanya (1-gram, 2-grams, dan 3-grams) yang unik dan disjoint.

1. **Pendahuluan**

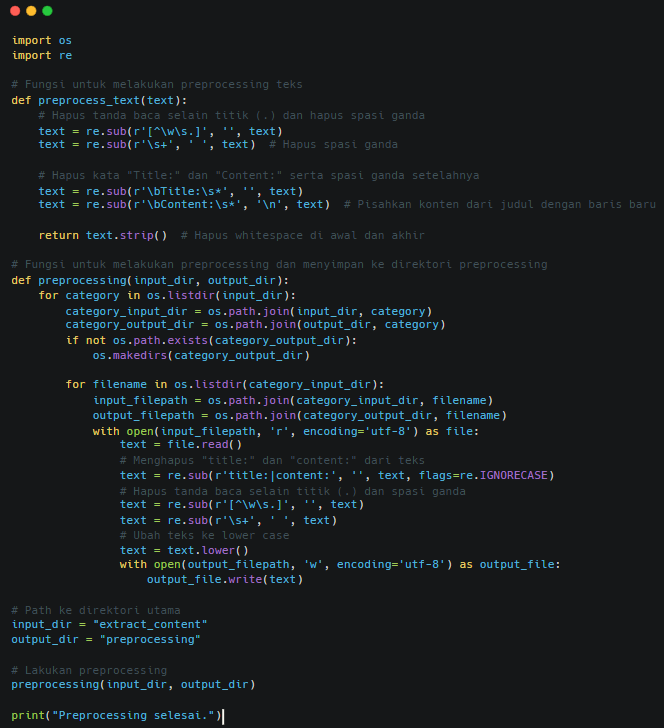
Pada Tugas 4 sebelumnya, dilakukan proses crawling terhadap lebih dari 4000 link URL untuk setiap kategori, yakni sepakbola dan berita, dari situs web berita Detik.com. Proses ini bertujuan untuk mengumpulkan sejumlah besar URL yang mengarah ke halaman-halaman berita terkait kedua kategori tersebut. Setelah proses crawling selesai, langkah selanjutnya adalah melakukan proses scraping pada setiap URL yang telah terkumpul. Proses scraping ini dilakukan untuk mengekstrak informasi yang relevan dari halaman web, seperti judul dan konten artikel.

Setelah scraping, file-file hasilnya disimpan dalam dua folder terpisah, yaitu html/sepakbola dan html/berita. Dalam masing-masing folder tersebut, terdapat file-file HTML yang berisi informasi dari halaman web yang telah di-scraper. Langkah terakhir yang dilakukan adalah eliminasi tag HTML dari konten yang diambil, sehingga hanya bagian judul dan konten artikel yang tersisa. Ini dilakukan untuk membersihkan teks dari kode HTML yang tidak relevan atau tidak diinginkan, sehingga data yang dihasilkan menjadi lebih mudah untuk diproses lebih lanjut. Hasilnya akan digunakan dalam menyelesaikan Tugas 5.

1. **Penjelasan Umum**

Langkah-langkah untuk membangun kamus one-gram, two-gram, dan three-gram seperti pada kode di atas dapat dijelaskan sebagai berikut:

1. Preprocessing Teks: Langkah pertama adalah melakukan preprocessing pada teks. Proses ini bertujuan untuk membersihkan teks dari karakter-karakter yang tidak diinginkan, seperti tanda baca selain titik, spasi ganda, dan kata-kata yang telah ditentukan sebelumnya seperti "Title:" dan "Content:". Preprocessing juga mencakup konversi teks menjadi huruf kecil (lowercase). Hasil preprocessing ini akan menjadi dasar untuk pembangunan kamus n-gram.
2. Generate One-Gram: Untuk membangun kamus one-gram, teks yang telah melalui preprocessing dibagi menjadi token-token individu, yaitu kata-kata. Kemudian, frekuensi kemunculan setiap kata dihitung. Hasilnya adalah sebuah kamus yang berisi setiap kata (unigram) beserta frekuensi kemunculannya dalam korpus teks.
3. Generate Two-Gram: Pembangunan kamus two-gram melibatkan proses yang mirip dengan pembangunan kamus one-gram, namun kali ini kata-kata digabungkan menjadi pasangan dua kata (bigram). Setiap bigram kemudian dihitung frekuensinya. Proses ini menghasilkan kamus yang berisi setiap pasangan dua kata beserta frekuensi kemunculannya dalam korpus teks.
4. Generate Three-Gram: Langkah terakhir adalah membangun kamus three-gram. Prosesnya serupa dengan pembangunan kamus one-gram dan two-gram, hanya saja kali ini kata-kata digabungkan menjadi triplet tiga kata (trigram). Setiap trigram dihitung frekuensinya, dan hasilnya adalah kamus yang berisi setiap triplet tiga kata beserta frekuensi kemunculannya dalam korpus teks.
5. Penanganan Duplikat: Selama proses pembangunan kamus, ada kemungkinan terdapat kata-kata atau pasangan kata yang muncul dalam lebih dari satu kategori atau dokumen. Untuk menghindari redundansi, terdapat penanganan untuk menghapus kata-kata duplikat di antara kategori-kategori tersebut. Hal ini dilakukan dengan membandingkan frekuensi kemunculan dan menghapus kata-kata yang memiliki rasio frekuensi kemunculan yang lebih besar dari sebuah threshold yang telah ditentukan.
6. Penyimpanan Hasil: Setelah proses pembangunan kamus selesai, hasilnya disimpan dalam file teks terpisah untuk setiap kategori atau dalam satu file tergantung pada kebutuhan. Setiap entri dalam file tersebut berisi token n-gram beserta frekuensi kemunculannya.
7. **Penjelasan Scripts**
8. **Preprocessing.py**



**Gambar C.a.1.** Script file preprocessing.py

1. Preprocessing Teks: Fungsi preprocess\_text melakukan preprocessing pada teks dengan langkah-langkah berikut:

* Menghapus tanda baca selain titik (.) dan menghapus spasi ganda.
* Menghapus kata "Title:" dan "Content:" serta spasi ganda setelahnya, kemudian memisahkan konten dari judul dengan baris baru.
* Menghapus whitespace di awal dan akhir teks.

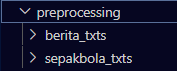
1. Preprocessing dan Menyimpan ke Direktori: Fungsi preprocessing melakukan preprocessing pada setiap file teks dalam direktori input, kemudian menyimpan hasilnya dalam direktori output. Langkah-langkahnya adalah:

* Iterasi melalui setiap kategori dalam direktori input.
* Untuk setiap file dalam setiap kategori, membaca teks dari file input.
* Melakukan preprocessing pada teks (seperti yang dijelaskan di atas).
* Menyimpan teks yang telah dipreproses ke dalam file output dengan format yang sama.

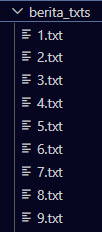
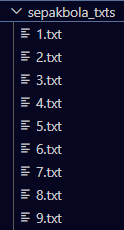
1. Eksekusi Preprocessing: Memanggil fungsi preprocessing dengan menyediakan path ke direktori input dan output. Setelah proses selesai, mencetak pesan "Preprocessing selesai."



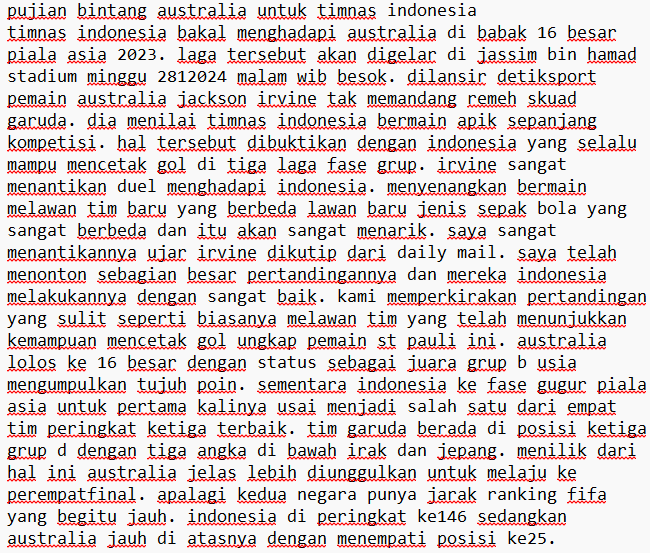
**Gambar C.a.2.** Output sukses melakukan preprocessing data



**Gambar C.a.3.** Directory yang dihasilkan

**Gambar C.a.4.** Hasil extracting halaman berita

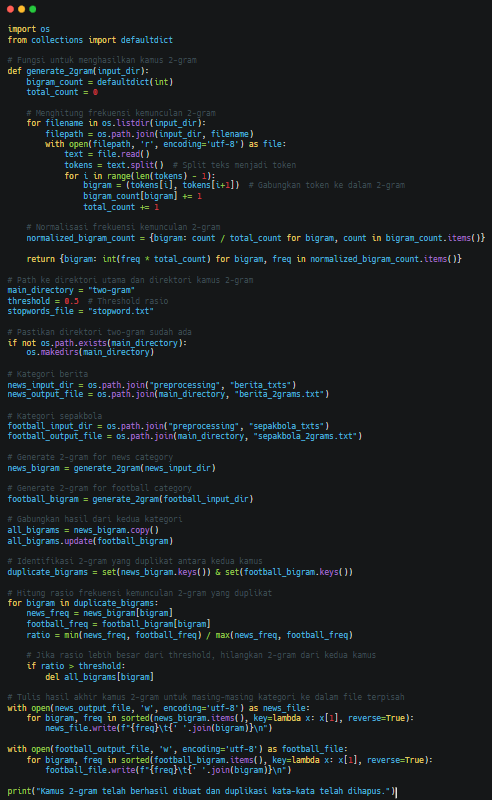


**Gambar C.a.5.** Hasil preprocessing halaman berita

1. **One-gram.py, two-gram.py, dan three-gram.py**



**Gambar C.b.1.** Scripts one-gram.py



**Gambar C.b.2.** Scripts two-gram.py



**Gambar C.b.3.** Scripts one-gram.py

Tahap-tahap dalam kode-kode tersebut dapat dijelaskan sebagai berikut:

1. Membuat Kamus Frekuensi N-Gram:

* Fungsi-fungsi generate\_1gram, generate\_2gram, dan generate\_3gram digunakan untuk menghitung frekuensi kemunculan n-gram (1-gram, 2-gram, dan 3-gram) dalam teks.
* Setiap fungsi membaca teks dari file-file dalam direktori yang sesuai dengan kategori (berita atau sepakbola) dan menghitung frekuensi kemunculan n-gram tersebut.
* Frekuensi kemunculan n-gram dinormalisasi dengan membaginya dengan total jumlah n-gram dalam teks.

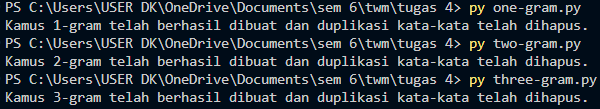
1. Penghapusan Duplikat:

* Setelah kamus frekuensi n-gram dibuat untuk kedua kategori (berita dan sepakbola), dilakukan identifikasi terhadap n-gram yang duplikat antara kedua kamus.
* Jika sebuah n-gram muncul dalam kedua kategori dan rasio kemunculannya melebihi threshold yang ditentukan, maka n-gram tersebut dihapus dari kedua kamus.

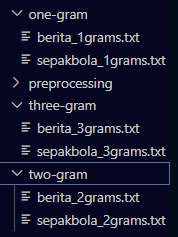
1. Penulisan Hasil Akhir:

* Setelah penghapusan duplikat, hasil akhir kamus n-gram untuk masing-masing kategori ditulis ke dalam file teks terpisah.
* Setiap baris dalam file tersebut berisi satu n-gram diikuti oleh frekuensi kemunculannya, dipisahkan oleh spasi tabulasi.

1. Setelah selesai, pesan output mencetak informasi bahwa proses telah selesai dan kamus n-gram telah berhasil dibuat serta duplikasi kata-kata telah dihapus.



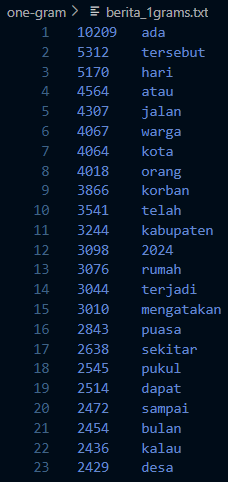
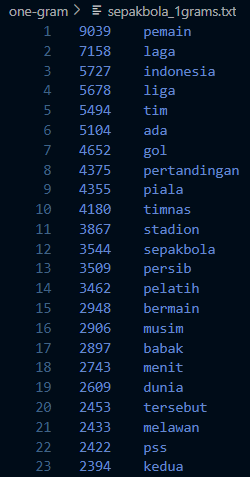
**Gambar C.b.4.** Output sukses membangun kamus one-gram, two-gram, dan three-gram



**Gambar C.b.4.** Directory yang dihasilkan

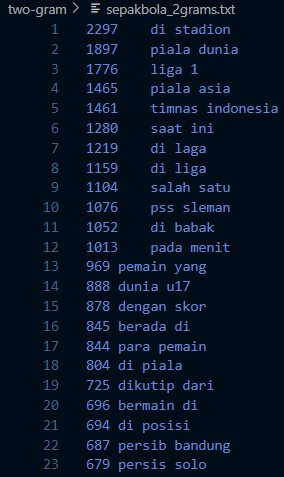
Output yang dihasilkan:

1. one-gram.py

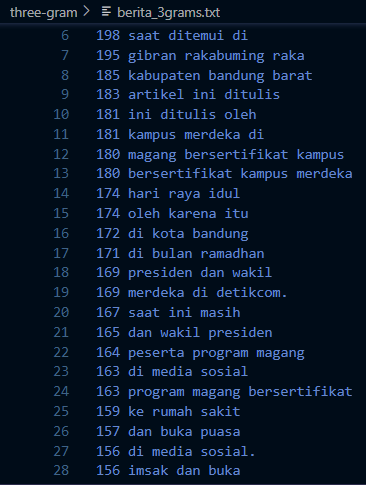
**Gambar C.b.5.** Hasil kamus one-gram kategori berita (kiri) dan sepakbola (kanan)

1. two-gram.py

**Gambar C.b.6.** Hasil kamus two-gram kategori berita (kiri) dan sepakbola (kanan)

1. three-gram.py

**Gambar C.b.7.** Hasil kamus three-gram kategori berita (kiri) dan sepakbola (kanan)

1. **Hasil**

Kode-kode di atas digunakan untuk menghasilkan tiga kamus n-gram (one-gram, two-gram, dan three-gram), masing-masing untuk kategori berita dan kategori sepakbola. Dalam konteks ini, n-gram adalah urutan n kata berturut-turut yang muncul dalam teks.Proses pembuatan kamus n-gram dimulai dengan membaca teks dari file-file dalam direktori yang sesuai dengan kategori (berita atau sepakbola). Kemudian, teks dibagi menjadi token-token (kata-kata) dan dihitung frekuensi kemunculan n-gram sesuai dengan nilai n yang ditentukan (1-gram untuk kata tunggal, 2-gram untuk pasangan dua kata, dan 3-gram untuk urutan tiga kata berturut-turut). Frekuensi kemunculan ini dinormalis asikan dengan membaginya dengan total jumlah n-gram dalam teks untuk mendapatkan proporsi kemunculan setiap n-gram.

Selanjutnya, dilakukan identifikasi terhadap n-gram yang duplikat antara kedua kategori. Jika sebuah n-gram muncul dalam kedua kategori dan rasio kemunculannya melebihi threshold yang ditentukan, maka n-gram tersebut dihapus dari kedua kamus.

Hasil akhir dari proses ini adalah dua file teks untuk setiap kategori (berita dan sepakbola), masing-masing berisi daftar n-gram beserta frekuensi kemunculannya. Setiap baris dalam file tersebut berisi satu n-gram diikuti oleh frekuensi kemunculannya, dipisahkan oleh spasi tabulasi. Ini membuat hasilnya mudah dibaca dan dianalisis untuk memahami hubungan antara kata-kata dalam teks kategori berita dan sepakbola, baik itu berupa kata tunggal (one-gram), pasangan kata (two-gram), maupun urutan tiga kata (three-gram).