STUDY HARD.
DO GOOD
AND THE
GOOD LIFE
WILL FOLLOW.

[T1] [PBA] Tugas 1 – Pemrosesan Bahasa Alami

A. ATURAN PENGERJAAN & PENGUMPULAN

Pengerjaan:

- Dikerjakan berkelompok **maksimal 4-5 orang (1 orang tidak dianggap berkelompok), apabila jumlah mahasiswa ganjil, terpaksa ada (1-2) kelompok yang berjumlah 5 anggota. Jadi mayoritas 4 anggota, bukan mayoritas 5 anggota**.
- Batas waktu dan pengumpulan melalui **BRONE**
- **Nama anggota dituliskan di bagian atas kode program dalam bentuk komentar:**

```
/*
* NIM_1 NAMA_1: peran_mahasiswa_1
* NIM_2 NAMA_2: peran_mahasiswa_2
* dst.
*/
```

- **Dilarang menjiplak/menyontek** dengan alasan, cara, sesedikit apapun. Kode program akan dicek dengan program pengecekan plagiarisme. Pengubahan variabel, letak kode, dll. tidak akan berpengaruh dan dianggap plagiarisme. **Plagiarisme (bahkan hanya satu baris saja) tidak akan ditolerir dan dapat mendapatkan nilai E.**
- **Baca dan pahami soal dengan sebaik-baiknya supaya tidak ada poin nilai yang terlewatkan. Apabila ada yang tidak dimengerti segera ditanyakan ke dosen.**

Pengumpulan (cek lagi Bagian E):

- Batas waktu pengumpulan di BRONE, **perhatikan jam *server***.
- ***Soft copy* melalui ke BRONE, lampirkan SEMUA FAIL** (.DOCX/PDF, .JAVA, dll.) **dengan terkompres ZIP/RAR** dengan format:

```
[TX][PBA-PRODI-KELAS] NamaMahasiswa.ZIP
```

  - **TX** berarti kode tugas, ada di baris pertama halaman ini, misal, T1 untuk Tugas 1, ..., TP untuk Tugas/Proyek Akhir, **PBA** adalah kode mata kuliah.
  - **PRODI** diisi inisial prodi: **TIF** (Teknik Informatika), **TKOM** (Teknik Komputer), **SI** (Sistem Informasi)
  - **KELAS** diisi kode kelas misal A, B, C, dst.
  - **NamaMahasiswa** maksudnya tidak perlu menuliskan semua nama anggota, tapi nama satu anggota saja, sebagai ketua kelompok/penanggung jawab unggahan.

  Contoh: `[T1][PBA-TIF-B] Elon Gates.ZIP`

  Di dalam fail .ZIP di atas berisi:

  - Fail **.DOCX/.PDF** untuk ***Source Code, Screenshot output,*** dan **penjelasan** kode (seluruh jawaban soal jadi satu dokumen). Gunakan templat yang disediakan di BRONE bagian "Informasi Dosen, Kontrak Kuliah, Nilai Keaktifan, Ketua Kelas, Template"
  - Fail **.py** dengan format: **T1_NoSoal_NamaMahasiswa.py** (dengan tanda _ (*underscore*))
  - Fail lainnya yang diperlukan

B. TUJUAN TUGAS

Pada tugas ini mahasiswa diharapkan mengetahui dan memahami cara untuk melakukan pemrosesan teks. Tugas ini menitikberatkan pada pemrosesan teks dasar menggunakan *Regular Expression* (Regex). Dalam tugas ini mahasiswa melakukan pencarian string, ekstraksi informasi sederhana menggunakan Regex. Teknik-teknik untuk melakukan pencarian, pembuatan pola string pencarian Regex, pemilihan struktur data yang tepat merupakan hal yang penting. Dengan menguasai pemrosesan teks dasar maka nantinya akan mempermudah mahasiswa untuk melakukan pengolahan data berupa teks terutama di bidang pengolahan bahasa alami.

C. DESKRIPSI TUGAS

Tugas ini menitikberatkan dalam melakukan pemrosesan teks di tingkat kata untuk kasus sederhana terutama untuk pemrosesan awal (*pre-processing*), seperti tokenisasi, *case folding, stopword removal,* dll. dan melakukan pencarian dan ekstraksi Informasi sederhana menggunakan Regex dari dokumen yang telah diberikan. Oleh karena itu tiap mahasiswa diwajibkan mengerjakan tugas dengan sebaik-baik mungkin.

D. SOAL DASAR PEMROSESAN TEKS (REGULAR EXPRESSION)

1. Buat program menggunakan Python (3.x) untuk memproses dokumen yang tersedia. Program ditulis untuk mengerjakan pertanyaan **nomor 4**.
2. Lakukan *pre-processing* apabila diperlukan, misalnya menghilangkan tanda baca, dll. (baca dan amati soal dengan baik sebelum melakukan *pre-processing*, karena ada tanda baca yang diperlukan).
3. Lakukan proses tokenisasi di tingkat kata. Silakan memilih sendiri metode yang digunakan untuk tokenisasi. Tokenisasi paling mudah adalah dengan Regex yang mana satu kata dibatasi oleh *'whitespace'.*
4. Dari dokumen-dokumen yang diberikan, cari informasi berikut:
   a. Berkas **doc-1.txt** terdiri dari beberapa referensi, misalnya:

```
Goldberg, Yoav (2016). "A Primer on Neural Network Models for Natural Language Processing". Journal of Artificial Intelligence Research. 57: 345–420. arXiv:1807.10854.

Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). Deep Learning. MIT Press.

Jozefowicz, Rafal; Vinyals, Oriol; Schuster, Mike; Shazeer, Noam; Wu, Yonghui (2016). Exploring the Limits of Language Modeling. arXiv:1602.02410.
```

   Dengan menggunakan Regex, ekstraklah ke *dictionary/dict()* semua **nama** (key=**authors**), **tahun** (*key*=**year**), dan **judul** (key=**title**) pada berkas tersebut. Bila tidak ditemukan tahun maka tidak perlu dituliskan. Simpan hasilnya pada berkas **a_judul.json** dengan format JSON.
   Gunakan module Python: `import json`. Contoh hasil:

```
[
    {
        "authors": "Goldberg, Yoav",
        "year": "2016",
        "title": "A Primer on Neural Network Models for Natural Language Processing"
    },
    {
        "authors": "Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron",
        "year": "2016",
        "title": "Deep Learning"
    },
    {
        "authors": "Jozefowicz, Rafal; Vinyals, Oriol; Schuster, Mike; Shazeer, Noam; Wu,
Yonghui",
        "year": "2016",
        "title": "Exploring the Limits of Language Modeling"
    },
    {
        "authors": "Choe, Do Kook; Charniak, Eugene",
        "title": "Parsing as Language Modeling"
    },
…
```


b. Cari **30 kata-kata unik (kecuali stopword) dan *case insensitive* (*case folding*) beserta frekuensinya** (terurut menurun/*descending*) yang muncul di dokumen **doc-2.txt** Simpan dalam **b_kataunik.txt.** Format penyimpanan:

`kata\tFrekuensi` → `\t` merupakan karakter tabulasi

Misalnya:

```
yang 100
di 75
…
```

c. Fail **doc_3.srt** berisi subtitle suatu film. Subtitle tsb. Menggunakan format:

1	Nomor baris
00:00:00,010 --> 00:00:40,010	Penanda waktu
Teks …	Teks percakapan, bisa lebih dari satu baris
<ENTER/NEWLINE>	Tiap akhir percakapan harus ada baris kosong

Lakukan proses *cleaning* pada file tsb. dengan cara menghilangkan semua isi yang tidak penting dengan Regex, tidak sekedar replace teks biasa (str.replace()), antara lain:

- Nomor baris
- Penanda waktu, dengan format mis. 00:00:00,000 --> 00:00:00,000
- Tag <i>, <font>, dll. dalam teks
- Newline kosong

Simpan dalam **c_subtitle.txt.**

Contoh hasil *cleaning* (mulai baris 2, baris 1 ada iklan):

```
Giliranku.
Aku dapat.
Bisa kau buat...
...sedikit lebih menantang?
Baik. Dengar.
…
```

5. Untuk setiap Regex yang dibuat, berikan penjelasannya dalam fail .DOCX dan tuliskan siapa yang bertanggung jawab dalam membuatnya.

E. PENGUMPULAN

Kompres semua fail (masukan, kode program, termasuk misal hasil *pre-processing,* dan keluaran) ke .ZIP atau .RAR yang berisi:

1. Source code Python (file .py atau Jupyter Notebook .ipynb)
2. Berkas DOCX berisi deskripsi hasil nomor 4.
   Tuliskan siapa yang bertanggung jawab mengerjakan setiap soalnya. Mahasiswa yang tidak mengerjakan tidak akan mendapatkan nilai meski berkelompok.
3. Berkas hasil eksekusi (**a_judul.json, b_kataunik.txt, c_subtitle.txt)**

Selamat mengerjakan sebaik-baiknya. Practice makes perfect.