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

Tugas 1 – Pemrosesan Bahasa Alami

A. ATURAN PENGERJAAN

- Dikerjakan berkelompok **maksimal 3-4 orang (1 orang tidak dianggap berkelompok)**.
- Batas waktu dan pengumpulan melalui Google Classroom
- Nama fail yang dikumpulkan dikompres **(.ZIP/.RAR) meski isinya satu fail** dengan format: **`[T1][PBA-Kelas] Nama-Pertama Ketua Kelompok`**.

  Misalnya:
  ```
  [T1][PBA-X] NamaKetua.zip atau
  [T1][PBA-X] NamaKetua.rar
  ```

  Nama pertama atau kedua di sini maksudnya tidak perlu menuliskan semua nama anggota, tapi nama salah satu anggota saja, sebagai penanggung jawab pengunggah.

- **Baca dan pahami soal dengan sebaik-baiknya supaya tidak ada poin nilai yang terlewatkan. Apabila ada yang tidak dimengerti segera ditanyakan.**
- **Plagiasi (bahkan hanya satu baris saja) tidak akan ditolerir dan akan mendapatkan nilai E.**

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 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 sub-judul, misalnya:
      - 1957-1960: Gemeentelijke Universiteit
      - 1961-1964: Upaya penegerian
      - 2006-sekarang: World Class Entrepreneurial University

      Ekstraklah semua sub-judul pada berkas tersebut menggunakan Regex. Simpan hasilnya pada berkas **a_subjudul.txt.** Format penyimpanan berkas:

      `tahun1 s.d. tahun2\tJudul` → \t merupakan karakter tabulasi

      Misalnya:

1957 s.d. 1960	Gemeentelijke Universiteit
1961 s.d. 1964	Upaya penegerian
2006s.d. sekarang	World Class Entrepreneurial University

   b. Amati tiap kemunculan tahun dengan format yang berbeda-beda kemudian ekstrak tahun yang muncul pada dokumen **doc-2.txt** (Regex yang digunakan hanya satu baris, bukan Regex yang terpisah-pisah)
      Misalnya:
      - tahun 671
      - abad ke-7
      - 605 Saka
      - dan masih banyak format tahun lainnya

      Simpan dalam berkas **b_abadtahun.txt**

   c. Cari **30 kata-kata unik (kecuali stopword) beserta frekuensinya** (terurut menurun/*descending*) yang muncul di dokumen **doc-1.txt**
      Simpan dalam **c_kataunik.txt.** Format penyimpanan:

      `kata\tFrekuensi` → \t merupakan karakter tabulasi

      Misalnya:

      ```
      yang 100
      di 75
      …
      ```

d. Fail **subtitle.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 **d_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 berkas (masukan, termasuk misal hasil *pre-processing* maupun keluaran) berisi:

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

Selamat mengerjakan sebaik-baiknya. Practice makes perfect.