

PROJECT BIG CHALLENGE

SINF-1001 PEMOGRAMAN

SEMESTER GANJIL 2025/2026

Dosen Pengasuh

Prof. Dr. Taufik Fuadi Abidin, S.Si., M.Tech
Fitria Nilamsari, M.Sc

PENDAHULUAN

Project Big Challenge adalah projek setara **Ujian Akhir Semester (UAS)**. Project dikerjakan oleh mahasiswa secara berkelompok (daftar kelompok dapat dilihat dibagian akhir penjelasan Project Big Challenge). Penilaian dari projek ini meliputi **laporan akhir dan video presentasi berdurasi antara 10 menit yang dipresentasikan oleh mahasiswa dalam kelompok**. Projek ini berbobot **30%** dari nilai akhir mata kuliah, sama dengan bobot **UAS**.

Setelah mengerjakan **Project Big Challenge** ini, mahasiswa diharapkan mampu:

1. Meningkatkan pemahaman terhadap konsep bahasa pemograman ANSI/C.
2. Mengerti cara menyelesaikan masalah menjadi suatu bentuk solusi menggunakan bahasa pemograman ANSI/C secara berkelompok (maksimum 2 orang).
3. Membaca data dalam file teks, memecahkan kalimat dalam bentuk kata-kata (token), mengelola kata berdasarkan huruf awal (alphabet), dan menghimpun data kata-kata dalam struktur data array dan struct.
4. Menyimpan hasil luaran dalam format binari, membaca kembali data yang telah disimpan dalam format binari tersebut, dan menampilkan hasilnya ke layar monitor.

Mahasiswa tidak dibenarkan memberikan hasil pekerjaannya kepada mahasiswa yang lain, namun diskusi sesama mahasiswa dibolehkan. Bila dalam penyelesaian Project Big Challenge ini, ada fungsi atau program yang diambil dari internet atau referensi lain maka sumber tersebut wajib ditulis di bagian informasi program. Namun, mahasiswa wajib memahami maksud dari setiap bagian yang dikutip tersebut.

PENGUMPULAN PROJECT BIG CHALLENGE

Project Big Challenge harus dikumpulkan paling telat pada tanggal **9 Desember 2025, pukul 23.59 WIB** via sistem e-learning. Dokumen yang harus dikumpulkan adalah **Laporan Penyelesaian Project Big Challenge** yang meliputi:

- Cover
- Permasalahan
- Metode Scoring
- Hasil
- Referensi
- Lampiran (jika ada)

Laporan ditulis menggunakan font Times New Roman 12, spasi tunggal, dan format PDF. **Tautan video presentasi** yang telah diunggah ke channel Youtube wajib disertakan dalam laporan di bagian **Hasil**. Selain laporan, semua kode program harus juga disertakan, dikompres dalam format **zip**. Nama file yang dikumpulkan ditulis dalam format, contoh: **Julia_2508107010101.zip**

PROBLEM

Anda dimintakan untuk membuat program dalam **bentuk modular** yang dapat **membaca data dari sebuah file teks, mengabaikan semua string dalam tag <url> ... </url>, menghimpun semua kata diantara tag <title> ... </title> dan tag <body> ... </body>** yang telah dibersihkan dari tanda baca, dibersihkan dari digit, dan diubah menjadi huruf kecil (*lower-case*). Selain itu, anda diminta untuk **menghimpun kata-kata dengan abjad awal yang sama beserta dengan jumlah kemunculannya (frekuensinya)**. Kemudian, setelah semua data dalam file teks diproses, **urut data per abjad berdasarkan frekuensinya, dari urutan terbesar ke terkecil (descending order)**. Hasil pengurutan itu kemudian **disimpan** dalam sebuah **file binari** dengan **susunan** sebagai berikut:

```
[abjad][jumlah_kata_dengan_abjad_tersebut][panjang_kata][kata][frekuensi]
[panjang_kata][kata][frekuensi][panjang_kata][kata][frekuensi]... dstnya sebanyak jumlah
kata dengan abjad a tersebut [abjad][jumlah_kata_dengan_abjad_tersebut][panjang_kata]
[kata][frekuensi][panjang_kata][kata][frekuensi][panjang_kata][kata][frekuensi]...
dstnya sebanyak jumlah kata dengan abjad b tersebut, .... dstnya sampai abjad z.
```

Dimana:

[abjad]	bertipe char, meliputi {a, b, c, ..., y, z}
[jumlah_kata]	bertipe integer
[panjang_kata]	bertipe integer
[kata]	bertipe string atau <i>array of char</i>

Catatan: *Tanda kurung [] tidak disimpan, hanya ditulis untuk memisahkan bagian saja untuk memudahkan pemahaman format penulisan ke dalam file tipe binari.*

Anda juga **diminta untuk menampilkan** maksimum **n** kata dengan nilai frekuensi terbesar ke layar monitor. Nilai **n** diinput oleh user via keyboard. Selain itu, anda juga **diminta untuk menampilkan kata** dalam frekuensi yang sama menurut kata dengan panjang terbesar terlebih dahulu. Misal, jika kata *aceh* frekuensinya 10, dan kata *alamat* juga memiliki frekuensinya 10, maka karena kata *alamat* panjangnya 6 dan kata *aceh* panjangnya 4, maka kata *alamat* ditampilkan terlebih dahulu di layar monitor, baru kata *aceh*. Contoh lain, kata *alamak* memiliki frekuensinya 12, dan kata *alamat* juga memiliki frekuensi 12, maka kata *alamat* ditampilkan terlebih dahulu baru kata *alamak* karena kata *alamat* > *alamak*.

Program yang dibuat harus menyediakan fungsionalitas minimal sebagai berikut:

Masukkan nama file input (txt): (*menunggu masukkan dari user*)

```
***** Petunjuk *****
```

Contoh file input:

https://elearning.usk.ac.id/pluginfile.php/243811/mod_assign/intro/Data-50K.txt

Gunakan file teks tersebut untuk **menguji hasil dari program Big Challenge** yang dikerjakan. Anda dapat menggunakan file teks yang lebih kecil sangat melakukan pengembangan dan ketika sudah selesai, dapat diuji menggunakan file tersebut.

Langkah-langkah yang harus dilakukan selanjutnya adalah:

1. Buka file input tersebut.
2. Baca baris per baris sampai akhir dari file dan hilangkan yang tidak diperlukan, termasuk tanda baca dan digit serta mengubah semua kata menjadi huruf kecil.
3. Tokenisasi kata-kata yang ada dalam setiap baris. Gunakan fungsi strtok <http://www.elook.org/programming/c/strtok.html>
4. Gunakan **array** dan **struct** untuk menghimpun kata, frekuensi, dan panjang katanya.
5. Gunakan metode pengurutan sederhana seperti *bubble sort*, *selection sort*, atau *insertion sort* untuk melakukan pengurutan.
6. Selanjutnya, MENU pilihan ditampilkan di layar monitor. Pilihan dapat ditentukan oleh user. Program dilanjutkan sesuai pilihan user.

***** Akhir dari Petunjuk *****

Pilihan:

- 1) Simpan luaran dalam file binari dalam susunan yang telah diuraikan sebelumnya.
- 2) Tampilkan di layar monitor maksimum n buah kata ($0 < n \leq 25$). Nilai n diinputkan oleh user via *keyboard*. Setelah nilai n diinputkan, tampilkan ke layar monitor kata - kata sesuai abjad sebanyak n kata beserta frekuensinya. Jika frekuensinya sama, maka kata yang lebih panjang terlebih dahulu dimunculkan ke layar monitor. Jika panjang kata dan frekuensi juga sama maka kata yang lebih besar ditampilkan terlebih dahulu. **Ingat, data harus dibaca dari file binari yang sudah disimpan pada poin (1).** Jika poin (1) belum dipilih, sementara pilihan (2) duluan dipilih, maka anda harus menyimpan luaran terlebih dahulu ke dalam file binari sesuai susunan yang telah diuraikan, kemudian membaca datanya kembali untuk menampilkan ke layar monitor.

Contoh keluaran untuk $n = 3$ adalah sebagai berikut:

```
abjad {kata (frekuensi)}
a   {alamak (10), alamat (10), aceh (5)}
b   {boikot (20), bollywood (5), bali (3)}
:
:
z   {} jika tidak ada kata dengan huruf z
```

Kolom pertama adalah **abjad**, kemudian diikuti dengan seluruh kata dengan abjad tersebut, dalam format **kata (frekuensi kata)** tersebut dalam kurung kurawal.

- 3) Selesai

Pilihan anda: _

Jika **pilihan (1)** dipilih maka program akan menyimpan keluaran dari data yang telah terurut berdasarkan abjad (*descending order*) dalam sebuah file binari dengan susunan yang telah diuraikan sebelumnya. **Penentuan nama file binari** boleh dilakukan secara manual (langsung ditulis dalam kode program) atau dapat pula ditentukan secara dinamis (diinputkan dan ditanyakan kepada user pada saat program dijalankan), atau boleh juga melalui *command line argument*.

Jika **pilihan (2)** dipilih maka program akan meminta kepada user nilai n ($0 < n \leq 25$) dan kemudian menampilkan sebanyak n buah kata dengan frekuensi terbesar ke layar monitor. Data n kata tersebut dibaca dari file binari yang dibuat pada saat pilihan 1 dipilih. Jika user langsung memilih **pilihan (2)** sebelum **pilihan (1)** dijalankan maka anda diminta untuk menyimpannya terlebih dahulu sama seperti **pilihan (1)** dipilih. Jika **pilihan (3)** yang dipilih maka program akan berakhir.

Kelompok Big Challenge akan ditentukan oleh dosen dan diinfokan melalui WAG.



(c) 2025/2026

