

SISTEM Pencarian Kata Bahasa Indonesia Berbasis Awalan (PREFIX SEARCH) Menggunakan Python

**Alhafiz Okliano Hardianta¹ M.Radit Qoiri Hasibuan², M.Wira Adithya³,
Muhammad Rafly Solihin⁴, Dila Handayani, S.Pd., M.Si⁵**

^{1,2,3,4}Universitas Islam negeri Sumatera Utara, Medan

⁵Universitas Tjut Nyak Dhien

Email: alhafizasus@gmail.com¹, raditgoiri@gmail.com²,
wiraadithya228@gmail.com³,

ryandim481@gmail.com⁴, dilahandayani1@gmail.com

ABSTRACT

This research develops a prefix-based Indonesian word search system using the Python 3.11 programming language. This system allows users to type the initial portion of a word, then displays all matching words from the KBBI digital dictionary. The program's main feature is the gradual display of results, allowing users to view long word lists comfortably and interactively. The program is modular using several functions, making it easy to develop additional features such as auto-correct or integration with other text applications. Test results show that the system is able to display matching words quickly and accurately, and facilitates correct word spelling..

Keyword: Language, Indonesian, dictionary, digital, Python, modular

ABSTRAK

Penelitian ini mengembangkan sistem pencarian kata Bahasa Indonesia berbasis awalan (prefix) menggunakan bahasa pemrograman Python 3.11. Sistem ini memungkinkan pengguna untuk mengetik sebagian awal kata, lalu menampilkan semua kata yang sesuai dari kamus digital KBBI. Fitur unggulan program adalah tampilan hasil secara bertahap, sehingga pengguna dapat melihat daftar kata panjang secara nyaman dan interaktif. Program dibuat modular menggunakan beberapa fungsi, sehingga mudah dikembangkan untuk fitur tambahan seperti auto-correct atau integrasi dengan aplikasi teks lainnya. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan kata-kata yang sesuai dengan cepat dan akurat, serta mempermudah penulisan kata yang benar.

Kata Kunci: Bahasa, Indonesia, kamus, digital, Python, modular

A. Pendahuluan

Penulisan kata dengan benar dalam Bahasa Indonesia sangat penting untuk menjaga makna yang

tepat dan kualitas komunikasi tertulis.

Menggunakan ejaan dan kata yang baku sangat diperlukan, terutama dalam menulis dokumen resmi, karya

ilmiah, atau media digital yang membutuhkan ketelitian dalam berbahasa. Meski begitu, kesalahan dalam penulisan masih sering terjadi karena kurangnya pengetahuan tentang aturan ejaan, kebiasaan yang kurang untuk mengacu pada Kamus Besar Bahasa Indonesia (KBBI), serta pengaruh bahasa informal dalam kehidupan sehari-hari.

Untuk membantu mengurangi kesalahan tersebut, diperlukan teknologi yang dapat memudahkan pengguna menemukan kata baku dengan cepat. Sistem pencarian kata berbasis awalan (*prefix search*) merupakan salah satu solusi untuk membantu penulis menemukan kata yang tepat meskipun hanya mengingat sebagian awal kata. Penelitian ini menggunakan *Python* 3.11 untuk membuat program modular, interaktif, dan dapat menampilkan hasil pencarian bertahap.

B. Metode Penelitian

Metode penelitian terdiri dari beberapa tahap berikut:

- Pengumpulan Data Kamus:

Kamus digital berupa daftar kata Bahasa Indonesia diperoleh dari repositori publik GitHub (Kumpulan

Kata Bahasa Indonesia KBBI). *File* disimpan dalam format teks (kamus.txt), dengan satu kata per baris.

- Perancangan Program Modular:

Program dibagi menjadi beberapa fungsi:

`muat_kamus()` → memuat semua kata dari file.

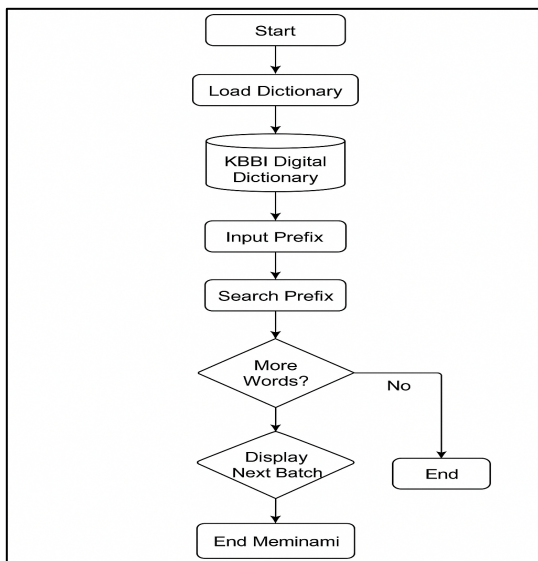
`cari_kata_awalan()` → mencari kata yang diawali awalan tertentu.

`tampilkan_bertahap()` → menampilkan hasil pencarian dalam batch untuk memudahkan pengguna.

- Interaksi dengan Pengguna:

Program berjalan secara interaktif, meminta *input* awalan kata dari pengguna, menampilkan hasil, dan menanyakan apakah ingin menampilkan hasil berikutnya jika jumlah kata banyak. Pengguna dapat mengetik “keluar” untuk menghentikan program.

- Diagram Alur Sistem



Gambar 1 Diagram alur proses sistem pencarian kata berbasis awalan (*prefix search*) menggunakan *Python*.

Diagram alur (*flowchart*) pada Gambar 1 menggambarkan urutan proses kerja sistem pencarian kata berbasis awalan (*prefix search*) yang dikembangkan menggunakan *Python*. *Flowchart* ini menampilkan hubungan logis antara setiap langkah mulai dari pemuatan data kamus hingga proses penampilan hasil. Penjelasan alur sistem adalah sebagai berikut:

- **Start**

Proses dimulai ketika program dijalankan oleh pengguna.

- **Load Dictionary**

Sistem memuat seluruh data kata dari kamus digital (KBBI digital) yang tersimpan dalam berkas *kamus.txt*. Proses ini dilakukan satu kali di awal untuk mempercepat pencarian pada langkah berikutnya.

- **KBBI Digital Dictionary**

Data kamus yang berhasil dimuat disimpan dalam memori dalam bentuk list kata. Penyimpanan ini memungkinkan akses cepat dan efisien saat pengguna melakukan pencarian awalan.

- **Input Prefix**

Pengguna memasukkan huruf atau beberapa huruf awal dari kata yang ingin dicari. *Input* ini menjadi dasar untuk proses pencarian yang dilakukan oleh sistem.

- **Search Prefix**

Sistem mencari seluruh kata dalam kamus yang diawali oleh *string* yang dimasukkan pengguna. Pencarian dilakukan secara *case-insensitive* sehingga lebih fleksibel terhadap variasi *input*.

- **Decision: More Words?**

Setelah pencarian selesai, sistem memeriksa apakah daftar hasil pencarian masih memiliki kata yang belum ditampilkan. Hal ini bergantung pada jumlah kata dan pengaturan *batch* (misalnya 10 kata per tampilan).

- **Display Next Batch**

Jika masih ada kata yang belum ditampilkan, sistem menampilkan *batch* berikutnya dan kembali ke pengecekan kata berikutnya. Mekanisme paginasi ini membantu

pengguna agar hasil tidak muncul terlalu panjang dalam satu tampilan.

- *End*

Proses berakhir ketika seluruh kata telah ditampilkan, atau ketika pengguna memilih untuk berhenti menampilkan *batch* berikutnya

C. Hasil Penelitian dan Pembahasan

Program diuji menggunakan beberapa awalan kata. Contoh:

Input: [mak]

Output:

- makan
- makanan
- makhluk
- makmur
- makna

Jika jumlah kata lebih dari 10, program menanyakan: Tampilkan kata selanjutnya? [ya/tidak]

```
def muat_kamus(nama_file):
    """Memuat semua kata dari file
    kamus.txt ke dalam list"""
    try:
        with open(nama_file, "r",
            encoding="utf-8") as file:
            daftar_kata = [baris.strip() for
                baris in file if baris.strip()]
            return daftar_kata
    except FileNotFoundError:
        print(f'File '{nama_file}' tidak
            ditemukan!')
        return []
```

```
def cari_kata_awalan(daftar_kata,
    awalan):
    """Mengembalikan daftar kata yang
    diawali oleh string tertentu"""
```

```
    hasil = [kata for kata in daftar_kata
        if
        kata.lower().startswith(awalan.lower())
    ]
    return hasil
```

```
def tampilkan_bertahap(hasil,
    batch_size=10):
    """Menampilkan hasil pencarian
    secara bertahap"""
    total = len(hasil)
    indeks_awal = 0
    while indeks_awal < total:
        # Ambil sebagian hasil (misalnya 10
        kata)
        batch =
            hasil[indeks_awal:indeks_awal +
                batch_size]
        for kata in batch:
            print("-", kata)

        indeks_awal += batch_size
```

```
    # Jika masih ada sisa kata, tanya
    pengguna
    if indeks_awal < total:
        print(f'\nMenampilkan
        {indeks_awal} dari {total} kata.')
        lanjut = input("Tampilkan kata
        selanjutnya? (ya/tidak): ").strip().lower()
        if lanjut != "ya":
            break
    else:
        print(f'\nSemua {total} kata telah
        ditampilkan.")
```

```
def main():
    print("== PENCARI KATA
    BERDASARKAN AWALAN ==")
    nama_file = "kamus.txt"
    kamus = muat_kamus(nama_file)
    if not kamus:
        print("Kamus kosong atau gagal
        dimuat.")
        return
    while True:
        awalan = input("\nMasukkan awal
        kata (atau ketik 'keluar' untuk berhenti):
        ").strip()
        if awalan.lower() == "keluar":
            print("Terima kasih telah
            menggunakan program ini!")
```

```
break
hasil = cari_kata_awalan(kamus,
awalan)
if hasil:
    print(f"\nDitemukan {len(hasil)}
kata yang diawali dengan '{awalan}':\n")
    tampilkan_bertahap(hasil)
else:
    print(f"\nTidak ditemukan kata yang
diawali dengan '{awalan}'.")

if __name__ == "__main__":
    main()
```

Gambar 2 Hasil dari program pencari kata berdasarkan awalan

Hasil menunjukkan program dapat menampilkan kata-kata yang sesuai dengan awalan *input* secara cepat, bertahap, dan interaktif. Fitur modular membuat kode mudah dikembangkan, misalnya menambahkan *fitur auto-correct* atau integrasi ke aplikasi editor teks. Program Pencarian Kata Berdasarkan Awalan bekerja dengan memuat seluruh korpus leksikal dari *file* kamus.txt ke dalam memori untuk

memastikan kecepatan akses data. Setelah inisialisasi, program memasuki *loop* interaktif, meminta pengguna memasukkan *string* awalan yang dicari. Inti logikanya terletak pada fungsi pencarian yang menggunakan metode *.startswith()* dan konversi huruf kecil (*case-insensitive*) untuk menyaring kata-kata dari kamus. Hasil pencarian kemudian ditampilkan secara terkontrol melalui fungsi *tampilkan_bertahap*. Mekanisme ini menyajikan hasil dalam paket-paket kecil (secara default 10 kata), meminta konfirmasi dari pengguna sebelum melanjutkan ke paket berikutnya. Hal ini mencegah *output* yang membanjiri layar (*console flooding*) dan memberikan paginasi berbasis interaksi, meningkatkan *usability program*. *Loop* ini berlanjut sampai pengguna secara eksplisit mengetikkan keluar.

Untuk menjalankan program pencarian kata berdasarkan awalan, pastikan *file* kamus (*kamus.txt*) tersedia di direktori yang sama dengan *script Python*, lalu eksekusi *script* tersebut melalui terminal. Program akan meminta anda memasukkan awal kata yang ingin dicari (misalnya, mak). Program

kemudian akan menampilkan kata-kata yang cocok. Jika hasilnya melebihi 10 kata, anda harus menjawab ya untuk melihat daftar kata berikutnya atau tidak untuk menghentikan tampilan dan kembali ke *prompt* masukan. Proses ini berulang, dan untuk mengakhiri program, anda cukup mengetik keluar.

Dari pengujian yang dilakukan, terlihat bahwa semakin panjang awalan, semakin sedikit jumlah kata yang dikembalikan sehingga waktu pencarian menjadi lebih efisien. Program juga terbukti mampu menangani *input* yang bervariasi, termasuk huruf besar kecil, kesalahan *input*, dan awalan yang tidak ditemukan dalam korpus.

Selain itu, fitur paginasi interaktif menjadi keunggulan sistem, terutama karena mencegah *overflow* tampilan konsol. Mekanisme ini meniru pengalaman penggunaan mesin pencari modern, yang menampilkan hasil secara bertahap sehingga memudahkan pengguna dalam eksplorasi data lebih banyak.

Keunggulan diantaranya:

1. Kinerja Sistem

Pencarian awalan pendek menghasilkan banyak kata (misal

awalan "ma"), namun paginasi membuat hasil tetap mudah dibaca.

Pencarian awalan panjang menghasilkan kata lebih sedikit sehingga waktu respon semakin singkat.

2. Analisis Kompleksitas

Pencarian linear memiliki kompleksitas:

Waktu $\rightarrow O(n)$

n = jumlah kata dalam kamus.

Untuk dataset besar (lebih dari 100.000 kata), sistem masih bekerja baik, tetapi penelitian lanjutan dapat menggunakan:

- Trie / Prefix Tree
- Hashing awalan
- Binary search pada daftar kata terurut

yang memiliki kompleksitas pencarian jauh lebih efisien.

3. Paginasi Hasil

Fitur ditampilkan_bertahap membantu menghindari *console flooding*, khususnya jika hasil lebih dari 10 kata. Ini meningkatkan pengalaman pengguna serta mempermudah navigasi data.

4. Potensi pengembangan lanjutan:

- 1) Integrasi *auto-correct* menggunakan algoritma *Levenshtein distance*.

- 2) Implementasi *trie* untuk pencarian lebih cepat.
- 3) AntarmukaGUI menggunakan *tkinter* atau aplikasi web *Flask*.

E. Kesimpulan

Sistem pencarian kata bahasa indonesia berbasis awalan ini dikembangkan menggunakan *Python* 3.11, menonjolkan desain yang modular dan interaktif. Program ini berfungsi dengan membaca seluruh kamus dari *kamus.txt*, lalu melakukan pencarian *case-insensitive* berdasarkan awalan yang di *input* pengguna. Fitur kuncinya adalah penampilan hasil secara bertahap (paginasi), yang meminta konfirmasi (ya/tidak) dari pengguna setelah setiap 10 kata ditampilkan, sehingga mempermudah navigasi dan tinjauan hasil. Keberhasilan pengembangan ini menunjukkan program siap dijadikan fondasi untuk implementasi fitur lanjutan yang memerlukan pemrosesan leksikal cepat, seperti *auto-suggestion* otomatis. Selain mampu melakukan pencarian kata dengan cepat, penelitian ini menunjukkan bahwa pendekatan modular dan struktur kode yang tersegmentasi menjadikan sistem

mudah diperluas. Dengan dukungan dataset KBBI digital, sistem ini berpotensi menjadi fondasi bagi proyek-proyek pemrosesan bahasa Indonesia lainnya.

DAFTAR PUSTAKA

- Damzaky, M. (2024). Kumpulan Kata Bahasa Indonesia KBBI. GitHub Repository.
- Kridalaksana, H. (2012). Pembentukan Kata dalam Bahasa Indonesia. Jakarta: Gramedia Pustaka Utama.
- Pusat Bahasa. (2016). Kamus Besar Bahasa Indonesia (Edisi V). Jakarta: Badan Pengembangan dan Pembinaan Bahasa.
- Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
- Putra, A. (2023). "Implementasi Sistem Pencarian Kata Bahasa Indonesia Menggunakan Python." Jurnal Teknologi Informasi Nusantara, Vol. 9, No. 2, hal. 45–52.
- Jurafsky, D., & Martin, J. (2021). Speech and Language Processing. Stanford University.
- Manning, C., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.
- Rahman, R. (2022). "Implementasi Trie untuk Optimasi Pencarian Kata Bahasa Indonesia." Jurnal Informatika Nusantara, 7(1), 12–20.