

[T2] Mesin Ketik Bu Sari

Struktur Data dan Algoritma
Departemen Teknik Komputer ITS

1 Deskripsi Masalah

Bu Sari adalah seorang sekretaris di kantor kelurahan yang baru saja mendapatkan mesin ketik listrik canggih untuk mengetik surat-surat penting. Mesin ketik ini memiliki fitur khusus yang dapat menyimpan riwayat pengetikan dan memungkinkan Bu Sari untuk membatalkan kesalahan pengetikan.

Mesin ketik Bu Sari memiliki buffer teks yang dapat menyimpan hingga 500 karakter. Bu Sari dapat melakukan operasi-operasi berikut pada mesin ketiknya:

1. **KETIK**: Menambahkan karakter di akhir buffer
2. **HAPUS**: Menghapus karakter terakhir dari buffer
3. **BATAL**: Membatalkan operasi terakhir (undo)
4. **LIHAT**: Melihat isi buffer saat ini
5. **CARI**: Mencari posisi kemunculan pertama sebuah kata

Mesin ketik ini memiliki memori terbatas, sehingga hanya dapat menyimpan maksimal 20 operasi dalam riwayat untuk fitur BATAL.

Karena Bu Sari sering membuat kesalahan ketik, ia sangat bergantung pada fitur BATAL untuk memperbaiki kesalahannya. Bantu Bu Sari menggunakan mesin ketik canggih ini!

2 Format Masukan dan Keluaran

2.1 Format Masukan

Masukan terdiri atas beberapa baris. Baris pertama berisi satu buah bilangan bulat positif N ($1 \leq N \leq 100$) yang merupakan banyak operasi. N baris berikutnya masing-masing berisi operasi yang akan dilakukan dengan format sebagai berikut:

- **KETIK karakter** - Menambahkan satu karakter ke buffer
- **HAPUS** - Menghapus karakter terakhir
- **BATAL** - Membatalkan operasi terakhir
- **LIHAT** - Menampilkan isi buffer

- CARI kata - Mencari posisi kemunculan pertama kata

Batasan:

- Karakter hanya berupa huruf (a-z, A-Z), angka (0-9), dan spasi
- Panjang kata untuk pencarian maksimal 50 karakter
- Buffer maksimal 500 karakter
- Riwayat operasi maksimal 20

2.2 Format Keluaran

Untuk setiap operasi, keluarkan respons sesuai format berikut:

- **KETIK:**
 - Jika berhasil: OK
 - Jika buffer penuh: BUFFER PENUH
- **HAPUS:**
 - Jika berhasil: OK
 - Jika buffer kosong: BUFFER KOSONG
- **BATAL:**
 - Jika berhasil: DIBATALKAN
 - Jika tidak ada operasi untuk dibatalkan: TIDAK ADA OPERASI
- **LIHAT:** Tampilkan isi buffer. Jika kosong: KOSONG
- **CARI:**
 - Jika ditemukan: DITEMUKAN posisi_awal
 - Jika tidak ditemukan: TIDAK DITEMUKAN

3 Contoh Masukan/Keluaran

Masukan	Keluaran
12	
KETIK H	OK
KETIK a	OK
KETIK l	OK
KETIK o	OK
LIHAT	Halo
HAPUS	OK
LIHAT	Hal
BATAL	DIBATALKAN
LIHAT	Halo
KETIK	OK
CARI lo	DITEMUKAN 2
CARI xyz	TIDAK DITEMUKAN

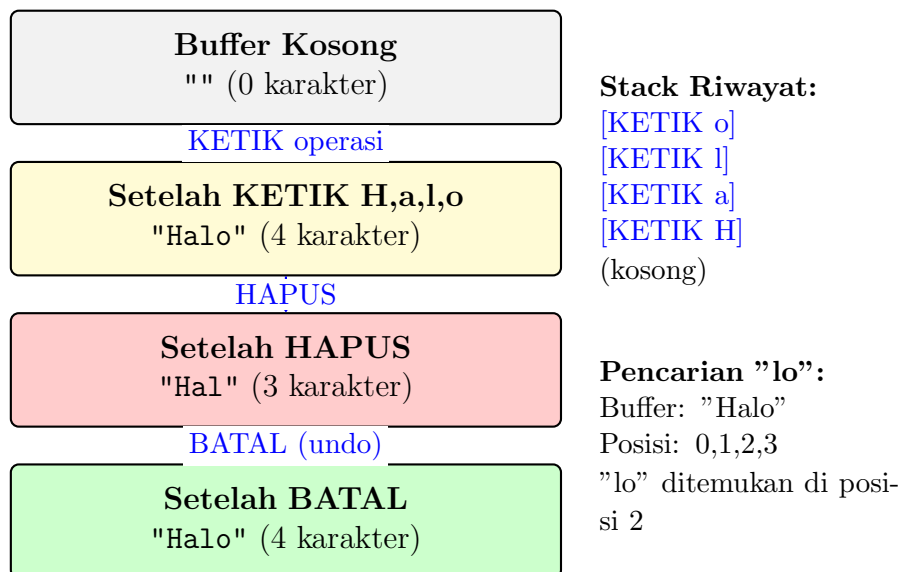
4 Penjelasan

Mari kita ikuti alur operasi pada contoh di atas:

1. **KETIK H, a, l, o**: Menambahkan karakter satu per satu ke buffer. Buffer menjadi "Halo".
2. **LIHAT**: Menampilkan isi buffer saat ini: "Halo".
3. **HAPUS**: Menghapus karakter terakhir 'o'. Buffer menjadi "Hal".
4. **LIHAT**: Menampilkan isi buffer: "Hal".
5. **BATAL**: Membatalkan operasi HAPUS terakhir. Buffer kembali menjadi "Halo".
6. **LIHAT**: Menampilkan isi buffer: "Halo".
7. **KETIK (spasi)**: Menambahkan spasi. Buffer menjadi "Halo ".
8. **CARI lo**: Mencari substring "lo" dalam "Halo ". Ditemukan di posisi 2.
9. **CARI xyz**: Mencari substring "xyz". Tidak ditemukan dalam buffer.

5 Ilustrasi Sistem

Mesin Ketik Bu Sari



6 Strategi Implementasi

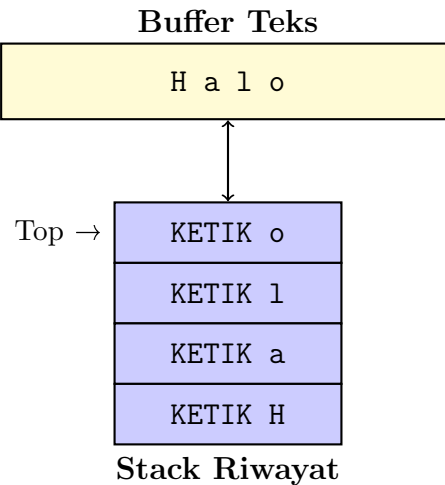
6.1 Struktur Data yang Digunakan

1. **String/Array** untuk menyimpan buffer teks
2. **Stack** untuk menyimpan riwayat operasi (untuk fitur BATAL)
3. **String matching** untuk operasi CARI

6.2 Algoritma Kunci

1. **Stack operations:** Push operasi baru, pop untuk BATAL
2. **String search:** Algoritma sederhana untuk mencari substring
3. **Buffer management:** Validasi kapasitas dan operasi dasar string

7 Contoh Implementasi Konsep



8 Catatan Implementasi

- Gunakan array atau string untuk buffer dengan validasi batas maksimal
- Implementasikan stack sederhana untuk riwayat operasi
- Untuk pencarian string, gunakan algoritma linear search sederhana
- Pastikan penanganan kasus edge: buffer kosong, buffer penuh, riwayat kosong
- Operasi BATAL hanya berlaku untuk KETIK dan HAPUS, tidak untuk LIHAT dan CARI
- Kompleksitas waktu yang diharapkan: $O(1)$ untuk operasi dasar, $O(n)$ untuk pencarian