

[T0] Perpustakaan Bu Ningsih - Contoh Domjudge

Struktur Data dan Algoritma
Departemen Teknik Komputer ITS

1 Pengantar

Ini adalah contoh soal untuk membantu Anda memahami cara menggunakan sistem domjudge dan format competitive programming. Soal ini menggunakan topik dari Part 1: **Array, String, dan Searching**.

Level kesulitan: 4/10 (Sedang - memerlukan pemahaman array dan string manipulation)

2 Deskripsi Masalah

Bu Ningsih adalah pustakawan di perpustakaan kampung yang baru saja mendapat bantuan komputer. Ia ingin membuat sistem sederhana untuk mengelola koleksi bukunya. Setiap buku memiliki kode unik berupa string yang terdiri dari huruf dan angka.

Bu Ningsih memerlukan bantuan untuk:

1. Mencari buku berdasarkan kode
2. Menghitung berapa banyak buku yang kodnya mengandung substring tertentu
3. Mengurutkan daftar buku berdasarkan kode secara alfabetis

Karena Bu Ningsih masih belajar komputer, sistem harus memberikan output yang jelas dan mudah dipahami.

3 Format Masukan dan Keluaran

3.1 Format Masukan

- Baris pertama berisi integer N ($1 \leq N \leq 100$): jumlah buku
- N baris berikutnya berisi kode buku (string, maksimal 20 karakter, hanya huruf dan angka)
- Baris selanjutnya berisi integer Q ($1 \leq Q \leq 50$): jumlah query
- Q baris berikutnya berisi query dengan format:
 - **CARI** `kode` - mencari apakah buku dengan kode tersebut ada
 - **HITUNG** `substring` - hitung buku yang kodnya mengandung substring
 - **URUT** - tampilkan semua kode buku terurut alfabetis

3.2 Format Keluaran

Untuk setiap query:

- **CARI:** ADA atau TIDAK ADA
- **HITUNG:** satu integer (jumlah buku yang mengandung substring)
- **URUT:** semua kode buku, satu per baris, terurut alfabetis

4 Contoh Masukan/Keluaran

Masukan	Keluaran
5 BK001 MA123 BK007 CS202 MA456	
4 CARI BK001 HITUNG BK CARI XY999 URUT	ADA 2 TIDAK ADA BK001 BK007 CS202 MA123 MA456

5 Penjelasan

1. **CARI BK001:** Buku dengan kode "BK001" ada dalam daftar
2. **HITUNG BK:** Ada 2 buku yang kodanya mengandung substring "BK" (BK001 dan BK007)
3. **CARI XY999:** Buku dengan kode "XY999" tidak ada dalam daftar
4. **URUT:** Menampilkan semua kode buku diurutkan secara alfabetis

6 Cara Menggunakan Domjudge

6.1 Langkah 1: Login ke Domjudge

1. Buka browser dan akses URL domjudge yang diberikan dosen
2. Login dengan username dan password yang telah diberikan
3. Pilih contest "Tugas Struktur Data dan Algoritma"

6.2 Langkah 2: Membaca Soal

1. Klik pada problem "T0 - Perpustakaan Bu Ningsih"
2. Baca deskripsi masalah dengan teliti
3. Perhatikan format input dan output
4. Download sample input/output jika tersedia

6.3 Langkah 3: Menulis Kode

Buat file dengan nama `solution.cpp` (atau sesuai bahasa yang digunakan):

Listing 1: Contoh Solusi dalam C++

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <algorithm>
5 using namespace std;
6
7 int main() {
8     int n;
9     cin >> n;
10
11     vector<string> books(n);
12     for (int i = 0; i < n; i++) {
13         cin >> books[i];
14     }
15
16     int q;
17     cin >> q;
18
19     while (q--) {
20         string operation;
21         cin >> operation;
22
23         if (operation == "CARI") {
24             string code;
25             cin >> code;
26
27             bool found = false;
28             for (int i = 0; i < n; i++) {
29                 if (books[i] == code) {
30                     found = true;
31                     break;
32                 }
33             }
34
35             cout << (found ? "ADA" : "TIDAK ADA") << endl;
36
37         } else if (operation == "HITUNG") {
38             string substring;
39             cin >> substring;
40
41             int count = 0;
42             for (int i = 0; i < n; i++) {
```

```

43         if (books[i].find(substring) != string::npos) {
44             count++;
45         }
46     }
47
48     cout << count << endl;
49
50     } else if (operation == "URUT") {
51         vector<string> sorted_books = books;
52         sort(sorted_books.begin(), sorted_books.end());
53
54         for (const string& book : sorted_books) {
55             cout << book << endl;
56         }
57     }
58 }
59
60 return 0;
61 }

```

6.4 Langkah 4: Testing Lokal

Sebelum submit, test kode Anda dengan sample input:

1. Compile: `g++ -o solution solution.cpp`
2. Run: `./solution < input.txt`
3. Bandingkan output dengan expected output

6.5 Langkah 5: Submit ke Domjudge

1. Kembali ke halaman domjudge
2. Pilih problem "T0 - Perpustakaan Bu Ningsih"
3. Click "Submit"
4. Pilih bahasa pemrograman (C++)
5. Upload file `solution.cpp`
6. Click "Submit"

6.6 Langkah 6: Melihat Hasil

1. Tunggu beberapa detik untuk proses judging
2. Lihat status submission:
 - **CORRECT**: Solusi benar untuk semua test case
 - **WRONG-ANSWER**: Output tidak sesuai
 - **TIME-LIMIT**: Program terlalu lambat
 - **RUN-ERROR**: Program crash/error

7 Tips Sukses di Domjudge

7.1 Format Input/Output

- **Baca input persis seperti format:** Jangan skip atau tambah input
- **Output format harus tepat:** Spasi, newline, huruf besar/kecil harus sesuai
- **Jangan print debug info:** Hanya print output yang diminta

7.2 Debugging

- Test dengan sample input/output dulu
- Buat test case sendiri untuk edge cases
- Cek boundary conditions ($N=1$, string kosong, dll)

7.3 Optimasi

- Perhatikan time limit dan constraints
- Untuk $N=100$, algoritma $O(N^2)$ masih aman
- Gunakan algoritma yang tepat untuk setiap operasi

8 Konsep yang Dipelajari

Soal ini mengajarkan:

1. **Array/Vector:** Menyimpan daftar buku
2. **String manipulation:** Mencari substring dengan `find()`
3. **Linear search:** Mencari buku dengan kode tertentu
4. **Sorting:** Mengurutkan array string dengan `sort()`
5. **Input/Output:** Membaca dan menulis data sesuai format

9 Variasi Soal Lanjutan

Setelah memahami soal dasar ini, Anda bisa coba variasi:

- Tambahkan informasi tahun terbit dan cari buku dalam rentang tahun
- Implementasikan binary search setelah sorting
- Tambahkan operasi insert/delete buku
- Gunakan hash table untuk pencarian $O(1)$

10 Troubleshooting Umum

Error	Solusi
WRONG-ANSWER TIME-LIMIT	Cek format output, case sensitivity, spasi extra Optimasi algoritma, hindari nested loop yang tidak perlu
RUN-ERROR	Cek array bounds, division by zero, null pointer
COMPILE-ERROR	Cek syntax, missing headers, typo

Selamat berlatih dan semoga sukses dalam tugas-tugas berikutnya!