Study Case
BNCC LnT C Programming
Sesi 3

📝 Deskripsi Study Case

Sistem Pemantauan Inventaris Gudang Otomatis 'QUANTUM'

Pabrik 'QUANTUM' memiliki gudang besar yang menyimpan berbagai jenis komponen elektronik. Anda ditugaskan untuk merancang sebuah sistem pemantauan inventaris otomatis yang dapat melacak jumlah, lokasi rak, dan status (tersedia/rusak) dari setiap komponen. Sistem ini harus efisien dalam penggunaan memori dan memungkinkan akses data yang cepat. Operator akan memasukkan data komponen baru secara berkala dan juga melakukan pencarian atau pembaruan data komponen yang sudah ada.

Tujuan: Menguji kemampuan mahasiswa dalam:

- Menggunakan *array* untuk menyimpan struktur data yang kompleks.
- Menerapkan *pointer* untuk mengakses dan memanipulasi elemen *array* secara efisien.
- Mengelola memori secara dinamis (opsional, untuk tingkat kesulitan lebih tinggi).
- Mengintegrasikan berbagai struktur kontrol (*if-else, switch-case*) dan perulangan (*for, while, do-while*) dalam skenario manajemen data.

Cara Kerja Sistem

1. Inisialisasi Sistem (Penggunaan Array dan Pointer Awal)

- Program akan menginisialisasi sebuah *array* dari struktur Komponen.

  Contoh a:

```
struct Komponen {
int id;
char nama[50];
int jumlah;
```

char lokasi_rak[10];

bool rusak;

}

Contoh b (tanpa struct):

int id_komponen[100];

char nama_komponen[100][50];

int jumlah_komponen[100];

- Ukuran *array* awal adalah konstan (misalnya, 100 komponen).
- Gunakan *pointer* untuk mengelola posisi kosong pertama dalam *array* untuk penambahan komponen baru.

2. Penambahan Komponen Baru (Pointer untuk Alokasi)

- Operator dapat memilih untuk menambahkan komponen baru.
- Program akan meminta input **ID Komponen, Nama Komponen, Jumlah, Lokasi Rak,** dan **Status Rusak** (0 untuk tidak rusak, 1 untuk rusak).
- Gunakan *pointer* untuk menempatkan data komponen baru ke dalam *array* pada posisi yang tersedia.
- Validasi input: **ID Komponen** harus unik; **Jumlah** harus positif. Jika tidak valid, operator harus mengulang input untuk komponen tersebut.

3. Pencarian dan Pembaruan Komponen (Pointer untuk Navigasi)

- Operator dapat mencari komponen berdasarkan **ID Komponen** atau **Nama Komponen.**
- Gunakan *pointer* untuk melakukan iterasi melalui *array* dan menemukan komponen yang cocok.
- Jika ditemukan, tampilkan detail komponen tersebut.
- Operator kemudian dapat memilih untuk memperbarui **Jumlah, Lokasi Rak,** atau **Status Rusak** dari komponen tersebut.
- Gunakan *pointer* untuk langsung memodifikasi data komponen yang ditemukan dalam *array*.

4. Laporan Inventaris (Pointer untuk Iterasi)

- Operator dapat meminta laporan inventaris lengkap.
- Program akan menampilkan daftar semua komponen yang tersimpan, termasuk **ID, Nama, Jumlah, Lokasi Rak,** dan **Status Rusak.**
- Gunakan *pointer* untuk melakukan iterasi melalui seluruh *array* dan menampilkan setiap elemen.
- Tampilkan juga ringkasan:
    - Total jumlah komponen yang tersedia (tidak rusak).
    - Jumlah komponen yang rusak.

5. Opsi Keluar

- Program akan terus berjalan dalam sebuah *loop* menu utama hingga operator memilih untuk keluar.

Constraints & Ranges

Variabel / Input	Syarat	Range (Note: Kalau ada string: wajib diperhatikan)
ID Komponen “ID”	Bilangan bulat positif, unik. (int)	$1 \leq ID \leq 9999$
Nama Komponen “NK”	String, tidak boleh kosong.	Maks. 49 karakter
Jumlah “J”	Bilangan bulat positif. (int)	$0 \leq J \leq 1000$
Lokasi Rak “LR”	String, format "A1-B2".	Maks. 9 karakter
Status Rusak “SR”	Bilangan bulat (0 atau 1). (int)	0 (Tidak Rusak), 1 (Rusak)
Ukuran Array Maksimal	Konstan. (int)	100

Contoh:

Input & Output

```
================================================
=== Sistem Pemantauan Inventaris QUANTUM (C) ===
================================================
1. Tambah Komponen Baru
2. Cari dan Perbarui Komponen
3. Laporan Inventaris Lengkap
4. Keluar
Pilih opsi (1-4):
```

```
Pilih opsi (1-4): 1

--- Tambah Komponen Baru ---
Masukkan ID Komponen (1-9999): 899
Masukkan Nama Komponen (Maks. 49 karakter): Kapasitor
Masukkan Jumlah (0-1000): 20
Masukkan Lokasi Rak (Maks. 9 karakter, e.g., A1-B2): A2
Masukkan Status Rusak (0=Tidak Rusak, 1=Rusak): 0
? Komponen 'Kapasitor' (ID: 899) berhasil ditambahkan ke Rak A2.
```

```
Pilih opsi (1-4): 2

--- Cari dan Perbarui Komponen ---

Cari berdasarkan:
1. ID Komponen
2. Nama Komponen
Pilih (1/2): 1
Masukkan ID Komponen yang dicari: 899

--- Detail Komponen Ditemukan ---
ID: 899 | Nama: Kapasitor | Jumlah: 20 | Lokasi: A2 | Rusak: Tidak (0)

Pilih yang akan diperbarui:
1. Jumlah
2. Lokasi Rak
3. Status Rusak
4. Batalkan
Pilih (1-4): 2
Masukkan Lokasi Rak Baru (Maks. 9 karakter): B2
? Lokasi Rak berhasil diperbarui.
```

```
Pilih opsi (1-4): 3

========================================================================
                          LAPORAN INVENTARIS QUANTUM
========================================================================
| ID   | Nama Komponen                          | Jumlah | Rak      | Status |
------------------------------------------------------------------------
| 899  | Kapasitor                              | 20     | B2       | TERSEDIA |
------------------------------------------------------------------------

--- Ringkasan Inventaris ---
Total Jenis Komponen Terdaftar: 1
Total Unit Tersedia (Tidak Rusak): 20
Total Unit Rusak (Siap Disposal/Repair): 0
Total Seluruh Unit di Gudang: 20
```

```
==================================================
=== Sistem Pemantauan Inventaris QUANTUM (C) ===
==================================================
1. Tambah Komponen Baru
2. Cari dan Perbarui Komponen
3. Laporan Inventaris Lengkap
4. Keluar
Pilih opsi (1-4): 4

==================================================
Terima kasih. Sistem Pemantauan QUANTUM dimatikan.
==================================================
```

Catatan: Output tidak harus sama yang penting cara kodingannya benar

📋 Ketentuan Pengerjaan

1. Dikerjakan secara individu
2. Dilarang menyalin hasil pekerjaan peserta lain
3. Disarankan tidak menggunakan bantuan AI agar menjadi latihan
4. Study Case ini menjadi syarat pengisian exit ticket sesi 4

📦 Format Pengumpulan

1. Kumpulkan dalam bentuk : zip - (berisi file .c dan screenshot outputnya)
2. Format penamaan : nama_kelasLnt_sesi keberapa

   Contoh: Darren Gavriel Suntara_C_Sesi 3