

Tugas Besar :

Sistem Manajemen Inventaris Gudang

Sistem Manajemen Inventaris Gudang (CLI) adalah aplikasi berbasis antarmuka baris perintah yang dirancang untuk mensimulasikan pengelolaan data inventaris pada skala gudang kecil. Program ini mengimplementasikan operasi dasar CRUD (Create, Read, Update, Delete) terhadap data barang, dilengkapi validasi logika bisnis dan mekanisme umpan balik pengguna.

Tujuan dan Fitur

Tujuan program untuk mensimulasikan sistem manajemen inventaris gudang skala kecil dalam antarmuka baris perintah (CLI).

Pengguna dapat:

- ✓ Menambahkan barang baru (nama, kode unik, kategori, stok awal),
- ✓ Melihat daftar seluruh barang (dengan opsi pencarian berdasarkan nama/kode/kategori),
- ✓ Memperbarui stok (masuk/keluar),
- ✓ Menghapus barang (dengan konfirmasi),
- ✓ Menampilkan laporan ringkasan: total barang, total stok, dan daftar barang stok kritis ($\text{stok} \leq 5$).

Program menyimpan data dalam memori (array/list of struct/record) selama sesi berjalan — tanpa file/database

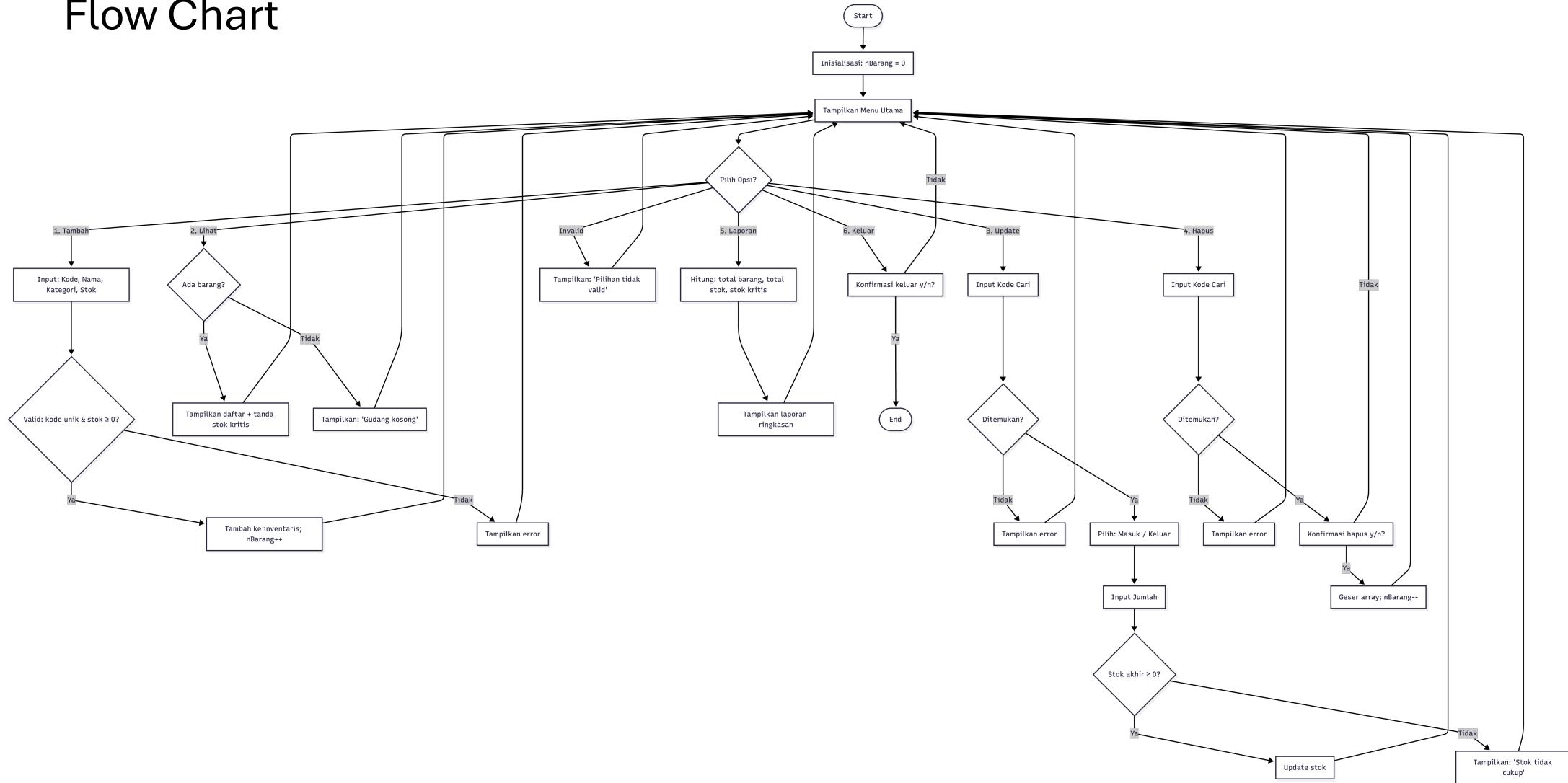
Semua operasi dilindungi validasi:

- ✓ Kode barang harus unik,
- ✓ Stok tidak boleh negatif,
- ✓ Input wajib tidak kosong,
- ✓ Konfirmasi sebelum hapus/perubahan

Struktur Alur Program

```
1 [Start]
2 |
3 |
4 [Tampilkan menu utama:
5   1. Tambah Barang
6   2. Lihat Daftar Barang
7   3. Update Stok (Masuk/Keluar)
8   4. Hapus Barang
9   5. Laporan Ringkasan
10  6. Keluar]
11 |
12 |
13 [Pilih opsi]
14 |
15  |- 1 → [Input & validasi data barang]
16    → Cek kode unik → Simpan ke list → Kembali ke menu
17 |
18  |- 2 → [Tampilkan tabel barang]
19    → [Opsional: Filter (nama/kode/kategori)?]
20 |
21  |- 3 → [Cari barang berdasarkan kode]
22    → Jika ditemukan: pilih [Masuk/Stok +] atau [Keluar/Stok -]
23    → Validasi: stok akhir ≥ 0
24 |
25  |- 4 → [Cari barang → Konfirmasi hapus → Hapus dari list]
26 |
27  |- 5 → [Hitung & tampilkan:
28    - Total jenis barang
29    - Total stok keseluruhan
30    - Daftar barang stok ≤ 5]
31 |
32  |- 6 → [Konfirmasi keluar → Simpan? (opsional) → End]
```

Flow Chart



Penjelasan Sub Menu

```

case pilih of
  '1':
    output("Tambah Barang Baru")
    output("Kode barang (unik, contoh: BRG-001): ")
    input(kodeInput)

    ditemukan ~ false
    for i ← 1 to nBarang do
      if inventaris[i].kode = kodeInput then
        ditemukan ~ true
      endif
    endfor

    if ditemukan = true then
      output("Kode barang sudah digunakan. Gagal menambahkan.")
    else
      output("Nama barang: ")
      input(namaInput)
      output("Kategori: ")
      input(kategoriInput)
      output("Stok awal: ")
      input(stokInput)

      if stokInput < 0 then
        output("Stok tidak boleh negatif.")
      else
        nBarang ← nBarang + 1
        inventaris[nBarang].kode ← kodeInput
        inventaris[nBarang].nama ← namaInput
        inventaris[nBarang].kategori ← kategoriInput
        inventaris[nBarang].stok ← stokInput
        output("Barang berhasil ditambahkan!")
      endif
    endif
  endif

```

```

  '2':
  if nBarang = 0 then
    output("Gudang masih kosong.")
  else
    output("Daftar Barang (" + str(nBarang) + " item):")
    output("KODE | NAMA | KATEGORI | STOK")
    output("-----|-----|-----|-----")
    for i ← 1 to nBarang do
      output(inventaris[i].kode + " | " +
             inventaris[i].nama + " | " +
             inventaris[i].kategori + " | " +
             str(inventaris[i].stok) +
             (if inventaris[i].stok ≤ 5 then " <- Stok kritis!" else ""))
    endfor
  endif

```

Case 1: Tambah Barang Baru

Mengimplementasikan prinsip CRUD (Create) dengan validasi integritas data (fail fast).

- Input: Kode unik, nama, kategori, stok awal.
- Validasi Integritas:
- Kode barang harus unik → cek duplikasi di array inventaris.
- Stok awal ≥ 0 → pencegahan nilai negatif.
- Output:
- Sukses: tambah ke inventaris, increment nBarang.
- Gagal: tampilkan pesan error spesifik.

Relasi dengan tujuan program : Memastikan data inventaris akurat dan konsisten sejak awal input

Before

Indeks	Kode	Nama	Stok
1	BRG-101	Baterai AA	50
2	BRG-102	Obeng Set	12

nBarang = 2



After

Indeks	Kode	Nama	Stok
1	BRG-101	Baterai AA	50
2	BRG-102	Obeng Set	12
3	BRG-103	Kabel USB-C	3

nBarang = 3

← Barang baru

Case 2: Lihat Daftar Barang

Operasi Read dalam CRUD, menampilkan data inventaris dengan format terstruktur dan indikator status.

- Kondisi Awal: Jika nBarang = 0 → tampilkan pesan “Gudang masih kosong.”
- Jika ada barang:
- Tampilkan header tabel: Kode | Nama | Kategori | Stok
- Iterasi for $i \leftarrow 1$ to nBarang untuk cetak setiap item.
- Tambahkan tag “<- Stok kritis!” jika stok ≤ 5 .

Relasi Tujuan Program: Memberikan visibilitas real-time atas kondisi

```

'3':
output("Masukkan kode barang: ")
input(kodeInput)

idx = -1
for i = 1 to nBarang do
    if inventaris[i].kode = kodeInput then
        idx = i
    endif
endfor

if idx = -1 then
    output("Barang tidak ditemukan.")
else
    output("Barang ditemukan: " + inventaris[idx].nama +
          " (Stok: " + str(inventaris[idx].stok) + ")")
    output("[1] Stok Masuk (tambah)")
    output("[2] Stok Keluar (kurangi)")
    input(pilih)

    if pilih != '1' and pilih != '2' then
        output("Pilihan tidak valid.")
    else
        output("Jumlah: ")
        input(jumlah)

        if jumlah <= 0 then
            output("Jumlah harus lebih dari nol.")
        else
            if pilih = '1' then
                inventaris[idx].stok = inventaris[idx].stok + jumlah
                output("Stok bertambah --> " + str(inventaris[idx].stok) + " unit.")
            else
                if inventaris[idx].stok - jumlah < 0 then
                    output("Stok tidak cukup! Sisa: " + str(inventaris[idx].stok))
                else
                    inventaris[idx].stok = inventaris[idx].stok - jumlah
                    output("Stok berkurang --> " + str(inventaris[idx].stok) + " unit.")
                endif
            endif
        endif
    endif
endif
endif

```

Case 3: Update Stok (Masuk/Keluar)

Operasi Update dalam CRUD — memodifikasi state data dengan validasi logika bisnis.

- Input: Kode barang → cari indeks.
- Validasi:
- Barang ditemukan? → jika tidak, error.
- Jumlah > 0? → pencegahan input nol/negatif.
- Stok akhir ≥ 0? → pencegahan stok negatif saat keluar.
- Output:
- Stok bertambah/kurang + konfirmasi nilai baru.
- Pesan error spesifik jika gagal.

Relasi Tujuan Program: Menjaga konsistensi data real-time

Barang: BRG-103
 Nama: Kabel USB-C
 Stok: 3

Hasil dari input
 kode barang



Contoh pilih operasi 2 : Stok
 Keluar (kurangi),
 dengan input angka
 pengurang =2

Barang: BRG-103
 Nama: Kabel USB-C
 Stok: 1
 Catatan: Stok ≤ 5 → status kritis

Output hasil operasi
 pengurangan stok

```
'4':
output("Kode barang yang akan dihapus: ")
input(kodeInput)

idx = -1
for i = 1 to nBarang do
    if inventaris[i].kode = kodeInput then
        idx = i
    endif
endfor

if idx = -1 then
    output("Barang tidak ditemukan.")
else
    output("Hapus barang '" + inventaris[idx].nama + "'? (y/n): ")
    input(konfirmasi)

    if konfirmasi = 'y' or konfirmasi = 'Y' then
        for i = idx to nBarang - 1 do
            inventaris[i] = inventaris[i + 1]
        endfor
        nBarang = nBarang - 1
        output("Barang berhasil dihapus.")
    endif
endif
```

Case 4: Hapus Barang

Operasi Delete dalam CRUD — menghapus data dari sistem dengan konfirmasi dan penyesuaian struktur array.

- Input: Kode barang → cari indeks.
- Validasi:
- Barang ditemukan? → jika tidak, tampilkan pesan error.
- Konfirmasi penghapusan (y/n)? → pencegahan hapus tidak sengaja.
- Proses:
- Jika dikonfirmasi, geser elemen setelah indeks ke kiri.
- Kurangi counter nBarang.
- Output: Pesan konfirmasi berhasil dihapus.

Relasi Tujuan Program: Menjaga kebersihan data dan menghindari akumulasi data usang

```
'5':
totalStok = 0
stokKritis = 0
for i = 1 to nBarang do
    totalStok += inventaris[i].stok
    if inventaris[i].stok <= 5 then
        stokKritis += stokKritis + 1
    endif
endfor

output("LAPORAN RINGKASAN:")
output("- Total jenis barang : " + str(nBarang))
output("- Total stok : " + str(totalStok))
output("- Barang stok kritis : " + str(stokKritis))
```

Case 5: Laporan Ringkasan

Operasi Read lanjutan, menghitung dan menyajikan metrik agregat dari seluruh data inventaris.

- Proses:
- Iterasi semua barang → hitung total stok.
- Hitung jumlah barang dengan stok ≤ 5.
- Output:
- Total jenis barang.
- Total stok keseluruhan.
- Jumlah barang stok kritis.

Relasi Tujuan Program: Memberikan gambaran makro atas kondisi Gudang,

```
'6':
output("Keluar dari program? (y/n): ")
input(konfirmasi)
if konfirmasi = 'y' or konfirmasi = 'Y' then
    output("Terima kasih! Sampai jumpa di gudang berikutnya.")
    break
endif
```

Case 6: Keluar dari Program

Operasi terminal sistem.

- Input: Konfirmasi keluar (y/n).
- Validasi:
- Hanya menerima 'y' atau 'Y' untuk keluar.
- Input lainnya → kembali ke menu utama.
- Output:
- Jika ya: pesan penutup + hentikan loop.
- Jika tidak: lanjutkan program.

```

Program SistemManajemenInventarisGudang

Kamus
const MAX_BARANG = 100
type Barang : record
    kode      : string
    nama     : string
    kategori : string
    stok      : integer
endrecord

inventaris : array [1..MAX_BARANG] of Barang
nBarang   : integer
pilih     : character
konfirmasi : character
kodeInput, namaInput, kategoriInput : string
stokInput, jumlah : integer
i, idx    : integer
ditemukan : boolean
totalStok, stokKritis : integer

Algoritma
nBarang ← 0

output("====")
output("  SISTEM MANAJEMEN INVENTARIS GUDANG")
output("====")

repeat
    output("[1] Tambah Barang")
    output("[2] Lihat Daftar Barang")
    output("[3] Update Stok (Masuk/Keluar)")
    output("[4] Hapus Barang")
    output("[5] Laporan Ringkasan")
    output("[6] Keluar")
    output("Pilihan Anda: ")
    input(pilih)

```

```

    |
    |
    |
otherwise:
    output("Pilihan tidak valid. Silakan coba lagi.")
endcase

output("-----")

```

```
until pilih = '6'
```

```
EndProgram
```

Deklarasi & Loop Utama

Definisi struktur data dan inisialisasi state awal sebelum masuk ke menu interaktif.

- Deklarasi Kamus:
- Konstanta MAX_BARANG → batas kapasitas inventaris.
- Tipe Barang → record dengan field: kode, nama, kategori, stok.
- Array inventaris → tempat penyimpanan data.
- Variabel kontrol: nBarang, pilih, konfirmasi, dsb.
- Inisialisasi:
- nBarang ← 0 → gudang kosong saat start.
- Start Loop:
- repeat ... until pilih = '6' → loop utama program hingga pengguna memilih keluar.

```
nBarang = 0
```

Output:

```
=====
| SISTEM MANAJEMEN INVENTARIS GUDANG
=====
```

Menu:

```
[1] Tambah Barang
[2] Lihat Daftar Barang
[3] Update Stok (Masuk/Keluar)
[4] Hapus Barang
[5] Laporan Ringkasan
[6] Keluar
Pilihan Anda: _
```

Ekspektasi tampilan menu CLI awal

Lari pelan pakai kebaya
Padi dituai menggunakan kampak
cukup sekian presentasi saya
semoga sesuai ekpektasi bapak