

# PENGENDALIAN ATAS DATA MASUKAN GUNA MENINGKATKAN KEAKURATAN INFORMASI

Djajasukma Tjahjadi

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI  
Jl. Ir. H. Juanda 96 Bandung 40132

---

## Abstrak

Sistem informasi bertujuan untuk menghasilkan informasi yang berguna bagi pengambilan keputusan. Agar dapat mengambil keputusan yang tepat, maka pengambil keputusan harus didukung dengan informasi yang andal (benar, akurat, lengkap, relevan, dan tepat waktu). Pada umumnya informasi dihasilkan melalui pengolahan data masukan. Makalah ini mengulas pengendalian atas data masukan (input control) dalam rangka meningkatkan kendalan informasi. Pada bagian akhir dicontohkan pula implementasi dari berbagai teknik pengendalian atas data masukan menggunakan bahasa pemrograman Visual Foxpro.

**Kata-kata kunci:** *input controls, accounting information system, validation, foxpro*

---

## 1. PENDAHULUAN

Dewasa ini sistem informasi memegang peranan penting dalam kegiatan pengelolaan perusahaan. Hal ini didorong dengan semakin pesatnya perkembangan teknologi informasi. Beberapa contoh misalnya sistem informasi perbankan berbasis web (internet banking), sistem penjualan elektronik (electronic sales system) di mana para pelanggan dapat melakukan transaksi dari tempat mereka masing-masing melalui jaringan komputer menunjukkan semakin berkembangnya dan pentingnya sistem informasi.

Tujuan dasar sistem informasi adalah menghasilkan berbagai informasi yang berguna bagi proses pengambilan keputusan, berdasarkan data masukan yang diterima. Misalnya data penjualan di sebuah toko serba ada (supermarket) yang diinput melalui alat pembaca barcode akan digunakan untuk proses update-stok, maupun berbagai macam format laporan penjualan, misalnya laporan penjualan per barang, per kategori, maupun per satuan waktu (hari, minggu, bulan, atau tahun).

Keakuratan data yang dihasilkan dalam laporan sangat bergantung pada proses pengolahan data, dan juga kualitas data masukan. Data masukan yang tidak akurat akan menghasilkan informasi yang salah, seperti jargon 'GIGO' (*Garbage In Garbage Out*).

Oleh karena itu makalah ini akan mengulas berbagai teknik pengendalian atas data masukan yang semuanya bertujuan untuk meningkatkan keakuratan data masukan dan pada akhirnya keakuratan informasi yang dihasilkan.

## 2. PEMBAHASAN

Pada bagian ini akan diuraikan pengendalian atas data masukan, sebagai bagian dari pengendalian pada sistem informasi berbasis komputer

### A. Pengendalian pada sistem informasi berbasis komputer

Pengendalian pada sistem informasi berbasis komputer terdiri atas 2 pengendalian utama, yaitu:

1. Pengendalian umum (*general controls*), yaitu pengendalian yang ditujukan untuk mengendalikan aktivitas pencatatan, pemrosesan, dan pelaporan data secara umum, dan berlaku untuk semua program aplikasi yang digunakan oleh suatu perusahaan.

Pengendalian umum ini meliputi antara lain:

- a. Pendokumentasian yang baik meliputi dokumentasi operasional (*standard operation procedure*), dokumentasi sistem, dan dokumentasi untuk kegiatan administrasi.
  - b. Perlindungan fisik atas komputer
  - c. Pengamanan atas data dan program
  - d. Sistem dan proses *backup*
  - e. Pengendalian yang dimiliki oleh peralatan, misalnya parity checking, duplicate circuitry, dual reading,
  - f. Pengendalian atas akses file/ data
  - g. *Computer security planning*
2. Pengendalian khusus/ aplikasi (*application controls*), yaitu pengendalian yang digunakan untuk program aplikasi tertentu. Pengendalian aplikasi umumnya memiliki tiga bentuk yaitu:
    - a. Pengendalian atas data masukan (*input control*), yaitu pengendalian yang ditujukan untuk menemukan, mencegah data masukan yang salah.
    - b. Pengendalian atas proses pengolahan data (*process control*), yaitu pengendalian yang ditujukan untuk menghasilkan proses pengolahan data yang benar

- c. Pengendalian atas data keluaran (*output control*), yaitu pengendalian yang ditujukan agar data keluaran yang dihasilkan bebas dari kesalahan, dan telah digunakan oleh pihak-pihak yang berwenang.

### B. Pengendalian atas Data Masukan

Ada berbagai macam teknik pengendalian yang dapat digunakan, agar data masukan bebas dari kesalahan. Teknik sering juga disebut sebagai *input validation routines*. Beberapa teknik yang sering digunakan ditampilkan dalam tabel berikut ini:

Teknik	Keterangan	Contoh
Completeness check	memaksa agar field-field tertentu harus diisi.	Tiap field pada data karyawan baru harus diisi.
Field format check	memeriksa apakah pe-ngisian data pada suatu field telah sesuai formatnya.	Tiap karakter yang diketikkan pada kode supplier haruslah numeric
Field length check	memeriksa apakah panjang data suatu field telah benar	Nrp mahasiswa haruslah selalu 7 digit
Field sign check	memeriksa kesesuaian tanda +/- pada data numeric	Nilai penjualan yang tertera pada faktur selalu positif
Limit check	memeriksa apakah nilai suatu data numeric berada dalam range yang benar	Jumlah kerja karyawan dalam satu minggu maksimum 60 jam
Reasonableness check	memeriksa kewajaran nilai suatu field dengan cara membandingkannya dengan nilai field lain.	Overhead cost suatu job tidak dapat melebihi 200% dari nilai labor costnya.
Valid code check	Memeriksa apakah suatu nilai terdaftar pada daftar yang sah	Kode vendor yang diinput pada transaksi pemesanan barang haruslah yang terdaftar pada vendor file.
Check digit	Memeriksa keakuratan suatu data, dengan melakukan perhitungan tertentu	program komputer menghitung kembali angka nomor rekening nasabah berds. rumus tertentu.
Combination field check	nilai suatu field dibandingkan dengan field lain untuk memeriksa keabsahan data	Kode transaksi dibandingkan dengan kode dept, karena transaksi tertentu hanya dapat dilakukan oleh dept. tertentu
Internal Label check	proses pembacaan internal label, untuk menentukan karakteristik suatu file	kode suatu file disimpan sebagai internal label. Program penggajian akan membaca internal label untuk memastikan bahwa file yang dibaca adalah file gaji.
Sequence check	data suatu field diperiksa urutannya	urutan nomor invoice pada file penjualan diperiksa

		sebelum diproses.
Record count check	penghitungan total record dalam suatu file untuk dibandingkan dengan input control	penghitungan record pada file jam kerja dibandingkan dengan kartu jam kerja
Hash total check	penjumlahan dari suatu field untuk dibandingkan dengan batch control total. Totalnya sendiri tidak mempunyai arti selain untuk tujuan pencocokkan.,	Penjumlahan nrp mahasiswa yang telah diinput dengan total nrp pada dokumen sumbernya
Financial total check	Penjumlahan nilai suatu field untuk dibandingkan dengan batch total.	Penjumlahan jumlah penjualan yang telah diinput dengan total jumlah barang yang tertera di faktur

**Check digit.** Teknik ini umumnya digunakan untuk memeriksa keakuratan data numerik. Dengan teknik ini suatu digit tambahan akan ditambahkan pada suatu data. Digit tambahan ini umumnya diletakkan pada digit terakhir. Contoh penggunaan check digit dengan metode Modulus 11 diilustrasikan sebagai berikut:

Account Number	1	2	4	0
Kalikan angka tiap digit dengan letak digitnya	x	x	x	x
	5	4	3	2
Jumlahkan digit demi digit angka hasil perkalian:	$5+8+(1+2)+0 = 16$			
Kurangi hasilnya dari kelipatan 11 yang terdekat:	$22-16 = 6$			
Jadi check digitnya = 6				
Account Number yang lengkap adalah	: 12406			

Cara ini umum digunakan karena memberikan keandalan yang tinggi. Tiap kali user mengetikkan account number 12406, maka komputer akan menghitung check digitnya, jika ternyata check digit yang dihitung tidak menghasilkan angka 6, maka kemungkinan besar terjadi salah penyetikan account numbernya.

### **C. Beberapa Contoh Implementasi Pengendalian atas Data Masukan Menggunakan Visual Foxpro**

Pada bagian ini akan dicontohkan beberapa teknik pemrograman menggunakan Visual Foxpro yang berguna sebagai pengendalian atas data masukan

#### **C.1 Completeness Check**

Validasi ini ditujukan agar field-field yang penting harus selalu diisi. Data tidak dapat disimpan, jika validasi ini belum lolos. Misalnya dalam Gambar 1 dicontohkan operator yang belum mengisi nama pelanggan, sehingga diberikan peringatan



Gambar 1 Completeness Check

### Teknik Pemrograman:

Untuk menerapkan teknik ini, digunakan fungsi `empty()` untuk memeriksa apakah field kosong/ tidak, sebagai berikut:

```
IF EMPTY(thisform.txtkd_cust.Value)
    MESSAGEBOX("Kode customer belum diisi",48,"Perhatian")
    RETURN
ENDIF
IF EMPTY(thisform.txtnama.Value)
    MESSAGEBOX("Nama belum diisi",48,"Perhatian")
    RETURN
ENDIF
```

### C.2 Field Format Check & Sequence Check

Adakalanya field yang diinput dengan pola/ format tertentu. Misalnya pada Gambar 1, saldo customer harus diisi dengan angka (bukan huruf), dan field lainnya (kode, nama, alamat, kota, telepon haruslah diisi karakter/ teks), maka teknik yang paling mudah adalah dengan menggunakan default value = nol untuk data numeric, dan default value="" untuk data karakter. Default value ini ditempatkan pada event forminit. Sehingga setiap kali form customer dipanggil, maka setiap field yang diberi format yang sesuai. Setiap kali terjadi penambahan customer, maka program otomatis mengusulkan nomor urut customer berikutnya (sequence check), namun user dapat mengubah jika ingin menggunakan kode customer yang lain.

**Teknik Pemrograman:**

```

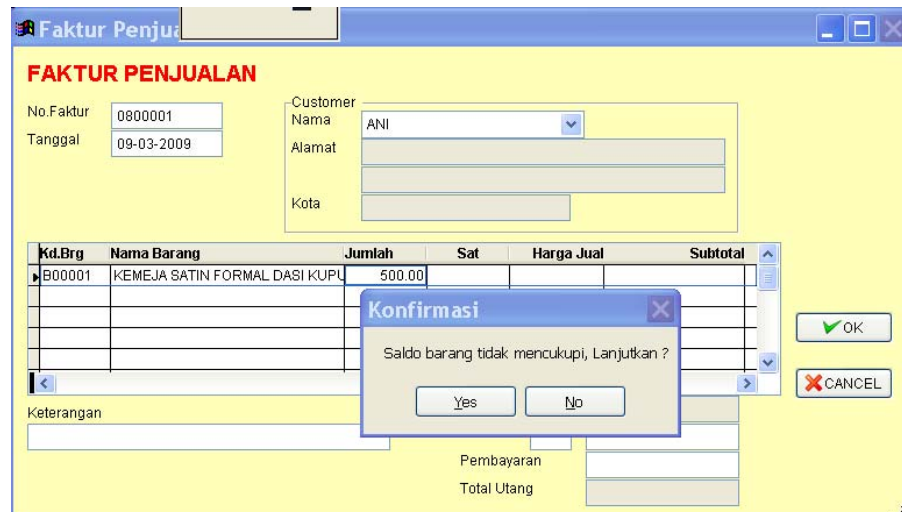
GO bottom
mno=VAL(RIGHT(customer.kd_cust,4))+1
thisform.txtkd_cust.Value="C"+REPLICATE("0",4-LEN(LTRIM(STR(mno))))+;
  LTRIM(STR(mno))
thisform.txtkd_cust.Enabled=.t.
thisform.txtnama.Value=""
thisform.txtalamat1.Value=""
thisform.txtalamat2.Value=""
thisform.txtkota.value=""
thisform.txttelepon.Value=""
thisform.txtsaldo.value=0
thisform.txtsaldo.Enabled=.t.

```

Tiga baris pertama pada program di atas digunakan untuk mencari nomor customer berikutnya secara berurutan. Teknik ini sangat berguna juga untuk mencari nomor dokumen secara berurutan (misalnya nomor faktur penjualan berikutnya)

**C.3. Limit Check**

Pengujian ini berguna untuk membatasi suatu angka agar berada pada batas yang ditentukan, misalnya jumlah barang yang dijual tidak dapat melebihi stoknya, seperti dicontohkan pada Gambar 2.



Gambar 2 Limit Check

**Teknik Pemrograman:**

Pengujian ini diterapkan dengan perintah IF sederhana yang ditempatkan pada objek text jumlah barang, seperti dicontohkan sebagai berikut:

```

IF this.Value>barang.saldo2+barang.awal
n=MESSAGEBOX("Saldo barang tidak mencukupi,Lanjutkan ?",;
4,"Konfirmasi")
IF n=6
RETURN .t.
ELSE

```

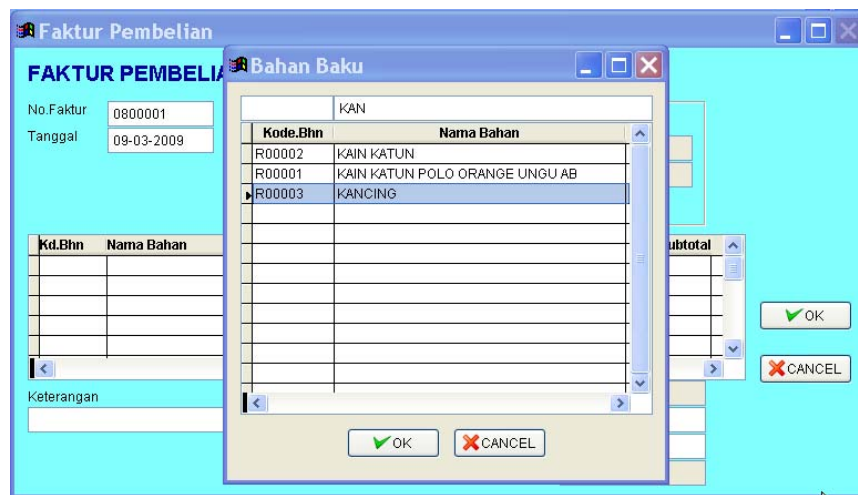
```

RETURN .f.
ENDIF
ENDIF
thisform.refresh

```

#### C.4 Valid Code Check

Pengujian ini merupakan pengujian penting pada setiap program aplikasi, karena dengan pengujian ini, maka integritas database dapat dipertahankan. Misalnya pada saat menginput pelanggan, hanya dapat dipilih pelanggan-pelanggan yang sudah terdaftar. Teknik yang paling umum adalah dengan menggunakan daftar pilihan (combobox) seperti ditunjukkan pada Gambar 2. Namun untuk data yang recordnya banyak (misalnya data barang) sebaiknya selain combobox, juga disediakan form khusus untuk melakukan pencarian berdasarkan nama, seperti ditunjukkan pada Gambar 3



Gambar 3 Valid Code Check

#### Teknik Pemrograman:

Pada dasarnya form pencarian barang adalah sebuah form biasa, dengan obyek textbox untuk menginput data barang yang dicari, dan grid untuk menampilkan data barang yang ada. Agar setiap kali user mengetikkan nama (seperti pada Gambar 3, user mengetikkan huruf "KAN" untuk mencari "KANCING"), program dapat langsung mencari data yang cocok, maka masukkanlah program pencarian ke dalam text pencarian, dengan menggunakan event "interactivechange"

### **3. KESIMPULAN**

Dari studi literatur yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Pengendalian data masukan sangat penting guna menghasilkan informasi yang akurat
2. Perlu adanya kombinasi teknik pengendalian atas data masukan agar validitas data lebih terjamin. Misalnya untuk data nomor rekening nasabah, perlu menggunakan completeness check (field tsb harus diisi), valid code check (nomornya harus terdaftar dalam database nasabah), dan juga check digit sebagai verifikasi
3. Implementasi konsep pengendalian data masukan dapat menggunakan bahasa pemrograman apapun.

### **4. DAFTAR PUSTAKA**

- [1].Bodnar, George H. and Hoopwood, William S., *Accounting Information Systems*, 4<sup>th</sup> ed., Allyn and Bacon, 1990.
- [2].Cushing, Barry E. and Romney, Marshall B., *Accounting Information Systems*, 5<sup>th</sup> ed., Addison-Wesley Publishing Company Inc., 1990.
- [3].Page, J. and Hooper, P., *Accounting and Information Systems*, Prentice-Hall, Inc., 1987