

BAB VII

PENANGANAN KESALAHAN dan HELP-DOCUMENTATION

7.1 KESALAHAN dan SLIP

Error (Kesalahan), tidak ada aplikasi yg berjalan sempurna sebelum melewati berbagai rentetan kesalahan semakin besar aplikasi yang dibuat, semakin banyak kesalahan yang dapat timbul.

Sebuah kompilator akan sering menemui program yang mengandung kesalahan, maka kompilator harus memiliki strategi apa yang harus dilakukan untuk menangani kesalahan - kesalahan tersebut

7.1.1. Jenis-Jenis Kesalahan

a. Kesalahan Leksikal

Misalnya kesalahan mengeja *keyword*,

contoh: then ditulis ten

b. Kesalahan Sintaks

Misalnya pada operasi aritmatika kekurangan jumlah *paranthesis* (kurung).

contoh : A:=X+(B*(C+D)

c. Kesalahan Semantik

- Tipe data yang salah, misal tipe data *integer* digunakan untuk variabel *string*.

Contoh : Var Siswa : Integer

Siswa := 'Aka' {tipe string}

- Variabel belum didefinisikan tetapi digunakan dalam operasi.

Contoh : B := B + 1 {B belum didefinisikan}

7.1.2. Jenis -Jenis Slip

- a. Kesalahan *Capture*
- b. Kesalahan Deskripsi
- c. Kesalahan *Data Driven*
- d. Pengaktifan Asosiatif
- e. Hilangnya Pengaktifan

f. Kesalahan Mode

7.2 PENANGANAN KESALAHAN

a. Prosedur penanganan kesalahan terdiri dari :

- Mendeteksi kesalahan
- Melaporkan kesalahan
- Tindak lanjut perbaikan / pemulihan

b. Pelaporan kesalahan yang dilakukan oleh sebuah kompilator yang menemukan kesalahan meliputi :

- Kode kesalahan
- Pesan kesalahan dalam bahasa natural
- Nama dan atribut *identifier*
- Tipe – tipe yang terkait bila *type checking*
- Contoh : *Error Message : Error 162 jumlah: unknown identifier*
 - Kode kesalahan = 162
 - Pesan kesalahan = *unknown identifier*
 - Nama identifier = *jumlah*

7.3. PETUNJUK PENCEGAHAN KESALAHAN

- a. Menghapus mode-mode atau menyediakan petunjuk yang terlihat untuk mode-mode tersebut.
- b. Gunakan teknik koding yang baik (warna, gaya).
- c. Memaksimalkan pengenalan, mengurangi hafalan.
- d. Merancang urutan gerak atau perintah yang tidak sama.
- e. Mengurangi kebutuhan untuk mengetik.
- f. Uji dan memantau kesalahan-kesalahan dan memperbaikinya.
- g. Memungkinkan pertimbangan ulang aksi-aksi yang dilakukan oleh user, misalnya memindahkan file dari *recycle bin*.

7.3.1. Petunjuk Memperbaiki Kesalahan

- a. Menyediakan tipe-tipe tanggapan yang sesuai.

- b. Query : bertanya pada user apa yang sudah dilakukan, kemudian melegalkan tindakan yang salah.
- c. Menyediakan fungsi “undo” dan pembatalan dari proses yang sedang berjalan.
- d. Meminta konfirmasi untuk perintah yang drastis dan bersifat merusak.
- e. Menyediakan pengecekan yang beralasan pada masukan data.
- f. Mengembalikan kursor ke area kesalahan, memungkinkan untuk melakukan perbaikan.
- g. Menyediakan beberapa kecerdasan buatan.
- h. Menyediakan akses cepat kepada bantuan untuk konteks-sensitif.

7.4. JENIS-JENIS DARI DOKUMENTASI/HELP

Tidak pernah ada perubahan untuk tampilan interaksi yang hamper mendekati sempurna. Sistem sederhana, user memanggil dan menggunakannya, dan user itu sendirikan memberikan nama untuk subsistem yang digunakannya tersebut. Hampir sebagian sistem yang memiliki banyak fitur membutuhkan *help*/bantuan.

7.4.1. Jenis-Jenis Bantuan

- a. Tutorial
- b. Review / referensi yang cepat
- c. Manual referensi (penjelasan lengkap)
- d. Bantuan untuk *context-sensitive* (spesifikasi tugas)

Ada sebagian pendapat menyatakan bahwa sistem yang interaktif dijalankan tanpa membutuhkan bantuan atau training. Hal ini mungkin ideal, akan tetapi jauh dari kenyataan. Pendekatan yang lebih membantu adalah dengan mengasumsikan bahwa user akan membutuhkan bantuan pada suatu waktu dan merancang bantuan (*help*) ke dalam sistem.

7.4.2. Merancang Help dan Dokumentasi

Hal-hal yang perlu diperhatikan dalam merancang *help* dan dokumentasi yaitu :

- a. Perancangan seharusnya tidak seperti “add-on” pada sistem. Secara ideal seharusnya merupakan bagian integral dalam sistem.
- b. Perancangan harus memperhatikan isi dari bantuan dan konteks sebelum teknologi tersedia.

7.4.3. Isu Presentasi dalam Help dan Dokumentasi

- a. Bagaimana help dan dokumentasi dipanggil? Pilihan pertama bagi perancang untuk membuat bagaimana bantuan dapat diakses oleh user. Bantuan ini dapat berupa command, tombol fungsi yang dapat memilih on atau off atau aplikasi yang terpisah.
- b. Bagaimana help dan dokumentasi ditampilkan? Dalam sistem window akan ditampilkan dalam window yang baru. Dalam sistem lain mungkin dalam layar yang penuh atau bagian dari layar. Alternatif lain dapat berbentuk pop-up box atau tingkat command line.
- c. Bagaimana help dan dokumentasi memiliki keefektifan? Tidak menjadi masalah teknologi apa yang digunakan untuk membuat help dan dokumentasi, tetapi yang perlu diperhatikan yaitu prinsip keefektifan.

7.4.4. Masalah yang Ada dalam Implementasi

- a. Para perancang harus membuat keputusan untuk implementasi berupa secara fisik maupun pilihan yang tersedia untuk user. Keputusan ini sudah termasuk dalam pernyataan command operating system, apakah berbentuk meta- command atau aplikasi. Hambatan fisik berupa screen space, kapasitas memori dan kecepatan.
- b. Masalah lain adalah bagaimana struktur data bantuan: apakah berbentuk single file, hierarchy file atau database

Soal dan Pembahasan

Soal :

Sebutkan jenis-jenis kesalahan

Pembahasan :

a. Kesalahan Leksikal

Misalnya kesalahan mengeja *keyword*,

contoh: then ditulis ten

b. Kesalahan Sintaks

Misalnya pada operasi aritmatika kekurangan jumlah *paranthesis* (kurung).

contoh : $A:=X+(B*(C+D)$

c. Kesalahan Semantik

- Tipe data yang salah, misal tipe data *integer* digunakan untuk variabel *string*.

Contoh : Var Siswa : Integer

Siswa := 'Aka' {tipe string}

- Variabel belum didefinisikan tetapi digunakan dalam operasi.

Contoh : B := B + 1 {B belum didefinisikan}