

## **Pertemuan : 3**

Pokok Bahasan dan Tujuan :

### **KRITERIA PEMILIHAN 4GL**

- Agar mahasiswa mengetahui kriteria yang harus digunakan untuk memilih 4GL
- 

### **Kriteria Pemilihan 4GL**

- ❖ Lingkungan aplikasi
- ❖ Kategori fungsi
- ❖ Default
- ❖ Evolutionary Level Syntax
- ❖ User Friendliness
- ❖ Stuktur
- ❖ Spesifikasi & Design
- ❖ Peralatan
- ❖ Perbandingan
- ❖ CheckList : report generator, query Language, graphics, Link, database, dll.

#### **Lingkungan Aplikasi**

- Untuk pemakai akhir/ DP professional.
- Penghitungan tertentu/ ad-hoc.
- Batch System, On-line System, atau perhitungan eksplorasi.
- Kerja Komersial/ kerja pengetahuan.
- Pendukung Keputusan.
- Volume transaksi rendah.
- Jenis Komputer.
- Ukuran basis data.
- Keberadaan basis data/ file on-line, sharing, dsb.
- Spesifikasi Kompleks.
- Prototype/ aplikasi akhir
- Access melalui aplikasi.
- Tersimpan pada paket aplikasi.
- Tersimpan pada Kantor otomatis.

#### **Kategori Fungsi**

- Simple query languages
- Complex query & update languages
- Report generators
- Graphics languages
- Decision support languages
- Application generators
- Specification languages
- Very High Level programming languages
- Parameterized application packages
- Application languages

Perlu dilihat cocok untuk end user, system analyst atau prof. programmers.

## Default

Pekerjaan membuat beberapa aplikasi berkurang oleh penggunaan default intelligent dalam 4GL. Pemakai harus menguji kekuatan default yang ada dalam 4GL.

## Evolusionary Level Syntax

Beberapa gelombang evolusi dalam pengembangan 4GL yang meningkatkan kekuatan dialognya :

- Operasi Off-line
- Dialog satu dimensi dengan terminal seperti mesin ketik.
- Dialog dua dimensi dengan terminal vdu sederhana.
- Dialog yang diorientasikan pada waktu respon desis detik dengan PC, workstation atau jaringan local.
- Dialog menggunakan grafik.
- Peralatan dengan keterkaitan expert-system.

## User Friendliness

Beberapa pertanyaan untuk melihat semua bahasa program user friendly atau tidak.

- Apakah bahasa tersebut cocok untuk end users, system analyst atau programmer.
- Apakah dirancang untuk dapat di install sendiri oleh users.
- Apakah syntax mudah dipelajari/diingat.
- Apakah harus diingat banyak mnemonics.
- Apakah bahasa prosedural, non-prosedural atau keduanya.
- Jika prosedural apakah codenya terstruktur.
- Apakah mudah dalam pembuatannya.
- Apakah terdokumentasi dengan baik(manual).
- Apakah memiliki layar terpisah untuk review data, procedures report error messages.

## Struktur

|                                     | 1                | 2 | 3 |   |
|-------------------------------------|------------------|---|---|---|
| - Avoidance of GO TO                | √                | √ |   |   |
| - Reform (nama subroutine)          | √                | √ | √ |   |
| Single etc. }<br>caution bracket    | - Do             | √ | √ |   |
|                                     | - If             | √ | √ |   |
|                                     | - If-else        | √ |   | √ |
|                                     | - Case           | √ |   | √ |
| Repetition }<br>Bracket             | - Do while       | √ | √ |   |
|                                     | - Do Until       | √ | √ | √ |
|                                     | - For (variable) | √ |   | √ |
| - For ( file/data )                 | √                | √ | √ |   |
| ESCAPE (one bracket )               |                  | √ | √ |   |
| ESCAPE (lebih one bracket)          | √                |   |   |   |
| Clear-End statement for all bracket | √                |   | √ |   |
| Loop initialization & control       |                  | √ |   |   |

1. IDEAL      2.NATURAL      3. MANTIS

Bahasa yang dirancang dengan pengertian dan teknis yang terstruktur sangat penting dalam pembuatan system yang kompleks. Berikut beberapa pertanyaan/feature untuk bahasa terstruktur. (Pada umumnya dihindari instruksi GO TO). Bahasa yang baik harus memiliki features seperti di atas.

### **Spesifikasi & Design**

Beberapa 4GL hanya merupakan bahasa pemrograman. Bahasa yang membantu dalam spesifikasi dan design mempunyai efek yang besar dalam pengembangan life-cycle daripada bahasa yang hanya dapat untuk pemrograman saja.

### **Peralatan**

Peralatan yang diperlukan dan dapat diuji kualitasnya:

- Fasilitas kreasi basis data.
- Bahasa query.
- Pembuatan laporan/grafik
- Bahasa prosedural
- Pointer layar.
- Pembuat dialog
- Peralatan untuk pembuatan dokumentasi.
- Peralatan untuk design system & spesifikasi (secara grafis).
- dsb.

### **Penanganan pada Perbandingan**

Dengan banyaknya bahasa pemrograman user harus selektif dan membandingkan satu dengan yang lain. Perlu pula membandingkan dengan bahasa generasi sebelumnya, seperti:

- Dapatkah bahasa tersebut mengganti COBOL seluruhnya baik fungsi maupun performance mesin.
- Jumlah line of code(bahasa procedural, untuk non-prosedural tidak berarti)
- Waktu pengembangan.

### **CheckList**

Untuk membantu pemilihan dibuat daftar pertanyaan/property, meliputi:

- Report Generator
  - Dapatkah dicetak pada stationary khusus, preprinted media.
  - Keamanannya.
  - Dapat di preview sebelum dicetak.
  - Ada subtotal, control total, dsb.
  - Format, dsb.
- Query Language
  - Apakah intruksinya mudah.
  - Pemakai "naïve" dapat mempelajarinya.
  - Pencarian data dengan criteria tertentu dan rentang tertentu.
  - dll.
- Graphics
  - Type grafik yang didukung: karakter, alpmosaic(vidioteks), vector(susunan baris pada layar), raster/pixel(dots).
  - Graphics berwarna.
  - Pembuatan graphics menggunakan default intelligent.

- Kualitas imaginasi.
  - Tersedianya Chart.
  - Dapatkah user memodifikasi grafik tersebut.
  - dll.
- Link ke tools lain
- Decision
  - Word processing
  - Financial
  - Otomatisasi kantor: mailbox, pengiriman elektronik, kalender management, dsb.
- Database
- Apakah 4GL memakai DBMS.
  - Dapatkah beroperasi dengan DBMS lain.
  - Apakah merupakan DBMS independent/bebas.
- Jika DBMS sendiri, perlu diperhatikan hal-hal sebagai berikut:
- Fleksibilitas
  - Syarat NW/SW
  - Efisiensi
  - Monitory
  - Security & Privacy
  - Restart & Recovery
  - Integrates
  - Performance
  - Portabilitas
- Kriteria Pemilihan 4GL lainnya
- Performance
  - Link ke program generasi keempat
  - Kamus data
  - Perpustakaan
  - Kontrol Integritas
  - Testing Aid
  - Dokumentasi
  - Security/keamanan (scramblede password, password basis data, password file, password record, user, dsb).
  - Auditibilitas
  - Karakteristik System
  - Training
  - Pricing (trial, Leasing, dsb)
  - Vendor

### **Evolusi Bahasa Masa Depan**

Perbedaan evolusi bahasa komputer dipengaruhi oleh:

- Perkembangan mesin(A/W) yang digunakan oleh pengembang aplikasi.
- Peningkatan respon desis detik dari perangkat(mempengaruhi dialog)
- Penggunaan grafik berwarna(CAD-CAM)

- Penekanan bahasa pokok pada bahasa spesifikasi bukan bahasa pemrograman.
- Dokumentasi spesifikasi sistem diorientasikan pada grafik.
- Teknik non-prosedural akan digunakan.
- Penggunaan default yang maksimum untuk meminimalkan pengembangan.
- Deteksi kesalahan dalam pengembangan aplikasi
- Aplikasi berbasis basis data.
- Sistem yang berdasarkan knowledgebase(system pakar), biasanya memerlukan bahasa untuk mengekspresikan aturan dan inference (USP/Prolog). Dan untuk meningkatkan user friendliness dapat digunakan cara:
  - ~ Pemrosesan bahasa manusia.
  - ~ Asistensi dalam operasi S/W yang complex.
  - ~ Asisten dalam penempatan data, pengetahuan, dsb.
  - ~ Diagnosa default, problem dan kesalahan operator.
  - ~ Asisten dalam pembuatan spesifikasi dari program dibuat otomatis.

### **Latihan :**

1. Sebutkan kriteria yang digunakan untuk memilih bahasa pemrogramman generasi keempat !
2. Jelaskan yang dimaksud dengan user friendlynness pada pemilihan pemrogramman 4GL !
3. Jelaskan yang dimaksud dengan checklist pada pemilihan pemrogramman generasi keempat.
4. Bagaimana struktur pemrogramman yang baik?
5. Jelaskan pengembangan aplikasi masa depan berbasis artificial dan knowledge base system.

### **Referensi :**

1. Suryadi H.S, *Pengantar Pemrograman Generasi ke 4*, Gunadarma, Jakarta, 1995