

Pertemuan : 2

Pokok Bahasan dan Tujuan :

Bagaimana 4GL digunakan

- Agar mahasiswa memahami penggunaan 4GL sebagai bahasa yang digunakan dalam pemrosesan data komersial
- Agar mahasiswa memahami penggunaan 4GL sebagai bahasa yang interaktif
- Agar mahasiswa mengetahui siklus pengembangan aplikasi/sistem.

Bagaimana 4GL digunakan

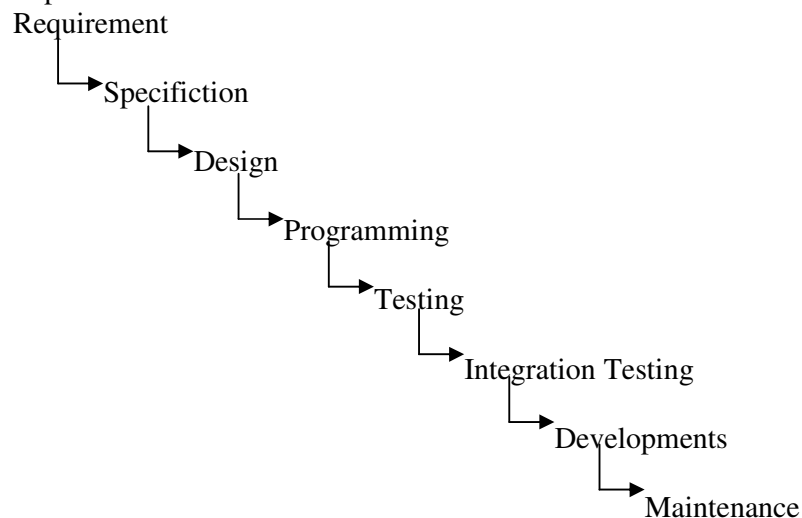
Problem dengan pemrosesan data

- Pemakai tidak dapat memperoleh aplikasi ketika memerlukannya(sering ada delay).
- Sukar/tidak mungkin mendapatkan perubahan manajemen dalam waktu tertentu.
- Terdapat kesalahan program(tidak dapat bekerja).
- System yang digunakan sering tidak memenuhi persyaratan pemakai.
- Sukar untuk memahami DP dan berkomunikasi secara tepat.
- Pemakai harus menandatangani spesifikasi, sulit pengendaliannya.
- Harga dan waktu banyak untuk pengembangan dan menjaga daripada untuk antisipasi dan implementasi.

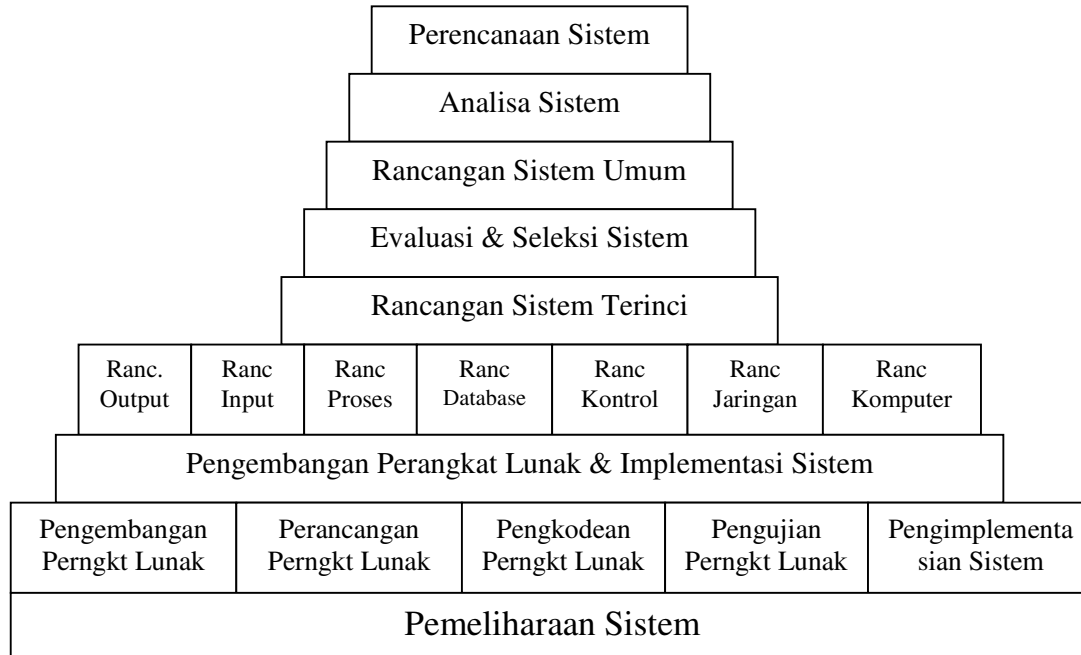
Perubahan Pengembangan Life Cycle

Sejarah life cycle tumbuh sebelum peralatan ini muncul:

- Bahasa non-prosedural.
- Teknik pembuatan code program secara otomatis.
- Bahasa spesifikasi.
- Peralatan grafik on-line untuk design.
- Bahasa untuk end user.
- Konsep Pusat Informasi.



Gb. Tradisional Development life cycle



Gb. System Development Life Cycle

Tipe life cycle

Beberapa tipe life cycle yang ada dalam penggunaannya dengan 4GL:

- System Ad-hoc
Pemakai/ analis akan menyaring, membuat, dan memanipulasi data. Tidak ada persyaratan analisis atau spesifikasi penulisan.
Sistem ini dirawat sepenuhnya oleh pembuat. Tidak ada life cycle formal.
- Prototype Cycle
Prototype dibuat oleh system analyst dan reaksi pemakai pada prototype menyebabkan terjadinya modifikasi.
- Prototype 3GL-Cycle
Diprogram ke 3GL untuk mendapatkan/ efisiensi mesin yang lebih besar.
- Spesifikasi dan 4GL Cycle
Analisa dan spesifikasi dengan life cycle tradisional, tetapi code dibuat dengan code generator.
- Lingkaran Bahasa Spesifikasi
Spesifikasi biasanya tidak konsisiten, ambigius dan tidak lengkap. Pada lingkungan ini Administrator data akan menjaga kamus dan data yang digunakan organisasi.

Pendekatan Fisik

Tidak hanya bahasa diubah, tetapi pendekatan fisik ke mesin juga berubah.

- Komputer Generasi I : Panel dan kabel dipasang.
- Komputer Generasi II : tidak mengatur program pada control.

- Komputer Generasi III : jarang melihat mesin.
Sekarang ini PC terhubung ke mainframe dengan LAN yang memberikan kecepatan transisi lebih cepat.

Gelombang 4GL

Gelombang pengembangan 4GL

1. Bahasa Off-line, digunakan pemakai untuk mengisi form.
ex: MARK IV, DMS.
2. Bahasa yang dapat beroperasi pada terminal seperti mesin ketik.
ex: SQL, NOMAD & DATATRIEVE.
3. Bahasa untuk monitor dumb vdu, pemakai berinteraksi dengan tampilan dua dimensi.
ex: MAPPER, FOCUS, RAMIS & QBE(Query By Example).
4. Bahasa yang dirancang untuk waktu tanggap desis detik dari PC atau LAN.
ex: MULTIPLAN 1 LOTUS, SYMPHONI & FRAMEWORK.
5. Bahasa yang menggunakan grafik untuk design program.
ex: CAD/CAM

Bagan System

Pemrosesan design yang kompleks dimulai dengan “bagan” dan berlanjut ke representasi detail. ex: arsitektur rumah, designer aplikasi komputer.

Properti bagan yang penting dapat dipahami oleh pemakai akhir. Tipe bagan yang digunakan antara lain:

- Prototype
- Diagram warnier
- DFD
- Notasi HOS
- Diagram HIPO
- Aksi Diagram Basis data

Catatan:

- Aksi diagram basis data menunjukkan logika dan relasi pada struktur basis data.
- DFD menunjukkan interaksi diantara pemrosesan yang terpisah.
- HOS menunjukkan penguraian fungsi yang formal.

Intepreter & Compiler

Code bahasa generasi III disebut ”Source Code” yang akan diterjemahkan oleh compiler untuk membentuk “Object Code’

- ✓ Compiler → offline
- ✓ Intepreter → Online(langsung menterjemahkan)

Intepreter kurang efisien dalam penggunaan mesin, namun demikian perbedaannya menjadi lebih kabur daripada 4GL.

Hampir semua 4GL disusun untuk operasi on-line dan dialog memerlukan tempat antara software dan pembuat aplikasi.

Aspek yang menarik dari compiler terpisah adalah prototype dapat dibuat pada PC untuk kemudian di compile pada main/miniframe.

Menentukan kesalahan system mungkin pada pengembangan aplikasi sangat penting. Kesalahan selama pengetesan program lebih mahal (10X) dibandingkan pada tahap aplikasi.

Latihan :

1. Jelaskan penggunaan bahasa pemrograman generasi ke 4 !
2. Jelaskan siklus pengembangan sistem/aplikasi komputer !
3. Jelaskan yang dimaksud dengan bagan system !.
4. Jelaskan yang dimaksud dengan intepreter dan kompiler !

Referensi :

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