

PENDAHULUAN

- Perkembangan bahasa untuk komputasi komersial berjalan lambat.
- Banyak bahasa dan piranti baru untuk menghasilkan peningkatan produktifitas DP, diantaranya:
 - Bahasa Generasi keempat.
 - Bahasa Produktifitas tinggi.
 - Bahasa Non Prosedural.
 - Pembuatan Aplikasi.
- Dengan bahasa generasi keempat, kemampuan komputer dapat diakses bagi semua orang, tanpa melakukan banyak pelatihan.
- Gaya bahasa:
 - Memiliki kemampuan yang handal dan ringkas.
 - Memiliki grafis yang bagus.
 - Cocok untuk pemakai terakhir dengan sedikit latihan.
- Revolusi komputer
Proses Instruksi lebih cepat dan mudah, karena:
 1. Komputer meningkat dalam hal kuantitas dan kecepatannya.
 2. Penyebaran komputer, tidak hanya DP menggunakan komputer.

Aplikasi akan secara meningkat dibuat oleh:

- Pemakai terakhir
- Konsultan bisnis
- System Analyst

System analyst memerlukan bahasa komputer yang handal. Dapat membuat aplikasi sendiri dengan cepat dan konsentrasi utama pada bisnis/aplikasi bukan pada coding.

Generasi Bahasa

I. Bahasa Mesin

II. Bahasa Assembly

III. Bahasa Tingkat Tinggi (Fortran, Cobol)

- Programmer dapat memberikan code program dengan tanpa pengetahuan instruksi mesin dan register.
- Memerlukan sejumlah besar baris code untuk system komersial dan di design untuk DP Profesional.

IV. Bahasa Generasi Keempat (4GL)

Tujuan:

- ❖ untuk mempercepat proses pembuatan aplikasi.
- ❖ untuk mempermudah aplikasi dan mempercepat perubahan → mengurangi harga perawatan.
- ❖ meminimalkan problem debugging.
- ❖ membuat bahasa lebih dikenal.

- 4GL bergantung basis data dan kamus data.
- Bahasa generasi III:
 - bervariasi syntaxnya.
 - konstruksi sama pada umumnya.
- Bahasa generasi IV, agak berbeda konstruksinya.

V. Bahasa Generasi Kelima –AI

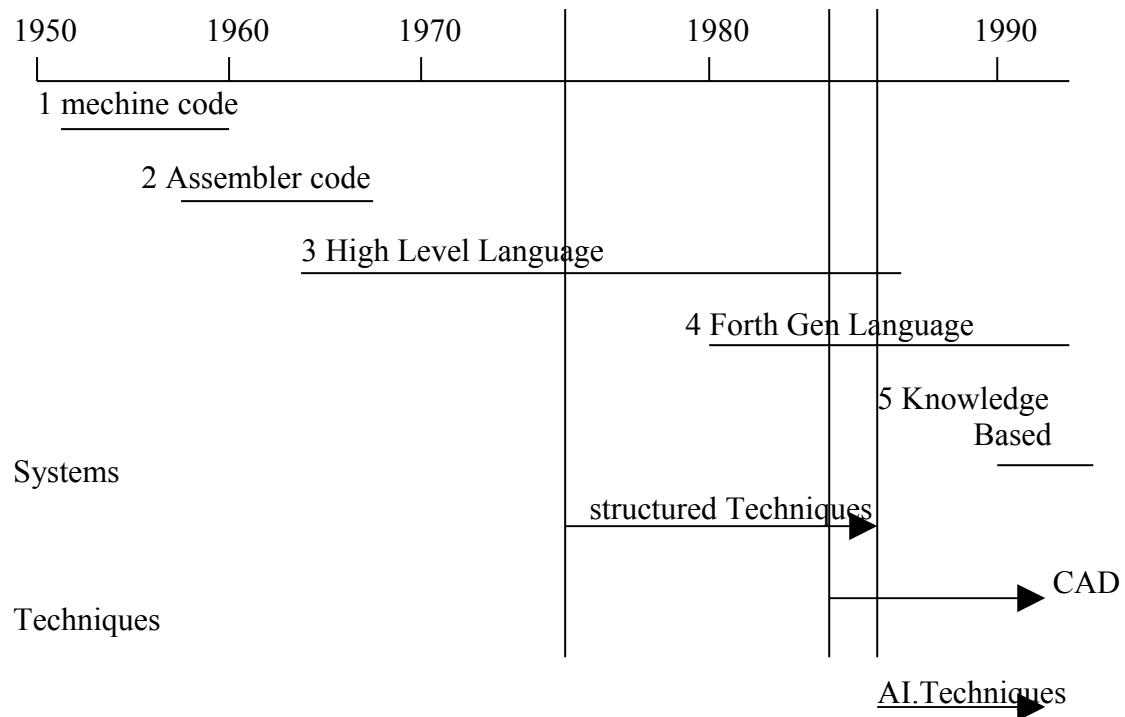
- memberikan code pengetahuan yang kompleks sehingga suatu mesin mungkin dapat menggambarkan kesimpulan.
- mengerjakan processor parallel yang tinggi.
- Bahasa Prosedural vs Bahasa Non Prosedural
 - Prosedural, “bagaiman” suatu diselesaikan.
 - Bahasa Non Prosedural, “Apakah” yang akan diselesaikan.
- Hampir semua bahasa query, paket grafik untuk non prosedural.
- Bahasa 4GL berkekuatan di query dan bahasa pemrograman tingkat tinggi.
- Bahasa generasi III hampir dapat membuat semua aplikasi, 4GL hanya untuk tugas tertentu saja.
- Batasan Pilihan
 - Feature pada setting generasi berbeda.
- Dialog satu dimensional:
 - Komputer saling menukar karakter stream dengan pemakai.
 - Tombol pemakai merupakan perintah dan operasi pada komputer.
- Dialog dua dimensional:
 - Pemakai merespon pada layar.
 - mengambil, memasukkan data/item pada layar.

Prinsip Pokok dalam design 4GL:

- ❖ Prinsip pekerjaan minimal
- ❖ Prinsip keterampilan minimal
- ❖ Prinsip pengindaran syntax dan mechanic asing
- ❖ Prinsip waktu kosong(delay)
- ❖ Prinsip usaha minimal (min error)
- ❖ Prinsip perawatan minimal (customize)
- ❖ Prinsip hasil maksimal

Ciri Bahasa generasi keempat:

- ❖ Tidak perlu training lama pada DP
- ❖ Bahasa untuk semua pemakai
- ❖ Bahasa query
- ❖ Pembuat aplikasi
- ❖ Mencakup bahasa procedural/ non prosedural



Karakter Pokok dari 4GL

Daftar kriteria pokok tersebut antara lain:

- Apakah diperuntukan untuk perhitungan rutin atau keputusan(pengambilan keputusan).
- Apakah diperuntukan untuk pemakai terakhir atau DP professional.
- Apakah memerlukan ketrampilan programmer atau tidak.
- Keistimewaan yang ada:
 1. Simple queries.
 2. Simple queries & Update.
 3. Komplek queries.
 4. Komplek queries & Update.
 5. Kemampuan membuat basis data secara cepat.
 6. Operasi intelligent basis data
 - Perubahan di satu sisi menyebabkan terjadinya operasi lain secara otomatis.
 7. Generasi layar data entry untuk operator key entry(dengan cek validasi).
 8. Generasi layar update.
 9. Bahasa prosedural yang memberikan kemampuan pemograman penuh.
 10. Teknik grafis untuk design aplikasi.
 11. Penyebaran manipulasi.
 12. Manipulasi multi dimensi matrik.
 13. Pembuatan laporan.
 14. Pembuatan grafik.
 15. Manipulasi grafik.
 16. Keputusan dukungan untuk pertanyaan what-if.
 17. Peralatan analisa matematika.

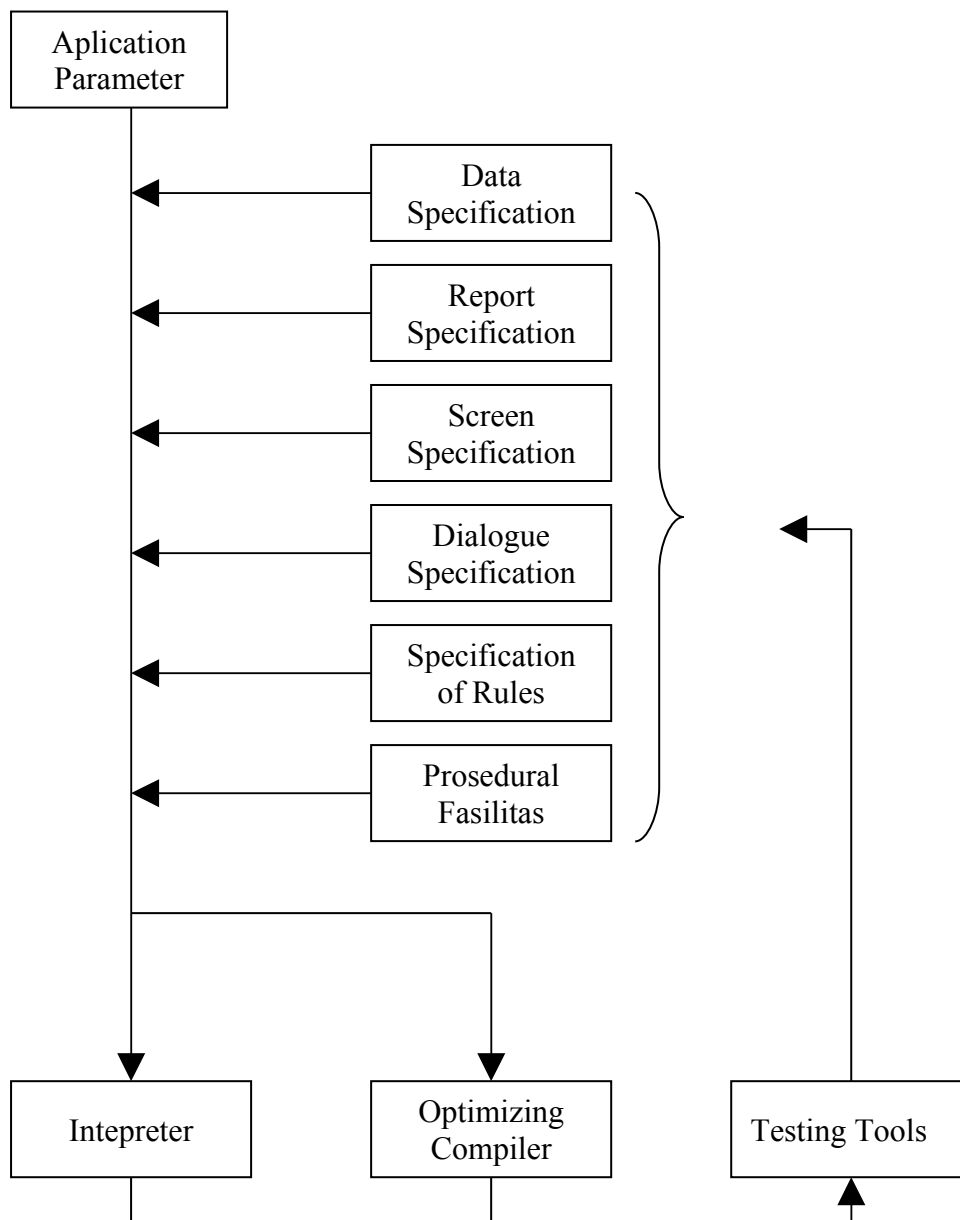
18. Peralatan analisa keuangan.
19. Peralatan pendukung lain
20. Manipulasi teks
21. Mailbox elektronik :
 - Apakah on-line/off-line
 - Apakah dijalankan untuk mainframe, miniframe atau PC.
 - Dapatkah meng-access mainframe atau remote basis data.
 - Apakah cukup mudah digunakan.
 - Dapatkah hasilnya diperoleh dengan cepat.

Properti 4GL:

Untuk semua bahasa dapat disebut generasi keempat harus mempunyai karakteristik sebagai berikut:

1. User-friendly.
2. Programmer non-profesional dapat memperoleh hasil.
3. Menggunakan system manajemen basis data secara langsung.
4. Program untuk semua aplikasidapat dibuat dengan satu perintah lebih sedikit daripada COBOL.
5. Code non-prosedural digunakan, jika mungkin.
6. Akan membuat asumsi default intelligent tentang yang diinginkan pemakai, jika mungkin.
7. Disusun unutm operasi on-line.
8. Memberi kekuatan dan semangat code terstruktur.
9. Mempermudah pemahaman dan perawatan code oranglain.
10. Pemakai non-DP dapat mempelajarinya dengan cepat.
11. Dirancang untuk debugging dengan mudah.
12. Prototype dapat dibuat dan dimodifikasi dengan cepat.
13. Hasilnya dapat diperoleh dengan perintah magnitude waktu yang lebih cepat dari COBOL/PLI.

Tidak semua bahasa yang memiliki property tersebut seluruhnya dan cocok untuk semua aplikasi. Harus dipilih bahasa pemograman tertentu untuk aplikasi tertentu.



Gb. Komponen 4GL untuk membuat aplikasi rutin

Komponen 4GL

4GL dengan tujuan umum yang baik mempunyai komponen non-prosedural yang mungkin disembunyikan pada fasilitas prosedural. Seperti pada gambar berikut:

- Application parameter, meliputi pemberian nama parameter, catalog, versi, pembuat dsb.
- Data specification, penentuan spesifikasi laporan berdasarkan data yang diambil.
- Screen specification, penentuan bentuk tampilan.
- Dialog specification, menentukan interaksi manusia dengan komputer.
- Specification Of Rules, aturan-aturan untuk penentuan keputusan. Biasanya dipisahkan dari badan aplikasi.

Dalam kaitan dengan data, laporan, generator layar, pembuat dialog dan penentu aturan adalah fasilitas prosedural. Hal ini memungkinkan struktur program ditentukan oleh : loops, kondisi, struktur kasus dan runtime yang sarangkan.

Bagaimana 4GL digunakan

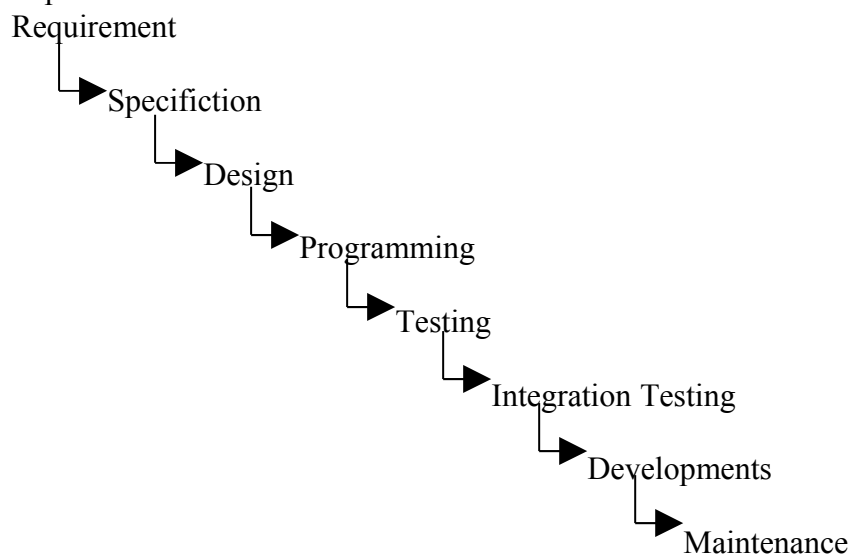
Problem dengan pemrosesan data

- 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.

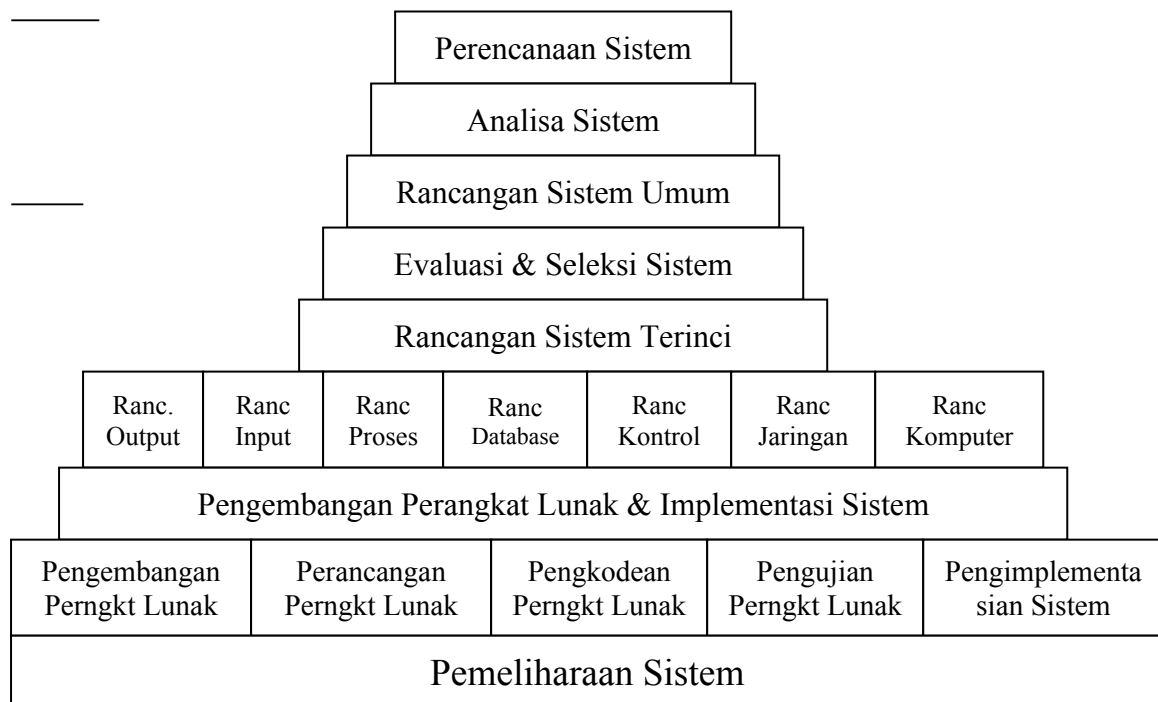
Perubahan Pengembangan Life Cycle

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.

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 cution bracket	- Do	√	√
	- If	√	√
	- If-else	√	√
	- Case	√	√
Repetition 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 imajinasi.
 - 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).

- Auditabilitas
- 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 desin 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 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.