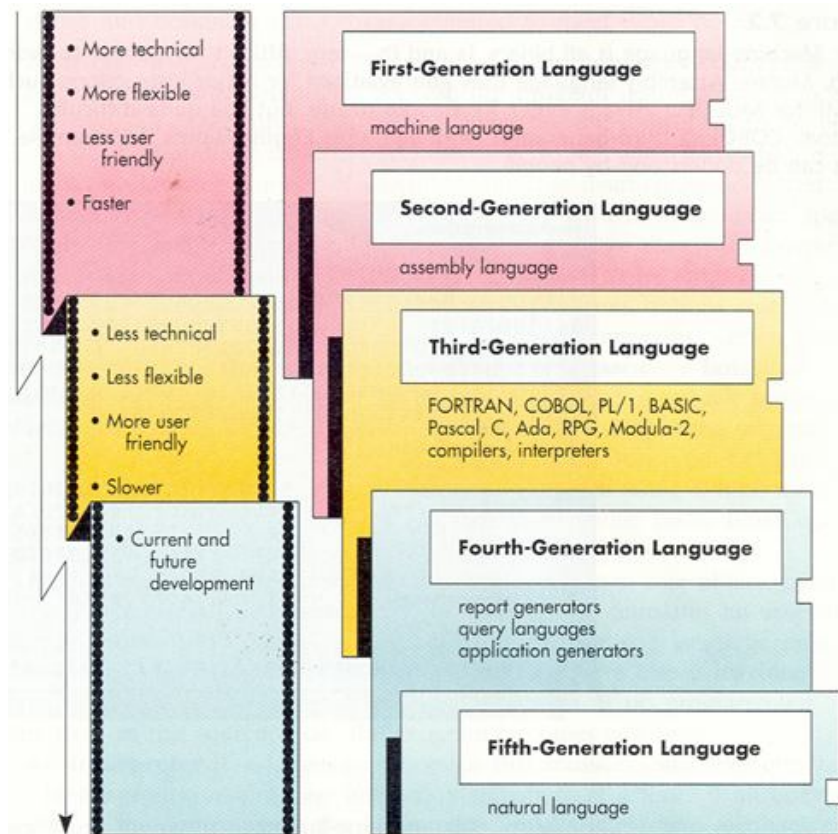


BAB VII

PENGEMBANGAN PERANGKAT LUNAK DAN BAHASA-BAHASA PEMROGRAMAN

Generasi Bahasa Pemrograman

Bahasa pemrograman adalah serangkaian aturan yang memberitahu komputer operasi apa yang akan dilakukan. Ada lima generasi bahasa pemrograman: (1) Machine language, (2) Assembly language, (3) High-level (procedural) language, (4) Very-high-level (Non-procedural) language, dan (5) Natural language.



Gambar-7.1: Lima Generasi Bahasa Pemrograman

Bahasa tingkat lebih rendah adalah bahasa yang lebih dekat dengan bahasa komputer yang menggunakan sistem biner 1 dan 0. Bahasa tingkat lebih tinggi adalah bahasa yang lebih dekat dengan bahasa manusia (pada umumnya adalah English-like). Dari generasi ke generasi, bahasa pemrograman mengalami peningkatan, bahasa generasi terbaru lebih mudah dipelajari dibandingkan dengan generasi sebelumnya. Bahasa

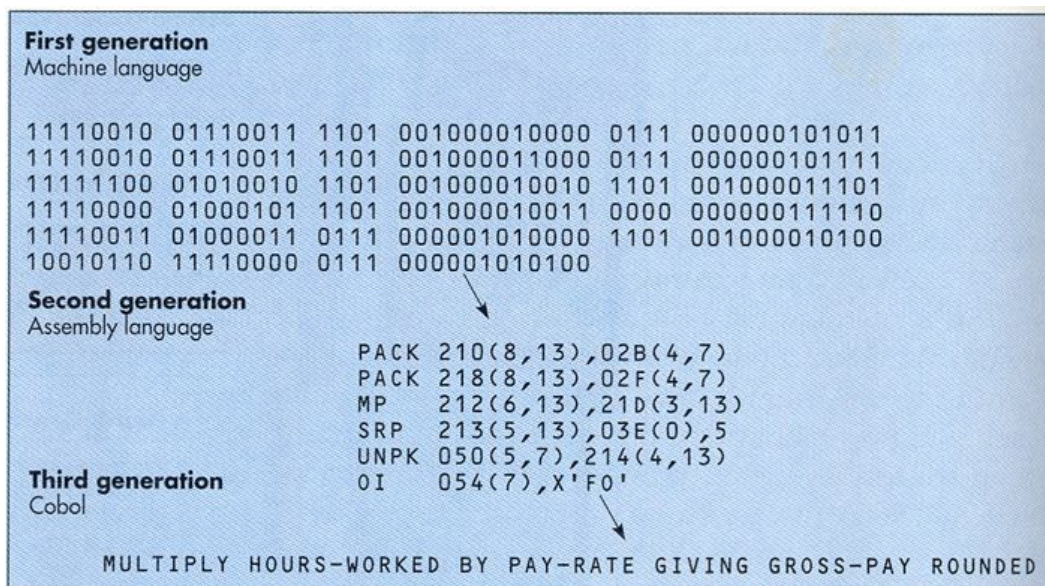
generasi awal menggunakan perbendaharaan kata (vocabulary) dan struktur kalimat (syntax) yang lebih kompleks dibandingkan yang terbaru.

Generasi Pertama: Machine Language

Machine Language (bahasa mesin) adalah bahasa dasar komputer, perintah dan data disajikan dalam bentuk *binary digit* (0 dan 1). Program yang ditulis dalam bahasa mesin tidak perlu diterjemahkan sebelum program tersebut dijalankan pada komputer. Bahasa ini tergantung pada mesin komputer, program hanya bisa dijalankan pada mesin yang programnya dirancang untuk mesin tersebut.

Generasi Kedua: Assembly Language

Bahasa Assembly menggunakan simbol sebagai singkatan untuk perintah, sebagai pengganti kombinasi bit-0 dan bit-1 yang panjang. Programmer bisa dengan mudah mengingat perintah, namun program dalam bahasa Assembly tetap sulit dikonversi untuk digunakan pada mesin komputer lain.



Gambar-7.2: Generasi Bahasa Pemrograman

Generasi Ketiga

Bahasa ini tampil sudah seperti teks sehari-hari dan formula matematik, dapat dijalankan pada berbagai jenis/merek komputer dengan atau tanpa perubahan. Contoh: COBOL (Common Business Oriented Language), FORTRAN (FORMula TRANslator), BASIC (Baginners All-purpose Symbolic Unstuction Code), C, RPG, PL/1, Pascal. Program harus diterjemahkan lebih dahulu oleh Language Translator ke dalam bahasa mesin (dari bentuk *source code* ke bentuk *object code*).

```

IF (XINVO .GT. 500.00) THEN
    DISCNT = 0.07 * XINVO
ELSE
    DISCNT = 0.0
ENDIF
XINVO = XINVO - DISCNT

```

Gambar-7.3: Contoh Pemrograman dalam FORTRAN

```

OPEN-INVOICE-FILE.
  OPEN I-O INVOICE FILE.

READ-INVOICE-PROCESS.
  PERFORM READ-NEXT-REC THROUGH READ-NEXT-REC-EXIT UNTIL END-OF-FILE.
  STOP RUN.

READ-NEXT-REC.
  READ INVOICE-REC
  INVALID KEY
    DISPLAY 'ERROR READING INVOICE FILE'
    MOVE 'Y' TO EOP-FLAG
    GOTO READ-NEXT-REC-EXIT.
  IF INVOICE-AMT > 500
    COMPUTE INVOICE-AMT = INVOICE-AMT - (INVOICE-AMT * .07)
    REWRITE INVOICE-REC.

READ-NEXT-REC-EXIT.
  EXIT.

```

Gambar-7.4: Contoh Pemrograman COBOL

```

10 REM   This Program Calculates a Discount Based on the Invoice Amount
20 REM   If Invoice Amount is Greater Than 500, Discount is 7%
30 REM   Otherwise Discount is 0
40 REM
50 INPUT "What is the Invoice Amount"; INV.AMT
60 IF INV.AMT 500 THEN LET DISCOUNT = .07 ELSE LET DISCOUNT = 0
70 REM   Display results
80 PRINT "Original Amt", "Discount", "Amt after Discount"
90 PRINT INV.AMT, INV.AMT * DISCOUNT, INV.AMT - INV.AMT * DISCOUNT
100 END

```

Gambar-7.5: Contoh Pemrograman BASIC

Generasi Keempat

Generasi keempat dikenal dengan bahasa berorientasi-masalah atau 4GLs, merupakan bahasa non-prosedural. Programmer atau pengguna dapat membuat spesifikasi apa yang diinginkan untuk dilakukan oleh komputer tanpa menguraikan bagaimana komputer melakukannya. Karena kemudahannya, yang bukan profesional komputer pun bisa mengembangkan perangkat lunak aplikasi tertentu. Query language: pengguna dapat bertanya tentang, atau menemu-balik informasi dari, berkas database

dengan menyusun pertanyaan dalam bahasa Inggris. Contoh: SQL (sekuel, Structured Query Language), QBE (Query-By-Example), dan Intellect.

Generasi Kelima

Disebut juga dengan bahasa natural mirip dengan bahasa *query*, dengan satu perbedaan dimana pengguna atau programmer tidak perlu mempelajari *vocabulary*, *grammar*, atau *syntax* tertentu.

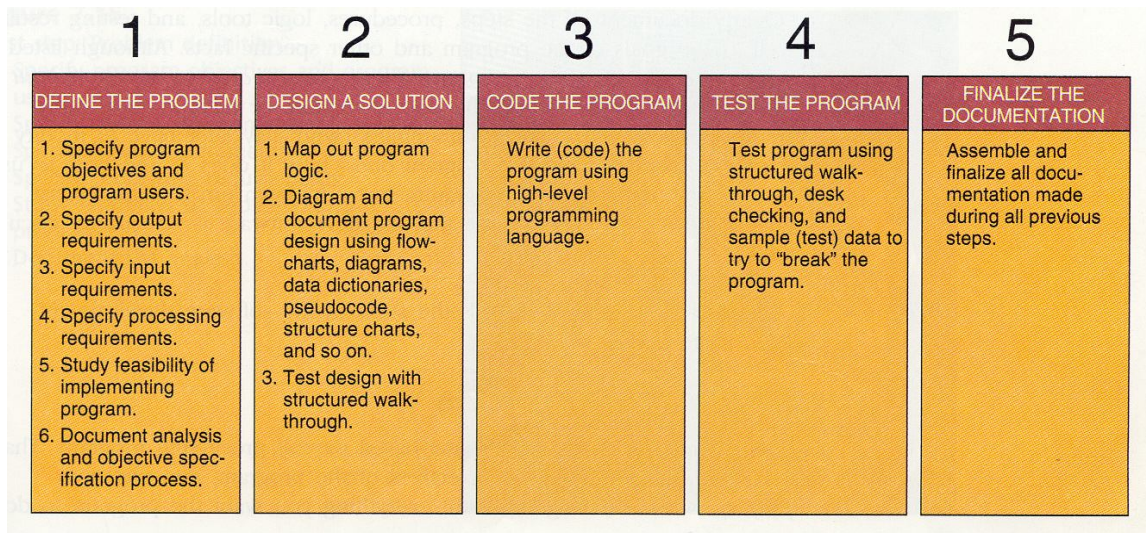
Pemrograman Berorientasi Obyek dan Visual

Pemrograman berorientasi obyek (OOP) adalah metode pemrograman yang mengkombinasikan data dan perintah-perintah untuk pemrosesan data cukup dengan kode pemrograman obyek atau *block preassemble*, yang dapat digunakan oleh program-program lain. Ada tiga konsep dari OOP yaitu *encapsulation*, *inheritance*, dan *polymorphism*. Beberapa contoh pemrograman berorientasi obyek: Smalltalk, C++, Turbo Pascal, dan Hypertalk.

Pemrograman visual memungkinkan *programmer* membuat koneksi antara obyek pemrograman dengan gambar, *pointing*, dan *clicking* pada diagram dan *icon*. Tujuan pemrograman visual adalah *programmer* dengan mudah membuat program dan lebih mudah digunakan oleh *non-programmer*. Beberapa contoh pemrograman visual adalah ObjectVision (dari Borland), Visual BASIC (Microsoft)

Pemrograman: Lima Langkah Produksi

Pemrograman dilakukan secara berurut dalam lima langkah proses: (1) *define the problem*, (2) *map out program logic and design a solution*, (3) *code the program*, (4) *test the program*, dan (5) *collate the documentation*.



Gambar-7.6: Lima Langkah Produksi Pemrograman