BUKU AJAR

PEMROGRAMAN BAHASA RAKITAN

oleh :

Adian Fatchur Rochim, ST, MT

Program Studi Sistem Komputer
Fakultas Teknik
Universitas Diponegoro
2009

KATA PENGANTAR

Puji syukur dipanjatkan kehadirat Allah SWT karena hanya atas izin-Nya, penyusunan buku ajar **Pemrograman Bahasa Rakitan**, dapat diselesaikan dengan baik.

Maksud disusunnya buku ajar ini untuk membantu mahasiswa didalam mengikuti perkuliahan di Program Studi Sistem Komputer Fakultas Teknik Universitas Diponegoro. Diharapkan dengan telah disusunnya buku ajar ini, perkuliahan akan dapat dilaksanakan dengan lebih terarah dan sistematis.

Secara umum materi Bahan-bahan Listrik ini meliputi :

1. Pengantar Bahasa Rakitan
2. Dasar – Dasar Mikroprosesor
3. Konversi Bilangan
4. Sintaks-sintaks instruksi Bahasa Rakitan
5. Interupsi – Interupsi pada Bahasa Rakitan

Tiada gading yang tak retak, penyusunan buku ajar inipun tentu masih terdapat kekurangan disana-sini, saran dan kritik yang membangun sangat diharapkan. Penyempurnaan akan dilakukan dengan menambah materi dari beberapa referensi buku, jurnal maupun artikel hasil penelitian.

Semoga buku ajar ini bermanfaat.

Semarang, Juli 2009

Penyusun

DAFTAR ISI

B A B I

PENGANTAR BAHASA RAKITAN DAN KONVERSI BILANGAN HEX, DEC DAN BINER

Tujuan Instruksional

Pada Bab ini dijelaskan kedudukan dan posisi bahasa rakitan dibandingkan dengan bahasa aras sangat rendah (bahasa mesin) dan bahasa tingkat menengah (C/C++) dan bahasa tingkat tinggi (Pascal dsb). Juga dijelaskan dalam bab ini mengenai representasi kode kode dalam bilangan biner, desimal dan hexadesimal. Yang diharapkan dalam bab selanjutnya akan lebih mudah mengikuti pembelajaran mengenai instruksi bahasa rakitan, mengingat kesemua data menggunakan pola pola konversi bilangan yang wajib bagi mahasiswa untuk menguasai terlebih dahulu bab ini.

1. Pendahuluan

Bahasa Rakitan termasuk ke dalam bahasa tingkat rendah dan merupakan bahasa dasar komputer. Bahasa ini memerlukan logika yang cukup rumit di samping instruksinya yang jauh berbeda dengan bahasa pemrograman lainnya. Program yang dihasilkan memiliki kecepatan
yang paling baik. Kelebihan dari bahasa rakitan adalah :

1. Memiliki fasilitas fungsi dan makro (ciri khas bahasa pemrograman yang menyebabkan pemrograman menjadi lebih mudah).
2. Program dapat dibuat secara modular (dipecah dalam modul-modul kecil dan dapat diintegrasikan kembali).
3. Ukuran program lebih kecil, sehingga lebih menghemat media penyimpan.
4. Lebih dekat ke hardware sehingga seluruh kemampuan komputer dapat dimanfaatkan secara maksimal.

Bahasa *rakitan* merupakan bahasa pemrograman yang posisinya di antara bahasa pemrograman lainnya adalah termasuk dalam bahasa pemrograman tingkat rendah karena bahasa ini berhubungan langsung dengan bahasa mesin. Sedangkan bahasa pemrograman Delphi berada di atas bahasa pemrograman *rakitan*, yang sering disebut OOP (*Object Orinted Programming*).

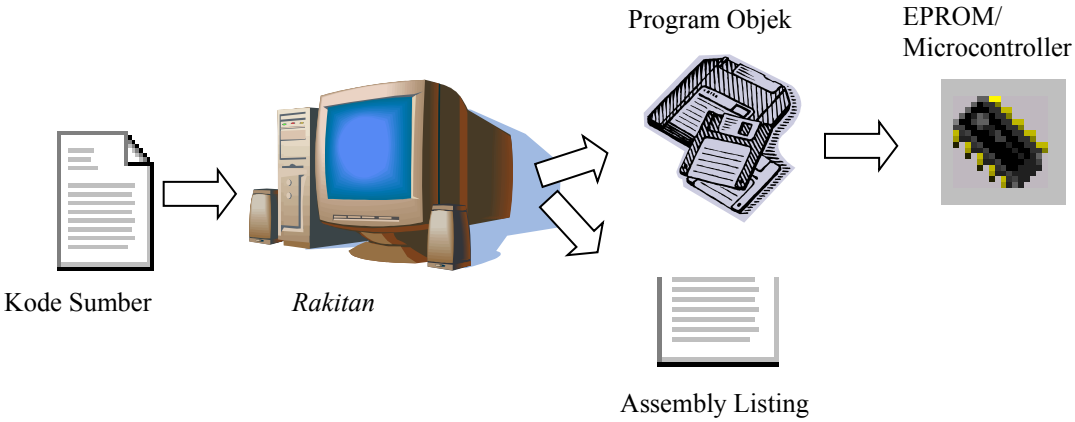
Bahasa mesin adalah kumpulan kode biner yang merupakan instruksi yang bisa dijalankan oleh komputer. Sedangkan bahasa *rakitan* memakai kode *mnemonic* untuk menggantikan kode biner, agar lebih mudah diingat sehingga memudahkan penulisan program.

Program yang ditulis dengan bahasa *rakitan* terdiri dari label; kode *mnemonic* dan lainnya, pada umumnya dinamakan sebagai program sumber (*source code*) yang belum bisa diterima oleh prosesor untuk dijalankan sebagai program tapi harus diterjemahkan terlebih dahulu menjadi bahasa mesin dalam bentuk kode biner.

Jika yang ditulis hanya bahasa *rakitan* saja maka biasanya program dibuat dengan program editor biasa, misalnya note pad pada Windows atau *sidekick* pada DOS, selanjutnya program sumber diterjemahkan ke bahasa mesin dengan menggunakan program *rakitan*. Hasil kerja program *rakitan* adalah "program objek" dan juga "*rakitan listing*". Tapi karena di sini bahasa *rakitan* ditulis bersama dengan bahasa Delphi maka program dibuat di dalam editor milik Delphi.

Program objek berisikan kode – kode bahasa mesin, kode – kode bahasa mesin inilah yang diumpankan ke memori – memori prosesor.

Perlu diperhatikan bahwa setiap prosesor mempunyai konstruksi yang berbeda – beda, instruksi untuk mengendalikan masing – masing prosesor juga berbeda – beda. Dengan demikian bahasa *rakitan* untuk masing – masing prosesor juga berbeda, yang sama hanyalah pola dasar cara penulisan program *rakitan* saja. Adapun bagan kerja proses *rakitan* dapat dilihat pada Gambar 2.2 :



Gambar 2.2 Bagan kerja proses *rakitan*

Dalam bahasa *rakitan* program sumbernya menganut prinsip 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian, yaitu bagian label, bagian *mnemonic*, dan bagian operan yang bisa lebih dari satu.

Label mewakili nomor memori program dari instruksi pada baris yang bersangkutan, misal pada saat menulis JUMP, label ini ditulis pada bagian operand untuk menyatakan nomor memori program yang dituju. Dengan demikian label selalu mewakili nomor memori program dan harus ditulis di bagian awal baris instruksi. Selain label dikenal pula *symbol*, yakni satu nama yang mewakili satu nilai tertentu dan nilai yang diwakili bisa apa saja tidak harus nomor memori program. Cara penulisan simbol sama dengan penulisan label, harus dimulai di huruf pertama dari baris instruksi.

Mnemonic merupakan singkatan perintah, dikenal dua macam *mnemonic*, yakni *mnemonic* yang dipakai sebagai instruksi mengendalikan prosesor, misalnya ADD, MOV, DJNZ dan lainnya. Ada pula *mnemonic* yang dipakai untuk mengatur kerja dari program *rakitan* misalnya ORG, EQU atau DB, *mnemonic* untuk mengatur kerja dari program *rakitan* ini dinamakan sebagai "*rakitan directive*".

Operan adalah bagian yang letaknya di belakang bagian *mnemonic*, merupakan pelengkap bagi *mnemonic*. Kalau sebuah instruksi diibaratkan sebagai kalimat perintah, maka *mnemonic* merupakan subjek (kata kerja) dan operan merupakan objek (kata benda) dari kalimat perintah tersebut.

2. Konversi Bilangan Hexadesimal, Biner dan Desimal sebagai modal awal belajar Bahasa Pemrograman Rakitan

Salah satu penyebab kesulitan mahasiswa untuk mempelajari bahasa Rakitan adalah hal – hal yang menyangkut konversi Bilangan biner dan hexadecimal. Meskipun dianggap sesuatu hal yang mengada – ada oleh sebagian anda bahwa : untuk apa bilangan bilangan direpresentasikan dengan orde yang bermacam-macam? Yang malah menyulitkan bagi pengguna bilangan. Sebetulnya tidak demikian. Coba anda bayangkan bagaimana merepresentasikan Bilangan desimal 4 (yang kita pakai sehari hari) dalam digital? Seperti yang anda ketahui dalam dunia digital representasi adalah 1 dan 0, karena secara fakta lebih mudah dan tegas perbedaan antara 1 dan 0. 1 adalah on dan 0 adalah off.

Disebut sebagai biner (1 dan 0) sehingga dari ungkapan saya diatas dapat dipahami mengapa biner dibutuhkan, sehingga desimal 4 adalah 100 (biner) kalau representasi pada nyala 3 buah led maka nyala padam dan padam. Kemudian muncul pertanyaan untuk apa bilangan hexadesimal? Coba bayangkan representasi bilangan desimal 65536 dalam biner..? maka anda dapat menulis sepanjang 16 bilangan biner. Hal ini sangat menyulitkan kala sudah masuk dalam pemrograman, sehingga muncul bilangan orde 16 atau hexadesimal yang berfungsi untuk menyederhanakan penulisan (mengecilkan ukuran representasi bilangan desimal dan biner) menjadi FF (hexadesimal) hanya membutuhkan dua karakter bilangan saja. Oleh sebab itulah mengapa biner dan hexadesimal dibutuhkan dalam representasi bilangan.

Oleh karena itu pemahaman konversi bilangan penting sebab penggunaan mereka menyederhanakan lain topik kompleks termasuk desain logika dengan aljabar boolean, untuk meyederhanakan representasi data.

Bab ini mendiskusikan beberapa konsep penting mencakup konversi biner dan hexadecimal, sistem, organisasi data biner (bytes), sistem nomorsign dan unsign, perhitungan, logis, pergeseran, dan operasi rotasi pada biner, bit dan representasi data, dan karakter ASCII. Bab ini wajib anda kuasai, sebelum anda menguasai sistem bilangan dan konversinya sebaiknya tidak melangkah ke bab berikutnya.

BILANGAN DESIMAL

Kita terbiasa dengan bilangan basis 10 (desimal) coba anda lihat angka ini 123 representasi dalam bilangan basis 10 adalah seperti berikut :

$$1*10^2 + 2*10^1 + 3*10^0 \text{ atau sama dengan}$$

$$100 + 20 + 3$$

kemudian coba perhatikan bilangan ini 123,456

direpresentasikan dalam desimal sebagai berikut

$$1*10^2 + 2*10^1 + 3*10^0 + 4*10.^{-1} + 5*10^{-2} + 6*10^{-3} \text{ atau sama dengan}$$

$$100 + 20 + 3 + 0,4 + 0,05 + 0,006$$

BILANGAN BINER

Sistem komputer menggunakan bilangan biner, dioperasikan menggunakan logika biner. Komputer merepresentasikan nilai-nilai menggunakan dua tingkatan voltase yang pada umumnya 0v dan + 5v. Dengan dua nilai seperti kita dapat merepresentasikan persisnya dua nilai-nilai berbeda. Ini bisa karena dimanapun dua nilai yang berbeda secara absolut, Dalam penggunaanya kita menggunakan nilai-nilai nol dan satu.

Contoh disini sangat mudah untuk mengkonversikan bilangan biner ke desimal. Anda perhatikan bilangan biner ini 11001010_2 direpresentasikan dengan cara seperti pada desimal diatas menjadi sebagai berikut :

$1*2^7 + 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$ sama dengan

$128 + 64 + 8 + 2$

202_{10}

Mudah kan?

Sedangkan untuk mengkonversi desimal ke biner agak sedikit lebih sulit, karena anda harus menemukan terlebih dahulu pangkat dua tertinggi berapakah yang sama dengan atau lebih kecil dari bilangan desimal yang akan dikonversi. Contoh anda perhatikan bilangan berikut : 1359 (desimal) coba kita konversikan..

- cari nilai pangkat dua yang sama dengan atau lebih kecil dari 1359, kita ambil misalnya $2^{10} = 1024$ -> lebih kecil dari 1359, kemudian
- kita naikkan satu tingkat $2^{11} = 2048$ ternyata 2048 lebih besar dari 1359, sehingga kita pilih 2 pangkat 10 (2^{10}).
- Kurangkan desimal yang akan dikonversi dengan nilai desimal dari pangkat dua yang lebih kecil dari bilangan tersebut sehingga, 1359-1024 adalah sama dengan 335
- Cari bilangan pangkat dua yang lebih kecil atau sama dengan 335 -> $2^8 = 256$.
- Kurangkan 335 dengan 256, 335 – 256 = 79
- Ulangi langkah keempat kita dapatkan pangkat dua yang mendekati adalah 64 shg 79-64 = 5
- Ulangi lagi langkah sebelumnya kita dapat 4, mjd 5-4 = 1

- Dari hasil langkah langkah tersebut diatas dilengkapi sampai nilai nomila terkecil kemudian diurutkan dan kemudian kita susun binernya menjadi seperti berikut ini
- 2^{10} 2^8 2^6 2^4 2^3 2^2 2^1 dan 2^0 selanjutnya kita urutkan mulai bilangan pangkat dua terbesar sampai yang terkecil, pangkat dua yang terepresentsi spt hitungan kita tadi kita urutkan dengan 1 yang terwakili dan 0 tidak terwakili menjadi urutan bilangan biner seperti ini : 1010101111

BILANGAN HEXADESIMAL

Bilangan Hexadesimal merupakan bilangan dengan orde 16, Dengan urutan sebagai berikut (paling kiri merupakan MSB dan paling kanan adalah LSB) :

F E D C B A 9 8 7 6 5 4 3 2 1 0. dimana bilangan F adalah 16 dalam orde 10 dan 1111 dalam orde biner. Konversi bilangan dari bilangan biner ke hexadesimal sangat mudah, bila dibandingkan konversi dari bilangan hexadesimal ke desimal. Cara mengkonversi bilangan hexadesimal ke biner adalah dengan cara menguraikan bilangan bilangan hexadesimal tersebut empat bit empat bit sesuai dengan konversi dalam biner. Contoh bilangan FE_{16} dalam konversi biner nya adalah F menjadi 1111_2 dan E menjadi 0001_2 sehingga $FE_{16} = 11110001_2$.

Cara mengkonversi ke bilangan desimal sama dengan biner Contoh:

3A hexa = (3 * 161) + (10 * 160)

= 48 + 10

= 58 desimal