Pertemuan 1

Waktu : 135 menit
Tujuan Pembelajaran : Mahasiswa mampu menjelaskan konsep algoritma dan struktur data
Substansi Materi : Konsep Algoritma dan Struktur Data

Tabulasi Kegiatan Perkuliahan

No	Tahap Kegiatan	Kegiatan Pengajar	Kegiatan Mahasiswa	Media & Alat	Waktu
1	Pendahuluan	1. Membuka Pertemuan 2. Menjelaskan peta kompetensi dan tujuan pembelajaran 3. Memberikan rancangan pembelajaran 4. Memberikan daftar referensi kuliah	Menyimak Bertanya	Papan Tulis	20 Menit
2	Penyajian Materi	1. Definisi Algoritma 2. Definisi Struktur Data 3. Konsep Algoritma dan Struktur Data 4. Contoh penerapan Algoritma dan Struktur Data	Menyimak Bertanya Menjawab Pertanyaan	Papan Tulis	80 Menit
3	Penutup	1. Menyimpulkan materi pertemuan 2. Memberikan tugas kecil 3. Menutup pertemuan	Menyimak	Papan tulis	35 Menit

MATERI KULIAH

Definisi Algoritma

Ditinjau dari asal usul katanya kata Algoritma sendiri mempunyai sejarah yang aneh. Orang hanya menemukan kata Algorism yang berarti proses menghitung dengan angka arab. Anda dikatakan Algorist jika anda menghitung menggunakan Angka Arab. Para ahli bahasa berusaha menemukan asal kata ini namun hasilnya kurang memuaskan. Akhirnya para ahli sejarah matematika menemukan asal kata tersebut yang berasal dari nama seorang ahli matematika dari Uzbekistan Abu Abdullah Muhammad Ibnu Musa Al-Khuwarizmi (770-840). Al-Khuwarizmi dibaca orang barat menjadi Algorism. Al-Khuwarizmi menulis buku yang berjudul Kitab Al Jabar Wal-Muqabala yang artinya "Buku pemugaran dan

pengurangan" (The book of restoration and reduction). Dari judul buku itu kita juga memperoleh akar kata "Aljabar" (Algebra). Perubahan kata dari Algorism menjadi Algorithm muncul karena kata Algorism sering dikelirukan dengan Arithmetic, sehingga akhiran –sm berubah menjadi –thm. Karena perhitungan dengan angka Arab sudah menjadi hal yang biasa. Maka lambat laun kata Algorithm berangsur-angsur dipakai sebagai metode perhitungan (komputasi) secara
umum, sehingga kehilangan makna kata aslinya. Dalam Bahasa Indonesia, kata Algorithm diserap menjadi Algoritma.

Kita bisa mendefinisikan algoritma sebagai berikut:
" Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan."

Dan kamus besar bahasa Indonesia (Balai Pustaka 1988) secara formal mendefinisikan algoritma sebagai berikut:
"Algoritma adalah urutan logis pengambilan putusan untuk pemecahan masalah."

Hubungan Algoritma dan Struktur Data

Program adalah kumpulan instruksi komputer, sedangkan metode dan tahapan sistematis dalam program adalah algoritma. Program ini ditulis dengan menggunakan bahasa pemrograman. Jadi bisa kita sebut bahwa program adalah suatu implementasi bahasa pemrograman. Beberapa pakar memberi formula bahwa:

program = struktur data + algoritma

Bagaimanapun juga struktur data dan algoritma berhubungan sangat erat pada sebuah program. Algoritma yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, semikian juga sebaliknya. Menilai Sebuah Algoritma ketika manusia berusaha memecahkan masalah, metode atau teknik yang digunakan untuk memecahkan masalah kemungkinan bisa lebih dari satu. Dan kita memilih mana yang terbaik diantara teknik-teknik itu. Hal ini sama juga dengan algoritma, yang

memungkinkan suatu permasalahan dipecahkan dengan metode dan logika yang berlainan. Lalu bagaimana mengukur mana algoritma yang terbaik ? Beberapa persyaratan untuk menjadi algoritma yang baik adalah:

- Tingkat kepercayaannya tinggi (realibility). Hasil yang diperoleh dari proses harus berakurasi tinggi dan benar.
- Pemrosesan yang efisien (low cost). Proses harus diselesaikan secepat mungkin dan jumlah kalkulasi yang sependek mungkin.
- Bersifat general. Bukan sesuatu yang hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.
- Bisa dikembangkan (expandable). Haruslah sesuatu yang dapat kita kembangkan lebih jauh berdasarkan perubahan requirement yang ada.
- Mudah dimengerti. Siapapun yang melihat, dia akan bisa memahami algoritma anda. Sulit dimengertinya suatu program akan membuat sulit pengelolaan.
- Portabilitas yang tinggi (portability). Bisa dengan mudah diimplementasikan di berbagai platform komputer.

Contoh Algoritma dan Implementasinya

Sebagai contoh sederhana, mari kita berlatih melihat permasalahan, mencoba menyusun algoritma, dan menerapkan dalam bahasa Pascal.

Permasalahan :

Bagaimana mengkonversi nilai yang diraih mahasiswa peserta kuliah Algoritma dan Struktur Data II menjadi Huruf Mutu (A / B / C / D / E) ?

Untuk memecahkan masalah tersebut maka dapat disusun algoritma sebagai berikut :

1. Buat satu variable (misalnya Nilai)
2. Bandingkan nilai dengan rentang huruf mutu dengan menggunakan selection
3. Tampilkan huruf mutu

Lalu jika diimplementasikan dengan menggunakan program Pascal, maka dibawah ini adalah listing programnya :

```
{ Program Huruf Mutu }
Uses crt;
Var n : integer;
Begin
    Clrscr;
    Write('Nilai Anda : ');
    Repeat
        GoToXY(13,1); ClrEol;
        Read(n);
    Until (n>=0) and (n<=100);
    Case n of
        85 . . 100 : write(' Anda mendapat Huruf Mutu A ');
        75 . . 84  : write(' Anda mendapat Huruf Mutu B ');
        65 . . 74  : write(' Anda mendapat Huruf Mutu C ');
        Else write(' Anda mendapat Huruf Mutu D ');
    End;
    ReadKey;
End.
```

Output :

```
Nilai Anda : 90
Anda mendapat Huruf Mutu A
```

Algoritma Tidak Tergantung Bahasa Pemrograman Dan Mesin Komputer

Notasi Algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman. Analoginya sama dengan resep membuat kue. Sebuah resep dapat ditulis dalam bahasa apapun. Bahasa Jepang, Inggris, Perancis, Indonesia, dan lain sebagainya. Apapun bahasanya, kue yang dihasilkan tetap sama asalkan semua aturan pada resep diikuti.

Mengapa demikian? Karena setiap juru masak (sebagai pemroses) dapat melakukan operasi dasar yang sama, seperti mengocok telur, menimbang berat gula, dan lain sebagainya. Demikian juga halnya dengan komputer. Meskipun setiap komputer berbeda teknologinya, tetapi secara umum semua komputer dapat melakukan operasi-operasi dasar dalam pemrograman seperti operasi pembacaan data, operasi perbandingan, operasi aritmatika, dan sebagainya. Perkembangan teknologi komputer tidak mengubah operasi-operasi dasar itu, yang berubah hanyalah kecepatan, biaya, atau tingkat ketelitian. Pada sisi lain setiap program dalam bahasa tingkat tinggi selalu diterjemahkan kedalam bahasa mesin sebelum akhirnya dikerjakan oleh CPU. Setiap instruksi dalam bahasa mesin menyajikan operasi dasar yang sesuai, dan menghasilkan efek yang sama pada setiap komputer.