

ARRAY PROCESSING

Pertemuan 6

Objectives

- To introduce arrays and the uses of arrays
- To develop pseudocode algorithms for common operations on arrays
- To illustrate the manipulation of single- and two-dimensional arrays

Outline

- 6.1 Array processing
- 6.2 Initialising the elements of an array
- 6.3 Searching an array
- 6.4 Writing out the contents of an array
- 6.5 Programming examples using arrays
- 6.6 Two-dimensional arrays

Array processing

Array adalah salah satu alat pemrograman paling kuat yang tersedia. Mereka menyediakan programmer dengan cara mengatur kumpulan item data homogen (yaitu, item yang memiliki tipe yang sama dan panjang yang sama) ke dalam struktur data tunggal. Array, kemudian, adalah struktur data yang terdiri dari sejumlah variabel yang semuanya memiliki tipe data yang sama, misalnya semua nilai ujian untuk kelas yang terdiri dari 30 siswa matematika. Dengan menggunakan array, satu nama variabel seperti 'skor' dapat dikaitkan dengan semua 30 nilai ujian.

Item data individu yang membentuk array disebut sebagai elemen larik. Elemen dalam array dibedakan satu sama lain dengan menggunakan indeks atau subskrip, diapit dalam tanda kurung, mengikuti nama array, misalnya 'skor (3)'.

Subskrip menunjukkan posisi elemen dalam array; skor (3) mengacu pada skor ujian ketiga, atau elemen ketiga dari skor array, dan skor (23) mengacu pada skor ujian ke-23. Subskrip atau indeks dapat berupa angka atau variabel, dan kemudian dapat digunakan untuk mengakses item apa pun dalam batas larik yang valid, misalnya:

scores (6), or
scores (index)

Array adalah struktur data internal; mereka diperlukan hanya selama durasi program di mana mereka didefinisikan. Mereka adalah mekanisme yang sangat nyaman untuk menyimpan dan memanipulasi kumpulan item data serupa dalam suatu program, dan Anda harus terbiasa dengan operasi yang paling umum dilakukan pada mereka. Array kadang-kadang disebut sebagai tabel.

Operations on array

Operasi yang paling umum dilakukan pada array adalah:

- memuat satu set nilai awal ke dalam elemen array
- memproses elemen array
- mencari array, menggunakan pencarian linear atau biner, untuk tertentu sebuah elemen
- menulis isi array ke laporan.

Biasanya, elemen array diproses secara berurutan, dimulai dengan elemen pertama. Ini dapat dicapai dengan mudah dalam pseudocode baik dengan loop DO atau loop DOWHILE.

Simple algorithms that manipulate arrays

Algoritma berikut melibatkan manipulasi sederhana dari array. Setiap algoritma ditulis menggunakan loop DO. Dalam setiap algoritma, array diberi nama 'array', subscript diberi nama 'indeks' dan isi array serta jumlah elemen telah ditetapkan. Jumlah elemen dalam array disimpan dalam variabel number_of_elements. Contoh

```
Find_sum_of_elements
  Set sum to zero
  DO index = 1 to number_of_elements
    sum = sum + array (index)
  ENDDO
  Print sum
END
```

Initialising the elements of an array

Dalam contoh ini, elemen array dicari untuk menentukan elemen mana yang terbesar. Algoritme dimulai dengan menempatkan elemen pertama dari array ke dalam variabel terbesar_elemen, dan kemudian melihat elemen lain dari array untuk melihat apakah ada nilai yang lebih besar. Nilai terbesar kemudian dicetak. Karena array adalah struktur data internal, nilai awal harus ditempatkan ke dalam array sebelum informasi apa pun dapat diambil darinya. Nilai awal ini dapat ditetapkan ke elemen array sebagai konstanta, atau dapat dibaca ke dalam array dari file.

Loading constant values into an array

Metode ini hanya boleh digunakan bila data dalam larik tidak mungkin diubah – misalnya, nama 12 bulan dalam setahun. Untuk menginisialisasi array seperti itu, buatlah sebuah array yang disebut month_table, yang berisi 12 elemen dengan ukuran yang sama. Kemudian tetapkan elemen array dengan nama bulan, satu per satu, sebagai berikut:

```
Initialise_month_table
month_table(1) = 'January'
month_table(2) = 'February'
:::
month_table(12) = 'December'
END
```

Perhatikan bahwa setiap elemen larik harus berukuran elemen terbesar – dalam hal ini September – jadi, nama bulan yang lebih pendek harus diisi dengan kosong (spasi).

Loading initial values into an array from an input file

Mendefinisikan elemen array sebagai konstanta dalam program tidak disarankan jika nilainya sering berubah, karena program perlu diubah setiap kali elemen array berubah. Prosedur umum adalah membaca nilai input ke dalam elemen array dari file input.

Pembacaan serangkaian nilai dari file ke dalam array dapat diwakili oleh loop DOWHILE sederhana. Loop harus berakhir ketika array penuh atau file input telah mencapai akhir file. Kedua kondisi ini dapat dipenuhi dalam klausa kondisi dari loop DOWHILE.

Dalam algoritma pseudocode berikut, nilai dibaca dari file input dan ditetapkan ke elemen array, dimulai dengan elemen pertama, hingga tidak ada lagi nilai input atau array penuh. Nama array adalah 'array', subscript adalah 'index', dan jumlah maksimum elemen yang dapat ditampung oleh array adalah max_num_elements.

```
Read_values_into_array
  Set max_num_elements to required value
  Set index to zero
  Read first input value
  DOWHILE (input values exist) AND (index < max_num_elements)
    index = index + 1
    array (index) = input value
    Read next input value
  ENDDO
  IF (input values exist) AND index = max_num_elements THEN
    Print 'Array size too small'
  ENDIF
END
```

Perhatikan bahwa pemrosesan akan berhenti ketika file input telah mencapai EOF atau array penuh. Pesan kesalahan akan dicetak jika ada lebih banyak item data input daripada elemen dalam array.

Searching an array

Operasi umum pada array adalah mencari elemen array untuk item data tertentu. Alasan untuk mencari array mungkin:

- untuk mengedit nilai input – yaitu, untuk memeriksa apakah itu adalah elemen yang valid dari sebuah array
- untuk mengambil informasi dari array
- untuk mengambil informasi dari elemen terkait dalam larik berpasangan.

Saat mencari array, merupakan keuntungan jika array diurutkan ke dalam urutan menaik, sehingga, ketika kecocokan ditemukan, sisa array tidak perlu dicari. Jika Anda menemukan elemen larik yang sama dengan entri input, kecocokan telah ditemukan dan pencarian dapat dihentikan. Juga, jika Anda menemukan elemen larik yang lebih besar dari entri input, tidak ada kecocokan yang ditemukan dan pencarian dapat dihentikan. Perhatikan bahwa jika entri array yang lebih besar dicari lebih sering daripada entri yang lebih kecil, mungkin ada keuntungan untuk mengurutkan array ke dalam urutan menurun. Sebuah array dapat dicari baik menggunakan pencarian linier atau pencarian biner.

A linear search of an array

Pencarian linier melibatkan melihat setiap elemen array, satu per satu, dimulai dengan elemen pertama. Lanjutkan pencarian sampai Anda menemukan elemen yang dicari atau Anda mencapai akhir array. Pencarian linier sering digunakan untuk memvalidasi item data. Algoritma pseudocode untuk pencarian linier dari sebuah array akan membutuhkan sebuah flag program bernama `element_found`. Bendera ini, awalnya disetel ke `false`, akan disetel ke `true` setelah nilai yang dicari ditemukan, yaitu, ketika elemen larik saat ini sama dengan item data yang dicari. Dalam algoritme berikut, item data yang dicari disimpan dalam variabel `input_value`, dan variabel `max_num_elements` berisi jumlah total elemen dalam array.

```
Linear_search_of_an_array
  Set max_num_elements to required value
  Set element_found to false
  Set index to 1
  DOWHILE (NOT element_found) AND (index <= max_num_elements)
    IF array (index) = input_value THEN
      Set element_found to true
    ELSE
      index = index + 1
    ENDIF
  ENDDO
  IF element_found THEN
    Print array (index)
  ELSE
    Print 'value not found', input_value
  ENDIF
END
```

Writing out the contents of an array

Elemen array dapat digunakan sebagai akumulator data, untuk ditulis ke laporan. Menuliskan isi array melibatkan mulai dengan elemen pertama dari array dan berlanjut sampai semua elemen telah ditulis. Ini dapat diwakili oleh loop DO sederhana. Dalam algoritma pseudocode berikut, nama arraynya adalah 'array' dan subscriptnya adalah 'index'. Jumlah elemen dalam array diwakili oleh number_of_elements.

```
Write_values_of_array
DO index = 1 to number_of_elements
    Print array (index)
ENDDO
END
```

Programming examples using arrays

Rancang program yang akan meminta dan menerima 18 nilai ujian dari tes matematika, menghitung rata-rata kelas, dan menampilkan semua skor dan skor rata-rata ke layar skor.

A Defining diagram

Input	Processing	Output
18 exam scores	Prompt for scores Get scores Calculate average score Display scores Display average score	18 exam scores average_score

B Control structures required

- 1 Array untuk menyimpan nilai ujian atau 'skor'
- 2 Indeks untuk mengidentifikasi setiap elemen dalam array
- 3 Lakukan DO loop untuk menerima skor
- 4 DO loop lain untuk menampilkan skor ke layer

C Solution algorithm

```
Process_exam_scores
  Set total_score to zero
  DO index = 1 to 18
    Prompt operator for score
    Get score
    scores (index) = score
    total_score = total_score + scores (index)
  ENDDO
  Compute average_score = total_score / 18
  DO index = 1 to 18
    Display scores (index)
  ENDDO
  Display average_score
END
```

Two-dimensional arrays

Dalam beberapa aplikasi bisnis, misalnya, ada kebutuhan untuk array multidimensi, di mana dua atau lebih subskrip diperlukan untuk menemukan elemen dalam array. Array dua dimensi dimuat dalam kolom dalam urutan baris; semua kolom untuk baris satu dimuat sebelum pindah ke baris dua dan memuat kolom untuk baris itu, dan seterusnya. Pembacaan serangkaian nilai dari file ke dalam array dua dimensi dapat diwakili oleh loop DO dalam loop DOWHILE.

```
Read_values_into_array
  Set max_num_elements to 24
  Set row_index to zero
  Read input file
  DOWHILE (input values exist) AND (row_index < 6)
    row_index = row_index + 1
    DO column_index = 1 to 4
      freight_charges (row_index, column_index) = input value
      Read input file
    ENDDO
  ENDDO
  IF (input values exist) AND row_index = 6 THEN
    Print 'Array size too small'
  ENDIF
END
```