MODUL PRAKTIKUM 16 - SKEMA PEMROSESAN SEKUENSIAL

ALGORITMA DAN PEMROGRAMAN 1

S1 INFORMATIKA

Telkom University

Fakultas Informatika
School of Computing
Telkom University

informatics lab

LEMBAR PENGESAHAN

Saya yang bertanda tangan di bawah ini:

Nama	: Prasti Eko Yunanto, S.T., M.Kom.
NIP	: 19890017
Koordinator Mata Kuliah	: Algoritma dan Pemrograman 1
Prodi	: S1 Informatika

Menerangkan dengan sesungguhnya bahwa modul ini digunakan untuk pelaksanaan praktikum di Semester Ganjil Tahun Ajaran 2024/2025 di Laboratorium Informatika, Fakultas Informatika, Universitas Telkom.

Bandung, 17 Agustus 2024

Fakultas Informatika
School of Computing
Telkom University
informatics lab

Mengesahkan,	Mengetahui,
Koordinator Mata Kuliah	Kaprodi S1 Informatika
Algoritma Pemrograman 1	
Prasti Eko Yunanto, S.T., M.Kom.	Dr. Erwin Budi Setiawan, S.Si., M.T.
NIP. 19890017	NIP. 00760045

MODUL 16. SKEMA PEMROSESAN SEKUENSIAL

(PENGAYAAN)

16.1 Pengantar Skema Pemrosesan Sekuensial

Dengan dipersenjatai bentuk perulangan dan bentuk percabangan, banyak problem komputasi yang dapat diselesaikan. Berikut ini beberapa skema (pola) yang umum ditemukan untuk pemrosesan data (secara sekuensial).

16.2 Pembacaan Data Tanpa Marker pada Akhir Rangkaian Data

Pola ini memperlihatkan bahwa semua data yang diberikan pada masukan adalah data yang harus diproses. Berikut contoh polanya

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(n)	fmt.**Scanln**(&n)
2	i = 1	i = 0
3	**while** i <= n **do**	**for** i < n {
4	**input**(dat)	fmt.**Scan**(&dat)
5	*{ kode untuk memproses dat }*	*// kode untuk memproses dat*
6	i = i + 1	i = i + 1
7	**endwhile**	}

Fakultas Informatika
School of Computing
Telkom University
informatics lab

Pada pola di atas, terlihat bahwa variabel ***dat*** yang diperoleh pada baris ke-4 adalah data valid yang pasti diproses pada baris ke-5. Baris ke-5 bisa berisi potongan kode apapun yang digunakan untuk memproses variabel ***dat*** tersebut.

Berikut contohnya pada potongan program pencarian nilai maksimum:

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(n)	fmt.**Scan**(&n)
2	max = *{BILANGAN_KECIL}*	max = *// BILANGAN_KECIL*
3	i = 1	i = 1
4	**while** i <= n **do**	**for** i <= n {
5	**input**(dat)	fmt.**Scan**(&dat)
6	**if** dat > max **then**	**if** dat > max {
7	max = dat	max = dat
8	**endif**	}
9	i =i + 1	i = i + 1
10	**endwhile**	}
11	**output**("Data terbesar", max)	fmt.**Println**("Data terbesar", max)

16.3 Pembacaan Data dengan Marker pada Akhir Rangkaian Data

Pada pola denga marker, terdapat data yang dipersiapkan khusus untuk menghentikan perulangan. Artinya semua data yang diberikan pada masukan adalah data yang valid, kecuali data yang terakhir, karena digunakan untuk menghentikan perulangan. Berikut contoh polanya:

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(dat)	fmt.**Scanln**(&dat)
2	**while** dat != MARKER **do**	**for** dat != MARKER {
3	*{kode untuk memproses dat}*	*// kode untuk memproses dat*
4	**input**(dat)	fmt.**Scanln**(&dat)
5	**endwhile**	}

Apabila kita memperhatikan pola dengan marker di atas, terlihat bahwa selalu ada pengkondisian pada while yang akan selalu menyecek nilai ***dat***. Semua nilai pada variabel ***dat***, baik yang diperoleh pada baris ke-1 ataupun baris ke-4 akan selalu dicek pada baris ke-2. Apabila variabel ***dat*** tidak berisi ***marker***, maka ***dat*** dapat diproses pada baris ke-3, sebaliknya apabila ***dat*** adalah ***marker***, maka perulangan akan berhenti, dan ***dat*** tidak akan diproses pada baris ke-3.

Fakultas Informatika
School of Computing

Perhatikan contoh potongan program mencari nilai maksimum berikut ini:

	Notasi Algoritma	Notasi dalam bahasa Go
1	max = *{BILANGAN_KECIL}*	max = *// BILANGAN_KECIL*
2	**input**(dat)	fmt.**Scan**(&dat)
3	**while** dat != MARKER **do**	**for** dat != MARKER {
4	**if** dat > max **then**	**if** dat > max {
5	max = dat	max = dat
6	**endif**	}
7	**input**(dat)	fmt.**Scan**(&dat)
8	**endwhile**	}
9	**output**("Data terbesar", max)	fmt.**Println**("Data terbesar", max)

Nilai ***marker*** bisa nilai berapapun, biasanya diberikan pada soal atau kita biasanya bisa memberikan nilai berdasarkan asumsi.

16.4 Kemungkinan Rangkaian Data Kosong, Hanya Ada Marker

Pola dengan marker pada contoh di atas memungkinkan terjadi bahwa data pertama yang diberikan pada masukan adalah marker, artinya tidak ada satu datapun yang valid. Kemungkinan ini disebut juga rangkaian data kosong atau kasus kosong.

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(dat)	fmt.**Scanln**(&dat)
2	**if** dat == MARKER **then**	**if** dat == MARKER {
3	*{kode untuk data kosong}*	*// kode untuk data kosong*
4	**else**	}**else**{
5	**while** dat != MARKER **do**	**for** dat != MARKER {
6	*{kode untuk memproses dat}*	*// kode untuk memproses dat*
7	**input**(dat)	fmt.**Scanln**(&dat)
8	**endwhile**	}
9	**endif**	}

Pada pola di atas ditambahkan struktur kontrol pengkondisian atau percabangan apabila ada kasus kosong yang terjadi. Berikut ini adalah contoh kode untuk mencari nilai maksimum.

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(dat)	fmt.**Scanln**(&dat)
2	**if** dat == MARKER **then**	**if** dat == MARKER {
3	**output**("tidak ada data")	fmt.Println("tidak ada data")
4	**else**	}**else**{
5	max = *{BILANGAN KECIL }*	max = *// BILANGAN KECIL*
6	**while** dat != MARKER **do**	**for** dat != MARKER {
7	**if** dat > max **then**	**if** dat > max {
8	max = dat	max = dat
9	**endif**	}
10	**input**(dat)	fmt.**Scanln**(&dat)
11	**endwhile**	}
12	**output**("Data terbesar", max)	fmt.**Println**("Data terbesar", max)
13	**endif**	}

Fakultas Informatika School of Computing Telkom University
informatics lab

16.5 Elemen Pertama Perlu Diproses Tersendiri/Kasus Khusus

Pada pola ini data pertama diproses terlebih dahulu secara khusus sebelum perulangan dilakukan. Apabila melihat contoh pencarian nilai maksimum di atas, terlihat bahwa nilai variabel ***max*** selalu diinisialisasi oleh sebuah nilai ***BILANGAN KECIL*** berapapun. Kekurangan dari pendekatan ini adalah kita harus mengetahui secara pasti nilai-nilai yang mungkin ada pada variabel ***dat***, yanga mana nilai pada variabel ***dat*** tersebut **TIDAK BOLEH** lebih kecil dibandingkan nilai dari ***BILANGAN KECIL*** yang digunakan saat inisialisasi. Sebagai contoh mencari nilai temperatur atau suhu maksimum. Pada kasus ini, berapa nilai dari ***BILANGAN KECIL*** yang akan digunakan? Tentu kita akan kesulitan menentukannya, karena temperatur bisa

bernilai negatif. Oleh karena itu, dengan menggunakan pola Kasus Khusus ini, penentuan ***BILANGAN KECIL*** tersebut data dapat terselesaikan.

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(dat)	fmt.**Scanln**(&dat)
2	**if** dat == MARKER **then**	**if** dat == MARKER {
3	*{kode untuk data kosong}*	*// kode untuk data kosong*
4	**else**	}**else**{
5	*{proses data pertama}*	*{proses data pertama}*
6	**input**(dat)	fmt.**Scan**(&dat)
7	**while** dat != MARKER **do**	**for** dat != MARKER {
8	*{kode untuk memproses dat}*	*// kode untuk memproses dat*
9	**input**(dat)	fmt.**Scanln**(&dat)
10	**endwhile**	}
11	**endif**	}

Pada pola tersebut, terlihat data atau elemen pertama yang diperoleh pada baris ke-1 diproses pada baris ke-5, selanjutnya data ke-2 didapat pada baris ke-6.

Berikut contoh potongan program untuk mencari nilai maksimum:

	Notasi Algoritma	Notasi dalam bahasa Go
1	**input**(dat)	fmt.**Scanln**(&dat)
2	**if** dat == MARKER **then**	**if** dat == MARKER {
3	**output**("tidak ada data")	fmt.**Println**("tidak ada data")
4	**else**	}**else**{
5	max = dat	max = dat
6	**input**(dat)	fmt.**Scan**(&dat)
7	**while** dat != MARKER **do**	**for** dat != MARKER {
8	**if** dat > max **then**	**if** dat > max {
9	max = dat	max = dat
10	**endif**	}
11	**input**(dat)	fmt.**Scanln**(&dat)
12	**endwhile**	}
13	**output**("Data terbesar", max)	fmt.**Println**("Data terbesar", max)
14	**endif**	}

Fakultas Informatika
School of Computing
Telkom University
informatics lab