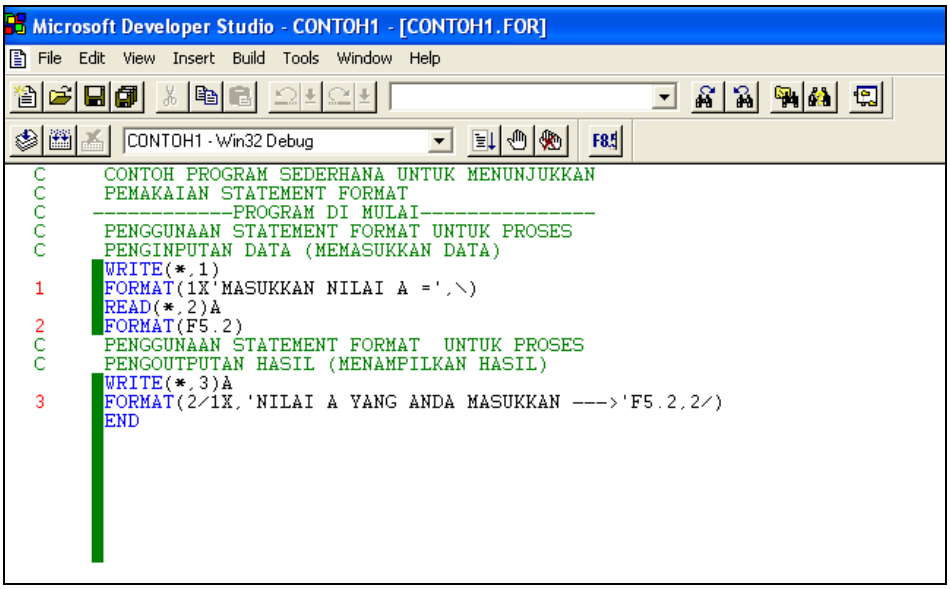
BUKU AJAR

Pemograman Komputer

Reni Suryanita, St, Mt
Mudjiatko, St, Mt

BUKU AJAR
MATA KULIAH
PEMROGRAMAN KOMPUTER
(TSS 2119 - 2 SKS)



OLEH :

RENI SURYANITA, ST, MT
MUDJIATKO, ST, MT

JURUSAN TEKNIK SIPIL
FAKULTAS TEKNIK
UNIVERSITAS RIAU

TINJAUAN UMUM MATA KULIAH

Matakuliah Pemrograman Komputer merupakan matakuliah yang disajikan pada semester tiga untuk mahasiswa Program Studi S1 Jurusan Teknik Sipil Fakultas Teknik Universitas Riau dengan beban 2 SKS.

Tujuan pemberian matakuliah Pemrograman Komputer ini adalah sebagai berikut :

1. Melatih mahasiswa untuk dapat berpikir logis sesuai dengan kasus yang dihadapinya yang diterapkan melalui diagram alir pemrograman.
2. Mahasiswa dapat membaca dan memahami program yang telah ada.
3. Mahasiswa dapat membuat program komputer sesuai dengan kasus yang diberikan kepadanya dengan mengikuti kaidah-kaidah penulisan program komputer yang benar.

Metode belajar yang dipakai untuk proses pembelajaran matakuliah Pemrograman Komputer ini lebih difokuskan kepada tugas mandiri dan kelompok yang dikerjakan oleh mahasiswa dengan bantuan komputer. Penyelesaian kasus dapat bantu oleh asisten mata kuliah di laboratorium Komputer Jurusan Teknik Sipil UNRI.

Mahasiswa diharapkan dapat menggunakan buku ini sebagai pegangan dalam mengikuti tatap muka di kelas.

Evaluasi hasil pembelajaran dapat diketahui dari nilai quis, tugas terstruktur, ujian tengah semester dan ujian akhir semester. Besaran nilai lengkap akhir diperoleh dengan cara menjumlahan nilai masing-masing item yang telah dikalikan dengan bobot prosentase masing-masing item tersebut. Nilai akhir adalah rerata dari nilai pada UTS dan UAS. Nilai lengkap akhir semester tersebut dinyatakan dengan nilai mutu kompetensi lulus A(sangat baik), B (Baik), C (Cukup),dan tidak kompeten (tidak lulus) dengan nilai E (gagal). Interval angka mutu untuk A =4, B=3, C=2, dan E=0.

RENCANA KEGIATAN PROGRAM PEMBELAJARAN (RKPP)
PRODI TEKNIK SIPIL S1
FAKULTAS TEKNIK-UNIVERSITAS RIAU

1. **Jurusan/Program Studi** : Teknik Sipil/ Teknik Sipil S1
2. **Mata Kuliah** : Pemrograman Komputer
3. **Kode Mata Kuliah** : TSS - 2119
4. **Jumlah Satuan Kredit Semester** : 2 SKS
5. **Jumlah Temu Muka** : 16 kali
6. **Deskripsi Mata Kuliah** :

Meliputi pembahasan tentang diagram alir dan logika program, syintax-syntax yang digunakan dalam pembuatan program dalam bahasa Fortran, cara penggunaan statement format, input-output, dimension (array), penggunaan statement kontrol (IF), Subroutine dan penggunaan FILE dalam pembuatan program komputer.

7. **Standard Kompetensi** :

Setelah mengikuti matakuliah ini mahasiswa diharap:

- mampu membuat dan menerapkan konsep-konsep dasar ilmu pengetahuan dan keteknikan khususnya konsep dasar bahasa Fortran dalam program komputer untuk menyelesaikan masalah-masalah ketekniksipilan,
- mampu menjaga relevansi ilmu khususnya pemograman komputer dalam menghadapi perkembangan terakhir di bidang rekayasa teknik sipil,
- mampu bekerja sama dalam tim multidisiplin.

8. **Sistem Penilaian**:
    - Tugas Terstruktur (tugas mandiri + tugas kelompok) : 20%
    - Quiz : 10%
    - Ujian Tengah Semester : 35%
    - Ujian Akhir Semester : 35%
9. **Sumber Belajar**:

- Jogiyanto, Teori dan Aplikasi Program Komputer Bahasa Fortran, Yogyakarta: Andi Offset, 1993
- Bambang Suryoatmono, Bahasa Fortran: Dari Fortran IV hingga Fortran Powerstation, Bandung, 1996
- Amrinsyah Nasution, Pemrograman dengan Bahasa Fortran, Jakarta: 1995.
- Buku Ajar Matakuliah Pemrograman Komputer, Prodi S1 Teknik Sipil 2007

KOMPETENSI DASAR	MATERI PEMBELAJARAN	INDIKATOR	ALOKASI WAKTU
Kemampuan menjelaskan filosofi dasar dalam membuat diagram alir untuk sebuah program komputer	Kontrak perkuliahan, RP/RKPP, Diagram Alir Pemrograman dan contoh Penggunaan serta latihan pembuatan diagram alir	Kejelasan filosofi diagram alir, dapat membuat diagram alir, kualitas dan kerapian tugas mandiri.	4 x 50 menit
Kemampuan menyebutkan bermacam-macam syntax dalam bahasa Fortran dan menggunakannya dalam program komputer.	Perkembangan bahasa Fortran, struktur program Fortran, konstanta, operator, ungkapan, nama, verb, unit Specifier, format specifier serta latihan	Dapat menyebutkan tipe konstanta, dapat menggunakan nama variabel dalam bahasa fortran dengan benar serta menggunakan operator bahasa Fortran dalam program sederhana. Kualitas tugas kelompok.	4 x 50 menit
Kemampuan menjelaskan fungsi dan kegunaan Statement Format dalam program bahasa Fortran.	Bentuk umum statement Format, edit diskripsi berulang, edit diskripsi tidak berulang dan latihan	Dapat menjelaskan fungsi dan kegunaan statement Format . Kualitas dan kerapian tugas mandiri.	2 x 50 menit
Kemampuan menggunakan statement Input dan Output dalam program bahasa Fortran	Statement READ, statement WRITE dan latihan penggunaan READ dan WRITE.	Dapat membuat program computer menggunakan statement input dan output.	4 x 50 Menit
Kemampuan menggunakan Statement Dimension (array) dalam program Fortran	Statement Dimension serta latihan penggunaan dalam program komputer.	Dapat membuat program komputer menggunakan statement Dimension (Array) dan kualitas tugas mandiri.	4x 50 menit
Kemampuan dalam pembuatan program komputer dengan menggunakan statement kontrol (IF)	Pendahuluan, statement GOTO, statement IF dan latihan penggunaan statement kontrol.	Dapat membuat program komputer dengan menggunakan statement kontrol (IF)	4x 50 menit
Kemampuan menggunakan statement SUBROUTINE dalam program komputer	Pendahuuan, statement SUBROUTINE, dan latihan dalam penggunaan SUBROUTINE	Dapat menggunakan statement SUB-ROUTINE dalam pembuatan program komputer. Kualitas tugas kelompok.	4x 50 menit
Kemampuan dalam pembuat-an program komputer dengan menggunakan FILE	Pendahuluan, Nama FILE, Statement OPEN FILE, Statement READ, Statement WRITE, Statement CLOSE dan latihan penggunaan FILE.	Dapat menggunakan statement OPEN FILE dan CLOSE dalam pembuatan program komputer. Kualitas tugas kelompok	2x 50 menit

KATA PENGANTAR

Puji Syukur Kami panjatkan kehadirat Allah SWT karena akhirnya kami bisa menyelesaikan Buku Ajar untuk matakuliah Pemrograman Komputer. Buku Ajar ini merupakan edisi yang pertama sehingga kemungkinan kekurangan di sana sini masih akan banyak dijumpai. Kritik membangun atau pun koreksi dari pembaca sangat kami harapkan.

Buku Ajar Pemograman Komputer ini ditujukan bagi mahasiswa semester 3 (tiga) Program S1 Jurusan Teknik Sipil agar dapat dijadikan pegangan dan panduan dalam mengikuti tatap muka perkuliahan di kelas dan pegangan bagi mahasiswa untuk dapat belajar mandiri. Selain itu buku ini dapat dijadikan acuan dan referensi bagi dosen dalam menyampaikan materi perkuliahan.

Adapun naskah Buku Ajar ini terdiri dari 8 (delapan) bab, yang berisi uraian umum tentang diagram pemrograman, statement format, statement input/output, statement kontrol, statement dimension (array) dan terakhir tentang pembuatan File. Materi disajikan dalam bentuk teori dan contoh penggunaan program. Setiap akhir bab pembahasan diikuti dengan soal latihan dengan harapan mahasiswa dapat lebih memahami materi dengan cara menyelesaikan kasus yang diberikan.

Semoga kehadiran buku ajar ini dapat mempermudah bagi mahasiswa dan pembaca dalam memahaminya dan bisa bermanfaat dengan baik.

Pekanbaru, 21 Juli 2007

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB I
DIAGRAM ALIR PEMROGRAMAN

KOMPETENSI DASAR :

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menjelaskan tujuan pembuatan diagram alir
2. Menjelaskan arti masing-masing bagan (lambang) pada diagram alir.
3. Membuat bagan alir perhitungan dalam rekayasa Teknik Sipil.

1.1. PENDAHULUAN

Sebelum membahas diagram alir, terlebih dahulu perlu kita pahami apa yang dimaksud dengan pemrograman. Pemrograman yaitu salah satu cara untuk merubah suatu permasalahan ke dalam bentuk sintak-sintak (aturan khusus) bahasa program. Proses perubahan tersebut harus dilakukan dengan dasar-dasar pemrograman yang baik yang artinya pemahaman mengenai alur data yang akan dipergunakan dan alur penyelesaian haruslah benar-benar valid dan sesuai dengan persamaan matematika yang ada.

Sebagai seorang programer pemula, anda perlu mengenal dan memahami diagram alir pemrograman agar dapat lebih mudah dalam pendalaman logika suatu permasalahan. Pembuatan diagram alir yang sesuai akan mempermudah dalam pembuatan list atau isi dari program itu sendiri.

Pemahaman diagram alir ini merupakan tahap awal dari beberapa tahap pemrograman. Kesalahanan dalam pembuatan atau penterjemahan suatu permasalahan kedalam diagram alir akan menimbulkan kesalahan di saat pembuatan list program dan berakibat fatal terhadap validitas dari program itu sendiri.

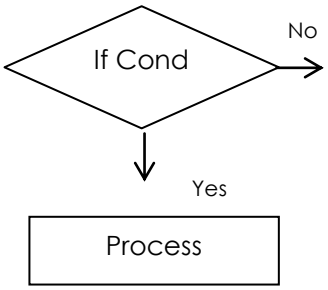
1.2. DIAGRAM ALIR

Diagram Alir (Flowchart) merupakan suatu bentuk diagram yang diwakilkan oleh bentuk – bentuk bagan (lambang), yang dapat mengartikan

sesuatu pembahasan. Diagram alir ini akan dihubungkan oleh suatu garis arah yang menggambarkan tahapan tahapan penyelesaian permasalahan.

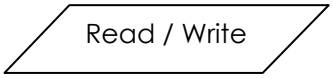
Bentuk – bentuk dasar dari bagan tersebut diperlihatkan sebagai berikut :

Diartikan sebagai awal suatu proses pemrograman

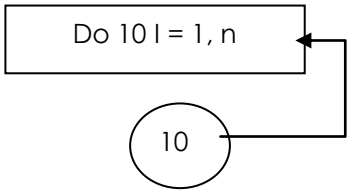


Diartikan sebagai bagan untuk proses pemilihan ya/tidaknya suatu proses akan dilaksanakan

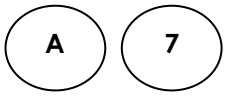
Diartikan sebagai tahap perhitungan-/pembahasan



Diartikan sebagai bagan untuk proses pembacaan data atau penulisan hasil proses perhitungan



Diartikan sebagai tahap perulangan proses diantara 2 bagan dan menggunakan nomor label 10



Diartikan sebagai tanda sambung dari sebuah program yang terpisah karena keterbatasan halaman atau tempat penulisan dan dapat dinyatakan dengan huruf atau angka.

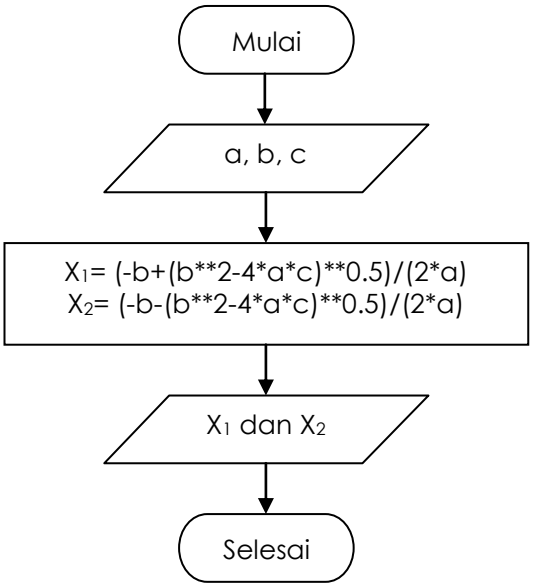
1.3. CONTOH PENGGUNAAN

Persamaan akar kuadrat dapat diselesaikan dengan rumus ABC untuk sebagai berikut :

$$X_{12} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Data apa sajakah yang diperlukan untuk mendapatkan nilai X ?

Kita memerlukan data nilai a, b, dan c untuk menghasilkan nilai X_1 dan X_2. Maka bentuk diagram alirnya adalah sebagai berikut:



Memulai Program

(Running program)

Baca data a, b dan c dari file masukan (input file)

Hitung nilai X_1 dan X_2 dari nilai a,b dan c yang ada

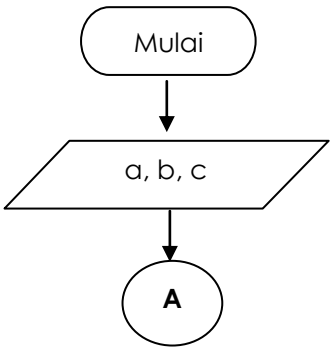
Tulis Hasil Perhitungan nilai X_1 dan X_2

Running Program selesai

Jika kita ingin membuat program yang dibatasi oleh nilai maka dapat kita nyatakan sebagai berikut:

”Jika nilai b = 5 maka nilai a = 2 dan c = 4”

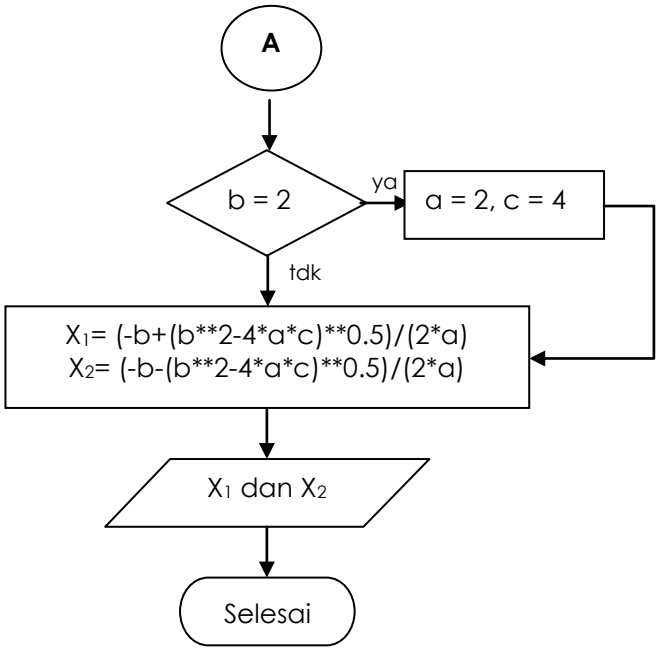
Bentuk diagram alir dengan pernyataan diatas adalah :



Mulai Running Program

Baca data a, b dan c dari input file

Tanda sambung



Tanda sambung

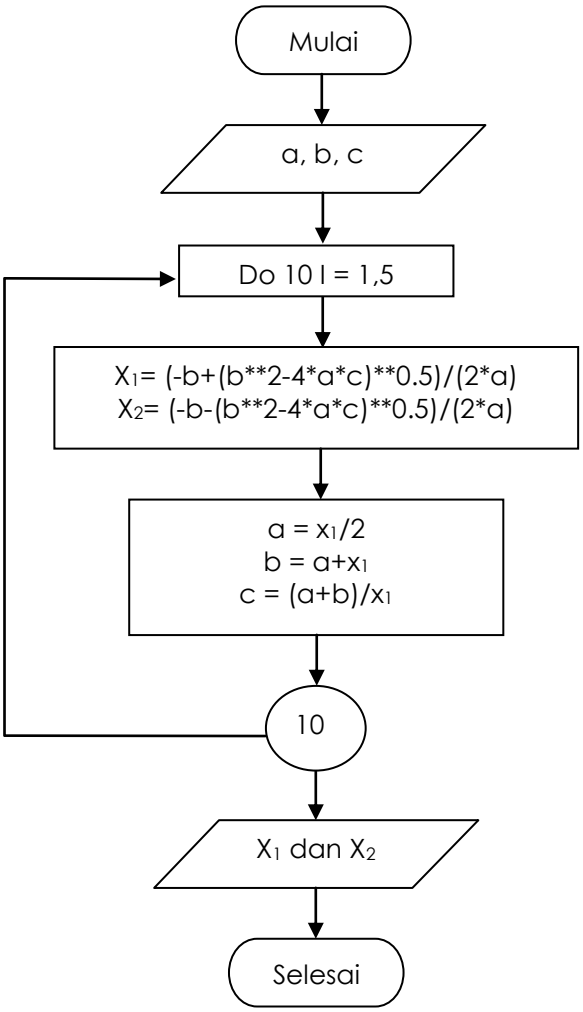
Apakah nilai b=2 ? Jika ya maka nilai a=2 dan c=4, jika tidak nilai a, b dan c tetap

Hitung nilai X_1 dan X_2 dari nilai a,b dan c yang ada

Tulis Hasil Perhitungan nilai X_1 dan X_2

Running Program selesai

Tentukan nilai ke lima dari soal diatas jika ditentukan untuk nilai a = $x_1/2$, b=a+x_1 dan c=(a+b)/x_1



Mulai Running Program

Baca data a, b dan c dari input file

Lakukan perulangan dengan menggunakan no. label 10 untuk menghitung nilai X_1 dan X_2

Hitung nilai X_1 dan X_2 dari nilai a,b dan c yang ada

Hitung nilai a, b dan c

Lanjutkan perulangan

Tulis Hasil Perhitungan nilai X_1 dan X_2

Running Program selesai

1.4. RANGKUMAN

- Diagram Alir (Flowchart) merupakan suatu bentuk diagram yang diwakilkan oleh bentuk – bentuk bagan (lambang), yang dapat mengartikan sesuatu pembahasan.
- Diagram alir dihubungkan oleh suatu garis arah yang menggambarkan tahapan tahapan penyelesaian permasalahan.
- Diagram alir dibuat sebelum kita membuat program komputer dan digunakan sebagai panduan penyusunan dan perhitungan program.

1.5. LATIHAN

Kerjakanlah latihan-latihan di bawah ini dan diskusikan hasilnya dalam kelompok.

a. Buatlah diagram alir untuk menghitung isi (volume) tabung dengan jari-jari alas bernilai 1 sampai 10.

b. Buat diagram alir untuk menentukan nilai ujian mahasiswa dengan kriteria nilai sebagai berikut.

   Nilai 80 < A≤ 100

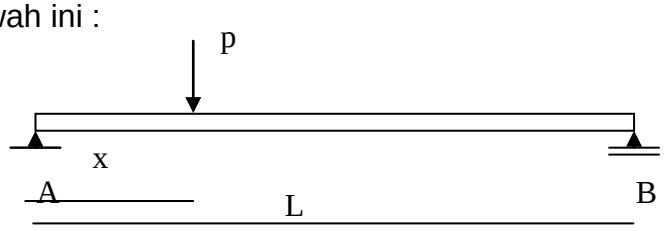
   Nilai 60 < B ≤ 80

   Nilai 50 < C ≤ 60

   Nilai 40 < D≤ 50

   Nilai E ≤ 40

c. Buat diagram alir untuk menghitung reaksi perletakan dari gambar dibawah ini :



d. Buatlah diagram alir untuk menghitung momen pada titik x, jika nilai x pada soal. c di atas berubah-ubah mulai dari x = 1 meter sampai x = L meter.

e. Buatlah sebuah diagram alir suatu perhitungan yang kamu pilih dan rencanakan sendiri untuk didiskusikan di dalam kelas.

BAB II
SEKILAS BAHASA FORTRAN

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

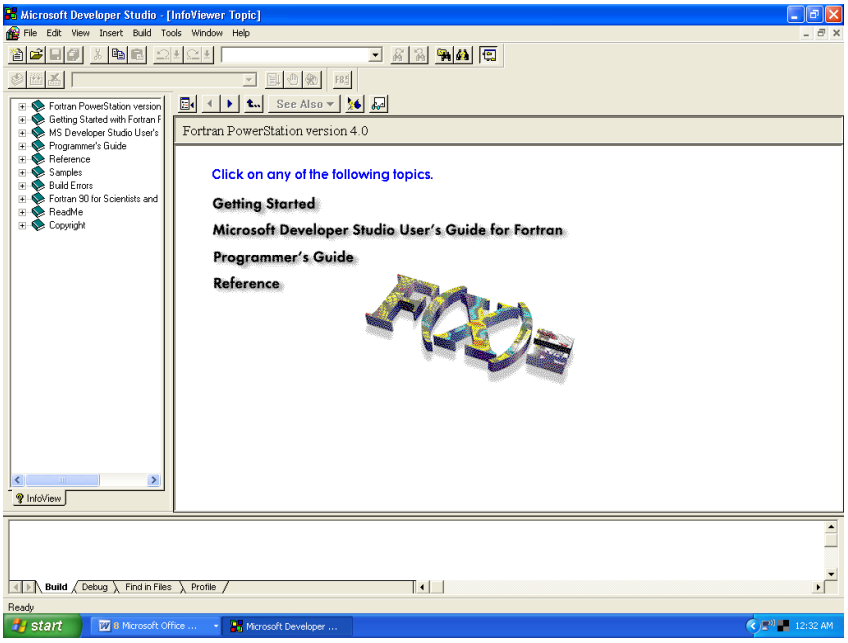
1. Menjelaskan struktur dasar bahasa Fortran
2. Menyebutkan contoh-contoh konstanta numerik dan konstanta karakter
3. Menjelaskan pengertian konstanta integer dan real
4. Menggunakan nama variabel dalam bahasa fortran dengan benar.
5. Menggunakan operator bahasa Fortran dalam program sederhana.

2.1. PERKEMBANGAN BAHASA FORTRAN

Bahasa Fortran merupakan suatu bahasa pemograman tingkat tinggi atau bahasa program yang berorientasi ke suatu masalah tertentu seperti rumus-rumus atau permasalahan teknik. Sedangkan kata Fortran berasal dari **For**mula **trans**lator.

Fortran pertama kali dikembangkan pada tahun 1950-an oleh IBM. Awalnya bahasa Fortran dikenal dengan sebutan Fortran 66 atau Fortran IV. Pada tahun 1977, bahasa program ini mengalami perkembangan menjadi Fortran 77 yang telah distandardisasikan oleh American National Standard Institute (ANSI). Kompiler Fortran untuk komputer mikro pertama kali dikembangkan oleh Microsoft pada tahun 1982. Versi kompilernya dimulai dengan versi 3.1 sampai versi 5.1. Pada Versi 5.1 ini telah banyak mengalami perkembangan misalnya adanya text editor (pengolah kata) dan bermacam-macam fungsi.

Pada perkuliahan pemrograman komputer ini kita akan menggunakan kompiler MS Fortran Power Station untuk PC dengan kecepatan 32-bit dan dikenal dengan Fortran 90.



Gambar.1. Tampilan awal layar Fortran PowerStation versi 4.0

2.2. STRUKTUR PROGRAM FORTRAN

Stuktur dasar program fortran dapat kita kategorikan berupa:

1. Metacommand sifatnya adalah optional didalam program fortran, artinya tidak harus ada.
2. Komentar dapat berupa tulisan bebas apapun yang berguna untuk memberi keterangan pada program sehingga memudahkan untuk membaca program tersebut.
3. Statement merupakan inti dari program yang berupa instruksi-intruksi kepada komputer.

Untuk dapat membuat program yang benar dalam bahasa fortran terlebih dahulu perlu diperhatikan posisi kolom-kolom di dalam program yaitu :

- Kolom ke-1 digunakan untuk komentar atau metacommand. Kolom ini bila diisi dengan karakter “C” atau “*” (asterik) menunjukkan bahwa baris tersebut berisi komentar bebas. Jika berisi karakter”$” (dollar) menunjukkan bahwa baris tersebut berisi metacommand.
- Kolom ke 1 sampai kolom ke 5 digunakan untuk menuliskan nomor label statement, berupa suatu angka yang menunjukkan letak dari suatu statement.

- Kolom ke 6 digunakan untuk indikasi sambungan statement dari baris sebelumnya. Baris sambungan harus diberi indikasi dengan cara meletakkan di kolom ke 6 karakter apapun kecuali blank atau 0.
- Kolom ke 7 sampai kolom ke 72 digunakan untuk menulis statement fortran.
- Kolom ke 73 sampai kolom ke 80 tidak digunakan oleh fortran sehingga kita dapat menulis komentar bebas yang menerangkan statement bersangkutan.

2.3. KONSTANTA

Konstanta merupakan nilai yang sudah pasti dan yang tidak akan berubah didalam program. Konstanta dikenal juga dengan istilah data.
Konstanta dapat berupa numerik, karakter maupun logika.

A. Konstanta Numerik

Konstanta numerik terdiri dari:

Konstanta integer : adalah nilai numerik bilangan bulat. Untuk nilai numerik integer 2 byte dapat berkisar dari -32767 sampai dengan 32767 dan untuk nilai numerik integer 4 byte berkisar dari -2147483647 sampai dengan +2147483647.

Contoh Integer :

-35 -7230 0 5 276584

Konstanta real ketepatan tunggal : merupakan konstanta numerik berbentuk nilai pecahan. Nilai positip berkisar antara +1.2 E-38 (berarti $1.2*10^{-38}$) sampai +3.4 E+38 dan untuk nilai negatip antara -3.37E+38 sampai -1.2 E-38 serta bilangan 0 (nol). Konstanta ini harus mengandung sebuah titik desimal dan dapat mempunyai angka desimal sebanyak 7 angka.

Contoh konstanta real ketepatan tunggal :

1.54 3.745E-3 256.E+11 .16524312

Konstanta real ketepatan ganda : merupakan konstanta numerik berbentuk nilai pecahan. Nilai positip berkisar antara +2.2D-308 (berarti 2.2 *10^{-308} sampai +1.8 D+308 dan untuk nilai negatip antara -1.8D+308 (berarti -1.8 * 10^{308}) sampai -2.2D-308 serta bilangan 0 (nol). Konstanta ini harus mengandung sebuah titik desimal dan dapat mempunyai angka desimal sebanyak 15 angka.

Contoh konstanta real ketepatan ganda :

.23D-45 -3.745245D-33 1256.D+11 .16524312D+110

B. Konstanta karakter

Konstanta ini merupakan nilai dari karakter-karakter ASCII yang ditulis dalam tanda petikan tunggal. Panjang maksimum dari konstanta karakter adalah 127 karakter. Untuk konstanta karakter, huruf besar dan huruf kecil tidak dianggap sama. Penggunaan variabel karakter dapat kita dilihat pada contoh berikut.

Contoh :

’SkS’	: bernilai SkS
’ ’	: bernilai blank sebanyak 2 karakter
’jum”at’	: bernilai jum’at
’Momen Mx’	: bernilai Momen Mx

C. Konstanta logika

Konstanta logika yang dapat dipergunakan adalah :

.TRUE. : untuk menyatakan logika benar

.FALSE.: untuk menyatakan logika salah

2.4. OPERATOR

Operator dalam program fortran dapat berupa operator aritmatika, operator hubungan dan operator logika.

A. Operator aritmatika

Operator aritmatika merupakan tanda operasi yang digunakan dalam perhitungan aritmatika seperti pada tabel berikut :

Tabel 2.1. Tanda Operator Aritmetika

Operator aritmatika	Maksud	Jenjang
**	Perpangkatan	1
*	Perkalian	2
/	Pembagian	2
+	Pertambahan	3
-	Pengurangan	3

B. Operator hubungan

Operator hubungan merupakan tanda operasi yang digunakan untuk menyatakan hubungan antara 2 buah elemen seperti pada tabel berikut :

Tabel 2.2. Tanda Operator Hubungan

Operator hubungan	Maksud
.LT.	Lebih kecil dari
.LE.	Lebih kecil sama dengan
.EQ.	Sama dengan
.NE.	Tidak sama dengan
.GT.	Lebih besar dari
.GE.	Lebih besar sama dengan dari

C. Operator logika

Operator Logika merupakan tanda operasi yang digunakan dalam operasi perbandingan logika dan diperlihatkan dalam tabel berikut :

Tabel 2.3. Tanda Operator Logika

Operator logika	Maksud	Jenjang
.NOT.	Tidak atau bukan	1
.AND.	Dan	2
.OR.	Atau	3
.EQV.	Kesamaan	4
.NEQV.	Ketidaksamaan	4

2.5. UNGKAPAN

Ungkapan dalam program fortran dapat berupa ungkapan aritmatika, ungkapan hubungan, ungkapan logika dan ungkapan karakter

A. Ungkapan arithmatika

Pada operator arithmatika, jenjang menunjukkan urutan dari elemen mana yang akan diproses terlebih dulu. Untuk merubah jenjang dapat kita gunakan tanda kurung buka ”(” dan tanda kurung tutup ”)” sehingga tidak menimbulkan kerancuan hasil perhitungan.

Contoh :

- A/D + B/C + D = $\frac{A}{D} + \frac{B}{C} + D$
- ((A*B+C)/(C/2*A))**1/6 = $\frac{\left(\frac{A*B+C}{\frac{C}{2}xA}\right)^{1}}{6}$
- Total = 6*(3.23*a+4.0 / (2. + 6**x))

  Ini berarti menghitung

  $$6\left(3.23a + \frac{4.0}{2+6^{x}}\right)$$

  dan menyimpan hasilnya pada variabel Total.

B. Ungkapan hubungan

Ungkapan hubungan membandingkan nilai dari dua numerik atau nilai dari dua karakter yang menghasilkan suatu nilai logika.

Contoh :

A.LT.B

Artinya : menunjukkan bagaimana hubungan antara nilai A dan B. Apakah nilai A lebih kecil dari nilai B atau tidak.

Jelaslah bahwa operasi yang menggunakan operator hubungan menghasilkan data berjenis logika.

Contoh berikut ini dapat menjelaskannya.

```
C23456789
      LOGICAL hasil
      REAL i , j
      DATA i/2.0/, j/3.0/
      Hasil =  i GT. j
      PRINT *,  hasil
      END
```

Selesai program di atas dijalankan variabel hasil akan dicetak di layar dengan simbol F yang berarti .FALSE. (salah), karena harga i (yaitu 2.0) lebih kecil dari pada harga j (yaitu 3.0). Kedua harga tersebut dimasukkan dengan menggunakan pernyataan DATA.

C. Ungkapan logika

Ungkapan logika dibentuk dengan menggunakan operator logika yaitu .NOT.,.OR. atau .AND. Ungkapan logika menyatakan ungkapan suatu nilai logika

Contoh :

(X.and.Y) .or.(T.and.Z)

artinya : nilai X dan Y atau nilai T dan Z

D. Ungkapan Karakter

Ungkapan karakter tidak boleh mnggunakan operator aritmetika.

Contoh :

Pangkat = ‘Sersan Mayor’

Jenis Kelamin = ‘Laki-laki’

Keterangan :

Pangkat, Jenis kelamin disebut dengan **nama variabel**

Sersan mayor, laki-laki disebut dengan **ungkapan karakter**

2.6. NAMA

Istilah nama dalam program bahasa Fortran digunakan untuk menunjukkan suatu variabel, larik (array), fungsi atau rutin bagian (subroutine). Untuk menghindari kesalahan pengetikan, sebaiknya nama variabel jangan dibuat terlalu panjang. Sebaliknya, untuk menunjang kejelasan suatu program, kita dapat menuliskan nama yang lebih panjang.

Adapun ketentuan yang perlu kita diperhatikan dalam penulisan nama adalah sebagai berikut:

a. Jumlah karakter maksimum adalah 31.
b. Tidak boleh mengandung karakter khusus, yaitu karakter-karakter selain huruf dan angka.
c. Karakter pertama harus berupa huruf alfabet (A sampai Z, dan a sampai z) dan karakter sisanya pada fortran 90 dapat berupa alfabet, angka (0 sampai 9), tanda garis bawah dan spasi.

   Contoh :

   Nilai_kuat_tekan

   Jumlah mahasiswa

   Tegangan1

   Regangan_2

Nama variabel digunakan untuk menyimpan suatu nilai konstanta atau hasil dari suatu ungkapan. Nama variabel dibedakan menjadi :

a. Variabel Integer

Variabel yang digunakan untuk menyimpan nilai numerik bulat. Bila tidak didefinisikan terlebih dahulu maka variabel integer tersebut harus ditunjukan oleh nama variabelnya yang diawali dengan huruf I, J, K, L, M dan N.

Besarnya memori yang dipergunakan dalam menyatakan variabel integer ini diatur dengan memberikan statement

Integer*2 variabel_integer untuk memori sebesar 2 byte

Integer*4 variabel_integer untuk memori sebesar 4 byte (default)

Contoh :

```
Integer*2 isi              {mendefinisikan variabel isi dengan 2 byte}
Isi = 275
Integer*2 luas, hasil {mendefinisikan variabel luas, hasil = 2 byte}
Integer*4 Total        {mendefinisikan variabel total = 4 byte}
```

b. Variabel Real

Variabel ini digunakan untuk menyimpan nilai numerik pecahan sebesar 4 byte. Nama variabel real ketepatan tunggal bila tidak didefinisikan terlebih dahulu harus diawali dengan huruf selain I, J, K, L, M atau N.

Kita dapat mendefinisikan variabel dengan nama yang diawali dengan karakter bebas yang tidak terikat dengan karakter apapun dengan mendefinisikan awal program menggunakan pernyataan :

Real variabel_real

Contoh :

```
Real Nilai              {mendifinisikan nilai real sebesar 4 byte}
Nilai = 256.358
```

Untuk mendefinisikan variabel yang dapat menyimpan nilai numerik pecahan sebesar 8 byte, nama variabel real harus diawali dengan huruf

selain I, J, K, L, M atau N dan harus sudah didefinisikan dengan statement type atau statement IMPLICIT variabel real.

Contoh :

```
IMPLICIT Hasil        {mendifinisikan nilai real sebesar 8 byte}
Hasil = 231.5D+38
```

c. Variabel Karakter

Variabel ini digunakan untuk menyimpan nilai karater. Nama dari variabel ini bebas diawali dengan huruf apapun dan mempunyai kapasitas default sebesar 4 byte. Tiap byte untuk variabel karakter dapat menyimpan sebesar 1 karakter. Bentuk definisi awal dari pernyataan charakter adalah :

Charakter*n variabel_karakter

Contoh :

Nama = ‘Muhammad’

Karena tidak didefinisikan maka informasi yang disimpan menjadi sebesar 4 byte atau hanya ‘muha’

```
Charakter Nama       {mendefinisikan Nama menjadi 1 byte}
Charakter*20 Nama    {mendefinisikan Nama menjadi 20 byte}
```

d. Variabel logika

Digunakan untuk menyimpan nilai logika yang mempunyai kapasitas default sebesar 4 byte. Untuk mendefinisikan variabel ini digunakan type dari variabel seperti :

Logical*n variabel_logika

Contoh :

```
Logical*2 status      {mendefinisikan status menjadi 2 byte}
```

2.7. VERB

Verb adalah kata kerja perintah yang terdapat dalam statement. Verb ini menunjukkan tindakan apa yang harus dilakukan oleh kompiler, misalnya verb WRITE menunjukkan perintah untuk mencetak sesuatu, READ untuk membaca data atau memasukkan nilai data atau CALL untuk memanggil

subroutine dan lain-lain. Untuk lebih jelasnya verb akan kita bahas pada bab selanjutnya.

2.8. UNIT SPECIFIER

Unit specifier merupakan nomor unit alat secara logika yang akan dipergunakan dalam operasi input atau output (I/O). Pernyataan yang menggunakan unit specifier yaitu READ, WRITE dan OPEN.

Unit Spesifier dapat berupa :

a. Tanda asterisk (*) atau 0 yang menunjukan bahwa nomor unit yang dipergunakan adalah bebas atau unit alat yang dipergunakan adalah keyboard atau layar.
b. Ungkapan integer yaitu nilai integer selain dari 0 yang menunjukan bahwa alat yang dipergunakan adalah file eksternal dapat berupa Name_file di disk atau printer. Nama file dapat berupa :
    - LPT1 atau prn menunjukan alat yang dipergunakan adalah printer
    - Con menunjukan alat yang dipergunakan adalah console
    - LPT2 menunjukan alat yang dipergunakan adalah printer
    - Com1 dan Com2 menunjukan alat yang dipergunakan adalah Comunication port
    - Nama file di disk yang menunjukan unit ralat yang dipergunakan adalah file di dalam disk.

Contoh :

```
WRITE(*,'(1X,A,\)') 'NAMA FILE DATA ANDA    : '
READ(*,'(A)') FIDA
OPEN(5,FILE=FIDA)
WRITE(*,'(1X,A,\)') 'NAMA FILE KELUARAN     : '
READ(*,'(A)')FILA
OPEN(6,FILE=FILA)
```

2.9. FORMAT SPECIFIER

Format specifier menunjukkan format yang akan dipergunakan oleh data input ataupun data output. Jika sebagai data input maka dipergunakan dalam statement Read dan jika sebagai data outpun digunakan pada statement Write.

Format Specifier dapat berupa :

a. Label statement yang menunjukan letak dari statement format yang mengatur bentuk dari data
b. Nama variabel integer
c. Ungkapan karakter
d. Tanda asterik (*)

Contoh :

```
WRITE(6,5)PIAS
5  FORMAT(1X,'JUMLAH PIAS UNTUK SALURAN  : ',F7.2,1X,'BUAH')
```

Penjelasan :

Lakukan penulisan nilai pias ke unit spesifier 6 dengan format sesuai dengan nomor label 5 yaitu : bergeser 1 spasi dari margin kiri, tulis JUMLAH PIAS UNTUK SALURAN :, sediakan ruang untuk bilangan real 7 karakter dan 2 desimal, bergeser 1 spasi dan tulis BUAH.

2.10.STATEMENT

Statement atau pernyataan dalam bahasa fortran dapat digolongkan kedalam dua bagian, yaitu statement yang terolah (executable statement) merupakan statement yang menyebabkan suatu operasi akan dilakuakan dan statement tidak terolah (nonexecutable statement) merupakan statement yang tidak melakukan suatu operasi.

Yang termasuk nonexecutable statement adalah

1. Statement FORMAT
2. Statement DATA dan statement PARAMETER
3. Statement spesifikasi
4. Statement PROGRAM, FUNCTION dan SUBROUTIN

Yang termasuk executable statement

1. Statement pengerjaan dan statement ASSIGN
2. Statement kontrol
3. Statement input/output

Peletakan statement didalam unit program fortran mempunyai urutan tersendiri antara satu jenis statement dengan statement yang lain. Tabel berikut menunjukkan urutan atau order dari statement di dalam unit program fortran.

Tabel 2.4. Urutan Statement Program Fortran

<table>
<tr><td colspan="4">Metacommand $DO66 dan $STORAGE</td></tr>
<tr><td colspan="2">Statement PROGRAM, FUNCTION atau SUBROUTINE</td><td rowspan="7">metacommand yang lain</td><td rowspan="8">baris komentar</td></tr>
<tr><td>Statement IMPLICIT</td><td rowspan="6">statement FORMAT</td></tr>
<tr><td>Statement spesifikasi yang lain</td></tr>
<tr><td>Statement DATA</td></tr>
<tr><td>Statement fungsi statement</td></tr>
<tr><td>Executable statement</td></tr>
<tr><td colspan="3">Statement END</td></tr>
</table>

2.11. RANGKUMAN

- Fortran berasal dari singkatan kata formula translator yaitu bahasa pemograman tingkat tinggi atau bahasa program yang berorientasi ke suatu masalah tertentu seperti rumus-rumus atau permasalahan teknik.
- Stuktur dasar program fortran dapat dikategorikan berupa metacommand, komentar dan statement yang merupakan inti dari program.
- Konstanta atau data dalam bahasa fortran merupakan nilai yang sudah pasti dan tidak akan berubah didalam program.
- Konstanta dapat berupa numerik, karakter maupun logika sedangkan operator dapat berupa operator aritmatika, operator hubungan dan operator logika.

2.12. LATIHAN

Kerjakanlah latihan berikut dan diskusikan hasilnya dalam kelompok.

1 Manakah nama valiabel yang tidak sah menurut bahasa Fortran?

a. Tegangan maksimum	b. KI_3	c. aLamat	d. 5_A
e. Luastepi	f. Nilail#1	g. W/c	h. N-asli
i. L3y	j. U4+2	k. Tegrata2	l. M_ Z

2. Tuliskan pernyataan FORTRAN untuk kalimat matematis berikut.

   a. Simpan hasil A^{3a} + 5B - 13 pada variabel h.

   b. Simpan hasil

$$3\sqrt{\frac{3a-2b}{4\sin(t)}}+3\cos\left(x-\frac{\pi}{2}\right)$$

   pada variabel HASIL.

   c. Simpan hasil $$d^{3}-\left[\frac{2a(3\sin(t)-1n2)}{4b-c^{2}}\right]$$

   pada variabel hasil.

3. Tuliskan kalimat matematis untuk pernyataan FORTRAN berikut

   a. Y=COS(A-DELTA/2.)+X**(3 . 0-B)

   b. TEMP = A(J+3)/K*D - SQRT(B+C/2.)

   c. b (i) = b (i-1) + delta*5/(2*T)

   d. SUM = SUM + A(I)

4. Jika A = 8.0 dan B = 10. 0, apakah hasil operasi logika berikut ini

a. A . GT. B	b. A . EQ. B	c. (B/2.0) .GT. A
d. .NOT. (A LE. B)	e.(A EQ. B) OR. (B.GT.A)	
f. (B.LE. A).EQV. ((A-.B).GE.0.0)		

5. Tulislah potongan program dalam bahasa FORTRAN untuk menghitung

$$\sum_{i=1}^{10} a_i$$

dan menyimpan ha da variabel Jumlah.

BAB III
STATEMENT FORMAT

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menggunakan statement Format untuk mengedit data pada alat-alat I/O pada pembuatan sebuah program komputer.
2. Membuat program sederhana dengan menggunakan format bahasa Fortran yang benar.

3.1. BENTUK UMUM STATEMENT FORMAT

Statement FORMAT dapat digunakan untuk mengedit pada alat-alat I/O yaitu mengatur letak, jenis dan panjang dari data yang akan dimasukan lewat alat input atau yang akan ditampilkan ke alat output.

Bentuk umum dari statement ini adalah :

<LStatement> FORMAT < Format_Spec>

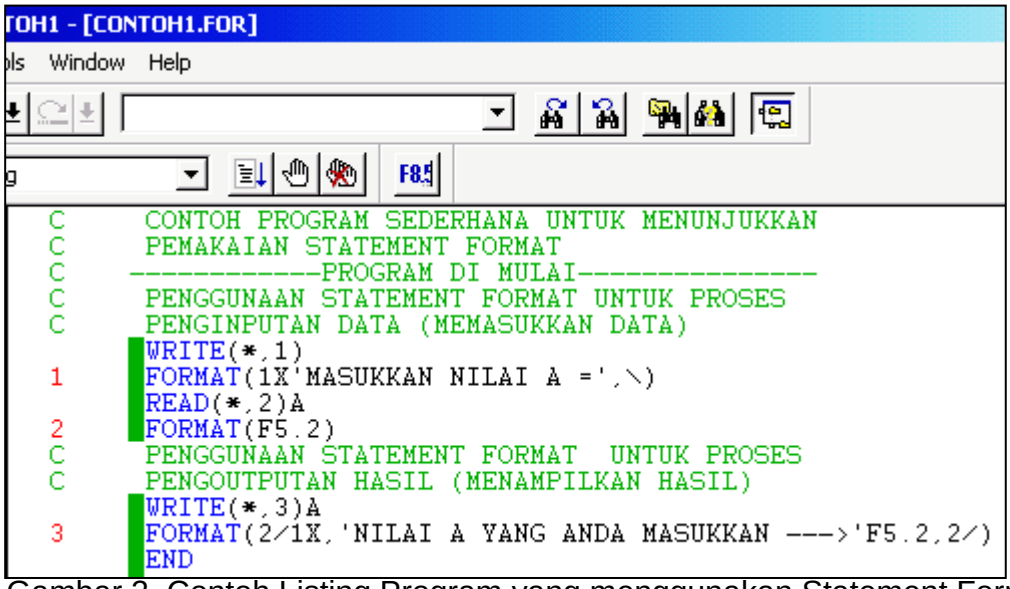
Penjelasan

<LStatement>	menunjukan label statement FORMAT yang akan digunakan oleh statement Read atau statement Write
< Format_Spec>	merupakan bentuk yang akan menyediakan informasi terhadap letak, jenis dan panjang dari data. Format ini harus ditulis dalam tanda kurung. Isi dari format spesification ini adalah edit descriptor.

Statement Format bebas diletakan dimana saja dalam program dan lebih baik jika statement ini dikumpulkan disuatu tempat di atas atau di bawah program asal sesuai dengan nomor label statement.

Contoh penulisan listing program menggunakan statement FORMAT dapat dilihat pada Contoh.1 dan Contoh.2. berikut:

Contoh.1.



Gambar.2. Contoh Listing Program yang menggunakan Statement Format

Penjelasan:

Kalimat pada baris dengan tanda C tidak akan diproses komputer dan berfungsi sebagai komentar.

WRITE (*,1) artinya tuliskanlah di monitor (tanda *) isi FORMAT 1.

READ (*,2) artinya masukkanlah dari keyboard (tanda *) nilai A dengan FORMAT 2.

WRITE (*,3) artinya tulislah di monitor nilai A dengan FORMAT 3

Isi masing masing format sebagai berikut:

FORMAT 1 berisikan MASUKKAN NILAI A =, 1X merupakan carriage control dan tanda \ (back slash) untuk menyatakan kursor tidak turun baris.

FORMAT 2 berisikan format untuk nilai A dengan tipe nilai real dua angka dibelakang koma.

FORMAT 3 berisikan 2 kali turun baris setelah itu ada carriage control dan tertulis di monitor NILAI A YANG ANDA MASUKKAN ---> , tipe nilai A adalah real 2 angka dibelakang koma dan turun lagi 1 baris.

Contoh. 2.

```
      Write(*,100)A,B
100   Format(1X,'Nilai A dan B ', F10.2, F10.2)
```

Penjelasan :

100 merupakan Label statement

1X,'Nilai A dan B ', F10.2, F10.2 merupakan Edit Descriptor dan Format Spesification

Edit Descriptor menunjukan informasi mengenai letak, tipe dan panjang dari masing-masing data yang akan dimasukan lewat alat input atau yang akan ditampilkan di alat output.

3.2. EDIT DISKIPSI BERULANG (REPEATEABLE EDIT DESCRIPTOR)

Merupakan edit descriptor yang dapat diulang dengan suatu nilai pengulangan tertentu. Yang termasuk repeatable edit descriptor yaitu : A, D, E, F, G, I dan L. Untuk D, E, F dan G dan I digunakan untuk data numerik sedangkan A digunakan untuk data alpanumerik atau data karakter dan L digunakan untuk data Logika.

a. Edit Descriptor I

   Dapat digunakan untuk input dan output dan digunakan untuk tipe data numerik integer. Bentuk umum dari penulisan edit descriptor ini adalah:

   <n>I <w>.<d>

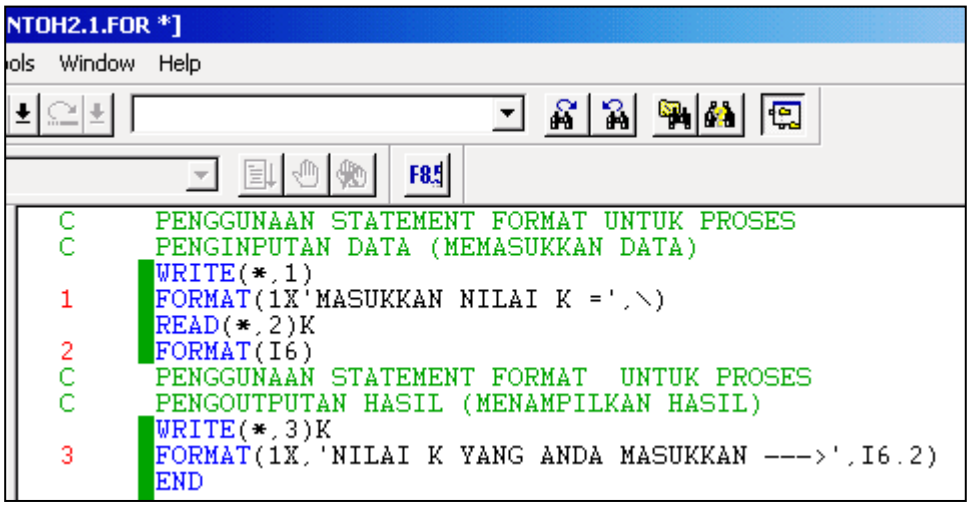
   Penjelasan :

   <n> Menunjukan banyaknya perulangan

   <w> Menunjukan panjangnya atau banyaknnya digit seluruh digit data (harus berupa bilangan integer bukan nol)

   <d> banyaknya angka nol dimuka sebagai pengganti blank.

Contoh 3.



Gambar.3. Contoh penulisan listing program untuk format bilangan Integer

Penjelasan:

Nilai K akan ditampilkan di monitor dengan format integer (bilangan bulat) 6 digit dengan 2 digit di depannya kosong.

Contoh.4.

```
C23456789
                K = 275
                I = 125
                J = 6125
                Write(*,10)K,I,J
  10            Format(1X,I10,I8,I5)
                End
```

Hasil running program :

```
                  275       25  6215
     x|xxxxxxxxxx|xxxxxxxx|xxxxx                    {jumlah spasi}
     1        10        8          5
```

b. Edit Descriptor F

Edit descriptor ini dapat digunakan untuk input dan output data. Bentuk umum dari penulisan edit descriptor ini adalah :

<n>F<w>.<d>

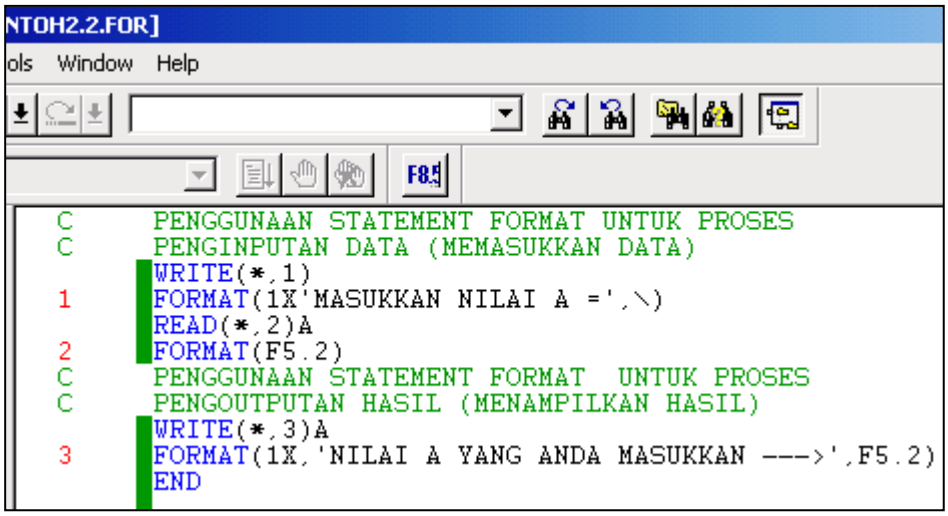
Penjelasan :

<n> Menunjukan banyaknya perulangan

<w> Menunjukan panjangnya atau banyaknnya digit seluruh digit data (harus berupa bilangan integer bukan nol)

<d> Menunjukan digit dibelakang titik desimal

Contoh.5.



Gambar.4. Contoh penulisan listing program untuk format bilangan Real

Penjelasan:

Nilai A yang akan masukkan dan ditampilkan di monitor mempunyai tipe real 5 digit dengan 2 angka dibelakang koma.

Contoh.6.

C23456789

```
A = 1
B = 3
C = 2
X1=(-B+(B**2-4*A*C)**0.5)/(2*A)
X2=(-B-(B**2-4*A*C)**0.5)/(2*A)
Write(*,'(1x,A,F8.3)')'Nilai X1 = ',X1
Write(*,'(1x,A,F8.3)')'Nilai X2 = ',X2
End
```

Hasil running Program :

Nilai X1 = -1.000

Nilai X2 = -2.000

c. Edit Descriptor E

Merupakan edit descriptor yang digunakan untuk tipe data numerik real ketepatan tunggal yang disajikan dengan bentuk eksponensial.

Bentuk umum dari Penulisan edit descriptor ini adalah :

<n>E<w>.<d>

Penjelasan :

<n> Menunjukan banyaknya perulangan

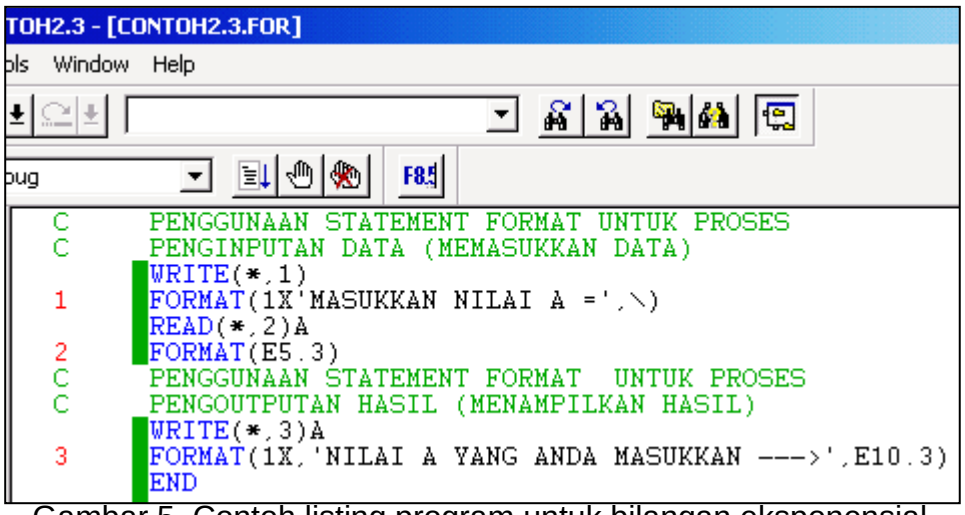
<w> Menunjukan panjangnya atau banyaknnya digit seluruh digit data

(harus berupa bilangan integer bukan nol)

<d> Menunjukan digit dibelakang titik desimal

Dapat digunakan untuk input dan output data.

Contoh. 7.



Gambar.5. Contoh listing program untuk bilangan eksponensial

Penjelasan:

Nilai A bertipe eksponensial dengan 3 angka dibelakang koma

Contoh.8.

```
      C23456789
                A = 0.275E+4
                B = 0.125E+4
                C = 0.6125E+4
                Write(*,10)A,B,C
          10    Format(1X,E10.3,E10.3,E12.4)
                End
```

Hasil running program :

```
          0.275E+4  0.125E+4    0.6215E+4
      x|xxxxxxxxxx|xxxxxxxxxx|xxxxxxxxxxxx        {jumlah spasi}
      1       10          10      12
```

d. Edit Descriptor A

Digunakan untuk tipe data alphanumerik atau tipe data karakter

Bentuk umum dari penulisan edit descriptor ini adalah :

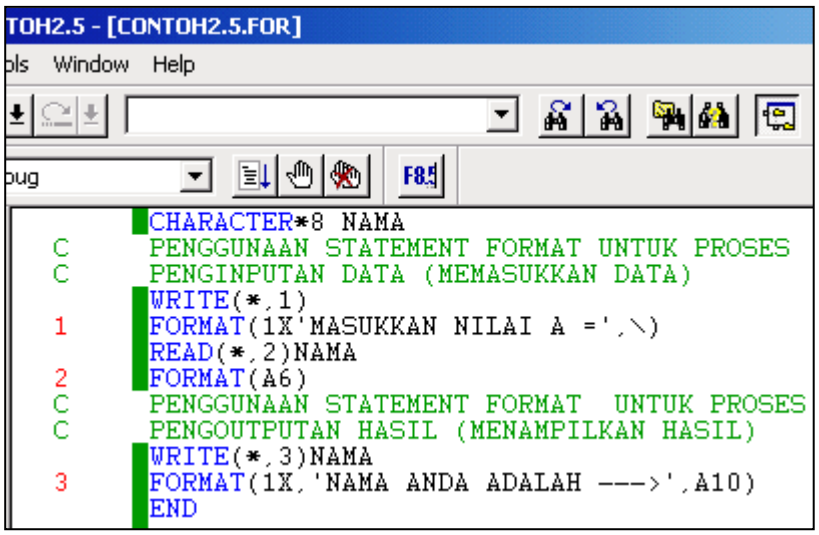
<n>A<w>

Penjelasan :

<n> Menunjukan banyaknya perulangan

<w> menunjukan panjang atau banyaknya digit seluruh digit data (harus berupa bilangan integer tidak nol)

Contoh 9.



Gambar.6.Contoh listing program untuk penulisan data karakter

Contoh.10

```
A = ‘Bahasa Indonesia”
B = ‘Bahasa Inggris’
Write(*,100)A,B
100   Format(1X,A16,1X,A16)
End
```

Hasil Running Program

```
  Bahasa  Indonesia   Bahasa Inggris
x|xxxxxxxxxxxxxxxx|xxxxxxxxxxxxxxxx|          {jumlah spasi}
 1       16                  16
```

3.3. EDIT DISKIPSI TIDAK BERULANG

Merupakan edit descriptor yang tidak dapat diulang dengan suatu nilai tertentu. Yang termasuk kdalam kelompok ini adalah konstanta karakter, X, /, \.

a. Edit descriptor konstanta karakter

   Hanya digunakan untuk menampilkan suatu konstanta karakter ke alat output, dan digunakan sebagai data output yang berhubungan dengan statement write saja dan tidak dapat digunakan untuk statement Read.

   Bentuk umum dari Kelompok ini adalah :

   ‘ Konstanta Katakter ’

   Penjelasan :

   ‘ Konstanta Katakter’ merupakan konstanta karakter bebas yang ditulis diantara tanda petik tunggal.

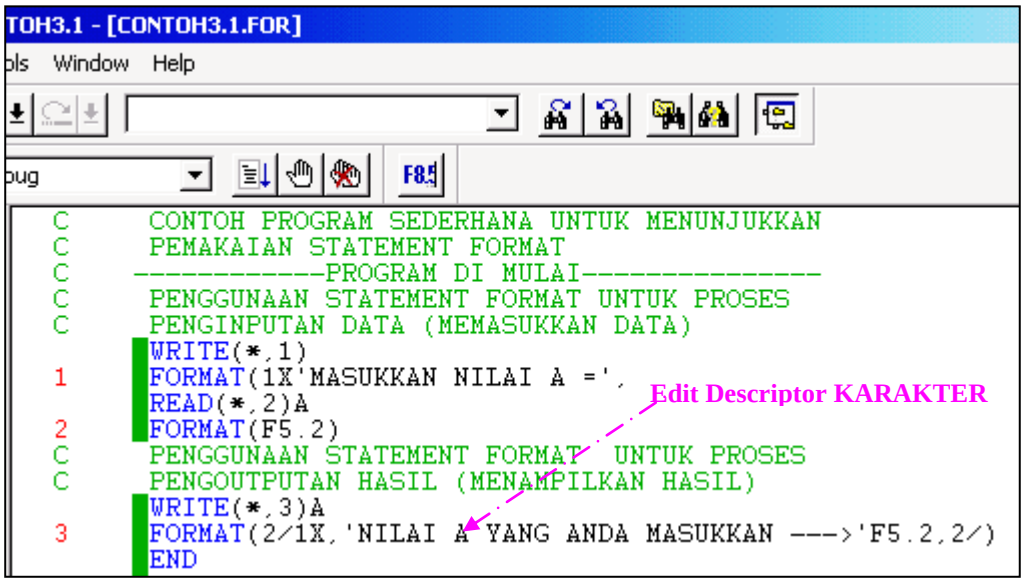
   Contoh.11.

```
C23456789
      Write(*,100)
100   Format(1X,’ Bahasa Fortran‘)
      End
```

   Hasil running program menjadi :

   Bahasa Fortran

Contoh.12.



Gambar.7. Contoh listing program untuk konstanta karakter

Penjelasan:

FORMAT 1 untuk menampilkan perintah memasukkan data nilai A

FORMAT 2 untuk menyimpan nilai A dengan format real F5.2

FORMAT 3 untuk menampilkan nilai A di monitor sesuai format yang perintahkan.

b. Edit Descriptor X

Digunakan untuk mengatur posisi data input dan output dengan memberi jarak sejumlah blank atau spasi tertentu. Edit descriptor ini digunakan untuk statement Read dan Write.

Bentuk umum dari Editor ini adalah :

<n>X

Penjelasan :

<n> menunjukan banyaknya spasi atau blank

Contoh.13.

```
C23456789
          Write(*,100)
100       Format(10X,'Bahasa Fortran',10X, 'Bahasa Fortran')
          End
```

Hasil running program menjadi :

xxxxxxxxx|Bahasa Fortran|xxxxxxxxxx|Bahasa Fortran

9 spasi 10 spasi

Catatan:

Satu spasi diawal hasil running program digunakan sebagai carriage control yaitu pengatur letak pada hasil di monitor atau printer.

c. Edit Descriptor /

Digunakan untuk slash editing, yaitu memberikan jarak spasi penampilan baris dengan baris yang lainnya. Spasi slash berarti memberikan jarak 1 spasi baris, // berarti memberikan 2 spasi baris dan seterusnya.

Contoh 14.

```
C23456789
      Write(*,100)
100   Format(10X,'Bahasa Fortran',/,10X, 'Bahasa Fortran')
      End
```

Hasil running program menjadi :

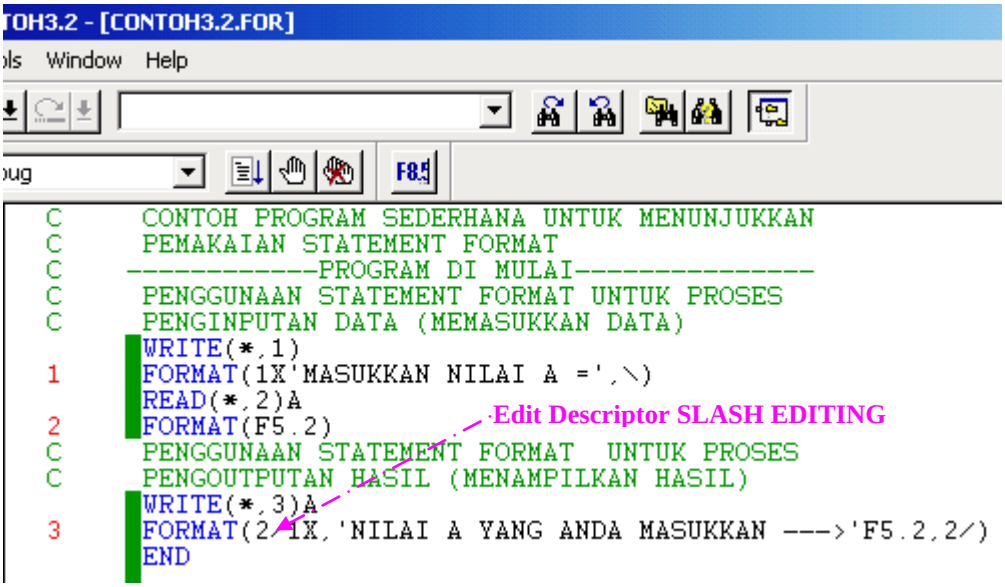
xxxxxxxxx|Bahasa Fortran

9 spasi {satu spasi sebagai carriage control}

xxxxxxxxxx|Bahasa Fortran

10 spasi

Contoh 14.



Gambar.8. Contoh listing program untuk format slash

Penjelasan :

Dalam FORMAT 3, 2/ berarti program akan menampilkan hasil terlebih dahulu turun baris sebanyak 2 baris dan setelah menampilkan hasil nilai A kembali turun 2 baris

d. Edit Descriptor \

Digunakan untuk menampilkan sesuatu tampilan di layar dan diikuti dengan pemasukan data pada baris yang sama, sehingga pemasukan data lebih interaktif.

Contoh 15.

```
Write(*,'(1x,A,\)')'Nilai A = '
Read(*,*)A
Write(*,'(1x,A,\)')'Nilai B = '
Read(*,*)B
Write(*,'(1x,A,\)')'Nilai C = '
Read(*,*)C
```

Hasil running program menjadi :

Nilai A = 5

Nilai B = 2

Nilai C = 4

Jika tanpa edit dscriptor maka akan menjadi :

```
 Write(*,'(1x,A)')'Nilai A = '
 Read(*,*)A
 Write(*,'(1x,A)')'Nilai B = '
 Read(*,*)B
 Write(*,'(1x,A)')'Nilai C = '
 Read(*,*)C
```

Hasil running program menjadi :

```
Nilai A =
5
Nilai B =
2
Nilai C =
4
```

3.4. RANGKUMAN

- Statement FORMAT digunakan untuk mengedit data pada alat-alat I/O pada pembuatan sebuah program komputer dengan menggunakan edit descriptor.
- Edit Descriptor menunjukan informasi mengenai letak, tipe dan panjang dari masing-masing data yang akan dimasukan lewat alat input atau yang akan ditampilkan di alat output.

3.5. LATIHAN

1. C23456789

```
        write (*,10)
10      format (’Bahasa Fortran’)
        End
```

Apa arti list program di atas dan bagaimana hasil tampilan program tersebut.

2. $FREEFORM {bentuk bebas penulisan tidak terikat nomor kolom}

```
A = 15.00
B = 25.00
C = A + B
Write (*,10) C
10 FORMAT (10X,’NILAI C ADALAH’, 2X , F5.2)
END
```

Bagaimanakah tampilan potongan program di atas?

3. c23456789

```
DATA1 = 4323.2
DATA2 = 675.421
DATA3 =  67.789
WRITE (*,8) DATA1, DATA2, DATA3
```

Tuliskan isi format 8 sehingga semua data dapat ditampilkan dengan spasi 2 antar data.

4. c23456789

```
NILAI1 = 234
NILAI2 = 5201
NILAI3 = 246
WRITE (*,10) NILAI1, NILAI2, NILAI3
```

Tuliskan isi format 10 sehingga semua data dapat ditampilkan sebagai

berikut:

00234_5201_0246

5. c23456789

```
A = 0.01E+3
B = 0.508E+2
C = 0.0115E+5
WRITE (*,12) A,B,C
```

Tuliskan isi format 12 sehingga semua data dapat ditampilkan dengan spasi 1 antar A - B dan spasi 2 antar B – C

6. c23456789

```
CHARACTER NAMA1*7, NAMA2*6
NAMA1 = ‘Pemrograman’
NAMA2 = ‘Komputer’
WRITE (*,1) NAMA1,NAMA2
1   FORMAT(1X,2(A8))
        END
```

Bagaimana tampilan program di atas jika program dijalankan (running program)

7. Buat program untuk menghitung :
   a. Luas sebuah lingkaran.
   b. Isi tabung
   c. Luas segitiga
   d. Luas trapesium

   Data masukan bisa anda definisikan sendiri.

BAB IV
STATEMENT INPUT DAN OUTPUT

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menjelaskan fungsi statement Input dan Output dalam program fortran
2. Menggunakan statement Input dan Output dalam pembuatan program komputer.

4.1. PENDAHULUAN

Statement Read dan Write digunakan untuk data transfer yaitu pengiriman data dari alat input atau pengiriman data ke alat output. Bentuk lain dari input/output yang lain seperti : Backspace, Open, Close, Endfile dan Rewind.

4.2. STATEMENT READ

Digunakan untuk mentransfer atau membaca data dari suatu file yang dapat berupa file di disk, di printer atau consule (layar dan keyboard).

Bentuk umum dari Statement Read adalah :

Read (<no_unit>,<no_label_format>,Rec = <no_rec>, End = <slabel1>, Err = <slabel2>) I/Olist

Penjelasan :

No_unit	adalah unit spesifier dari alat input yang akan dipergunakan
No_label_format	format spesifier dari data yang akan dibaca yang menunjukan tipe, letak dan panjang dari data
No_rec	Hanya digunakan untuk sistem pembacaan data dari file di disk secara dirct acces
slabel1	statement label yang pertama yang digunakan hanya

	untuk pembacaan file dari disk
slabel2	stetement label ke dua yang digunakan untuk mendeteksi bila ada kesalahan pembacaan data
I/O list	Input dan Output list yang merupakan kumpulan atau sebuah variael, nama larik atau elemen dari larik yang ditulis dengan pemisahan koma

Penulisan nomor_unit dan nomor_format adalah secara berurutan dan tidak boleh terbalik. Sedangkan penulisan lainnya tidak spesifik ditentukan secara berurut. Penulisan statement Read untuk penggunaan yang berhubungan dengan pembacaan data dari alat input keyboad mempunyai bentuk umum sebagai berikut:

Read(<no_unit>,<no_label_format>) I/O list

Contoh :

	Read(*,*) A, B, D	Berarti baca nilai A, B dan D dari keyboard dan menggunakan format bebas
	Read(*,5) A, B, D	Berarti baca nilai A, B dan D dari keyboard dengan nomor_format 5 yaitu : dengan menggeser penulisan 1 spasi dari margin kiri, 2 buah tempat untuk 5 karakter dengan 2 desimal dan tempat untuk 5 karakter bilangan integer
5	format(1X,2F5.2,I5)	

Penulisan nomor_label_format dapat dihilangkan penulisannya dan diganti dengan ungkapan karakter yang langsung menunjukan tipe, letak dan panjang data. Penulisan seperti ini akan menghilangkan penggunaan format.

Contoh :

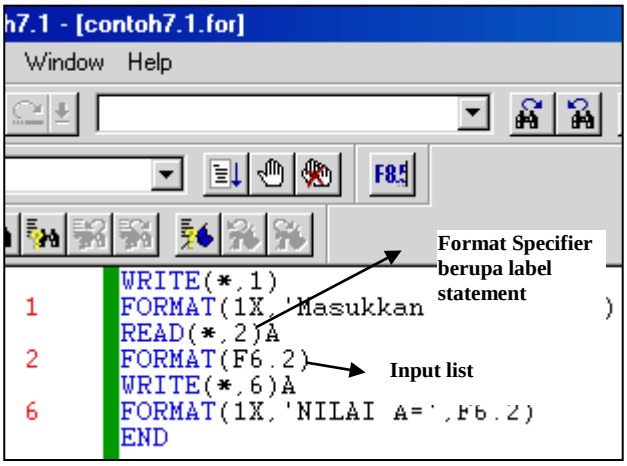
Read (*,’(1X,2F5.2,I5)’) A, B, D

Pembacaan data dengan menggunakan Read dapat digunakan untuk pembacaan dengan fungsi do list atau perulangan seperti yang diperlihatkan pada contoh dibawah :

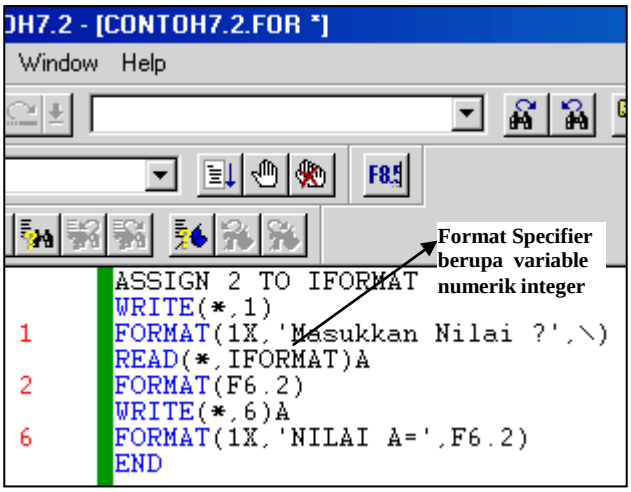
Integer*2 X(3)	Integerkan nilai X dan 3 buah elemennya
Read(*,'(I5,2(1X,i5))')(X(I), I=1,3)	Baca nilai X(1), X(2) dan X(3) dengan format masing-masing 5 karakter bilangan integer

Contoh penggunaan READ:

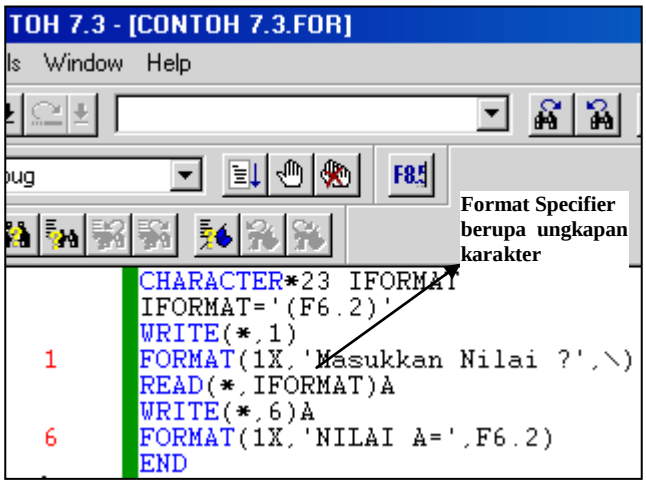
- Format specifier berupa label statement



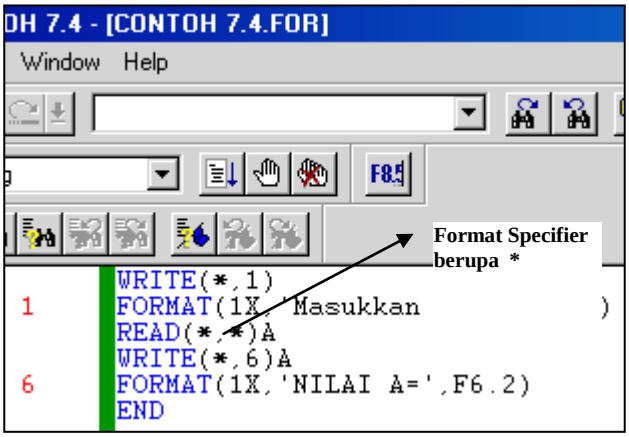
- Format specifier berupa variable numerik integer



- Format specifier berupa ungkapan karakter



- Format specifier berupa *



4.3. STATEMENT WRITE

Digunakan untuk mentransfer atau menampilkan data dari I/O list ke suatu file di disk, printer atau consule yang ditunjukan oleh no_unit.

Bentuk umum dari statement Write adalah

Write(<no_unit>,<no_label_format>,Rec = <no_rec>, Err = <slabel>) I/O list

Penjelasan :

No_unit	adalah unit spesifier dari alat output yang akan dipergunakan
No_label_format	format spesifier dari data yang akan ditulis / ditransfer yang menunjukan tipe, letak dan panjang dari data
no_rec	Hanya digunakan untuk sistem penulisan/transfer data

	dari file di disk secara dirct acces
Slabel	stetement label ke dua yang digunakan untuk mendeteksi bila ada kesalahan penulisan/transfer data
I/O list	Input dan Output list yang merupakan kumpulan atau sebuah variabel, nama larik atau elemen dari larik yang ditulis dengan pemisahan koma menunjukan data yang akan ditampilkan / ditransfer

Contoh :

```
C23456789
      A = 5.2645
      B = 445.3665
      C = 1135.254
      Write(*,5) A, B, C
  5    Format(1x,‘Nilai A = ’,F7.2,1x,’Nilai B = ‘,F7.2,1x,’Nilai C =’,
      1F7.2)
       End
```

Hasil running program diatas menjadi :

```
      Nilai A =    5.27 Nilai B =  445.37 Nilai C = 1135.25
```

Jika Output List berupa kumpulan Variabel dan konstanta karakter, maka bentuk penulisan dari write dapat ditulis sebagi berikut :

```
C23456789
      A = 5.2645
      B = 445.3665
      C = 1135.254
      Write(*,5)’Nilai A =’,A, ’Nilai B =’,B, ’Nilai C =’,C
  5    Format(3(1x,A10,F7.2)
      End
```

Hasil running program diatas menjadi :

Nilai A = 5.27 Nilai B = 445.37 Nilai C = 1135.25

Jika format spesifier berupa ungkapan karakter yang langsung menunjukan tipe, letak dan panjang dari data sehingga nomor format tidak lagi digunakan seperti diperlihatkan pada contoh berikut :

```
C23456789
      A = 5.2645
      B = 445.3665
      C = 1135.254
      Write(*,'(1x,''Nilai A = '',F7.2,1x,''Nilai B = '',F7.2,1x,
     1''Nilai C = '',F7.2)')A,B,C
      End
```

Hasil running program diatas menjadi :

Nilai A = 5.27 Nilai B = 445.37 Nilai C = 1135.25

Bentuk lain dapat ditulis sebagai berikut :

```
C23456789
      A = 5.2645
      B = 445.3665
      C = 1135.254
      Write(*,' (3(1x,A10,F7.2)')'Nilai A = ',A,'Nilai B = ',B,'Nilai C = ',C
      End
```

Hasil running program diatas menjadi :

Nilai A = 5.27 Nilai B = 445.37 Nilai C = 1135.25

Edit deskriptor A dapat juga tanpa disebutkan panjang dari karakternya sehingga dapat ditulis sebagai berikut:

```
C2345678
      Write(*,’ (3(1x,A,F7.2)’)’Nilai A = ’,A,’Nilai B = ’,B,’Nilai C = ’,C
       End
```

Contoh penggunaan dalam program :

1. Program menghitung Nilai pangkat dari suatu bilangan.

```
$freeform
Integer*2 X
real X1,X2
Write(*,*)'Tabel Menghitung Nilai pangkat  '
Write(*,*)'================================'
Write(*,*)'  X     X Kuadrat   X Pangkat 3 '
Write(*,*)'================================'
do 10 i = 1,15
x=i
x1=x**2
x2=x**3
Write(*,'(i3,2(4x,F10.3))')x,x1,x2
10 continue
Write(*,*)'================================'
end
```

Hasil running program diatas menjadi :

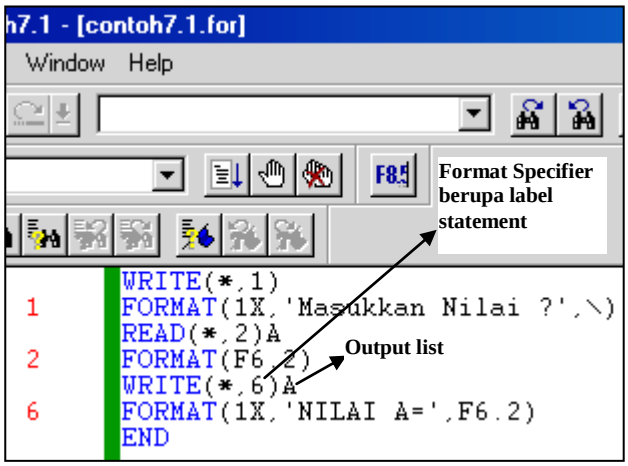
```
Tabel Menghitung Nilai pangkat
    ================================
     X     X Kuadrat   X Pangkat 3
    ================================
     1        1.000        1.000
     2        4.000        8.000
     3        9.000       27.000
     4       16.000       64.000
     5       25.000      125.000
```

6	36.000	216.000
7	49.000	343.000
8	64.000	512.000
9	81.000	729.000
10	100.000	1000.000
11	121.000	1331.000
12	144.000	1728.000
13	169.000	2197.000
14	196.000	2744.000
15	225.000	3375.000

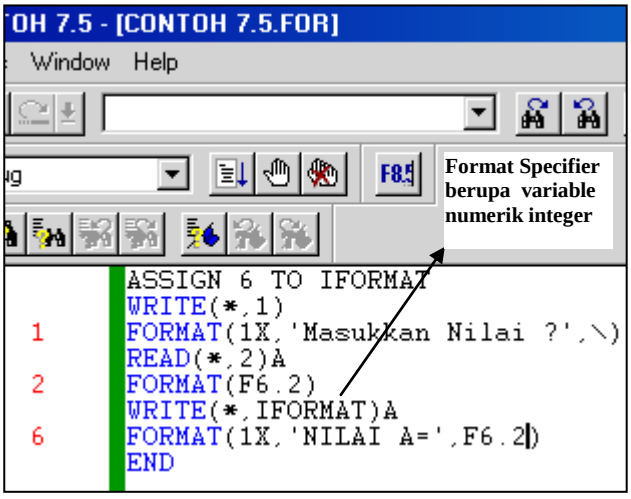
==========

Contoh Penggunaan WRITE

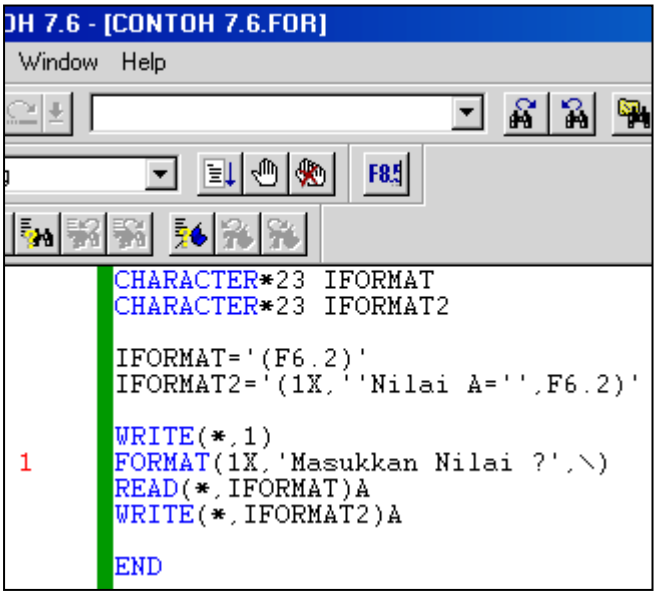
- Format specifier berupa label statement



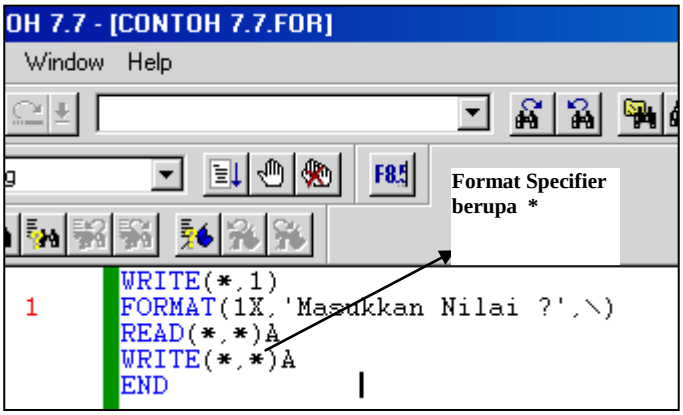
- Format specifier berupa variable numerik integer



- Format specifier berupa ungkapan karakter



- Format specifier berupa *



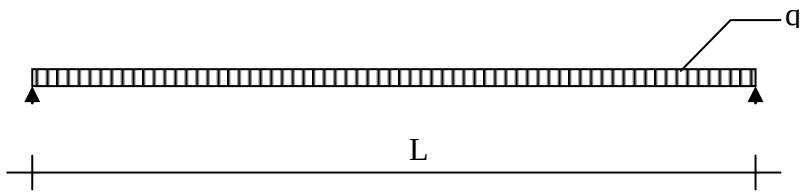
4.4. RANGKUMAN

- Statement Read digunakan untuk mentransfer atau membaca data dari suatu file yang dapat berupa file di disk, di printer atau consule (layar dan keyboard).
- Statement Write digunakan untuk mentransfer atau menampilkan data dari I/O list ke suatu file di disk, printer atau consule yang ditunjukan oleh no_unit.

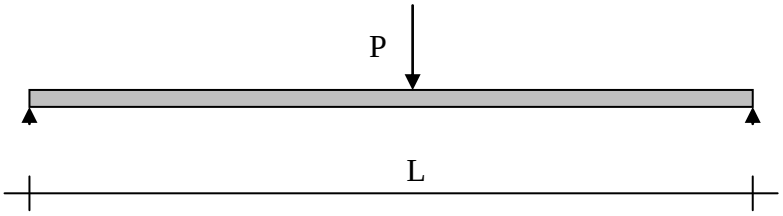
4.5. LATIHAN

Buat program untuk menghitung reaksi perletakan dan bidang gaya dari bentuk konstruksi berikut :

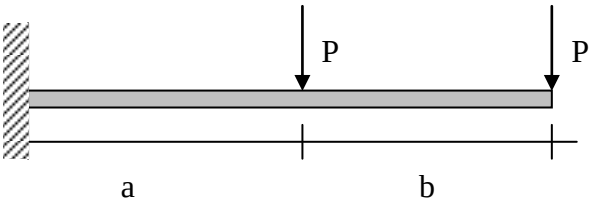
a.



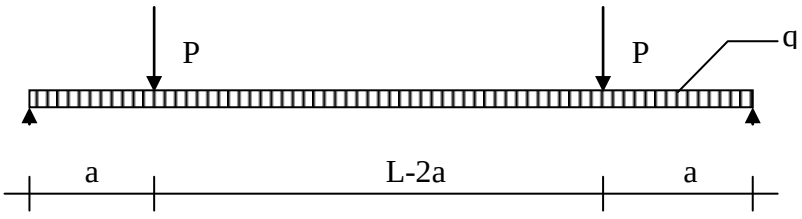
b.



c.



d.



BAB V
STATEMENT SPESIFIKASI

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menjelaskan fungsi statement Statement Spesifikasi Dimension (Array), Implicit dan Type dalam program fortran
2. Menggunakan statement spesifikasi Dimension (Array), Implicit dan Type dalam pembuatan program komputer.

5.1. STATEMENT DIMENSION

Statement dimension merupakan statement spesifik yang menunjukan fungsi larik atau dimensi dari suatu data yang mempunyai beberapa elemen. Umumnya statement ini digunakan bersamaan dengan fungsi do list atau perulangan.

Kegunaan dari statement ini adalah untuk mendefinisikan suatu nama yang dibentuk oleh pembuat program yang merupakan suatu larik serta sekaligus menentukan jumlah elemn-elemennya.

Bentuk umum dari statement ini adalah :

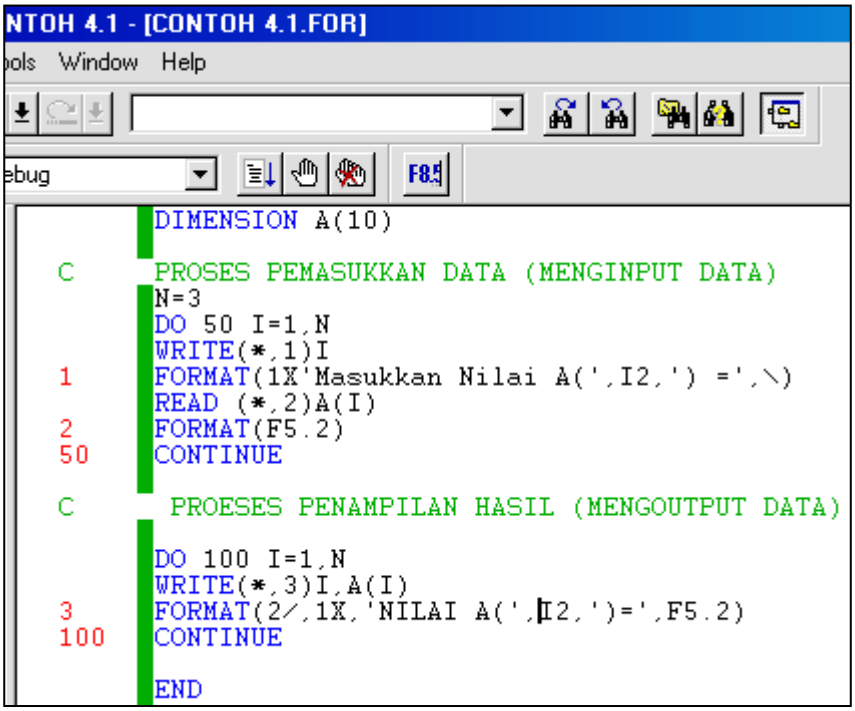
Dimension <array>(<dim>), <array>(<dim>)

Penjelasan :

<Array>	Nama dari suatu larik yang didefinisikan
<dim>	Deklarasi dimensi yang menunjukan jumlah elemen dari larik

Suatu larik dapat mempunyai dimensi sampai 7 buah dimensi dan maksimum ukuran seluruh larik adalah 64 kb.

Contoh 1.



Penjelasan:

Dimension A (10) menunjukkan larik A berdimensi satu dengan kapasitas memori untuk A sebesar 2*10 = 20 byte.

Contoh 2.

Integer*2 B, C

Dimension A(100,50), B(10,10), C(50)

Penjelasan:

Menunjukan bahwa A adalah larik yang berdimensi dua dengan kapasitas memori sebesar 2*100*50 = 10000 byte, B dengan kapasitas 2*10*10 = 200 byte serta C mempunyai kapasitas sebesar 2*50 = 100 byte.

Jika pendimensian yang digunakan dalam program mempunyai kapasitas lbih dari 64 kb, maka akan muncul error yang akan menjelaskan bahwa kapasitas maksimum untuk pendimensian adalah sebesar 64 kb.

Deklarasi dimensi dari suatu larik menunjukan jumlah elemen dari larik tersebut dan juga merupakan batas maksimum dari suatu dimensi dengan batas bawahnya adalah nol.

Dari contoh diatas dapat dijelaskan bahwa : batas maksimum dari elemen pada larik A adalah 10*50 = 500 buah. Begitu juga dengan larik B yang mempunyai batas maksimum dari jumlah elemen adalah sebesar 100 buah.
Penulisan batas maksimum ini harus merupakan batas dari penulisan di list program. Jika penulisan di list program lebih besar dari batas maksimum maka akan muncul error yang menjelaskan bahwa pendimensian lebih besar dari batas yang tersedia.

Contoh 3.

```
C23456789
      Dimension A(25)
      Total = 0
      Do 10 I = 1, 26, step 1
      C = A(i)*6+5
      Total = Total + C
10    Continue
      Write(*,'(1x,A,F10.3)')'Nilai Total Data = ',Total
      End
```

Hasil running program akan menunjukan terjadinya error dimana nilai A(26) overflow atau melebihi. Hal ini disebabkan oleh kapasistas dari A melebihi dari batas maksimum dari jumlah elemen A.

Contoh 4.

Tentukan total nilai dan standar deviasi dari data-data statistik berikut :

Data Ke	Nilai Data
1	25.6
2	44.9
3	32.6
4	55.8
5	23.8
6	100.6

7	26.58
8	44.7
9	89.5
10	36.2

Persamaan dari Standar deviasi dtunjukan oleh persamaan berikut :

$$SD = \sqrt{\frac{\left(\sum_{i=1}^{n} X_i - \overline{X}\right)}{n}}$$

dengan : Xi = data ke i

$\overline{X}$ = rata-rata nilai X

n = jumlah data

SD= Standar deviasi

List program dapat dilihat seperti di bawah ini

```
$freeform
C Program menghitung nilai Rerata dan Standar Deviasi
      Dimension Data(11)
      Real Data, Total, Sigma, Rerata, Sd
      Write(*,*)'Masukan Data dan Menghitung Nilai Total'
      Total = 0
      Do 10 I = 1, 10
      Write(*,'(1x,A,I2,A,\)')'Data Ke (',I,') = '
      Read(*,*)Data(i)
      Total = Total + Data(i)
   10 Continue
      Rerata = Total/10
      Sigma=0
      do 20 i = 1,10
   20 Sigma=Sigma+(Data(i)-Rerata)**2
      Write(*,*)'Menghitung Standar Deviasi'
      Sd=(Sigma/10)**0.5
      Write(*,'(1x,A,F10.2)')'Total Data      = ',Total
```

```
      Write(*,'(1x,A,F10.2)')'Rerata Data     = ',Rerata
      Write(*,'(1x,A,F10.2)')'Standar Deviasi = ',Sd
      End
```

Hasil Running Program adalah sbb:

Data Masukan

Data Ke	Nilai Data
1	25.60
2	44.90
3	32.60
4	55.80
5	23.80
6	100.60
7	26.58
8	44.70
9	89.50
10	36.20

Menghitung Standar Deviasi

Total Data = 480.28

Rerata Data = 48.03

Standar Deviasi = 25.48

5.2. STATEMENT IMPLICIT

Bentuk umum :

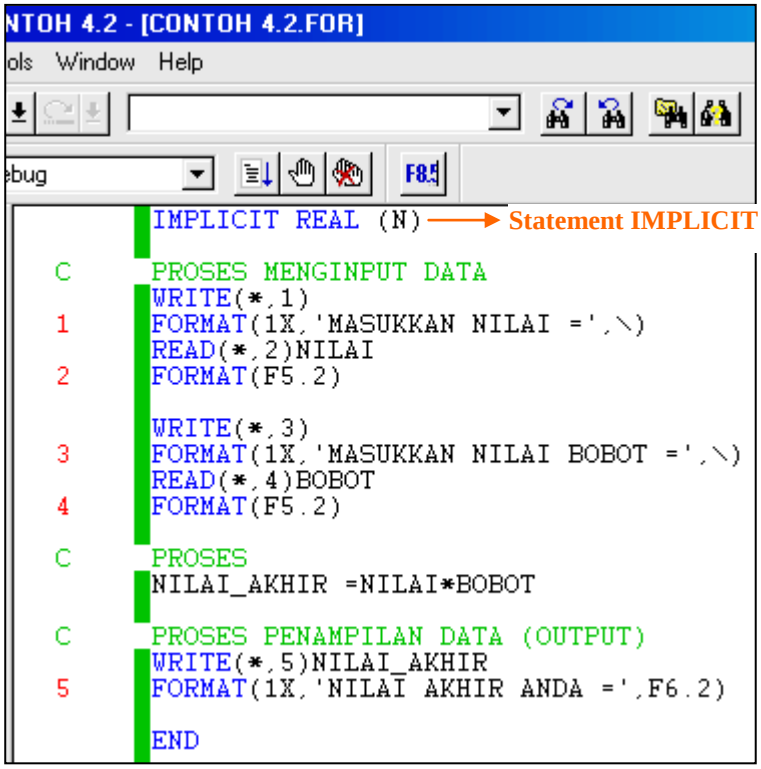
IMPLICIT <type>(<a>[.<a>]…)[,<type(<a>[,<a>],…)…]

Penjelasan :
<type> adalah salah satu dari tipe:
INTEGER
INTEGER *2
INTEGER *4
REAL
REAL *4
REAL *8
DOUBLE PRECISION
LOGICAL

LOGICAL *2
LOGICAL *4
CHARACTER

<a> adalah salah satu huruf atau range dari huruf bila berbentuk
Range dari huruf harus dari urutan kecil ke besar.

Contoh.



5.3. STATEMENT TYPE

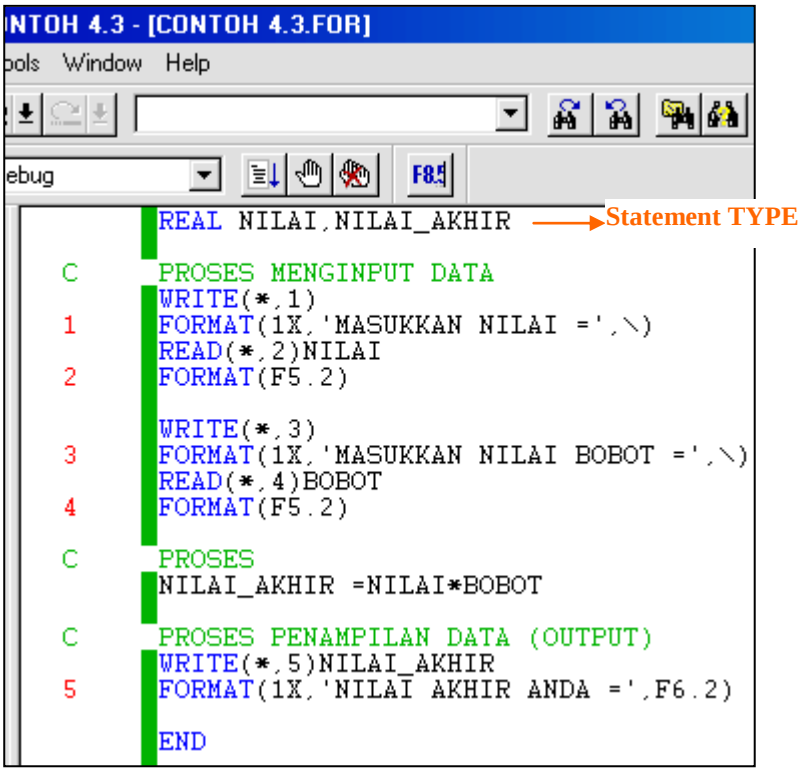
Bentuk umum :

<type><name> [,<name>] …

Penjelasan :
<type> adalah salah satu dari tipe:
INTEGER
INTEGER *2
INTEGER *4
REAL
REAL *4
REAL *8
DOUBLE PRECISION
LOGICAL
LOGICAL *2
LOGICAL *4
CHARACTER

<name> adalah nama variable,nama larik, nama fungsi statement
atau nama fungsi eksternal.

Contoh



5.4. RANGKUMAN

Statement Spesifikasi Dimension dapat digunakan untuk menyatakan larik atau jumlah dimensi suatu variabel.
Statement Implicit dan Type digunakan untuk menyatakan bentuk dari tipe dari variabel.

5.5. LATIHAN

1. Buat program untuk menghitung penjumlahan matriks dari 2 Matriks A dan B.
2. Buat program untuk menghitung perkalian dua matrik dengan orde yang berbeda

3. Buatlah program komputer untuk menghitung rata rata dari 6 buah data A dan rata-rata dari 3 buah data B sebagai berikut:

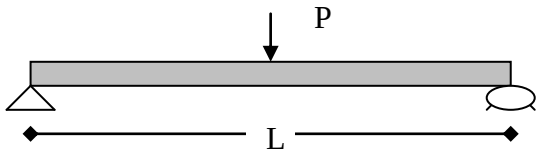
   A(1) = 2.3 A(4) = 3.6 B(1) = 10

   A(2) = 4.2 A(5) = 5.8 B(2) = 15

   A(3) = 3.5 A(6) = 7.7 B(3) = 17

   Gunakan Statement DIMENSION.

4. Buatlah program komputer menggunakan DIMENSION untuk menghitung momen maksimum akibat beban P dari struktur berikut:



   DATA :

P(i)	L(i)
5 ton	12 meter
7 ton	12 meter
8 ton	12 meter
10 ton	12 meter

5. Buatlah program komputer untuk menampilkan nama, nilai dan nilai rata-rata ujian akhir semester (UAS) dari 5 orang mahasiswa dengan data sebagai berikut:

NO	NAMA MAHASISWA	NILAI UAS
1	Sridewi	90.5
2	Dwiguna	70.7
3	Wicaksono	84.0
4	Ananta	67.8
5	Wawan	87.0

BAB VI
STATEMENT KONTROL

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menjelaskan fungsi statement Kontrol dalam program fortran
2. Menggunakan statement Kontrol dalam pembuatan program komputer.

6.1. PENDAHULUAN

Pada bab ini akan dibahas 2 bagian dari statement kontrol yaitu statement loncatan (goto) dan statement untuk menyeleksi suatu kondisi (If condition).

6.2. STATEMENT GOTO

Statement ini merupakan statement loncatan yaitu digunakan untuk meloncat statement ke suatu statement lainnya yang tertentu. Statement ini dapat digolongkan menjadi :

a. Statement Goto tak bersyarat (Unconditional Goto)

   Digunakan untuk mengontrol proses menuju k suatu statement lainnya yang ditunjukan oleh <slael> tanpaa ada syarat yang diberikan.

   Bentuk Umum dari statement ini adalah :

   Goto <slabel>

   Penjelasan :

   <slabl> Statement label dari sutau executable statement yang harus berada pada satu unit program yang sama dengan statement Goto bersangkutan.

   Contoh Penggunaan :

```
C23456789
      5 Write(*,'(A)')'Statement Goto'
      Goto 5
      End
```

Hasil Running Program adalah sbb:

Statement Goto
Statement Goto
Statement Goto
Statement Goto
Statement Goto
Statement Goto
Statement Goto
Statement Goto^C Tanda C menunjukan pemutusan perintah atau dengan Ctrl + C

Tampak bahwa perintah penulisan di atas dilakukan secara berulang-ulang tanpa henti, hal ini disebabkan karena begitu program selesai melaksanakan eksekusi maka perintah goto akan memerintahkan untuk meloncat ke statement write. Begitu seterusnya.
Program diatas dapat dilakukan pembatasan dalam hal melakukan perintah penulisan yaitu dengan menggunakan perintah perulangan.

Contoh

```
C2345678
      5  Write(*,'(A)')'Statement Goto'
         i=i+1
         if(i.eq.10)stop
         Goto 5
         End
```

Hasil Running Program adalah sbb:

Statement Goto
Statement Goto
Statement Goto
Statement Goto
Statement Goto
Statement Goto

Statement Goto
Statement Goto
Statement Goto
Statement Goto

b. Statement Goto pengerjaan (assigned Goto)

Digunakan untuk meloncat ke suatu label statement <slabel> yang ditunjukan oleh isi dari <name>.

Bentuk umum dari statement Goto diperlihatkan sbb. :

Goto <name>,<slabel>

Penjelasan :

<name> nama dari variabel integer yang diisi label statement dengan perintah assign

Jadi salah satu nilai dari <slael> harus sama dengan nilai dari <name>. Kalau nilai <name> tidak sama dari salah satu nilai <slabl., maka proses tidak akan meloncat ke suatu label statement apapun, tetapi akan dilanjutkan ke statement berikutnya.

Contoh :

```
C23456789
      Assign 15 to loncat
      Goto Loncat, (15)
      Write(*,'(1x,A)')'Tidak Ditampilkan'
   15 Write(*,'(1x,A)')'Ditampilkan'
      Write(*,'(1x,A)')'Ini Juga Ditampilkan'
      End
```

Hasil dari Running Program

```
Ditampilkan
Ini Juga Ditampilkan
```

c. Statement Goto bersyarat atau Goto terhitung (Computed Goto)

Digunakan untuk mengontrol loncatan dari proses ke suatu labl statement <slabl> tertentu, tergantung dari nilai ungkapan integer <i>.

Bentuk umumdari statement Goto diperlihatkan sbb:

Goto (<slabel>,<slabel>), <i>

Penjelasanan :

<slabel> merupakan statement dari suatu statemnt eksekusi yang berada pada unit program yang sama dengan statement Computed Goto tersebut.

<i> merupakan ungkapan integer

Statement Computed Goto akan menuju k <slabel> yang pertama bila ungkapan <i> bernilai I, akan menuju ke <slabel> kedua bila ungkapan <i> bernilai 2 dan selanjutnya.

Contoh :

Program menghitung Isi dan Luas dari bangun silinder, Segitiga dan Kubus.

```
C2345678
C Program menghitung Luas dan Isi dari Banguna
C Segitiga, Silinder dan Kubus
C ================================================
      Character*1 Cetak
      Real Isi,Luas, Tinggi, Jari,Sisi
      Write(*,'(1x,A)')'Pilih Jenis Bangun'
      Write(*,*)
      Write(*,'(1x,A)')'1. Menghitung Isi Silinder'
      Write(*,'(1x,A)')'2. Menghitung Luas Segitiga'
      Write(*,'(1x,A)')'3. Menghitung Isi Kubus'
      Write(*,*)
      Write(*,'(1x,A)')'4. Selesai'
   15 Write(*,'(/,1x,A,\)')'Pilihan Nomor (1/2/3/4) ? '
      Read(*,*)nomor
```

```
C Fungsi Goto
      Goto(1000,2000,3000,4000)Nomor
      Cetak = 5
      Write(*,'(1x,A,A)')'Salah Pilih  !!!!!', Cetak
      GOto 15

C Menghitung Isi Silinder
 1000 Write(*,'(1x,A,\)')'Jari-Jari Lingkaran Dasar = '
      Read (*,*)jari
      Write(*,'(1x,A,\)')'Tinggi SIlinder          = '
      Read (*,*)Tinggi
      Isi = 3.14*jari**2*tinggi
      Write(*,'(1x,A,F6.2)')'Isi Silinder          = ',Isi
      Goto 4000

C Menghitung Luas Segitiga
 2000 Write(*,'(1x,A,\)')'Panjang Alas             = '
      Read (*,*)Alas
      Write(*,'(1x,A,\)')'Tinggi                   = '
      Read (*,*)Tinggi
      Luas = 0.5*alas*tinggi
      Write(*,'(1x,A,F6.2)')'Luas Segitiga         = ',Luas
      Goto 4000

C Menghitung Isi Kubus
 3000 Write(*,'(1x,A,\)')'Panjang Sisi             = '
      Read (*,*)Sisi
      Isi = Sisi**3
      Write(*,'(1x,A,F6.2)')'Isi Kuus              = ',Isi
      Goto 4000

C Selesai
```

```
4000  Continue
      End
```

Running Progam menjadi

Pilih Jenis Bangun

1. Menghitung Isi Silinder
2. Menghitung Luas Segitiga
3. Menghitung Isi Kubus

4. Selesai

Pilihan Nomor (1/2/3/4) ? 1

Jika nomor 1 yang kita pilih, maka akan muncul :

Jari-Jari Lingkaran Dasar = 5

Tinggi SIlinder = 2

Isi Silinder = 157.00

6.3. STATEMENT IF

Statement If merupakan statement yang digunakan untuk menyeleksi suatu kondisi atau syarat dan proses akan melakukan suatu tindakan tertentu bila kondisi yang diseleksi benar dan akan melakukan tindakan yang lainnya bila yang diseleksi salah atau tidak benar.

Statement ini digolongkan menjadi :

a. Statement If Logika

   Digunkan untuk menyeleksi suatu ungkapan logika atau ungkapan hubungan dan jika kondisinya benar (.true.) maka Statement yang mengikutinya akan diproses. Sebaliknya bila ungkapan logika atau ungkapan hubungan yang diseleksi tersebut salah (.false.), maka proses akan dilanjutkan ke statement berikutnya.

   Bentuk umum Statement adalah :

If(<expression>)<statement>

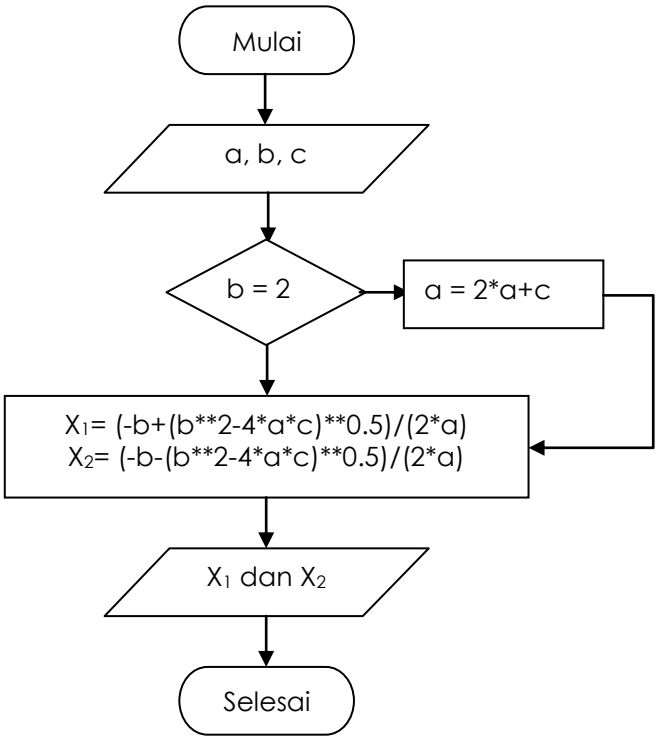
Penjelasan :

<expression> Ungkapan logika atau ungkapan hubungan yang akan diseleksi

<statement> Merupakan executable statement, kecuali statement do, block if, elseif, else, endif, end atau statement if logika yang lainnya

Contoh :

Progam menghitung nilai X1 dan X2 dari persamaan ABC dengan batasan jika nilai B=2 maka nilai A = 2A+C.



Dalam bentuk List program menjadi :

C23456789

C Program menghitung Nilai X1 dan X2 dari Persamaan ABC

```
C Diprogram Oleh Mudjiatko
C ================================================
      Real A,B,C, X1,X2
      Write(*,'(1x,A,\)')'Nilai A = '
      Read(*,*)A
      Write(*,'(1x,A,\)')'Nilai B = '
      Read(*,*)B
      Write(*,'(1x,A,\)')'Nilai C = '
      Read(*,*)C
      If(B.eq.2) A=2*A+C
      X1=(-B+(B**2-4*A*C)**0.5)/(2*A)
      X2=(-B-(B**2-4*A*C)**0.5)/(2*A)
      Write(*,'(1x,A,F5.2)')'Nilai X1 = ',X1
      Write(*,'(1x,A,F5.2)')'Nilai X2 = ',X2
      end
```

Hasil Running Program didapat :

```
Nilai A = 2
Nilai B = 5
Nilai C = 3
Nilai X1 = -1.00
Nilai X2 = -1.50
Dan jika dengan b = 2
Nilai A = 2
Nilai B = 2
Nilai C = 5
Nilai X1 = -1.45
Nilai X2 =  3.45
```

b. Statement If Aritmetika

Digunakan untuk menyeleksi kondisi dari suatu ungkapan aritmetika, apakah bernilai negatif, nol atau positif.

Bentuk umum dari statement ini adalah :

If(<expression>)<slabel1>,<slabel2>,<slabel3>

Penjelasan :

<expression> Ungkapan aritmetika real atau integer

<slabel> label statement dari suatu executable statement yang harus berada pada unit program yang sama dengan statement IF aritmetika yang bersangkutan.

Bila nilai ungkapan aritmetika yang diseleksi bernilai negetif, maka program akan menuju ke <slabel1>, bila nol maka akan menuju <slabel2> dan bila positif akan menuju <slabel3>.

c. Statement If Blok

Digunakan untuk menyeleksi suatu kondisi dan mengambil tindakan apa yang harus dilakukan dalam bentuk blok-blok statement.

Dapat terdiri dari statement IF – Then, Else, Elseif dan Endif. Bentuk umum dari statement ini adalah :

IF (<Ungkapan>) Then

Elseif(<Ungkapan>) Then

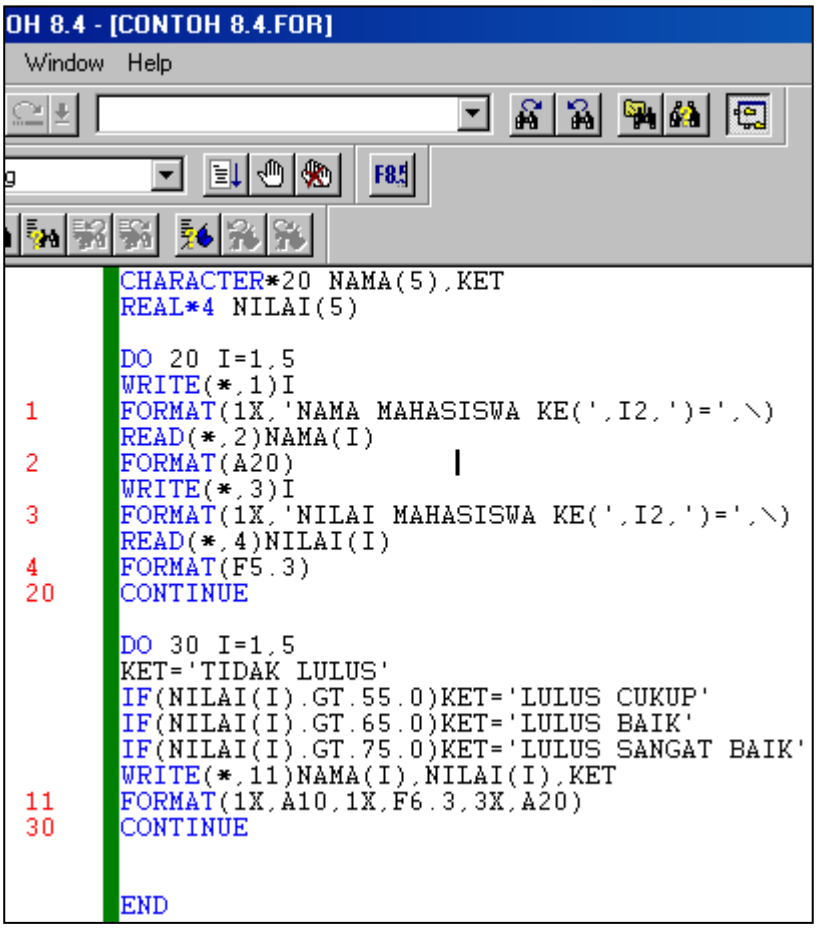
Else

Endif

Penjelasan :

(<Ungkapan>) Ungakapan logika atau ungkapan hubungan

Contoh-contoh Penggunaan Statement IF



6.4. RANGKUMAN

- Statement GOTO merupakan statement loncatan yaitu digunakan untuk meloncatkan statement ke suatu statement lain.
- Statement If merupakan statement yang digunakan untuk menyeleksi suatu kondisi atau syarat dan proses akan melakukan suatu tindakan tertentu bila kondisi yang diseleksi benar dan akan melakukan tindakan yang lainnya bila yang diseleksi salah atau tidak benar

6.5. LATIHAN

Diskusikanlah latihan berikut dalam kelompok.

Buat daftar nilai dari Mahasiswa teknik sipil, jika

nilai A	80 < Nilai ≤ 100	Kriteria “Sangat Memuaskan”
nilai B	65 < Nilai ≤ 80	Kriteria “Memuaskan”
nilai C	50 < Nilai ≤ 65	Kriteria “Kurang Memuaskan”
nilai D	35 < Nilai ≤ 50	Kriteria “Sangat Kurang Memuaskan”
nilai E	Nilai ≤ 35	Kriteria “Gagal”

BAB VII
STATEMENT SUBROUTINE

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menjelaskan fungsi statement Subroutine dalam program fortran
2. Menggunakan statement Subroutine dalam pembuatan program komputer.

7.1. PENDAHULUAN

Unit program dalam bahasa fortran dapat berupa ;

a. Program Utama

   Suatu program dapat berupa sebuah dan hanya sebuah saja program utama. Program utama meruakan program utama yang tidak diawali dengan statement subroutine atau statement function.

b. Program fungsi atau function

   Suatu unit program bagain yang berupa fungsi diawali dengan statement Function dan unit program ini harus berada bersama-sama dengan prjogram utama yang terletak dibawah program utama dan tidak boleh dikompilasi secara terpisah.

c. Program bagian berupa rutin bagian atau subroutine

   Suatu program rutin bagian diawali dengan statement Subroutine yang dapat dipergunakan dan dipanggil oleh program utama dengan statement Call. Program utama dan sekaligus memberikan nama dari program utamanya.

7.2. STATEMENT SUBROUTINE

Guna statement subroutine ini adalah untuk mengidentifikasi bahwa suatu unit program adalah suatu rutin bagian, serta sekaligus memberikan nama dan argumen-argumen.

Bentuk umum penulisan dari subroutine ini adalah :

SubRoutine <subroutine_name>(<name_arg>,<name_arg>,<name_arg>, …)

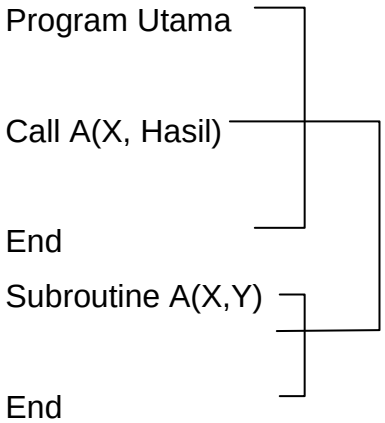
Penjelasan :

<subroutine_name> Nama dari rutin bagian

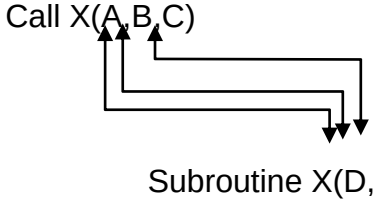
<name_arg> nama argumen atau disebut juga dengan istilah dummy argumen

Beberapa ketentuan dari subroutine :

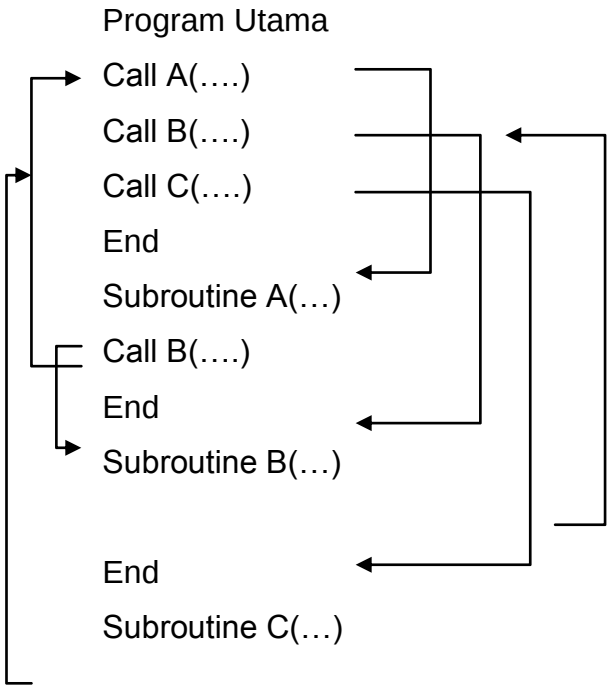
a. Subroutine merupakan unit program tersendiri yang diawali dengan statement Subroutine dan diakhiri dengan statement End atau Return serta dipergunakan oleh unit program yang lain dengan statement Call.



b. Subroutine dapat berisikan statement apaupun kecuali statement Program, statement Subroutine yang lainnya atau statement Fuction
c. Nama argumen tidak boleh tampak di statement common, equvalence, intrinsic atau data.
d. Argumen sesungguhnya yang tampak di statement Call harus sesuai urutannya, jumlahnya dan tipenya dengan dummy argument yang tampak di statement Subroutine. Antara nama argumen sesungguhnya dengan dummy argument noleh sama ataupun boleh tidak sama.



e. Didalam suatu subrootine dapat memeanggil subroutine yang lainnya.



f. Subroutine dapat tidak mengandung dummy argument yang berarti tidak ada data yang dikirim ke subroutine dan tidak ada hasil yang dikirim ke pemanggil subroutine.

Contoh :

```
C23456789
C**************************************************************
C     MAIN PROGRAM
C**************************************************************
      INTEGER A
      CALL MUKA
      CALL DATA
      READ(5,1)A
    1 FORMAT(I1)
      IF(A.EQ.1)CALL PERSEGI
```

```
      IF(A.EQ.2)THEN
      CALL TRAPESIUM
      ELSE
      CALL PENUTUP
      ENDIF
      END
C*****************************************************************
C     SUBROUTINE MUKA
C*****************************************************************
      SUBROUTINE MUKA
      CHARACTER READY*12
      WRITE(*,1)
    1 FORMAT(2x,'PROGRAM MENGHITUNG KARAKTERISTIK
     1ALIRAN DARI PENAMPANG SUNGAI ',/,2X,'DENGAN'
     2PERSAMAAN MANNING ',/,2X,'1.PERSEGI',
     3/2X,'2.TRAPPESIUM',/,2X,'3.SEGITIGA',/,2X,'4.HELP')
      WRITE(*,'(2X,A,\)')'DATA ANDA SUDAH SIAP (YES/NO/HELP) ? : '
      READ(*,'(A5)')READY
      IF(READY.EQ.'NO'.OR.READY.EQ.' ')CALL PENUTUP
      IF(READY.EQ.'HELP')CALL TOLONG
      if(READY.ne.'YES'.AND.READY.NE.'yes')CALL PENUTUP
      END
C*****************************************************************
C     SUBROUTINE DATA
C*****************************************************************
      SUBROUTINE DATA
      LOGICAL ADA
      CHARACTER FIDA*12,FILA*12,HURUF*1
   50 WRITE(*,'(/,2X,A,\)')'NAMA FILE DATA ANDA    : '
      READ(*,'(A)')FIDA
      IF (FIDA.EQ.' ') CALL PENUTUP
      INQUIRE(FILE=FIDA,EXIST=ADA)
      IF (ADA)THEN
```

```
      GOTO 60
      ELSE
      WRITE (*,1)
 1    FORMAT(/,2X,' FILE TIDAK ADA DI DIRECTORY (LIHAT' HELP) ')
      GOTO 50
      ENDIF
60    WRITE(*,'(/,2X,A,\)')'NAMA FILE KELUARAN     : '
      READ(*,'(A)')FILA
      IF (FILA.EQ.' ') CALL PENUTUP
      INQUIRE(FILE=FILA, EXIST=ADA)
      IF (ADA)THEN
70    WRITE (*,2)
 2    FORMAT(/,2X,'FILE KELUARAN SUDAH ADA DI DIRECTORY ')
      WRITE (*,'(20X,A,\)') 'OVERWRITE (Y/T) ?: '
      READ (*,'(A1)') HURUF
      IF ((HURUF.EQ.'T').OR.(HURUF.EQ.'t'))GOTO 60
      IF ((HURUF.NE.'Y').AND.(HURUF.NE.'y'))GOTO 70
      ENDIF
      OPEN(5,FILE=FIDA)
      OPEN(6,FILE=FILA)
      END

C*****************************************************************
C     SUBROUTINE PERSEGI
C*****************************************************************
      SUBROUTINE PERSEGI
      REAL H,B,AN1,AN2,AN3,PJG,NTOT,NT1,NT2,NT3,NCO
      READ(5,*)H,B,AN1,AN2,AN3
      NT2=B*AN2**(1.5)
      NT1=H*AN1**(1.5)
      NT3=H*AN3**(1.5)
      NTOT=NT1+NT2+NT3
      PJG=2*H+B
```

```
      NCO=(NTOT/PJG)**(2./3.)
      WRITE(*,'(2X,A)')'BENTUK PENAMPANG SALURAN PERSEGI'
      WRITE(*,6)NCO
    6 FORMAT(2X,'KEKASARAN KOMPOSIT : ',F7.5)
      END

C*****************************************************************
C     SUBROUTINE TRAPESIUM
C*****************************************************************
      SUBROUTINE TRAPESIUM
      REAL D1,D2,C1,C2,C3,M1,M3,M5,M7,N1,N2,N3,N4,N5,N6,N7
      REAL PNSUM,PSUM,NCO,P1,P2,P3,P4,P5,P6,P7,
     1PNSU1,PNSU2,PNSU3
      READ(5,*)D1,D2,C1,C2,C3,M1,M3,M5,M7,N1,N2,N3,N4,N5,N6,N7
      P1=((D1*M1)**2+D1**2)**(.5)
      P2=C1
      P3=((D2*M3)**2+D2**2)**(.5)
      P4=C2
      P5=((D2*M5)**2+D2**2)**(.5)
      P6=C3
      P7=((D1*M7)**2+D1**2)**(.5)
      PSUM=P1+P2+P3+P4+P5+P6+P7
      PNSU1=N1**1.5*P1+N2**1.5*P2+N3**1.5*P3
      PNSU2=N4**1.5*P4+N5**1.5*P5+N6**1.5*P6
      PNSU3=N7**1.5*P7
      PNSUM=PNSU1+PNSU2+PNSU3
      NCO=(PNSUM/PSUM)**(2./3.)
      WRITE(*,'(2X,A)')'BENTUK PENAMPANG SALURAN TRAPESIUM'
      WRITE(*,18)NCO
   18 FORMAT(2X,'KEKASARAN KOMPOSIT : ',F7.5)
      END
C*****************************************************************
C     SUBROUTINE PENUTUP
```

```
C*************************************************************
      SUBROUTINE PENUTUP
      WRITE(*,1)
    1 FORMAT(///,2x
     1'********** ANDA SALAH DALAM PENULISAN DATA !!! ‘
      END

C*************************************************************
C     SUBROUTINE HELP
C*************************************************************
      SUBROUTINE TOLONG
      INTEGER READY
      WRITE(*,'(//,2X,A)')'BIMBINGAN PENULISAN DATA MASUKAN '
      WRITE(*,'(2X,A,\)')'BENTUK SALURAN 1/2 : '
      READ(*,*)READY
      IF(READY.EQ.1)CALL HELP1
      IF(READY.EQ.2)CALL HELP2
      END

C*************************************************************
C     SUBROUTINE HELP1
C*************************************************************
      SUBROUTINE HELP1
  100 WRITE(*,1)
    1 FORMAT(//,2X,'PROGRAM INI  UNTUK MENGHITUNG SUATU '
      1'NILAI KEKASARAN',/,2X,'DARI SALURAN BENTUK PERSEGI.'
      2/,2X,'DATA YANG DIBUTUHKAN  SALURAN DENGAN '
      3'SUSUNAN ',/,2X,'SEBAGAI BERIKUT :',/,2X
      4'1. PILIHAN JENIS SALURAN DENGAN 1',/,2X,'2. KEDALAMAN'
      5'SALURAN DARI PERMUKAAN AIR ',/,2X,'3. LEBAR DASAR  ',/,2X
      6'4. KEKASARAN SALURAN')
      STOP
      END
```

```
C*************************************************************
C     SUBROUTINE HELP2
C*************************************************************
      SUBROUTINE HELP2
  200 WRITE(*,2)
    2 FORMAT(//,2X,'PROGRAM INI UNTUK MENGHITUNG SUATU '
     1'NILAI KEKASARAN ',/,2X,' SALURAN BENTUK TRAPESIUM.'
     2' BENTUK TRAPESIUM DIPERSIAPKAN MENGGUNAKAN AMBAL
     3'DUA SISI DAN DENGAN KEMIRINGAN SALURAN TERTENTU'
     4' DATA YANG DIBUTUHKAN BERURUTAN DARI '
     5'KIRI SALURAN',/,2X,'DENGAN SUSUNAN BERIKUT :',/,2X
     6'1. PILIHAN JENIS SALURAN ADALAH 2',/,2X
     7'2. KEDALAMAN DARI PERMUKAAN AIR PERAMBAL',/,2X
     8'3. LEBAR AMBAL DAN DASAR SALURAN',/,2X,
     9'4. KEMIRINGAN SALURAN',/,2X,'5. KEKASARAN SALURAN')
      STOP
      END
```

7.3. RANGKUMAN

- Unit program dalam bahasa fortran dapat berupa program utama, program fungsi dan program bagian (SUBROUTINE)
- Statement subroutine berfungsi untuk mengidentifikasi bahwa suatu unit program adalah suatu rutin bagian, serta sekaligus memberikan nama dan argumen-argumen.

7.4. LATIHAN

1. Buat program untuk menghitung debit dan kecepatan aliran dari bebrapa persamaan kecepatan seperti berikut :

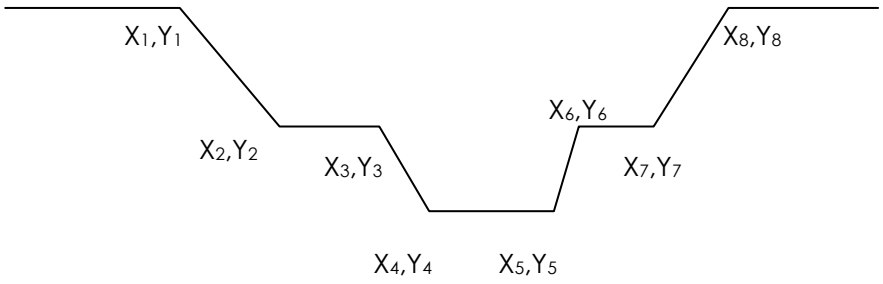
   Manning : $V = \frac{1}{n} R^{2/3} S^{1/2}$

   Chezy : $V = C\sqrt{RS}$

Strikler : $V = K_s R^{2/3} S^{1/2}$

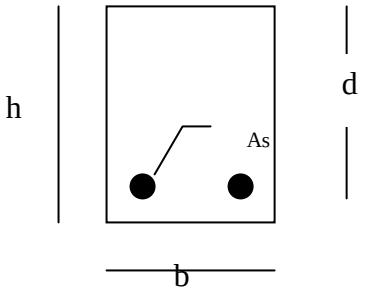
Dan Debit ditentukan dengan Q = V.A

A merupakan luas penampang saluran yang dihitung dengan persamaan sebagai berikut.



$$A = \left[\frac{\sum_{i=1}^{n} X_i * Y_{i+1} - \sum_{i=1}^{n} X_{i+1} * Y_i}{2} \right]$$

2. Buatlah program untuk menghitung momen tahan nominal (Mn) dari penampang balok di bawah ini.



Rumus momen nominal sebagai berikut:

$$Mn = A_s . fy \left[d - \frac{a}{2} \right]$$

Gunakan SUBROUTINE untuk masukan data, SUBROUTINE untuk proses dan SUBROUTINE untuk hasil.

BAB VIII
FILE

KOMPETENSI DASAR:

Pada akhir matakuliah ini mahasiswa diharapkan dapat:

1. Menjelaskan fungsi FILE dalam program fortran
2. Menggunakan statement OPEN FILE dan CLOSE dalam pembuatan program komputer.

8.1. PENDAHULUAN

File adalah media penyimpan data yang bersifat tetap. File dapat berupa eksternal atau file yang mempunyai alat input dan output (I/O) sendiri, maupun internal yang berarti bahwa suatu variabel atau karakter yang berfungsi sebagai sumber atau tujuan dari bebreapa operasi input dan output.

Istilah pengambilan atau perekaman data dari atau ke file disebut dengan access.

8.2. NAMA FILE

File yang disimpan didalam disk harus diberi nama agar terhindar akan terjadinya kesalahan jenis data yang akan dipergunakan juga dapat membedakan dengan file lainnya. Syarat pemberian nama untuk program fortran terdiri dari :

1. Maksimum panjangnya nama file hanya 8 karakter dan tidak boleh mengandung karakter khusus,
2. Dapat ditambahkan extension sepanjang 3 karakter, juga tidak boleh mengadung karakter khusus.

8.3. STATEMENT UNTUK OPERASI FILE

Statement untuk operasi file dibagi menjadi :

1. Statement Open

   Digunakan untuk membuka file ke dalam memori sehingga dapat dimanfaatkan untuk pengambilan data atau penulisan data dari atau ke file tersebut.

   Bentuk format umum dari statement open adalah :

   Open(<no. unit>,file = '<nama file data>',Status='<status>')

   Penjelasan :

   No. unit : adalah nomor yang menunjukan spesifikasi alat yang digunakan berupa nilai integer dan harus diletakan sebagai argumen pertama. Pada statement read dan write akan menggunakan nomor unit ini jikalau data yang akan digunakan diambil dari file tertentu dan jikalau data hasil perhitungan akan disimpan dalam file tertentu.

   Nama File data : merupakan nama file yang berisikan data yang akan dipergunakan dalam pengambilan data maupu dalam penulisan data. File ini dapat berupa file text atau jenis file lainnya asalkan disesuaikan dengan sisitem pembacaan atau penulisan data dari program yang dibuat.

   Status : merupakan identitas dari file yang akan dipergunakan. Jika file yang akan digunakan merupakan file lama yang sudah berisikan data maka satatus yang digunakan adalah old. Status Old tersebut merupakan default dari program fortran. Jika tidak disebutkan jenis status tersebut maka fortran akan langgung menganggap bahwa file data yang dipergunakan berstatus Old. Sedangkan New merupakan status

untuk file yang akan digunakan dalam kondisi baru yang berarti bahwa semua data yang ada pada file tersebut akan hilang. Status New digunakan untuk file yang berlum pernah dibuat.

2. Statement Close

   Merupakan statement yang digunakan untuk menutup file yang sudah tidak dipergunakan kembali. File tersebut dibuka oleh statement open diatas.

   Format yang digunakan untuk melaksanakan statement close tersebut adalah :

   Close(<no. unit>, Status = '<status>')

   Penjelasan :

   No. unit : merupakan nomor unit yang akan di tutup, sama dengan nomor unit yang digunakan pada statement open.

   Status : merupakan konstanta karakter yang menunjukan status dari file yang akan ditutup, dapat berupa Keep atau Delete. Keep berarti bahwa file yang ditutup akan disimpan sedangkan Delete berarti file yang ditutup dihapus dari disk.

3. Statement Write

   Seperti yang sudah dijelaskan pada bab terdahulu bahwa statement write dapat digunakan untuk menulis data kedalam suatu file tertentu dengan menggunakan no. unti yang ditentukan dalam statement Open File.

   Bentuk format dari Statement Write ini seperti diperlihatkan pada bab sebelumnya dalah :

   Write(<no. unit>, <format>) < I/O List >

   Penjelasan :

   No. unit : merupakan nomor unit dari file yang sudah terbuka oleh statement open file yang khusus digunakan untuk

menuliskan data hasil proses perhitungan ke dalam file dan disimpan ke dalam disk.

Format : Merupakan bentuk susunan dari jenis data yang akan dituliskan ke dalam disk. Penjelasan format lebih ditail pada bab sebelumnya.

I/O List : Merupakan isi dari data yang akan dituliskan ke dalam file. Baik berupa karekter ataupun berupa nilai dari hasil proses perhitungan.

Contoh Penggunaan Statement Write :

```
Write ( 1 ) A,B,C
```

Menunjukan perintah menuliskan nilai dari variabel A,B dan C kedalam file tanpa menggunakan format dengan menggunakan nomor unit 1 yang sudah terbuka sebelumnya.

```
Write ( 1, '(A20,A15,F7.2)' ) A,B,C
```

Menujukan perintah menuliskan nilai A dengan menggunakan format karakter yang berjumlah 20 karekter, nilai B dengan menggunakan format karakter yang berjumlah 15 karakter dan menuliskan nilai C dengan menggunakan format bilangan untuk 7 karakter dan 2 desimal. Penulisan ke dalam file data menggunakan nomor unit 1 yang sudah dibuka pada statement open file.

4. Statement Read

Seperti yang sudah dijelaskan pada bab terdahulu bahwa statement read dapat digunakan untuk membaca data dari suatu file tertentu dengan menggunakan no.unit yang ditentukan dalam statement Open File.

Bentuk format dari Statement Read ini diperlihatkan sbb :

```
Read(<no. unit>, <format> ) < I/O List >
```

Penjelasan :

No. unit : merupakan nomor unit dari file yang sudah terbuka oleh statement open file yang khusus digunakan untuk membaca data dari file.

Format : Merupakan bentuk susunan dari jenis data yang akan digunakan untuk membaca data dari file. Penjelasan format lebih ditail pada bab sebelumnya.

I/O List : Merupakan isi dari variabel yang akan digunakan untuk membaca data dari file.

8.4. Contoh Penggunaan

```
      REAL PJS,PIAS,T,TAS
       INTEGER READY
       CHARACTER FILA*12,FIDA*12
       WRITE(*,1)
     1 FORMAT(1X,'PROGRAM MENGHITUNG KOSENTRASI SEDIMEN
                  PADA SALURAN',
      1/1X'SEPANJANG L DALAM WAKTU T DENGAN PERSAMAAN
                  PARABOLA DAN',
      2/1X'SKEMA EXPLISIT.'
      3/1X'PROGRAM MENGGUNAKAN PROSES OPEN FILE DATA,
                  DENGAN URUTAN',
      4/1X'INPUT DATA SEBAGAI BERIKUT : ',
      5/1X'PANJANG SALURAN (METER), JUMLAH PIAS SALURAN
                  (MAX 12), LAMA,'
      4/1X'PENGAMATAN (DET), TAHAP PENGAMATAN, KECEPATAN
                  ALIRAN (M/DT)',
      5/1X'DAN KOEFISIEN DIFFUSI.',//)
       WRITE(*,'(1X,A,\)')'APAKAH DATA ANDA SUDAH SIAP (Y=1/N=2) ?
                  : '
       READ(*,*)READY
       IF(READY.EQ.2)STOP
```

```
      WRITE(*,'(1X,A,\)')'NAMA FILE DATA ANDA    : '
      READ(*,'(A)')FIDA
      WRITE(*,'(1X,A,\)')'NAMA FILE KELUARAN     : '
      READ(*,'(A)')FILA
      OPEN(5,FILE=FIDA)
      OPEN(6,FILE=FILA)
      READ(5,*)PJS,PIAS,T,TAS,U,H
      CALL HITUNG(PJS,PIAS,T,TAS,U,H)
          CLOSE(5)
          CLOSE(6)
      END
C*************************************************************
C     SUBRUTINE HITUNG
C*************************************************************
      SUBROUTINE HITUNG(PJS,PIAS,T,TAS,U,H)
      REAL PJS,PIAS,T,TAS,DX,DT,A,C,H
      INTEGER AM,AN
      DIMENSION P(100,100),PA(100,100)
      WRITE(6,1)
    1 FORMAT(1X,'PROGRAM MENGHITUNG KOSENTRASI SEDIMEN
                        PADA SALURAN',
     1/1X'SEPANJANG L DALAM WAKTU T DENGAN PERSAMAAN
                        PARABOLA DAN',
     2/1X'SKEMA EXPLISIT.',
     8/1X'DIPROGRAM OLEH  : MUDJIATKO',/2X
     9'              : 9430/I-1/68 /1997',//1X,'INPUT DATA :')
      WRITE(6,4)PJS
    4 format(1X,'PANJANG SALURAN ( METER )  : ',F7.2,1X,'METER')
      WRITE(6,5)PIAS
    5 FORMAT(1X,'JUMLAH PIAS UNTUK SALURAN  : ',F7.2,1X,'BUAH')
      WRITE(6,6)T
    6     FORMAT(1X,'LAMANYA PROSES PENGAMATAN :
                        ',F7.2,1X,'DETIK')
```

```
  WRITE(6,7)TAS
 7 FORMAT(1x,'JUMLAH PIAS UNTUK WAKTU    : ',F7.2,1X,'BUAH')
  WRITE(6,11)U
11 FORMAT(1X,'KECEPATAN ALIRAN          : ',F7.2,1X,'M/DETIK')
  WRITE(6,9)H
 9 FORMAT(1X,'KOEFISEIN DIFFUSI         : ',F7.2,/)
  WRITE(6,'(1X,A)')'HASIL HITUNGAN : '
  DX=PJS/PIAS
  DT=T/TAS
  WRITE(6,8)DX,DT
 8   FORMAT(1X,'BEDA   JARAK               :   ',F6.2,1X,'M',1X,'     ===>
                      HORIZONTAL',
  1/1X,'BEDA WAKTU      : ',F6.2,1X,'DET',1X,'===> VERTIKAL')
  AM=0
  DO 60 I=1,PIAS+1
  WRITE(6,'(4X,(I2,\,6X))')AM
  AM=AM+1
60 CONTINUE
  AN=0
  WRITE(6,'(/,I2,\)')AN
  DO 10 I=1,PIAS+1
  P(1,I)=0
  P(1,1)=1
10 CONTINUE
  WRITE(6,'(1x,100(F5.2,1X))')(P(1,I),I=1,PIAS+1)
  DO 20 J=1,TAS
  P(J+1,1)=1
  A=DT*U/DX
  DO 30 K=2,PIAS+1
  P(J+1,K)=A*(P(J,K-1)-P(J,K))+P(J,K)
  PA(J,K)=P(J+1,K)
30 CONTINUE
20 CONTINUE
```

```
      DO 50 N=1,TAS
      AN=AN+1
      PA(N,1)=1
      C=H*DT/DX**2
      DO 40 M=2,PIAS
      P(N+1,M)=C*(PA(N,M-1)-2*PA(N,M)+PA(N,M+1))+PA(N,M)
   40 CONTINUE
      WRITE(6,'(I2,\)')AN
      WRITE(6,'(1x,100(F5.2,1X))')(P(N+1,M),M=1,PIAS+1)
   50 CONTINUE
      END
```

8.5. RANGKUMAN

- File merupakan media penyimpan data yang bersifat tetap.
- File dapat berupa eksternal yang mempunyai alat input dan output (I/O) sendiri, maupun internal yang berfungsi sebagai sumber atau tujuan operasi input dan output.
- Statement untuk FILE terdiri dari READ, WRITE, OPEN dan CLOSE.

8.5. LATIHAN

1. Buat Program untuk menghitung luas penampang saluran dengan menggunalkan open file.

2. Buatlah Program untuk menghitung gaya dalam dari suatu perletakan sederhanan (Simple beam).

DAFTAR PUSTAKA

Amrinsyah Nasution, Pemrograman dengan Bahasa Fortran, Jakarta: 1995

Bambang Suryoatmono, Bahasa Fortran: Dari Fortran IV hingga Fortran Powerstation, Bandung, 1996

Jogiyanto, Teori dan Aplikasi Program Komputer Bahasa Fortran, Yogyakarta: Andi Offset, 1993

.

Penerbit :
Pusat Pengembangan Pendidikan Universitas Riau
Gedung Rektorat Unri Lt.4 Kampus Binawidya
Simpang Baru, Pekanbaru 28293 Riau, Indonesia
Phone/Fax: +62761 567092,
E-mail: pusbangdik@unri.ac.id
www.ruedc.unri.ac.id

ISBN 978-979-1222-28-0