



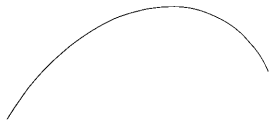
APLIKASI Excel untuk **REKAYASA**

TEKNIK SIPIL

Guntar Pangaribuan

Teguh Santoso
2008.04.

Aplikasi
Excel
untuk
Rekayasa Teknik Sipil



Aplikasi **Excel** untuk **Rekayasa Teknik Sipil**

Guntar Pangaribuan

Penerbit PT Elex Media Komputindo
Kelompok Gramedia, Jakarta

Kutipan Pasal 44, Ayat 1 dan 2, Undang-Undang Republik Indonesia tentang HAK CIPTA:

Tentang Sanksi Pelanggaran Undang-Undang Nomor 6 Tahun 1982 tentang HAK CIPTA, sebagaimana telah diubah dengan Undang-Undang No. 7 Tahun 1987 jo. Undang-Undang No. 12 Tahun 1997, bahwa:

1. Barangsiapa dengan sengaja dan tanpa hak mengumumkan atau memperbanyak suatu ciptaan atau memberi izin untuk itu, dipidana dengan pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.100.000.000,- (seratus juta rupiah).
2. Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran Hak Cipta sebagaimana dimaksud dalam ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.50.000.000,- (lima puluh juta rupiah).

Aplikasi Excel Untuk Rekayasa Teknik Sipil
Guntar Pangaribuan
© 2004, PT Elex Media Komputindo, Jakarta
Hak cipta dilindungi undang-undang
Diterbitkan pertama kali oleh
PT Elex Media Komputindo
Kelompok Gramedia, Anggota IKAPI, Jakarta 2004

121040386
ISBN: 979-20-5506-1

Cetakan pertama : April 2004
Cetakan kedua : Juli 2004

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta
Isi di luar tanggung jawab percetakan

KATA PENGANTAR

Microsoft Excel adalah program spreadsheet atau pengolah angka yang paling populer dan banyak digunakan saat ini. Disukai banyak kalangan karena pengoperasiannya yang relatif mudah dengan hasil yang memuaskan. Dalam lingkup Teknik Sipil, Excel sudah mendapat tempat tersendiri bagi profesi mereka.

Salah satu kekuatan Excel adalah bahasa macro-nya. Hal ini tentu tidak lepas dari aplikasi Visual Basic (Visual Basic for Application) yang bekerja dengan Excel. Macro yang selama ini kita kenal, umumnya digunakan sebagai otomasi langkah-langkah pekerjaan dalam aplikasi perkantoran. Namun dalam hal lain, juga dapat digunakan untuk aplikasi perhitungan. Di sini terdapat kombinasi yang unik antara spreadsheet dan Visual Basic yang ternyata banyak memberikan kemudahan bagi pemakai jika membuat program perhitungan di Excel.

Bekerja dengan Excel berarti kita menginginkan suatu penyelesaian yang cepat dan praktis, namun diharapkan dengan hasil yang maksimal pula. Jadi, tinggal bagaimana memanfaatkan fasilitas-fasilitas yang ada sebaik mungkin. Tidak ada aturan baku memang dari Microsoft tentang aplikasi Excel untuk rekayasa teknik. Sebab, apabila kombinasi antara fungsi-fungsi Excel dan aplikasi VB diterapkan, aplikasinya akan sangat luas. Penulis mencoba menjembatani antara keduanya yang kemudian dirangkum ke dalam buku ini. Diharapkan materi-materi yang disampaikan dapat menjadi referensi yang berguna bagi para pembaca.

Cikarang, September 2003

Guntar Pangaribuan

xl4eng@hotmail.com

DAFTAR ISI

Kata Pengantar.....	v
Daftar Isi.....	vii

HAB 1 DASAR-DASAR EXCEL 1

1.1 Worksheet dan Workbook.....	1
1.2 Tipe Data	2
1.3 Formula	5
1.4 Fungsi Built-In	7
1.5 Formula Array	10
1.6 Format Data dan Sel	10
1.7 Pesan Error	11
1.8 Mencetak Worksheet.....	12
1.9 Membuat Grafik	13

HAB 2 FUNGSI..... 15

2.1 Fungsi Numerik.....	15
2.2 Fungsi Logika.....	17
2.3 Fungsi Pembacaan Tabel.....	20
2.4 Fungsi Teks	21
2.5 Fungsi Pemakai	22
2.5.1 Membuat Fungsi dengan Macrosheet.....	23
2.5.2 Membuat Fungsi dengan Aplikasi Visual Basic.....	24
2.5.3 Alur Program	26
2.5.4 Contoh-Contoh Fungsi Pemakai.....	31
2.6 Circular Reference.....	41

HAB 3 FUNGSI UNTUK ANALISIS DATA 43

3.1 Pemodelan Data.....	43
3.1.1 Regresi Linear	43

3.1.2	Regresi Polinomial	46
3.1.3	Contoh Regresi dalam Grafik	47
3.1.4	Interpolasi	56
3.2	Data Statistik	59
3.2.1	Histogram dan Distribusi Kumulatif	59
3.2.2	Range, Mean, dan Standard Deviation	62
3.2.3	Distribusi Normal	63

BAB 4 PROGRAM MATRIKS 67

4.1	Pengertian Matriks	67
4.1.1	Tipe Matriks	68
4.1.2	Operasi Matriks	70
4.2	Macro Program	73
4.2.1	Membuat Macro Program	73
4.2.2	Program Untuk Operasi Matriks	75
4.2.3	Tip	80
4.3	Metode Matriks Untuk Analisis Struktur	83
4.3.1	Struktur Atas	83
4.3.2	Struktur Bawah	85

BAB 5 PROGRAM EXCEL UNTUK REKAYASA TEKNIK SIPIL 87

5.1	Struktur Frame Balok	87
5.2	Struktur Rangka Batang	100
5.3	Balok di Atas Medium Elastis	117
5.4	Dinding Penahan Tanah	124
5.5	Pondasi Pelat	127
5.6	Konsolidasi Satu Dimensi	135

BAB 6 SCRIPT FILE AUTOCAD 141

BAB 7 LISTING PROGRAM 151

7.1	Visual Basic for Application	151
7.2	Petunjuk Pemakaian Program	156
7.3	Listing Program	158

Daftar Pustaka 183

BAB 1

DASAR-DASAR EXCEL

Pada dasarnya, pekerjaan yang dilakukan dengan Excel adalah pekerjaan memasukkan data yang kemudian diproses untuk mendapatkan hasil yang diinginkan. Prinsipnya sama seperti kita memasukkan data pada sebuah kalkulator elektronik. Hanya saja data yang dimasukkan di sini terdiri atas berbagai tipe dan ditambah dengan fasilitas-fasilitas Excel yang ada memungkinkan untuk menyajikan tampilan angka dan teks berikut tabel, grafik hingga database. Hal ini yang membuat program menjadi utuh untuk membuat teks laporan, berikut analisis perhitungannya.

Setiap versi dari Excel terus dikembangkan dengan selalu mengadakan perubahan dan penambahan fasilitas baru, namun tetap menjaga kompatibilitas dengan versi sebelumnya. Namun perubahan-perubahan yang dilakukan tidak mengubah fitur dasar software ini sebagai lembar kerja untuk aplikasi perhitungan. Dalam pembahasan buku ini kita akan bekerja dengan Microsoft Excel 2000.

1.1 Worksheet dan Workbook

Pada saat pertama kali membuka Excel, secara default Book1 adalah nama workbook (buku kerja) yang pertama. Workbook ini terdiri atas 3 buah worksheet (halaman kerja) dengan nama Sheet1, Sheet2, dan Sheet3. Worksheet Excel ini juga disebut sebagai spreadsheet, yaitu lembar untuk pengolahan angka.

Gambar 1.1 memperlihatkan elemen-elemen dari sebuah workbook. Pada bagian atas halaman terdapat title bar yang menampilkan nama workbook. Di bawahnya terdapat menu bar dengan menu **File – Edit – View** dan seterusnya yang di dalamnya terdapat menu pull-down dengan sejumlah perintah yang spesifik nama menunya. Contoh nama perintah-perintah di dalam menu pull-down disajikan pada Tabel 1.1. Toolbar memperlihatkan sejumlah icon (gambar kecil) sebagai shortcut untuk perintah-perintah di

dalam menu. Formula bar memperlihatkan data dan formula dalam sel aktif. Pada bagian bawah di sebelah kiri dari workbook terdapat tab sheet yang memperlihatkan nama worksheet.

Workbook dapat dimodifikasi dengan menampilkan toolbar-toolbarnya. Tampilan ini biasanya bergantung pada kebutuhan kita saat bekerja. Untuk kebutuhan suatu aplikasi perhitungan, kita optimalkan saja seperti yang tampak pada Gambar 1.1. Langkah-langkahnya sebagai berikut.

1. Klik menu **View > Toolbars**, lalu pilih **Standard, Formatting** dan **Visual Basic** yang ditandai dengan tanda centang ☒ di sebelah kiri pilihan ini.
2. Klik **View > Formula Bar**.

Worksheet dibagi oleh baris dan kolom. Kolom diberi nomor dari A sampai Z, kemudian AA, AB hingga IV. Baris diberi nomor dari 1 hingga 65536. Perpotongan kolom dan baris membentuk sel-sel sebagai tempat untuk diisi data. Setiap sel berdiri sendiri dan mempunyai alamat yang ditunjukkan oleh kolom dan baris, misalnya B4 adalah sel yang berada pada kolom B dan baris ke-4. Alamat sel yang aktif dapat dilihat dalam box referensi sel di bawah toolbar. Kumpulan sel-sel disebut range sel, membentuk array memanjang horizontal atau secara vertikal. Misalnya range A1:A5 adalah kumpulan sel dari A1 hingga A5 membentuk array 5 x 1, atau range A1:C5 membentuk array 5 x 3, dan seterusnya.

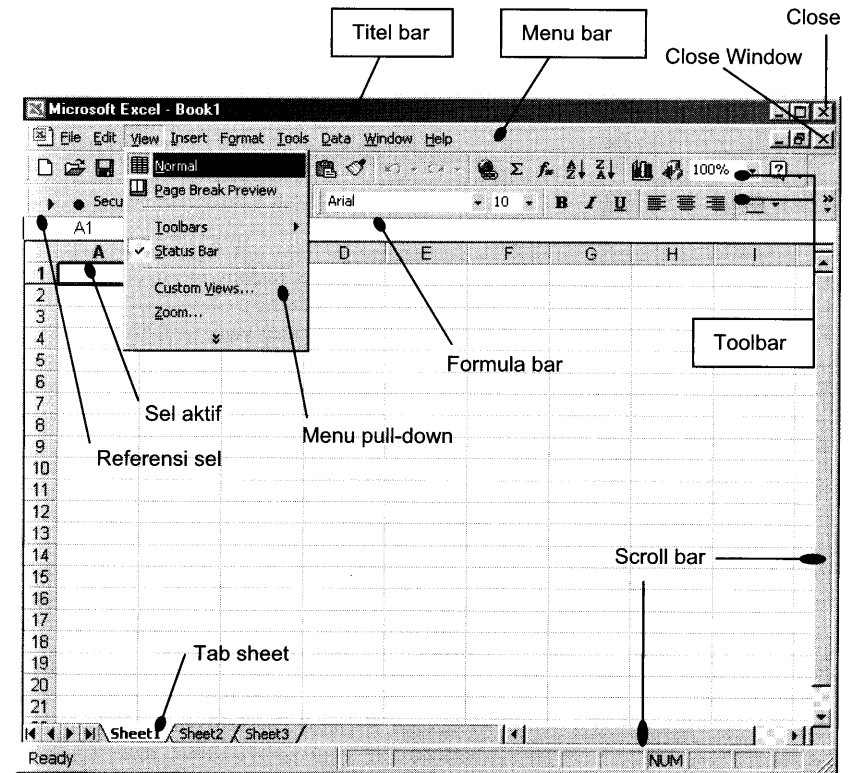
Nama worksheet dapat diganti dengan nama lain, dengan mengklik dua kali pada tab sheet; tekan Delete untuk menghapus namanya, kemudian tulis nama barunya. Atau dengan mengklik kanan pada tab sheet kemudian dari menu shortcut klik **Rename** untuk mengganti namanya.

1.2 Tipe Data

Data input dalam spreadsheet dapat dibedakan menjadi beberapa tipe, yaitu:













- a. Teks : karakter alphabet dan teks: A, B, -Z, AB, A2, Komputer, ...
- b. Angka : data numerik: 1, 2, 3, 0, -1, -2, 4, 5.85, ...
- c. Tanggal : data tanggal biasanya mengacu pada setting penanggalan komputer atau diformulasikan dalam perhitungan.
- d. Jam : data jam biasanya mengacu pada setting waktu komputer atau diformulasikan dalam perhitungan.

- c. Formula : ekspresi matematik yang menghitung dua atau lebih nilai yang menghasilkan sebuah nilai baru.
- f. Fungsi : fungsi yang digunakan untuk berbagai aplikasi perhitungan, seperti keuangan (financial), matematika dan trigonometri, statistik, database, logika, dan lain-lain.



Gambar 1.1 Workbook Excel 2000

Tabel 1.1 Perintah-Perintah Dasar Excel

Nama	Shortcut	Keterangan
File > Save		menyimpan (save) file dengan nama workbook yang sedang dikerjakan.
File > Save As	Alt+FA	menyimpan file dengan nama lain ke dalam folder dan direktori, atau ke sebuah disket.
File > New		membuka workbook baru.
File > Open		membuka workbook yang ada.
File > Close		menutup workbook (icon Close Window).
File > Exit		menutup program dan keluar dari Excel (icon Close).
File > Print		mencetak worksheet melalui mesin printer.
Edit > Undo		membatalkan pekerjaan yang terakhir.
Edit > Redo		mengulangi pekerjaan yang terakhir
Edit > Copy		meng-copy data pada sel atau range sel ke lokasi lain. Untuk memilih data yang akan di-copy atau cut, klik pada sel atau range sel tempat data tersebut.
Edit > Cut		memindahkan data ke lokasi lain. Perintah seperti Cut, Copy, dan Paste dapat ditemukan dalam menu pop-up dengan mengklik tombol kanan mouse pada sel yang disorot.
Edit > Paste		menempatkan data ke dalam sel atau range sel yang baru. Perintah Paste dilakukan setelah ada perintah Cut atau Copy. Untuk menempatkan data, klik lokasi barunya, kemudian tekan Enter atau klik Paste  .
Insert > Cells	Alt+IE	menyisipkan sel baru di posisi pointer (sel yang disorot).
Insert > Column	Alt+IC	menyisipkan kolom baru di posisi pointer.
Insert > Row	Alt+IR	menyisipkan baris baru di posisi pointer.

Format > Cell	Ctrl+I	memformat isi sel, terdiri atas format angka, teks, font yang dipakai, garis tepi, warna, dan proteksi worksheet.
Format > Row	Alt+OR	memformat baris, terdiri atas pengaturan tinggi baris, menyembunyikan baris dan menampilkannya kembali.
Format > Column	Alt+OC	memformat kolom, terdiri atas pengaturan lebar kolom, lebar otomatis (autofit selection), menyembunyikan dan menampilkan kolom, dan informasi tentang lebar kolom standar.

Tabel 1.2 Operator dan Relasi Matematika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
^	Pangkat
Relasi	
=	Sama dengan
>	Lebih besar
<	Lebih kecil
<>	Tidak sama dengan
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

1.3 Formula

Berdasarkan definisinya, sebuah formula adalah ekspresi matematika untuk menghitung hasil dari dua atau lebih nilai. Formula dapat terdiri atas bilangan, operator matematika, fungsi-fungsi, referensi sel atau range sel. Sel dan range sel biasanya diberi nama, misalnya "A" untuk B2, atau

”B” untuk range D2:D6. Penamaan sel akan dibahas sebentar pada bagian ini. Semua formula selalu dimulai dengan tanda sama dengan (=).

Contoh:

=B2+B3 menjumlahkan data pada sel B2 dengan sel B3

=B4*B5 mengalikan data pada sel B4 dengan sel B5

=SUM(D2:D4) menjumlahkan data bilangan sel D2, D3, dan D4

Ketika suatu formula akan kita copy, yang perlu diperhatikan adalah jenis sel dalam formula tersebut. Sel relatif adalah sel yang apabila di-copy akan menyesuaikan diri dengan tempat barunya. Sel C1 yang berisi formula =A1+B1 akan menjadi =C2+D2 pada E2. Hasilnya seperti yang diperlihatkan dalam Gambar 1.2.

Sel absolut adalah sel yang tidak mengalami perubahan alamatnya apabila di-copy ke tempat lain. Notasinya dengan menambahkan \$ di depan nama kolom dan nomor baris, misalnya \$A\$4. Menambahkan \$ pada nama kolom atau nomor baris saja, hanya akan mengubah salah satu alamatnya. Kolom A pada \$A4 akan tetap apabila di-copy, namun nomor barisnya akan menyesuaikan dengan lokasi barunya. Demikian sebaliknya dengan sel A\$4. Sel-sel demikian disebut sel semi-absolut.

Formula yang menyatakan relasi sel-sel atau range sel, misalnya =B4+C30 atau =SUM(A4:C20), lebih sulit dibaca dibanding relasi matematik dengan nama variabel yang lebih praktis, misalnya = x + n atau SUM(B). Sel dan range sel dibuat namanya melalui menu **Insert > Name > Define > Names**. Nama ini berlaku untuk semua sheet dalam satu workbook. Apabila hanya mewakili pada sheet tertentu, misalnya Sheet1, maka penulisannya adalah **Sheet1!x**, yaitu untuk variabel x dalam Sheet1. Penggunaan dengan variabel-variabel ini sangat direkomendasikan, karena akan sangat menyederhanakan suatu formula. Nama-nama yang dipakai berikut lokasinya dapat dibuat ke dalam list dan dicetak dalam worksheet dengan **Insert > Name > Paste > Paste List**. Mencetak list ini akan berguna apabila kita bekerja dengan banyak variabel dalam beberapa sheet.

	A	B	C	D	E	F
1			=A1+B1			
2					=C2+D2	
3			=\$A\$1+B1			
4					=\$A\$1+D2	
5			=\$A1+B1			
6					=\$A2+D2	
7			=A\$1+B7			
8					=C\$1+D8	
9						
10						

Gambar 1.2 Meng-copy formula dan hasilnya dalam spreadsheet

1.4 Fungsi Built-In

Excel mempunyai fungsi-fungsi built-in untuk membangun formula yang kompleks, beberapa di antaranya diperlihatkan dalam Tabel 1.3. Fungsi seperti matematika dan trigonometri atau statistik misalnya, adalah fungsi yang paling umum dipakai dalam proses pengukuran di laboratorium. Sebelum menggunakan sebuah fungsi, sebaiknya dipahami benar fungsi dan argumennya. Referensi mengenai ini ada pada menu Help atau tekan F1.

Tabel 1.3 Beberapa Fungsi Built-in Excel

A. Fungsi Matematika dan Trigonometri	
ABS(bil)	nilai absolut dari bilangan
ACOS(bil)	arccosinus dari bilangan (dalam radian)
ACOSH(bil)	invers hiperbolic cosinus dari bilangan
ATAN(bil)	arctangen dari bilangan
EXP(bil)	eksponensial dari bilangan
FACT(bil)	factorial hingga ke bilangan
INT(bil)	pembulatan bilangan ke bawah ke integer terdekat
LOG(bil, dasar)	logaritma bilangan untuk bilangan dasar ditentukan

LOG10(bil)	logaritma bilangan untuk bilangan dasar 10
MDETERM(array)	determinan matriks suatu array
MINVERS(array)	invers matriks suatu array
MMULT(array1,array2)	perkalian matriks dari 2 data array
PI()	nilai pi = 3.141592654
RAND()	bilangan random antara 0 dan 1
SIGN(bil)	tanda dari bilangan, 1 jika bilangan positif, 0 jika bilangan 0, -1 jika bilangan negatif
SIN(bil)	sinus dari bilangan
SINH(bil)	hiperbolic sinus dari bilangan
SQRT(bil)	mencari akar pangkat 2 dari bilangan
SUM (bil1,bil2,...)	penjumlahan dari bilangan
SUMPRODUCT (array1,array2)	jumlah produk dari komponen array yang sama
TAN(bil)	tangen dari bilangan
TANH(bil)	hiperbolik tangen dari bilangan
B. Fungsi Informasi	
ISBLANK(nilai)	TRUE jika nilai adalah kosong
ISLOGICAL(nilai)	TRUE jika nilai adalah sebuah logika
ISNUMBER(nilai)	TRUE jika nilai adalah sebuah bilangan
ISTEXT(nilai)	TRUE jika nilai adalah teks
C. Fungsi Statistik	
AVERAGE(bil1,bil2,...)	nilai rata-rata dari argumen bilangan
COUNT(nilai1,nilai2,...)	menghitung jumlah data bilangan dalam list argumen
COUNTA(nilai1,nilai2,...)	menghitung jumlah sel yang tidak kosong
LINEST(data y,data x...,c,stat)	mencari parameter garis linear, di mana c = konstanta
MAX(bil1,bil2,...)	nilai maksimum dalam list argumen
MIN(bil1,bil2,...)	nilai minimum dalam list argumen

SLOPE(data y,data x)	slope (kemiringan) garis regresi linear
INTERCEPT(data y,data x)	perpotongan garis regresi linear terhadap aksis-Y
TREND(data y,data x, x',const)	menentukan nilai regresi sepanjang garis linear
D. Fungsi Lookup and Reference	
COLUMNS(array)	menentukan nomor kolom suatu array
INDEX(array,baris,kolom)	memilih komponen suatu array
ROWS(array)	menentukan nomor baris suatu array
TRANSPOSE(array)	menentukan transpose suatu array
E. Fungsi Logika	
AND(logika1,logika2,...)	TRUE jika semua argumen logika adalah TRUE
IF(logika1, nilai jika benar, nilai jika salah)	
NOT(logika)	pembalikan logika dalam argumen: NOT(TRUE) = FALSE
OR(logika1, logika2)	TRUE jika salah satu argumennya TRUE
F. Fungsi Teks	
CHAR(bil)	karakter berdasarkan kode bilangan
EXACT(teks1,teks2)	fungsi untuk memeriksa apakah teks1 identik dengan teks2
FIND(teks1,teks2,start)	mencari nomor teks1 dalam teks2 dengan nomor start
LEFT(teks, jumlah karakter)	pengambilan teks mulai dari kiri
LEN(teks)	menentukan jumlah karakter dalam teks
RIGHT(teks, jumlah karakter)	pengambilan teks mulai dari kanan
TRIM(teks)	menghilangkan spasi dalam teks, kecuali satu spasi antarkata
UPPER(teks)	membuat semua huruf teks menjadi huruf kapital

1.5 Formula Array

Formula array adalah formula yang dapat mengerjakan beberapa perhitungan kemudian menghasilkan satu atau beberapa hasil. Formula ini bekerja atas dua atau lebih set data dalam argumen array. Setiap array masing-masing harus mempunyai jumlah kolom dan baris yang sama. Sebagai contoh, fungsi built-in LINEST. Fungsi ini mencari parameter garis regresi linear dari data x dan y , seperti yang diperlihatkan dalam Gambar 1.3.

	A	B
1	x	y
2	-1.0	-5.0
3	-0.5	0.0
4	1.0	5.0
5	2.0	4.0
6	3.0	0.5
7	4.0	-5.0
8	5.0	-12.0
9	Slope	Y-intercept
10	-1.204	0.537

	A	B
1	x	y
2	Data	Data
3		
4		
5		
6		
7		
8		
9	Slope	Y-intercept
10	=INDEX(LINEST(B2:B8,A2:A8),1)	
	=INDEX(LINEST(B2:B8,A2:A8),2)	

Gambar 1.3 Contoh formula array untuk regresi linear dan hasilnya

1.6 Format Data dan Sel

Format data dan sel dapat dilakukan dari menu **Format > Cells**. Pada kotak dialog Format Cells terdapat beberapa lembar tab, yaitu **Number**, **Alignment**, **Font**, **Border**, **Patterns** dan **Protection**. Format bilangan dalam tab **Number** dibagi dalam beberapa kategori seperti general, number, date dan sebagainya. Misalnya untuk kategori custom atau pilihan pemakai diperlihatkan dalam Tabel 1.4. **Alignment** adalah pengaturan untuk letak teks dalam sel seperti alignment vertikal dan horizontal, kontrol teks dan orientasinya dalam sel. **Font** adalah pengaturan yang berhubungan dengan karakter, seperti jenis huruf yang digunakan (Arial, Times New Roman, Tahoma, ...), style (italic, bold, ...), ukuran (8, 9, ...), warna dan sebagainya.

Border untuk membuat garis tepi, tipe garis (lurus, putus-putus, tipis, tebal), dan warnanya. **Patterns** untuk membuat tampilan sel dengan arsiran (horizontal, diagonal, titik, ...), warna arsiran dan warna background sel. **Protection** digunakan untuk memilih kondisi (locked atau tidak) data dan formula (hidden atau tidak) dalam sel apabila proteksi worksheet diaktifkan. Untuk memproteksi worksheet, pilihlah menu **Tools > Protection > Protect Sheet**.

Tabel 1.4 Contoh kategori custom format untuk bilangan

Format	Tampilan
0.00	2343.00
0.00E+00	2.34E+03
###0.0+0	2.3E+3
#,##0.00	2,343.00
#,##0.00%	234,300.00%
0.0 "m"	2343.0 m

Lebar kolom dan tinggi baris dapat disesuaikan besarnya dengan men-drag mouse (menahan tombol kiri mouse sambil menyeret) saat posisi pointer berada di batas-batas judul kolom dan baris (ada tanda "+"). Cara lain dengan mengklik kanan mouse pada judul kolom atau baris, dan dari menu shortcut klik **Row > Height** atau **Column > Width**, kemudian masukkan nilai yang diinginkan pada kotak teks yang disediakan. Atau melalui menu **Format > Row > Height** atau **Format > Column > Width**.

1.7 Pesan Error

Pesan error akan disampaikan apabila suatu formula tidak bekerja sebagaimana mestinya, misalnya argumen dalam formula tidak lengkap, tipe data tidak sesuai, atau salah dalam mengembangkan logika. Hal lain misalnya pembagian dengan 0 (nol) atau lebar kolom tidak cukup lebar untuk tampilan suatu hasil. Pesan dimulai dengan tanda # kemudian diikuti jenis kesalahannya. Tabel 1.5 memperlihatkan beberapa pesan error yang mungkin muncul.

Tabel 1.5 Pesan Error

Pesan	Keterangan
#####	Sel berisi data bilangan, tanggal atau waktu yang lebih lebar dari selnya, atau hasil minus dalam format tanggal dan waktu.
#N/A	(Not Available) Referensi data tidak cocok dan tidak dapat digunakan oleh fungsi atau formula.
#DIV/0!	Pembagian dengan nol.
#NAME?	Excel tidak mengenali teks dalam formula.
#REF!	Referensi sel yang digunakan tidak berlaku.
#VALUE!	Tipe argumen yang digunakan dalam fungsi adalah salah. Argumen di sini dapat berupa nilai numerik, teks, nama, label, referensi sel, range sel, dan fungsi.

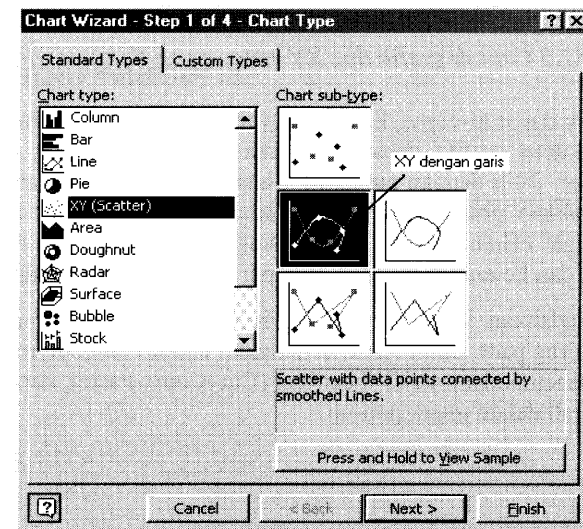
1.8 Mencetak Worksheet

Sebelum mencetak worksheet pada kertas, terlebih dulu kita menentukan dengan memblok area yang akan dicetak (dengan **Shift + "→↓"** atau men-drag mouse). Setelah itu ditetapkan dengan **File > Print Area > Set Print Area**. Hasil pencetakannya dapat dilihat dengan **File > Print Preview** atau mengklik **Print Preview** pada toolbar. Pencetakan kemudian dilakukan dengan **File > Print** atau dengan mengklik Print ikon.

Batas-batas pencetakan halaman yang juga dapat ditentukan dengan **View > Page Break Preview**. Tampilan ini memperlihatkan setiap data yang akan dicetak berikut nomor halamannya. Anda juga dapat menentukan batas-batas pencetakan halaman men-drag mouse pada garis batas yang berwarna biru. Konfigurasi tampilan halaman kemudian diatur dengan **File > Page Setup**. Pada kotak dialog Page Setup terdapat beberapa lembar tab, yaitu **Page**, **Margins**, **Header/Footer** dan **Sheet**. Tab **Page** untuk memilih orientasi dan ukuran kertas, serta skala worksheet dalam kertas; **Margins** untuk mengatur batas-batas teks dalam kertas; **Header/Footer** untuk membuat catatan-catatan pinggir; dan **Sheet** untuk mengatur tampilan sheet seperti mengeluarkan grid, nama baris, kolom, serta urutan pencetakan.

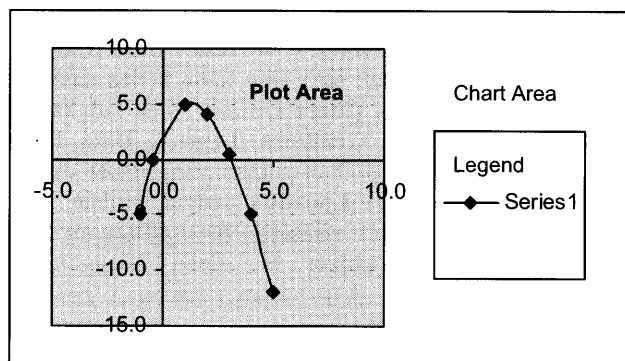
1.9 Membuat Grafik

Sekumpulan data dapat disampaikan secara baik melalui gambar grafik untuk memperlihatkan korelasi antardatanya. Dalam Excel terdapat banyak tipe untuk penggambaran grafik, seperti Pie, Bar, Column, Area, Line, dan sebagainya, ditambah dengan tampilan 3 dimensi. Langkah-langkah untuk membuat grafik adalah dengan **Insert > Chart** atau dengan mengklik **Chart Wizard** pada toolbar sehingga keluar kotak dialog **Chart Wizard** seperti diperlihatkan pada Gambar 1.4. Pada langkah awal kita menentukan tipe grafik pada **Standard Types** atau tipe lainnya pada Custom Types. Sebagai contoh, di sini kita akan mengambil data dari Gambar 1.3 untuk membuat grafik dengan Standard Types. Selanjutnya dipilih tipe **XY (Scatters)** dengan sub-tipe **connected by smoothed lines**. Klik **Next > Data Range >** pilih data pada A2:B8 (pilihan **Series** untuk satu atau lebih series data) **> Series In Column > Next >** keluar kotak **Chart Option**. Pada langkah ini terdapat lembar tab, yaitu **Titels**, **Axes**, **Gridlines**, **Legend**, **Data Label** yang di dalamnya terdapat textbox dan pilihan-pilihan yang dapat diisi. Kemudian klik **Next >** keluar kotak **Chart Location** dengan pilihan **As object in: Sheet1** atau **As new sheet: Chart1 > Finish**. Jika grafik ingin ditempatkan di dalam worksheet, pilih **As object in**. Hasil penggambaran grafik diperlihatkan pada Gambar 1.5.



Gambar 1.4 Kotak dialog Chart Wizard

Gambar grafik dibagi menjadi tiga bidang gambar, yaitu **Chart Area**, **Plot Area** dan **Legend Area**, seperti yang diperlihatkan dalam Gambar 1.5. Anda dapat melakukan pengaturan sendiri pada gambar grafik yang sudah dibuat dengan dua cara. Cara pertama dengan mengklik dua kali pada grafik, dan pengaturan selanjutnya menggunakan kotak dialog **Format**. Sedangkan cara kedua adalah dengan mengklik kanan bidang gambar, dan pengaturan selanjutnya menggunakan perintah-perintah dalam kotak menu. Sebagai contoh, dengan mengklik dua kali pada salah satu aksis maka akan keluar kotak dialog **Format Axis** dengan tab **Pattern**, **Scale**, **Font**, **Number** dan **Alignment**.



Gambar 1.5 Contoh grafik tipe XY dengan sub-tipe smoothed lines

Gambar grafik dapat di-copy, dipindahkan maupun diubah ukurannya. Untuk mengubah ukuran grafik, dapat dilakukan dengan men-drag mouse pada bidang gambar. Pilih dengan mengklik bidang yang akan diubah ukurannya, kemudian arahkan pointer pada kotak hitam kecil (di sudut atau tengah-tengah) hingga keluar tanda "↕↔". Drag mouse ke arah dalam untuk memperkecil dan ke arah luar untuk memperbesar bidang gambar.

Untuk memindahkan grafik ke tempat lain dalam sheet, dapat dilakukan dengan men-drag pada Chart Area. Sedangkan untuk mengopi, memindahkan grafik dan menempatkannya di sheet lain (Cut-Copy-Paste), dapat dilakukan dengan perintah dalam menu pop-up.

BAB 2

FUNGSI

Sebuah fungsi bekerja berdasarkan argumen yang diberikan. Argumen ditulis sesudah nama fungsi, diberi tanda kurung, dan merupakan nilai yang digunakan oleh fungsi untuk mengerjakan operasinya. Argumen yang digunakan dapat merupakan nilai numerik, teks, referensi atau range sel, nama, label maupun fungsi.

Fungsi-fungsi dalam Excel dikelompokkan ke dalam kategori fungsi untuk berbagai aplikasi perhitungan seperti matematika dan trigonometri, statistik, finance, logika, dan sebagainya. Selain itu, Excel juga menyediakan macrosheet dan aplikasi Visual Basic untuk membuat fungsi yang didefinisikan sendiri oleh pemakainya.

2.1 Fungsi Numerik

ABS

Menghasilkan nilai absolut dari suatu bilangan.

Penulisan:

ABS(bilangan)

Contoh:

ABS(4) = 4

ABS(-4) = 4

SQRT(ABS(-81)) = 9

INT

Pembulatan bilangan ke bawah ke integer terdekat.

Penulisan:

INT(bilangan)

Contoh:

INT(8.9) = 8

INT(-8.9) = -9

TRUNC

Membuat bilangan menjadi sebuah integer dengan menghilangkan bagian desimalnya. Apabila hanya ingin menghilangkan beberapa bagian dari desimalnya, masukkan jumlah desimal yang diinginkan sesudah bilangan.

Penulisan:

TRUNC(bilangan, jumlah desimal)

Contoh:

TRUNC(8.9) = 8

TRUNC(-8.9) = -8

TRUNC(PI()) = 3

TRUNC(PI(),3) = 3.141

ROUND

Pembulatan bilangan dengan jumlah desimal yang diinginkan.

Penulisan:

ROUND(bilangan, desimal)

Contoh:

ROUND(3.25,1) = 3.3

ROUND(3.247,1) = 3.2

ROUND(3.247,0) = 3

ROUND(32.47,-1) = 30

ROUNDDOWN

Pembulatan bilangan ke bawah dengan jumlah desimal yang diinginkan.

Penulisan:

ROUNDDOWN(bilangan, jumlah desimal)

Contoh:

ROUNDDOWN(3.3,0) = 3

ROUNDDOWN(66.8,0) = 66

ROUNDDOWN(3.14159,3) = 3.141

ROUNDDOWN(-5.24621,2) = -5.24

ROUNDDOWN(1550.24621,-2) = 1500

ROUNDUP

Pembulatan bilangan ke atas dengan jumlah desimal yang diinginkan.

Penulisan:

ROUNDUP(bilangan, jumlah desimal)

Contoh:

ROUNDUP(3.3,0) = 4

ROUNDUP(66.8,0) = 67

ROUNDUP(3.14159,3) = 3.142

ROUNDUP(-5.24621,2) = -5.25

ROUNDUP(1550.24621,-2) = 1600

2.2 Fungsi Logika

IF

Fungsi ini mempunyai dua hasil nilai, yaitu TRUE (benar) dan FALSE (salah), dengan tes logika apabila terpenuhi, akan dikerjakan dari nilai yang pertama. Apabila hasil logikanya adalah sebuah teks, maka sebelum dan sesudah teks harus diberi tanda petik ("teks").

Penulisan:

IF(tes logika, nilai jika benar, nilai jika salah)

Contoh 1:

Apabila nilai ujian siswa ≤ 55 maka dinyatakan gagal dan di atas nilai itu lulus.

	A	B	C
1	Nama	Nilai	Keterangan
2	Magda Denawati	51	GAGAL
3	Jeni Arnis	93	LULUS
4	Joe Manik	90	LULUS
5	Malik Hasan	70	LULUS
6	Anto Sirat	37	GAGAL
7			

C2:=IF(B2<=55,"GAGAL","LULUS")

Hasil copy sel C2
ke C3 s.d. C6

Contoh 2: Tes logika bertingkat

Siswa dengan kriteria gagal namun nilainya >45 masih dapat mengambil tugas perbaikan.

	A	B	C
1	Nama	Nilai	Keterangan
2	Magda Denawati	51	PERBAIKAN
3	Jeni Arnis	93	LULUS
4	Joe Manik	90	LULUS
5	Malik Hasan	70	LULUS
6	Anto Sirat	37	GAGAL
7			

C2:=IF(B2<=45,"GAGAL",IF(B2<=55,"PERBAIKAN","LULUS"))

Hasil copy sel C2
ke C3 s.d. C6

AND

Hasilnya adalah TRUE jika semua argumennya TRUE, dan menjadi FALSE jika satu atau lebih argumennya FALSE. Biasanya dipakai bersama fungsi IF.

Penulisan:

AND(logika1, logika2, ...)

Contoh:

AND(2*2=4,2+2=4) hasilnya TRUE

AND(2<100,4<100,102<100) hasilnya FALSE

Contoh:

Masih dengan siswa yang sama seperti contoh di atas, namun untuk mata kuliah yang berbeda, di mana sekarang diberikan tugas. Nilai tugas hanya berpengaruh untuk memperbaiki nilai ujian siswa antara > 40 dan ≤ 55 . Rumusan yang dipakai, 2 x Nilai Ujian + 1 x Nilai Tugas dibagi 3, dengan pembulatan.

	A	B	C	D
1	Nama	Nilai Ujian	Nilai Tugas	Nilai Akhir
2	Magda Denawati	60	56	60
3	Jeni Arnis	55	72	61
4	Joe Manik	60	65	60
5	Malik Hasan	40	60	40
6	Anto Sirat	46	64	52
7				

D2:=IF(AND(B2>40,B2<=55),
ROUND((2*B2+C2/3,0),B2)

Hasil copy sel D2
Ke sel D3 s.d. D6

COUNTIF

Menghitung jumlah sel dalam range sel yang sesuai dengan kriteria yang diberikan.

Penulisan:

COUNTIF(range sel, kriteria)

Contoh:

Mencari jumlah siswa dengan kriteria nilai tertentu dari data tabel.

COUNTIF(D2:D6,60) = 2

COUNTIF(D2:D6,">60") = 1

OR

Hasilnya TRUE jika salah satu argumennya TRUE, dan menjadi FALSE jika semua argumennya FALSE. Biasanya dipakai bersama fungsi IF.

Penulisan:

OR(logika1, logika2, ...)

Contoh:

OR(2*2=4,2+2 =4)

hasilnya TRUE

OR(2<100,4<100,102<100)

hasilnya TRUE

2.3 Fungsi Pembacaan Tabel

VLOOKUP

Mencari nilai yang sesuai nilai acuan berdasarkan nomor indeks kolom dalam data tabel yang disusun secara vertikal.

Penulisan:

VLOOKUP(nilai acuan, range tabel, nomor indeks kolom, logika)

Contoh:

Hasil perhitungan suatu pengujian dipengaruhi oleh ukuran dan berat alat-alat (tools) yang dipakai dalam pengujian tersebut. Berikut contoh fungsi yang digunakan untuk membaca data ukuran dan berat ring berdasarkan kalibrasi terakhir yang dilakukan.

	A	B	C	D	E	F
1	KALIBRASI RING					
2						
3	No. Ring	Tinggi (cm)	Diameter (cm)	Berat (gram)	Luas (cm ²)	Volume (cm ³)
4	1	1.950	5.000	32.850	19.635	38.288
5	2	2.000	5.000	32.000	19.635	39.270
6	3	2.000	5.000	33.440	19.635	39.270
7	4	2.000	5.000	34.980	19.635	39.270
8	5	1.950	5.000	33.700	19.635	38.288
9	6	1.925	4.985	32.000	19.517	37.571
10	7	2.000	5.000	33.000	19.635	39.270
11	8	1.975	5.125	34.000	20.629	39.711
12						
13						

	No. Ring	Diameter (cm)	Berat (gram)
14			
15	3	5.000	33.440
16	7	5.000	33.000

12	A	B	C	D
13				
14	No Ring	Diameter (cm)	Berat (gram)	
15	3	=VLOOKUP(B15,\$A\$4:\$F\$11,3)	=VLOOKUP(B15,\$A\$4:\$F\$11,4)	
16	7	=VLOOKUP(B16,\$A\$4:\$F\$11,3)	=VLOOKUP(B16,\$A\$4:\$F\$11,4)	

Logika mempunyai nilai TRUE atau FALSE. Jika TRUE (atau dikosongkan), data pada kolom pertama harus diurut semakin membesar (...,-2,-1,0,1,2,...) karena VLOOKUP dapat memberikan hasil yang salah. Jika FALSE, data dalam tabel tidak perlu diurut. Untuk kondisi TRUE, apabila nilai acuan tidak benar-benar sama dengan data tabel, VLOOKUP akan mencari nilai yang terdekat. Sebaliknya, FALSE akan mencari nilai yang benar-benar sama, kalau tidak ditemukan, hasilnya adalah #N/A.

HLOOKUP

Mencari nilai yang sesuai nilai acuan berdasarkan nomor indeks baris dalam data tabel yang disusun secara horizontal. Penerapan fungsi ini sama seperti VLOOKUP.

Penulisan:

HLOOKUP(nilai acuan, range tabel, nomor indeks baris, logika)

2.4 Fungsi Teks

LEFT, RIGHT, MID

Mengubah suatu karakter dengan jumlah karakter yang diinginkan.

=LEFT(teks, jumlah karakter)

Pengambilan karakter mulai dari kiri.

=**RIGHT**(teks, jumlah karakter)

Pengambilan karakter mulai dari kanan.

=**MID**(teks, posisi awal karakter, jumlah karakter)

Pengambilan karakter dari bagian tengah.

Contoh:

	A	B	C
1	ENGINEERING	ENG	B1:=LEFT(A1,3)
2	745.56	RING	B2:=RIGHT(A1,4)
3		ENGINEER	B3:=MID(A1,1,8)
4		745.5	B4:=LEFT(A2,5)

CONCATENATE

Menggabungkan data dari dua atau lebih sel ke dalam satu sel yang baru.

Contoh:

	A	B	C	D
1				
2	Civil	Engineer		Civil Engineer
3				
4		Titik		Koordinat
5	x	y	z	
6	2	4	0	(2,4,0)
7	2	4	-1.5	(2,4,-1.5)
8	2 m	4 m	-1.5 m	(2,4,-1.5)
9				

D2:=CONCATENATE(A2," ",B2)

=CONCATENATE("(",A6," ",B6," ",C6,")")

Hasil copy sel D6 ke sel D7 dan D8

2.5 Fungsi Pemakai

Selain fungsi-fungsi yang sudah ada di Excel, adakalanya kita memerlukan suatu fungsi untuk mengerjakan perhitungan yang kita tentukan sendiri. Fungsi seperti ini disebut fungsi pemakai atau user-defined function.

Fungsi ini akan ditambahkan oleh Excel ke dalam koleksi fungsinya dan dapat digunakan seperti halnya sebuah fungsi built-in.

Sebagian besar fungsi pemakai dibuat dengan mengombinasikan antara formula dan fungsi built-in. Fungsi pemakai dalam Excel dapat dikerjakan dengan menggunakan macrosheet dan aplikasi Visual Basic. Macrosheet terlihat seperti spreadsheet, namun di dalamnya berisi fungsi-fungsi dan perintah sebagai ganti data input. Bahasa yang digunakan di sini merupakan bahasa program yang orisinal dari spreadsheet. Sedangkan Visual Basic adalah bahasa program dalam lingkup Microsoft Windows yang mirip dengan bahasa Basic atau bahasa komputer lainnya, namun banyak dilengkapi perintah-perintah visual. Gambar 2.1 memperlihatkan fungsi COSD untuk fungsi cosinus sudut dalam derajat yang dibuat dengan Macrosheet dan Visual Basic.

	A
1	COSD
2	=RESULT(1)
3	=ARGUMENT("X",1)
4	=RETURN(COS(PI()*X/180))

Const Pi = 3.141592654

Function Cosd (x)

Cosd = Cos(Pi * x / 180)

End Function

Gambar 2.1 Fungsi dalam Macrosheet (kiri) dan Visual Basic (kanan)

2.5.1 Membuat Fungsi dengan Macrosheet

Sebelum menuliskan fungsi, terlebih dahulu kita menyisipkan sebuah macrosheet ke dalam workbook. Klik kanan mouse pada tab sheet, dan dari kotak menu klik **Insert > Excel 4.0 Macro > OK** sehingga tampil macrosheet dengan nama Macro1. Sheet ini sepintas mirip dengan spreadsheet namun kolom-kolom yang terdapat di sini cukup lebar sebagai tempat menuliskan fungsi. Penulisan bahasa macro di sini mempunyai struktur yang spesifik, yaitu pada baris pertama ditempatkan nama fungsi (contoh COSD dalam Gambar 2.1). Kemudian pada baris kedua adalah definisi tipe outputnya, bisa bilangan, teks, fungsi logika, referensi, pesan error, sebuah array atau kombinasinya dengan memakai statement **RESULT**, diikuti tipe data. Beberapa jenis tipe data dapat dilihat dalam Tabel 2.1. Jika result menghasilkan sebuah bilangan atau teks, tipe datanya = 3. Jika data tipe dikosongkan, secara default diasumsikan = 7 (untuk bilangan, teks dan logika). **RESULT** dikerjakan sebelum mengisi formula.

Seperti fungsi built-in, fungsi pemakai mempunyai variabel input, yang secara individu didefinisikan oleh statement **ARGUMENT** pada kolom di bawah **RESULT**. **ARGUMENT** menentukan nama dan tipe data setiap input variabelnya. Nama sebagai teks ditulis di antara 2 tanda petik diikuti koma kemudian tipenya (contoh: "Data", 1). Sama seperti **RESULT**, definisi input dalam **ARGUMENT** bisa bilangan, teks, fungsi logika, referensi, pesan error, sebuah array atau kombinasinya.

Setelah **ARGUMENT**, pada langkah berikut pemakai (user) menentukan formula-formula yang digunakan dalam operasi utama sebagai fungsi variabel inputnya. Formula-formula ini memakai variabel input yang didefinisikan **ARGUMENT**, merujuk nama atau lokasi sel dalam worksheet. Misalnya "x" di dalam suatu kolom pada worksheet untuk sejumlah data sudut COSD. Langkah fungsi ini diakhiri dengan statement **RETURN**, untuk mengakhiri perhitungan dan kembali ke worksheet. Selanjutnya menamakan macro dengan **Insert > Name > Define** atau **Create**, agar dapat dijalankan.

Tabel 2.1 Tipe Data untuk Fungsi Dalam RESULT dan ARGUMENT

Nilai	Tipe Data
1	Bilangan
2	Teks
4	Logika
8	Referensi
16	Error
64	Array

2.5.2 Membuat Fungsi dengan Aplikasi Visual Basic

Fungsi pemakai dalam Visual Basic (VB) dikerjakan dengan terlebih dulu menyisipkan sebuah module dengan **Tools > Macro > Visual Basic Editor > Insert > Module**. Jendela Visual Basic Editor akan memperlihatkan sebuah module di mana macro dikerjakan dengan nama Module1. Penulisan fungsi dimulai dengan kata **Function** yang diikuti namanya dan diakhiri dengan **End Function**. Tipe fungsi dapat merupakan Single/Double atau Variant bergantung pada tipe data yang dihasilkan, apakah sebuah bilangan atau sebuah array. Tipe data dalam VB dapat dibedakan menjadi beberapa macam, seperti disajikan pada Tabel 2.2. Pemilihan tipe data ini selain untuk mendefinisikan sebuah fungsi dan argumennya, juga berhubungan dengan

pemakaian memori komputer oleh VB yang besarnya bergantung pada jangkauan nilai dari tipe data tersebut.

Sesudah fungsi dan namanya, kemudian kita mendefinisikan sejumlah argumen. Jika argumen ini tidak didefinisikan, akan digolongkan sebagai Variant. VB sendiri mempunyai banyak fungsi built-in yang mempunyai kata yang mirip dengan fungsi built-in dalam macrosheet. Beberapa fungsi built-in macrosheet dapat digunakan dengan memakai awalan **Application**, seperti diperlihatkan pada Gambar 2.2. Jadi dengan VB, kapabilitas dari macrosheet dapat lebih ditingkatkan dan lebih diperluas. Fungsi dalam VB akan secara otomatis terhubung dengan workbook tanpa perlu mendefinisikan namanya.

Perhatikan fungsi pemakai Garis pada Gambar 2.2, kemudian buatlah fungsi itu dan jalankan di dalam worksheet. Bandingkan hasilnya dengan fungsi **Linest** pada Gambar 1.3. Hasil keduanya ternyata sama. **Linest** adalah sebuah fungsi built-in Excel, sedangkan fungsi Garis dibuat dalam Visual Basic memakai fungsi built-in macrosheet.

Tabel 2.2 Tipe Data Dalam Visual Basic

Tipe Data	Jangkauan Nilai
Integer (bilangan bulat)	-32,768 s.d. 32,767
Long (bilangan bulat)	-2,147,483,648 s.d. 2,147,483,547
Single (bilangan desimal)	-3.402823E38 s.d. -1.401298E-45 (nilai -) 1.401298E-45 s.d. 3.402823E38 (nilai +)
Double (bilangan desimal)	-1.79769313486232E308 s.d. -4.94065645841247E-324 (-) 4.94065645841247E-324 s.d. 1.79769313486232E308 (+)
Currency (nilai mata uang)	-922,337,203,685,477.5808 s.d. 922,337,203,685,477.5807
String (teks)	0 s.d. 2 miliar karakter
Byte (bilangan bulat)	0 s.d. 255
Boolean (logika)	TRUE atau FALSE
Date (tanggal dan waktu)	1 Januari 100 s.d. 31 Desember 9999
Object (referensi objek)	mengacu pada objek
Variant	semua tipe data

```

Function GARIS (Y,X) As Variant
    Slope = Application. SLOPE (Y,X)
    Intercept = Application. INTERCEPT (Y,X)
    Garis = Array (Slope, Intercept)
End Function

```

Gambar 2.2 Fungsi built-in macrosheet dalam Visual Basic

2.5.3 Alur Program

Sebuah program dibuat untuk dapat menangani setiap perubahan kondisi, baik dari data input yang diberikan maupun perubahan yang terjadi selama proses program. Oleh karena itu, diperlukan kontrol agar prosesnya berada pada statement yang ditetapkan. Berikut ini beberapa pemilihan alur dalam program yang dapat dipakai dengan memperhatikan kondisi yang ada. Semua perintah-perintah di bawah ini dikerjakan dalam Visual Basic, dan beberapa di antaranya juga mirip dengan yang dipakai dalam macrosheet.

Pengulangan

Do While ... Loop

Mengulang suatu perintah selama kondisinya masih bernilai True. Apabila sudah bernilai False, program akan keluar dari pengulangan dan pindah ke kode program berikutnya. Bentuk yang lain adalah **Do ... Loop While**.

Penulisan:

Do While condition

```

[Statements]
[Exit Do]
[Statements]

```

Loop

Contoh:

```

Sub Command()
    II = 1
    Do While II <= 5
        Cells(1 + II, 1) = II
        II = II + 1
    Loop
End Sub

```

Do Until ... Loop

Mengulang suatu perintah hingga kondisi menjadi True. Apabila sudah bernilai True, pengulangan akan dihentikan. Bentuk yang lain adalah **Do ... Loop Until**.

Penulisan:

Do Until condition

```

[Statements]
[Exit Do]
[Statements]

```

Loop

Contoh:

```

Sub Command()
    II = 1
    Do Until II > 10
        Cells(1 + II, 1) = II
        II = II + 1
    Loop
End Sub

```

For ... Next

Mengulang suatu perintah dengan nilai yang ditetapkan.

Penulisan:

For counter = start To end [Step step]

```

[Statements]
[Exit For]
[Statements]

```

Next [counter]

Contoh:

```

Sub Command()
    IJ = 1
    For II = 1 To 5
        Cells(1 + II, 1) = IJ
        IJ = IJ + 1
    Next II
End Sub

```

While ... Wend

Menjalankan suatu statement dalam pengulangan selama kondisinya masih bernilai True. Dalam macrosheet dikerjakan dengan **While...Next**.

Penulisan:

While condition

[Statements]

Wend

Contoh:

```
Sub Command()  
  II = 1  
  While II <= 5  
    Cells(1 + II, 1) = II  
    II = II + 1  
  Wend  
End Sub
```

For Each ... Next

Mengulang suatu perintah sebanyak anggota dalam data array, dengan jumlah yang tidak diketahui.

Penulisan:

For Each element In array

[Statements]

[Exit For]

[Statements]

Next [element]

Contoh:

```
Function Command(Yarray, Xarray) As Variant  
  Dim N As Integer, x As Single, y As Single  
  N = 0  
  x = 0  
  y = 0  
  For Each c In Xarray  
    N = N + 1  
    x = x + Xarray(N)  
    y = y + Yarray(N)  
  Next  
  a = x * x
```

```
b = y * x  
Command = Array(a, b)  
End Function
```

Pencabangan

If ... Then ... Else

Penggunaannya sama seperti fungsi built-in logika dalam spreadsheet, tetapi di sini cara penulisan yang dipakai dalam pemrograman.

If condition (1) Then

Statement (1)

ElseIf condition (2) Then

Statement (2)

Else

Statement (3)

End If

Pencabangan ini biasanya juga dipakai dalam suatu pengulangan. Apabila kondisinya bernilai True, program kemudian keluar dengan perintah **Exit**. Bentuk **If ...** juga dapat disertai dengan **And**, **Or**, atau **Not** untuk menyatakan ekspresi dan relasi antara dua ekspresi.

For ... or Do Until ...

Statement (1)

If expression (1) And or Or expression (2) Then

Statement (2)

Exit For or Exit Do or Exit Function or Exit Sub

End If

Next ... or Loop

GoSub ... GoTo ... Return

Digunakan pada alur program yang terdiri atas satu atau lebih sub-program. Sub-program adalah perintah yang dibuat untuk melakukan tugas-tugas yang spesifik. Manfaatnya agar penulisan program secara keseluruhan menjadi

lebih terfokus. Prosedur sub diberi nomor sebagai alamat dari baris sub ini, dan pada bagian akhir diberi perintah Return untuk kembali ke statement sebelumnya. Perintah GoTo dipakai untuk melengkapi alur program ini.

Contoh:

```

20      ""
      GoTo 40
      e = KM
      c = e * a:Return
40      ""
      a = NM
      GoSub 20
      F = h * c
      ""
      a = NP
      GoSub 20
      G = j * c
      Hasil = F + G
End Sub

```

Call ... Sub

Perintah ini mirip dengan GoSub, GoTo, dan Return. Namun di sini sub diberi nama dan digunakan kata Call untuk memanggil dan memproses prosedur sub-nya.

Contoh 1: Prosedur Sub → Function

```

Sub Luas(x, y, L)
    L = x * y
End Sub

Function Volume(a, b, t) As Variant
    If a <= 0 Or b <= 0 Then
        Volume = "#VALUE!"
        Exit Function
    Else
        Call Luas(a, b, L)
        Volume = L * t
    End If
End Function

```

Contoh 2: Prosedur Sub → Sub

```

Sub Luas(x, y, L)
    L = x * y
End Sub

Sub Vol()
    Dim a, b, t As Single
    a = Cells(12, 6)

```

```

b = Cells(12, 7)
t = Cells(12, 8)
If a <= 0 Or b <= 0 Then
    Cells(12, 9) = "#VALUE!"
    Exit Sub
Else
    Call Luas(a, b, L)
    Volume = L * t
    Cells(12, 9) = Volume
End If
End Sub

```

2.5.4 Contoh-Contoh Fungsi Pemakai

Berikut ini beberapa contoh fungsi pemakai beserta penggunaannya dalam spreadsheet. Contoh 2, 3, dan 4 mungkin sudah tidak begitu asing apabila kita sering membuat formula dengan fungsi built-in. Namun fungsi yang didefinisikan sendiri akan menjadi sangat spesifik dengan argumen dan formula yang dibuat oleh pemakainya.

Contoh 1

Menghitung luas persegi panjang di mana Luas = Panjang x Lebar.

Dalam macro hal ini diekspresikan sebagai fungsi (a,b) = a x b.

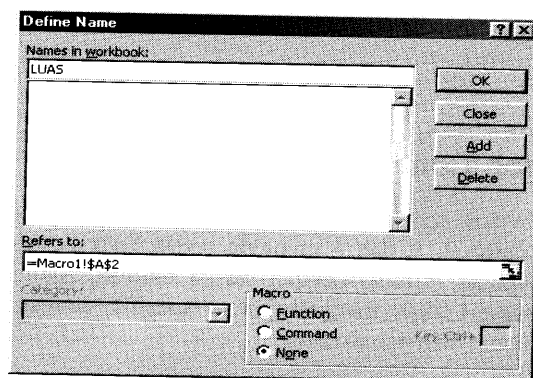
Ini contoh pertama di mana kita akan membuat sebuah fungsi pemakai. Langkah pertama, menyisipkan macrosheet dengan **Insert > MS. Excel 4.0 Macro > OK**. Kemudian tulis macro-nya sebagai berikut.

	A
1	LUAS
2	=RESULT(1)
3	=ARGUMENT("a",1)
4	=ARGUMENT("b",1)
5	=RETURN(a*b)

Setelah itu kita harus mendefinisikan nama fungsi ini agar dikenal oleh Excel. Langkah-langkahnya sebagai berikut.

1. Tempatkan pointer pada nama fungsi, yaitu **LUAS** pada sel A1.
2. Kemudian klik **Insert > Define > Name > OK**. Nama fungsi secara otomatis akan mengacu ke sel A1, yaitu **LUAS**.

3. Apabila Anda ingin menambahkan fungsi ini ke dalam menu **Function**, pada option button klik **Function** atau **Command**, kemudian pilih kategori untuk fungsi Anda atau sebuah shortcut key untuk itu.



Sekarang kembali ke spreadsheet. Anda dapat langsung menggunakan fungsi LUAS dengan merujuk pada alamat sel atau nama sel. Pemakaian fungsi LUAS dalam worksheet dikerjakan seperti berikut ini.

	A	B	C	D	E
1	a	b	Luas		
2		2	3	6	C2=LUAS(a,b)
3					

Contoh 2

Menghitung luas bidang yang dibatasi garis linear $y = mx + c$, aksis-X, $x = a$ dan $x = b$, untuk $c \geq 0$. Hal ini akan dilakukan dengan penyelesaian integral.

$$L = \int_a^b (mx + c) dx$$

Dalam macro, L diekspresikan sebagai:

$$\text{fungsi } (m, c, a, b) = (\frac{1}{2} \cdot m \cdot b^2 + c \cdot b) - (\frac{1}{2} \cdot m \cdot a^2 + c \cdot a).$$

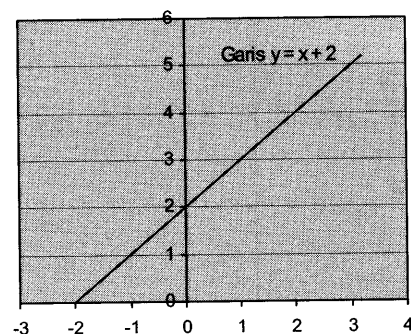
Diminta mencari luas yang dibatasi garis $y = x + 2$, aksis X, $a = -1$ dan $b = 3$, dan juga pesan error-nya.

Macro

	A
1	BIDANG
2	=RESULT(1)
3	=ARGUMENT("m",1)
4	=ARGUMENT("c_",1)
5	=ARGUMENT("a",1)
6	=ARGUMENT("b",1)
7	=-c_/m
8	=IF(m<0,IF(OR(a>A7,b>A7),RETURN(#VALUE!)))
9	=IF(m>0,IF(OR(a<A7,b<A7),RETURN(#VALUE!)))
10	=(1/2)*m*a^2+(c_*a)
11	=(1/2)*m*b^2+(c_*b)
12	=RETURN(A11-A10)

Contoh soal dalam grafik

Luas bidang yang dibatasi garis $y = x + 2$, aksis X, $a = -1$, $b = 3$



Pemakaian fungsi dalam worksheet

	A	B	C	D	E	F	G	H
1	m	c	a	b	Luas			
2	1	2	-1	3	12	E2:=BIDANG(m,c,a,b)		
3	1	2	-3	3	#VALUE!	E3:=BIDANG(m,c,a,b)		
4	-0.5	2	1	5	#VALUE!	E4:=BIDANG(m,c,a,b)		
5								

Contoh 3

Seorang dosen menetapkan tingkatan nilai untuk suatu mata kuliah sebagai berikut, A untuk nilai 80 – 100, B = 65 – 79, C = 51 – 64, D = 40 – 50 dan F untuk nilai di bawah 40. Juga akan dipakai tanda + dan – untuk setiap tingkatannya.

Macro

	A
1	NILAI
2	=RESULT(2)
3	=ARGUMENT("G",1)
4	=ARGUMENT("Ax",1)
5	=ARGUMENT("Bx",1)
6	=ARGUMENT("Cx",1)
7	=ARGUMENT("Dx",1)
8	=IF(G>=(2*100/3+Ax/3),RETURN("A+"))
9	=IF(G>=(100/3+2*Ax/3),RETURN("A"))
10	=IF(G>=Ax,RETURN("A-"))
11	=IF(G>=(2*Ax/3+Bx/3),RETURN("B+"))
12	=IF(G>=(Ax/3+2*Bx/3),RETURN("B"))
13	=IF(G>=Bx,RETURN("B-"))
14	=IF(G>=(2*Bx/3+Cx/3),RETURN("C+"))
15	=IF(G>=(Cx/3+2*Bx/3),RETURN("C"))
16	=IF(G>=Cx,RETURN("C-"))
17	=IF(G>=(2*Cx/3+Dx/3),RETURN("D+"))
18	=IF(G>=(Cx/3+2*Dx/3),RETURN("D"))
19	=IF(G>=Dx,RETURN("D-"))
20	=RETURN("F")

Pemakaian fungsi dalam worksheet

	A	B	C
1	Nama Mahasiswa	Nilai	Huruf
2	Magda Denawati	51	C-
3	Jeni Arnis	93	A+
4	Joe Manik	90	A
5	Malik Hasan	70	B
6	Anto Sirat	37	F

C2:=NILAI(B2,80,65,51,40)

hasil copy sel C2 ke sel C3 s.d. C6

Apabila seorang dosen kemudian menetapkan standar lain untuk mata kuliah yang lain, maka cukup dengan mengganti argumen nilai dari fungsi NILAI dalam worksheet.

Contoh 4

Mencari distribusi tegangan di bawah pusat pondasi pelat $b \times l$. Diketahui beban awal $Q = 90$ ton bekerja pada pusat pondasi ukuran 3.0×3.0 meter. Distribusi tegangan diminta digambarkan pada grafik dalam setiap interval kedalaman 0.5 meter.

Contoh soal ini akan diselesaikan dengan persamaan elastis dari Boussinesq – Newmark (1935). Persamaan untuk mendapatkan tegangan di bawah sudut pelat persegi pada kedalaman y adalah:

$$\sigma_y = \sigma_o \frac{1}{4\pi} \left[\frac{2MN\sqrt{V}}{V+V_1} \cdot \frac{V+1}{V} + \tan^{-1} \left(\frac{2MN\sqrt{V}}{V-V_1} \right) \right]$$

di mana

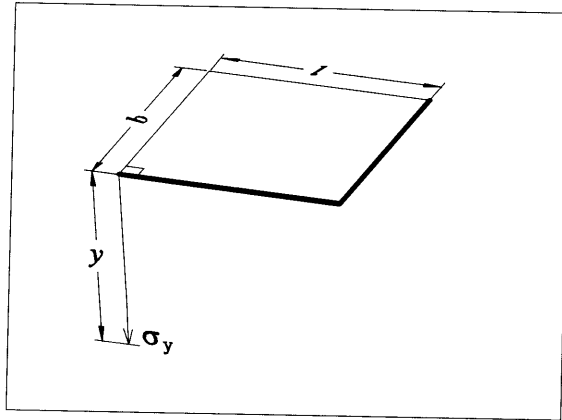
$$M = b/y$$

$$N = l/y$$

$$V = M^2 + N^2 + 1$$

$$V_1 = (MN)^2$$

Apabila $V_1 > V$ maka arctan dalam persamaan perlu ditambah dengan π . Macro akan dibuat dengan aplikasi Visual Basic dan diberi nama Bouss.



Macro

```
Function Bouss(p, b, l, y) As Variant
Const Pi = 3.141592654
m = b / y
n = l / y
v = m ^ 2 + n ^ 2 + 1
vs = (m * n) ^ 2
rad = (2 * m * n * Sqr(v)) / (v - vs)
If vs > v Then at = atn(rad) + Pi Else at = atn(rad)
Bouss = p / (4 * Pi) * (((2 * m * n * Sqr(v)) / (v + vs)) * ((v + 1) / v)) + at)
End Function
```

Macro ini merupakan sebuah fungsi untuk mencari besarnya tegangan σ_y pada setiap kedalaman di bawah sudut pelat persegi panjang. Langkah pertama adalah menjadikan beban terpusat Q menjadi beban merata, dengan membagi Q dengan luas pelat. Kemudian pelat dibagi menjadi beberapa buah bidang dan untuk beban yang terpusat, diperoleh 4 bidang yang sama besar. Jadi untuk empat sudut yang ikut berkontribusi diperoleh tegangan = $4 \sigma_y$.

Pemakaian fungsi BOUSS dalam worksheet dikerjakan seperti berikut.

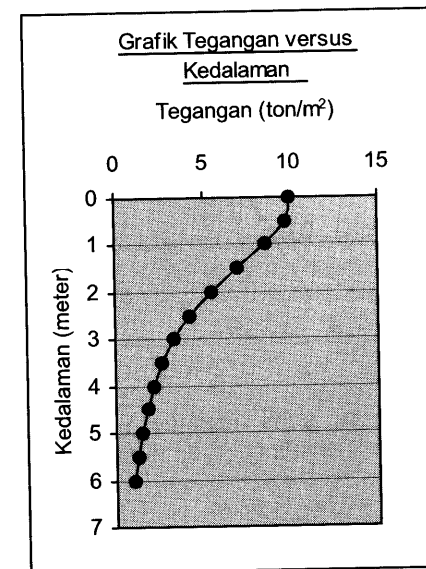
	A	B	C	D	E
1	p	b	l	y	py
2	10	1.5	1.5	0.00*	10.00
3				0.50	9.76
4				1.00	8.63
5				1.50	7.01
6				2.00	5.49
7				2.50	4.28
8				3.00	3.36
9				3.50	2.68
10				4.00	2.17
11				4.50	1.79
12				5.00	1.49
13				5.50	1.26
14				6.00	1.08

unit ton-meter

E2:=4*BOUSS(\$A\$2,\$B\$2,\$C\$2,D2)

Hasil copy sel E2
ke sel E3 s.d. E14

Catatan, *: pembagian dengan 0 tidak diperkenankan dalam perhitungan, oleh karena itu gunakan pendekatan = 0 (ambil misalnya 0.0001).

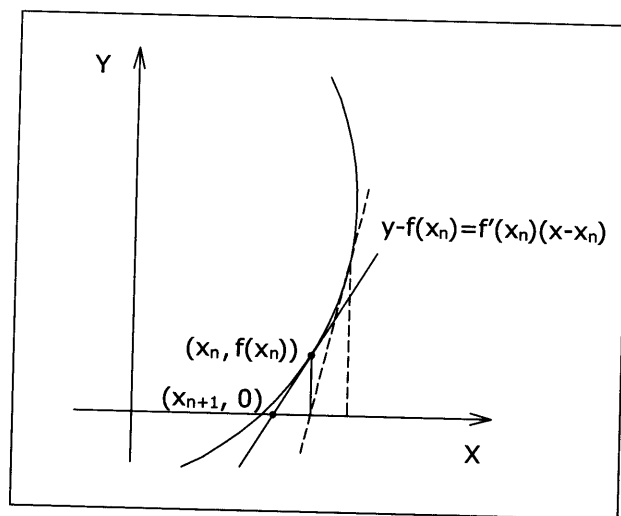


Orientasi kedalaman tanah dalam grafik dibuat bertambah besar ke bawah. Untuk mengerjakan hal ini, klik dua kali pada salah satu nilai aksis-Y hingga keluar kotak dialog **Format Axis**. Klik tab **Scale > Values in reverse order**.

Contoh 5

Mencari akar x pada persamaan: $ax^4 + bx^3 + cx^2 + dx + e = 0$.

Hal ini akan diselesaikan secara numerik dengan metode Newton-Raphson. Proses perhitungan pada metode ini dikerjakan dalam suatu iterasi, yaitu pendekatan nilai x secara bertahap. Fungsi persamaan digambarkan dalam grafik dan dimisalkan memotong aksis-X pada titik p dan $f(x) = 0$ atau $(p, 0)$. Pada tahap awal diambil nilai x_1 untuk mendapatkan garis singgung pada $y = f(x)$ yang memotong aksis-X di titik $(x_2, 0)$, di mana x_2 adalah nilai pendekatan untuk p . Proses pendekatan berikutnya dilakukan dengan nilai x_2 untuk mendapatkan x_3 dan seterusnya hingga $x_n \rightarrow p$.



Jadi secara induksi diperoleh garis singgung: $y - f(x_n) = f'(x_n)(x - x_n)$. Garis ini memotong aksis-X di titik $(x_{n+1}, 0)$, di mana:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Proses perhitungan dengan metode Newton-Raphson ini kemudian dibuat ke dalam macro dan diberi nama TRIAL. Dalam contoh berikut macro akan dipakai untuk mencari nilai x pada persamaan:

$$-2.79x^3 - 12.11x^2 + 24.51x + 36.68 = 0.$$

Bentuk persamaan ini dapat ditemukan dalam mencari kedalaman tancapan dinding turap dengan metode tumpuan tanah bebas (Free Earth Method). Penyelesaiannya dalam worksheet dikerjakan seperti berikut.

	A	B	C	D	E	F	G	H	I
1	ax^4	bx^3	cx^2	dx	e	$= 0$			
2	a	b	c	d	e	$x =$			
3	0	-2.79	-12.11	24.51	36.68	2.23247	F3=TRIAL(a,b,c,d,e)		
4									

Macro

```
Function TRIAL(a, b, c, d, e) As Variant
n = 1000
I = 1
x = 100
Do
    funct = (a * x ^ 4) + (b * x ^ 3) + (c * x ^ 2) + (d * x) + e
    deriv = (4 * a * x ^ 3) + (3 * b * x ^ 2) + (2 * c * x) + d
    xn = x - (funct / deriv)
    If funct = 0 Or (Abs(xn - x) < 0.00001) Then
        TRIAL = xn
    Exit Function
    End If
    I = I + 1
    x = xn
Loop Until I = n
TRIAL = "Divergen!"
Exit Function
End Function
```

Contoh 6

Tabel X adalah data bilangan random dari 0 – 100. Anda diminta menghitung banyaknya data dalam Tabel X dengan kriteria nilai dari Y hingga Z.

	A	B	C	D	E	F	G
1			Tabel X				
2	77.58	73.27	22.21	70.40	81.40	52.57	98.34
3	68.58	21.86	62.37	37.61	51.24	12.41	18.55
4	60.92	22.26	35.01	23.76	63.34	32.81	83.16
5	98.27	63.85	30.97	34.98	64.44	70.49	17.96
6	71.23	79.63	74.21	100.00	37.95	3.94	95.27
7	10.62	20.69	5.82	74.81	95.09	37.48	0.74

Macro

```
Function DIST(Xarray, Y, Z) As Variant
Dim N, Num As Integer
N = 0
For Each c In Xarray
    N = N + 1
    If Xarray(N) >= Y And Xarray(N) <= Z Then Num = Num + 1
Next
DIST = Num
End Function
```

Pemakaian fungsi dalam worksheet

	I	J	K	
2	0.0	50.0	19	K2:=DIST(X,J2,K2)
3	50.0	100.0	23	K3:=DIST(X,J3,K3)
4	0.0	100.0	42	K4:=DIST(X,J4,K4)

Argumen (X) di sini adalah argumen array, sehingga penulisannya juga dapat menjadi (A2:G7) atau range tertentu: (B2:C7), (A7:G7) dan sebagainya. Satu kemudahan membuat fungsi dengan spreadsheet karena memang ditunjang oleh worksheetnya untuk pembuatan nama variabel, penulisan fungsi dan menempatkan hasilnya.

Catatan:

1. Karakter "c" dan "r" tidak dapat digunakan untuk nama variabel dalam macrosheet, karena C dan R mewakili kata perintah untuk penentuan kolom (Columns) dan baris (Rows). Apabila Anda mendefinisikannya dalam workbook, secara default Excel akan mengubahnya menjadi "c_" dan "r_".
2. Apabila sebuah fungsi mempunyai lebih dari satu hasil jawaban (hasil array), maka digunakan fungsi Index untuk menurunkan hasilnya. Contohnya seperti yang diperlihatkan formula array GARIS (Gambar 2.2) atau LINEST (Gambar 1.3) yang mempunyai dua indeks hasil secara berurutan untuk SLOPE dan INTERCEPT. Contoh sederhana misalnya, sebuah fungsi AKAR untuk mencari akar-akar real x_1 dan x_2 pada persamaan $ax^2 + bx + c = 0$. Penulisan macro hasil fungsi menjadi: AKAR = Array(x1, x2) dalam VB atau RETURN(x1_:x2_) dalam macrosheet.

2.6 Circular Reference

Ketika formula terhubung dengan selnya sendiri baik secara langsung atau tidak langsung, maka kondisi ini disebut circular reference. Excel secara default tidak dapat memproses formula yang berisi circular reference sampai kita mengaktifkan **Iteration** pada **Tools > Options > Calculation**. Iterasi adalah pengulangan perhitungan sampai kondisi numeriknya terpenuhi. Proses perhitungannya akan dijalankan berdasarkan jumlah iterasi yang dimasukkan pada textbox **Maximum iterations** dan batasan (maksimum) untuk hasil dua iterasi terakhir pada **Maximum change**. Excel akan berhenti menghitung bila salah satu kondisi sudah terpenuhi atau mana yang dicapai lebih dulu.

Dalam aplikasinya, circular reference dapat digunakan misalnya untuk penyelesaian stabilitas lereng cara Bishop dimana faktor keamanan berada pada kedua ruas persamaan. Atau penyelesaian untuk rembesan (seepage) di bawah bendungan dengan metode finite difference. Sel dengan circular reference diperlihatkan dalam spreadsheet seperti berikut:

Contoh 1

	A	B
1	4	
2	=B2	=(A1+A2)/3

Hasil = 1 x iterasi

	A	B
1	4	
2	0	1.33333

5 x iterasi

	A	B
1	4	
2	1.97531	1.99177

100 x iterasi

	A	B
1	4	
2	2	2

Secara singkat, dapat dijelaskan di sini mengenai metode iterasi Excel untuk menghitung contoh 1. Formula pada sel A2:B2 saling terkait dan mempunyai bentuk: $x = f(x)$. Dalam iterasinya, dihitung melalui persamaan: $x_{n+1} = f(x_n)$, di mana: x_n = perkiraan nilai x pada iterasi ke n
 x_{n+1} = perkiraan nilai x yang baru pada iterasi ke $n+1$
 n = jumlah iterasi

Formula pada sel A2:B2 ditulis: $x_{n+1} = (4 + x_n)/3$.

Pada iterasi pertama ($n=1$), x_1 diambil = 0 $\rightarrow x_2 = (4 + 0)/3 = 1.33333$

Pada $n = 2$, x_2 diambil = 1.33333 $\rightarrow x_3 = 1.77778$

Pada $n = 5$, $x_5 = 1.97531 \rightarrow x_6 = 1.99177$

Dan seterusnya.

Excel akan mengerjakan proses ini berdasarkan jumlah iterasi (n) dan batasan ($|x_{n+1} - x_n|$) yang Anda tetapkan, dan akan berhenti bila hingga salah satu terpenuhi. Analogi ini sama seperti pada proses fungsi TRIAL, namun metode yang dipakai berbeda.

Contoh 2

	A	B	C
1	0	10	10
2	10	=(B1+B3+A2+C2)/4	=B2
3	10	=(B2+B4+A3+C3)/4	=B3
4	0	0	0

1 x iterasi

	A	B	C
1	0	10	10
2	10	5	5
3	10	3.75	3.75
4	0	0	0

5 x iterasi

	A	B	C
1	0	10	10
2	10	8.64128	8.64128
3	10	6.18032	6.18032
4	0	0	0

100 x iterasi

	A	B	C
1	0	10	10
2	10	8.75	8.75
3	10	6.25	6.25
4	0	0	0

Dalam formula sel B2, nilai awal yang diambil untuk B3 dan C2 adalah = 0. Selanjutnya nilai B2 akan dipakai sebagai nilai awal dalam formula sel B3, dimana C3 = 0. Nilai-nilai yang didapat kemudian dipakai sebagai perkiraan nilai yang baru pada iterasi berikutnya.

BAB 3

FUNGSI UNTUK ANALISIS DATA

Excel mempunyai banyak fungsi built-in yang digunakan untuk menganalisis sekumpulan data dari hasil suatu pengujian. Metode yang dipakai juga cukup bervariasi. Dalam bab ini akan diterangkan pemakaian fungsi-fungsi ini, namun dibatasi hanya pada beberapa fungsi yang umum dipakai. Kumpulan data yang ditinjau adalah data model dan data statistik.

3.1 Pemodelan Data

Pemodelan data merupakan hal yang umum dilakukan dalam bidang sains dan teknik. Cara ini digunakan untuk mencari parameter-parameter teoritis yang memberikan hubungan paling tepat antara teori dan hasil pengujian. Tiga cara yang akan dibahas di sini adalah regresi linear, regresi polinomial, dan interpolasi.

3.1.1 Regresi Linear

Regresi linear adalah menentukan garis lurus yang fit atau paling mendekati dengan sejumlah data titik, dengan memberikan hubungan linear antara dua variabel. Metode yang digunakan untuk mendapatkan garis tersebut adalah dengan metode kuadrat terkecil (least square method). Jadi regresi linear terdiri atas titik-titik yang fit dengan sejumlah (n) data titik (x_i, y_i) yang dituliskan ke dalam persamaan garis lurus:

$$y = Ax + B \quad (3.1)$$

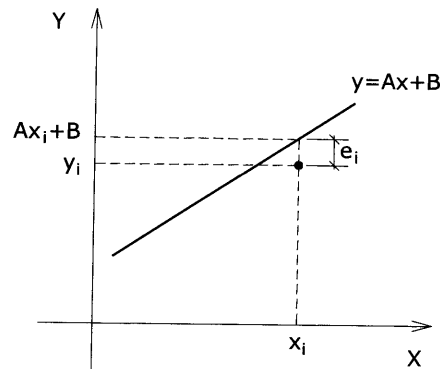
di mana:

A = slope (kemiringan) garis

B = perpotongan garis lurus dengan aksis-Y

Akurasi dari kesesuaian antara garis lurus dengan sejumlah data titik dapat dievaluasi dengan deviasi total E , yang adalah jumlah dari pangkat dua, jarak antara data titik dan titik-titik yang fit.

$$E = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (Ax_i + B - y_i)^2 \quad (3.2)$$



E akan minimum apabila turunan pertamanya terhadap A dan $B = 0$:

$$\frac{\partial E}{\partial A} = 0 \quad \text{dan} \quad \frac{\partial E}{\partial B} = 0 \quad (3.3)$$

Dari Persamaan 3.2 dan 3.3:

$$\frac{\partial E}{\partial A} = 2 \sum_{i=1}^n x_i (Ax_i + B - y_i) = 0$$

dan,

$$\frac{\partial E}{\partial B} = 2 \sum_{i=1}^n (Ax_i + B - y_i) = 0 \quad (3.4)$$

Persamaan 3.4 adalah sistem dua persamaan linear dengan A dan B yang tidak diketahui:

$$A \sum_{i=1}^n x_i^2 + B \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i \quad (3.5)$$

$$A \sum_{i=1}^n x_i + Bn = \sum_{i=1}^n y_i$$

Nilai A dan B adalah:

$$A = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$

$$B = \frac{\sum_{i=1}^n y_i \sum_{i=1}^n x_i^2 - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (3.6)$$

Dalam worksheet, regresi linear dikerjakan dengan fungsi built in SLOPE, INTERCEPT atau LINEST. Macronya dapat dibuat seperti berikut.

	REGRESIL	: Nama fungsi untuk regresi linear
	=RESULT(64)	: Hasil fungsi sebuah array
	=ARGUMENT("Y",64)	: Data y dalam array
	=ARGUMENT("X",64)	: Data x dalam array
n	=COUNT(X)	: Jumlah data dalam array x dan y
Sx	=SUM(X)	: $x_1 + x_2 + \dots + x_n$
Sy	=SUM(Y)	: $y_1 + y_2 + \dots + y_n$
Sxx	=SUMPRODUCT(X,X)	: $x_1^2 + x_2^2 + \dots + x_n^2$
Sxy	=SUMPRODUCT(X,Y)	: $x_1 y_1 + x_2 y_2 + \dots + x_n y_n$
D	=n*Sxx-Sx^2	
	=IF(D=0,RETURN("#DIV/0!"))	: Pesan error
Slope	=(n*Sxy-Sx*Sy)/D	: Nilai A
Intercept	=(Sy*Sxx-Sx*Sxy)/D	: Nilai B
	=RETURN(Slope:intercept)	: Mengeluarkan hasil fungsi

3.1.2 Regresi Polinomial

Fungsi polinom (suku banyak) mempunyai bentuk:

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0 \quad (3.7)$$

di mana m = bilangan bulat positif dan disebut fungsi polinom dengan orde m . Persamaan 3.7 dapat ditulis:

$$f(x) = \sum_{j=0}^m a_j x^j \quad (3.8)$$

Regresi polinomial terdiri atas titik-titik yang fit dengan sejumlah (n) data titik (x_i, y_i) menjadi bentuk polinomial. Regresi linear adalah bentuk khusus dari regresi polinomial dengan $m = 1$, $A = a_1$, dan $B = a_0$. Akurasi kesesuaian antara garis polinomial dengan data titiknya dievaluasi dengan deviasi total E , sebagai penjumlahan pangkat dua, jarak antara data titik dan titik-titik yang fit.

$$E = \sum_{i=1}^n [f(x_i) - y_i]^2 \quad (3.9)$$

E akan minimum jika turunan pertamanya terhadap $a_j = 0$:

$$\frac{\partial E}{\partial a_j} = 0 \quad \text{untuk } j = 0, 1, \dots, m \quad (3.10)$$

ini berarti,

$$\sum_{i=1}^n [f(x_i) - y_i] \frac{\partial f(x_i)}{\partial a_j} = 0 \quad \text{untuk } j = 0, 1, \dots, m \quad (3.11)$$

Melalui beberapa manipulasi aljabar, dari persamaan 3.11 kemudian dapat ditentukan koefisien a_0, a_1, a_2, \dots hingga orde m . Selain E , nilai R^2 juga digunakan sebagai ukuran kesesuaian titik:

$$R^2 = 1 - \frac{\sum_{i=1}^n [y_i - f(x_i)]^2}{\sum_{i=1}^n y_i^2 - \frac{1}{n} \left(\sum_{i=1}^n y_i \right)^2} \quad (3.12)$$

Nilai R^2 bervariasi dari 0 s.d. 1, dan tingkat kesesuaian yang maksimum adalah $R^2 = 1$. Makin tinggi orde polinom, nilai R^2 makin mendekati hingga = 1 ($R^2 = 1$ jika garis regresi berimpit dengan datanya).

3.1.3 Contoh Regresi dalam Grafik

Regresi Linear

Gambar 3.1 menunjukkan pemakaian regresi linear yang diterapkan pada data yang dihasilkan dari pengujian kuat geser. Cara grafis ini dipakai untuk mencari parameter kuat geser, yaitu kohesi (c) dan sudut geser dalam (ϕ) dari massa tanah. Pada umumnya, penggambaran data yang dihasilkan dari pengujian jarang merupakan garis lurus untuk menyatakan selubung keruntuhan. Oleh karena itu, diperlukan garis yang fit untuk mewakili data tersebut. Keadaan tegangan dari massa tanah tersebut tidak akan berada di atas selubung keruntuhan. Dalam uji *direct shear*, selubung keruntuhan adalah suatu garis linear yang dinyatakan dalam persamaan:

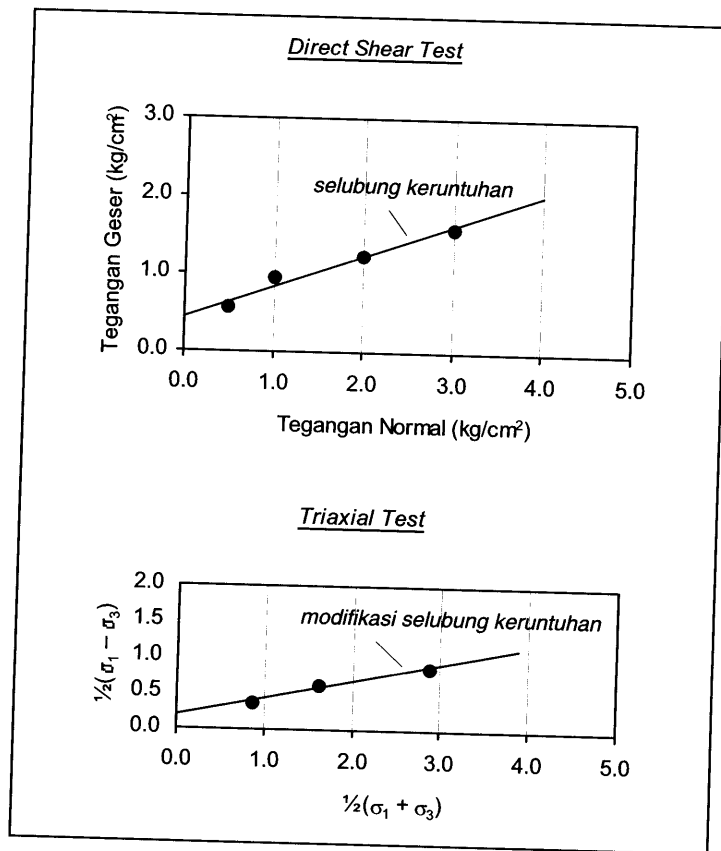
$$\tau = c + \sigma_n \tan \phi$$

di mana:

- τ tegangan geser pada saat runtuh
- c kohesi
- σ_n tegangan normal
- ϕ sudut tahanan geser

Kekuatan geser (τ) di sini adalah fungsi linear terhadap tegangan normal, di mana c dan $\tan \phi$ masing-masing menyatakan perpotongan garis dengan aksis-Y dan slope dari garis.

Hasil yang diperoleh dari uji *direct shear* adalah sejumlah data (x_i, y_i) dari tegangan geser pada saat runtuh untuk setiap tegangan normal. Garis regresi kemudian digambar ke dalam grafik dengan Trendline. Langkah-langkahnya dengan mengklik kanan mouse pada saat pointer berada pada salah satu data titik hingga keluar sebuah kotak menu. Dari kotak menu, klik **Add Trendline > Linear > Options > Forecast > isi Forward dan Backward** untuk memperpanjang trendline > **OK**. Hasilnya seperti tampak pada Gambar 3.1. Trendline adalah persamaan garis yang dikerjakan melalui Persamaan 3.1 sampai 3.12.



Gambar 3.1 Regresi dalam grafik

Perpotongan garis pada aksis-Y adalah nilai c dalam pengujian, dan dihitung dengan formula $=\text{INTERCEPT}(Y,X)$. Slope garis sama dengan $=\text{SLOPE}(Y,X)$. Slope dan intercept adalah nilai A dan B pada Persamaan 3.6. Sudut geser ϕ kemudian dapat dihitung menjadi (dalam derajat):

$$= \text{ATAN}(\text{SLOPE}(Y,X)) * 180/\text{PI}()$$

Beberapa fungsi dapat dipakai untuk maksud yang sama:

$=\text{INDEX}(\text{LINEST}(Y,X),1)$ untuk Slope,

$=\text{INDEX}(\text{LINEST}(Y,X),2)$ untuk Y-intercept,

$=\text{TREND}(Y,X,0,\text{TRUE})$ adalah nilai y yang koresponding dengan $x = 0$.

Kekuatan geser juga dapat dinyatakan dalam tegangan utama besar σ_1 dan kecil σ_3 dalam uji *triaxial*. Dengan memplot $\frac{1}{2}(\sigma_1 - \sigma_3)$ terhadap $\frac{1}{2}(\sigma_1 + \sigma_3)$ untuk setiap kondisi tegangan, akan diperoleh sejumlah data titik (x_i, y_i) . Kondisi tegangan yang fit dengan sejumlah data titik di sini disebut sebagai modifikasi selubung keruntuhan, yang merupakan garis linear yang dinyatakan dalam persamaan:

$$\frac{1}{2}(\sigma_1 - \sigma_3) = a + \frac{1}{2}(\sigma_1 + \sigma_3) \tan \alpha$$

Parameter a dan $\tan \alpha$ masing-masing adalah Y-intercept dan slope dari garis ini. Parameter c dan ϕ kemudian diperoleh melalui hubungan:

$$\phi = \sin^{-1}(\tan \alpha)$$

$$c = \frac{a}{\cos \phi}$$

Regresi dalam Grafik Log

Pada contoh kedua, kita akan mencari relasi data dalam satu series di mana aksis-X-nya dalam skala log seperti pada Gambar 3.3. Di sini Persamaan 3.1 dikerjakan menjadi:

$$y = A \ln(x) + B,$$

di mana:

A dan B adalah konstanta

Sebagai contoh, data yang diambil uji tanah di laboratorium untuk mencari batas cair (liquid limit), yaitu **kadar air tanah pada 25 pukulan**. Berikut beberapa keterangan yang diperlukan.

Kadar air (W) =

$$[(\text{berat tanah asli} - \text{berat tanah kering}) / (\text{berat tanah kering})] \times 100\%.$$

Kadar air' (W') = data W yang fit dalam hubungan semi-log.

Formula yang digunakan untuk mencari data titik yang fit antara serie data kadar air dan N (pukulan) dapat dilihat pada Gambar 3.2. Garis regresi dibuat ke dalam grafik dengan **Add Trendline > Logarithmic > OK**, atau dengan menggunakan **Chart Wizard** dari serie data $N - W'$. Pada $N = 25$ pukulan, titik yang fit diperoleh = 70.93%. Cara lain dengan memasukkan nilai x ke dalam persamaan Trendline $y = -10.887 * \ln(x) + 105.97$. Untuk

menampilkan persamaan ini, klik **Options > Display Equation** pada kotak dialog Add Trendline.

	A	B	C	D	E	F	G
1	LIQUID LIMIT TEST						
2							
3	No.	Berat	Can +	Can +	Pukulan	Kadar	Kadar
4	Set	Can* (gr)	Tanah (gr)	Tanah Kering (gr)		Air (%)	Air' (%)
5		Wc	Ww	Wd	N	W	W'
6	1	8.3	52.46	33.29	14	76.71	77.24
7	2	2.8	55.61	32.96	19	75.10	73.92
8	3	2.8	57.14	35.12	29	68.13	69.31
9	4	7.8	52.72	34.62	36	67.49	66.96

Liquid Limit (%) = 70.93

Slope of flow line = 0.151

3	Kadar	Kadar
4	Air (%)	Air' (%)
5	W	W'
6	=100*(Ww-Wd)/(Wd-Wc)	=TREND(W,LOG10(N),LOG10(E6))
7	=100*(Ww-Wd)/(Wd-Wc)	=TREND(W,LOG10(N),LOG10(E7))

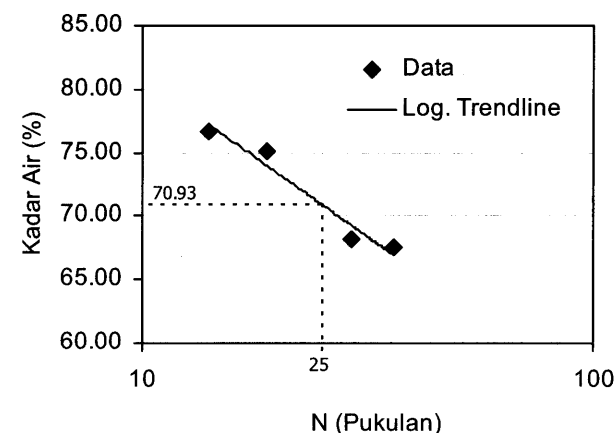
Liquid Limit (%) = =TREND(W,LOG10(N),LOG10({25}))

Slope of flow line = =-SLOPE(LOG(W),LOG(N))

* Can: wadah untuk diisi tanah.

Gambar 3.2 Contoh data uji Liquid Limit dan formula yang digunakan

Liquid Limit Sheet



Gambar 3.3 Contoh regresi dalam grafik semi-log

Pada contoh ketiga, kita akan bekerja dengan grafik dalam skala log-log. Untuk itu regresi yang digunakan adalah **Power** Trendline, di mana persamaan umumnya adalah:

$$Y = 10^c X^a, \text{ c dan a = konstanta}$$

Seperti juga pada contoh sebelumnya, fungsi built-in TREND, SLOPE, INTERCEPT dan LINEST akan digunakan bersama Trendline. Untuk menggambarkan regresi ke dalam grafik adalah dengan **Add Trendline > Power**.

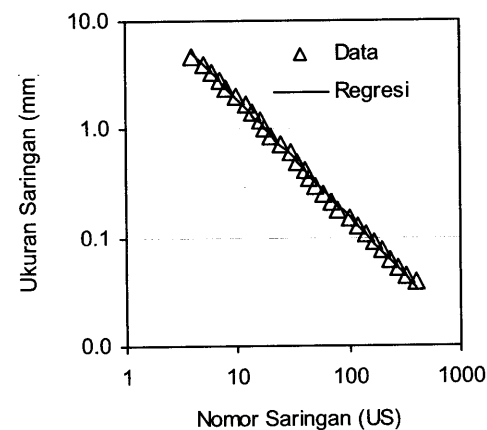
Dari data nomor saringan (US sieve) dan ukuran saringan (mm), kemudian dicari ukuran yang fit dengan menggunakan formula seperti yang diperlihatkan pada Gambar 3.4. Hasilnya kemudian dibuat ke dalam grafik seperti tampak pada Gambar 3.5. Formula slope dan intercept untuk serie ini adalah =SLOPE(LOG(d),LOG(N)) dan =INTERCEPT(LOG(d),LOG(N)). Slope dan intercept adalah konstanta *c* dan *a* dalam persamaan relasi Power. Persamaan Trendline dalam grafik Gambar 3.6 akan sama dengan persamaan yang dibuat fungsi SLOPE dan INTERCEPT dalam worksheet.

	A	B	C	D
1	Nomor	Ukuran	Ukuran	
2	Saringan	Saringan	Saringan	
3		US (mm)	yang Fit	
4	N	d	(mm)	Formula :
5	4	4.750	5.098	=10^TREND(LOG10(d),LOG10(N),LOG10(A5))
6	5	4.000	4.002	=10^TREND(LOG10(d),LOG10(N),LOG10(A6))
7	6	3.350	3.284	=10^TREND(LOG10(d),LOG10(N),LOG10(A7))
8	7	2.800	2.778	dst.
9	8	2.360	2.404	
10	10	2.000	1.887	
11	12	1.700	1.548	
12	14	1.400	1.310	
13	16	1.180	1.133	
14	18	1.000	0.997	
15	20	0.850	0.890	
16	25	0.710	0.699	
17	30	0.600	0.573	
18	35	0.500	0.485	
19	40	0.425	0.420	
20	45	0.355	0.369	
21	50	0.300	0.329	
22	60	0.250	0.270	
23	70	0.212	0.229	
24	80	0.180	0.198	
25	100	0.150	0.155	
26	120	0.125	0.127	
27	140	0.106	0.108	
28	170	0.090	0.087	
29	200	0.075	0.073	
30	230	0.063	0.063	
31	270	0.053	0.053	
32	325	0.045	0.043	
33	400	0.038	0.035	

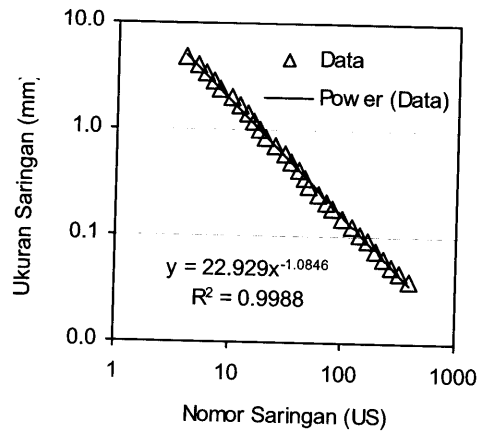
	A	B	C	D
34		Slope =	-1.08461	=SLOPE(LOG(d),LOG(N))
35		Intercept =	1.36039	=INTERCEPT(LOG(d),LOG(N))
36				

Sumber: Bardet, JP, *Experimental Soil Mechanics*, 1997.

Gambar 3.4 Data nomor saringan, ukuran saringan dan ukuran yang fit dalam hubungan log-log, berikut formula-formula yang digunakan.



Gambar 3.5 Grafik serie data dari Gambar 3.4 di mana fungsi TREND digunakan untuk mencari ukuran yang fit



Gambar 3.6 Grafik serie data dari Gambar 3.4 di mana Power Trendline digunakan untuk mencari ukuran yang fit

Regresi Polinomial

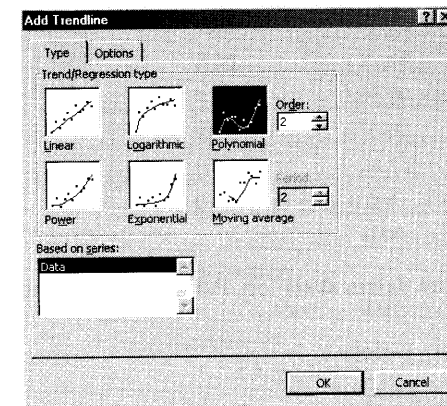
Gambar 3.7 memperlihatkan satu serie data titik x dan y , di mana koordinat- X adalah suhu ($^{\circ}\text{C}$) dan koordinat- Y adalah massa jenis air (gr/cm^3). Dengan mem-plot data x dan y , akan diketahui bahwa sebaran data yang dihasilkan memiliki trend yang melengkung. Regresi polinomial dipakai untuk mendapatkan garis lengkung yang fit dengan sejumlah data titik.

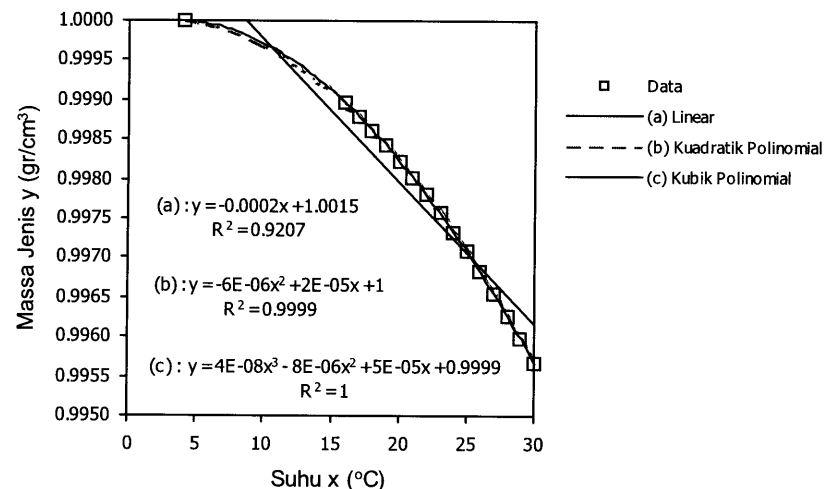
Regresi dalam grafik dibuat dengan **Add Trendline > Type > pilih Linear, Polynomial Order 2 dan Order 3 > Option > Display Equation > Display R-squared**. Hasilnya seperti yang diperlihatkan pada Gambar 3.8. Terlihat bahwa regresi polinomial lebih baik daripada regresi linear untuk serie data ini. Titik-titik yang fit yang dibuat oleh regresi orde 2 dan 3 tampak seperti benar-benar berimpit dengan datanya dan hanya dibedakan oleh nilai R^2 , di mana orde 3 adalah lebih baik. Excel 2000 menyediakan tipe regresi polinomial hingga orde 6.

	A	B
	Suhu ($^{\circ}\text{C}$)	Masa Jenis Air (gr/cm^3)
1	X	Y
2		
3	4	1.00000
4	16	0.99897
5	17	0.99880
6	18	0.99862
7	19	0.99844
8	20	0.99823
9	21	0.99802
10	22	0.99780
11	23	0.99757
12	24	0.99733
13	25	0.99708
14	26	0.99682
15	27	0.99655
16	28	0.99627
17	29	0.99598
18	30	0.99568

Sumber: Bardet, JP, *Experimental Soil Mechanics*, 1997.

Gambar 3.7 Contoh sebuah serie data untuk analisis regresi





Gambar 3.8 Analisis regresi dengan Trendline dari data Gambar 3.7

3.1.4 Interpolasi

Dalam suatu perhitungan engineering, kadang-kadang hanya diperlukan suatu interpolasi data tanpa perlu menganalisis data dengan regresi. Di sini kita mencari nilai y yang koresponding dengan nilai x yang diketahui dalam satu segmen garis, berdasarkan tabulasi sejumlah data (x_i, y_i) , di mana x_i secara kontinyu mungkin bertambah atau berkurang. Interpolasi linear nilai y yang koresponding dengan x adalah:

$$y = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} (x - x_i) \quad (3.13)$$

di mana $(x - x_i)(x - x_{i+1}) \leq 0$ dan $x_i \neq x_{i+1}$

Jika axis-X dikerjakan dalam skala log, Persamaan 3.13 menjadi:

$$\ln(y) = \ln(y_i) + \frac{\ln(y_{i+1}) - \ln(y_i)}{x_{i+1} - x_i} (x - x_i) \quad (3.14)$$

di mana $(x - x_i)(x - x_{i+1}) \leq 0$ dan $x_i \neq x_{i+1}$

Persamaan 3.14 dapat juga ditulis:

$$\ln \frac{y}{y_i} = \frac{x - x_i}{x_{i+1} - x_i} \ln \frac{y_{i+1}}{y_i} \quad (3.15)$$

atau sama dengan:

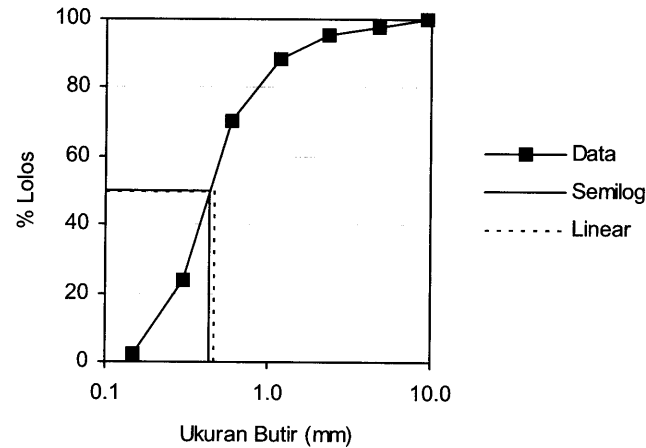
$$y = y_i \left(\frac{y_{i+1}}{y_i} \right)^{\frac{(x - x_i)}{(x_{i+1} - x_i)}} \quad (3.16)$$

Contoh Interpolasi

Dalam uji analisis ayakan (sieve analysis), diambil contoh tanah pasir seberat 1000 gr yang diletakan di dalam mesin ayakan. Ukuran saringan yang dipakai dan hasilnya setelah pengujian dipresentasikan dalam Tabel 3.1. Prosentasi lolos dihitung berdasarkan akumulasi berat yang tertinggal.

Tabel 3.1 Hasil Uji Sieve Analysis

	A	B
1	Ukuran Ayakan	% Lewat
2	(mm)	Ayakan
3	9.50	100.00
4	4.75	97.80
5	2.36	95.48
6	1.18	88.50
7	0.60	70.60
8	0.30	24.10
9	0.15	2.40
10		0.00
Interpolasi semilog		$D_{50} = 0.441$
Interpolasi linear		$D_{50} = 0.467$



Gambar 3.9 Interpolasi linear dan semilog dari data Tabel 3.1

Di sini kita akan mencari nilai prosentasi lolos yang koresponding dengan ukuran butir dari material. Misalnya untuk mencari penyebaran ukuran butir dalam 50% berat contoh pasir atau ukuran D_{50} . Nilai yang bersesuaian dapat diperoleh melalui Persamaan 3.13 dan 3.16, atau dengan membuat macro seperti berikut.

LOGINTER	: interpolasi semilog
=RESULT(1)	
=ARGUMENT("Value",1)	: data 1, sebuah bilangan
=ARGUMENT("X",64)	: data 2, sebuah array
=ARGUMENT("Y",64)	: data 3, sebuah array
=IF(OR(ROWS(X)<>ROWS(Y),COLUMNS(X)>1,COLUMNS(Y)>1),RETURN(#VALUE!))	: pesan error
=FOR("I",1,ROWS(X)-1)	
=IF(AND((INDEX(X,I+1)-Value)*(INDEX(X,I)-Value)<=0,INDEX(X,I+1)<>INDEX(X,I)))	: syarat persamaan
=RETURN(INDEX(Y,I)*(INDEX(Y,I+1)-INDEX(Y,I))^(Value-INDEX(X,I))/(INDEX(X,I+1)-INDEX(X,I)))	: Persamaan 3.16

=END.IF()
=NEXT()
=RETURN(#VALUE!)

Interpolasi semilog dalam Tabel 3.1 adalah =LOGINTER(50,B3:B9,A3:A9). Macro interpolasi linear dikerjakan dengan mengganti Persamaan 3.16 dengan Persamaan 3.13 dalam macro LOGINTER.

Analisis saringan di sini dipakai untuk mengetahui baik atau buruknya gradasi pasir. Gradasi dari pasir atau partikel berbutir (lanau hingga kerikil) dapat diukur dengan koefisien keseragaman (C_U) dan koefisien kelengkungan (C_C) yang dirumuskan sebagai:

$$C_U = \frac{D_{60}}{D_{10}}$$

$$C_C = \frac{D_{30}^2}{D_{60} D_{10}}$$

Makin besar nilai C_U maka makin besar pula rentang distribusi butiran. Sedangkan partikel yang bergradasi baik biasanya memiliki nilai C_C antara 1 sampai 3.

3.2 Data Statistik

3.2.1 Histogram dan Distribusi Kumulatif

Langkah awal suatu penyajian data statistik adalah membuat hasil pengujian ke dalam gambar histogram untuk melihat penyebaran ukurannya. Dalam histogram, data di-plot terhadap distribusinya. Sebagai contoh, dari uji kuat tekan beton karakteristik diperoleh data secara acak dari 40 sampel berikut ini.

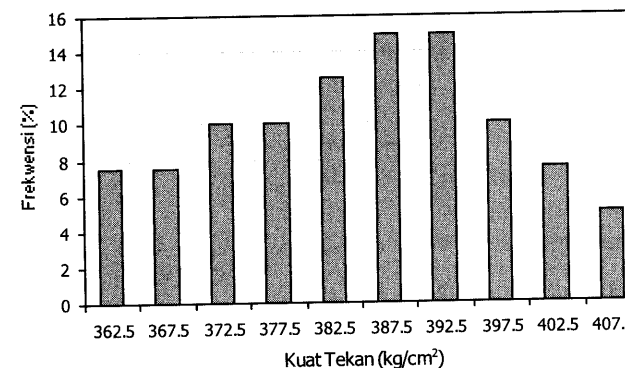
Hasil Uji Kuat Tekan (kg/cm ²)			
376.28	399.30	375.41	370.41
394.43	409.47	393.53	371.43
387.19	374.34	386.30	372.44
390.35	392.29	389.46	390.30
401.27	386.78	390.37	384.82
379.95	361.23	379.08	362.28
380.97	402.90	380.10	400.86
389.33	368.22	388.44	366.35
384.64	399.20	383.76	407.46
396.34	365.36	398.83	363.51

Hasil uji ini kemudian dibagi ke dalam beberapa kelas dengan interval (diambil) = 5 kg/cm², kemudian ditabulasikan seperti pada Tabel 3.2. Banyaknya data setiap kelas disebut frekuensi kelas. Untuk mempermudah proses pendistribusikan data ini, dapat dilakukan dengan menggunakan macro. Gunakan pencabangan **If A >= B And A <= C Then** dalam looping **For ... Next**, di mana A adalah data hasil uji, dan kriterianya adalah range nilai dari B sampai C.

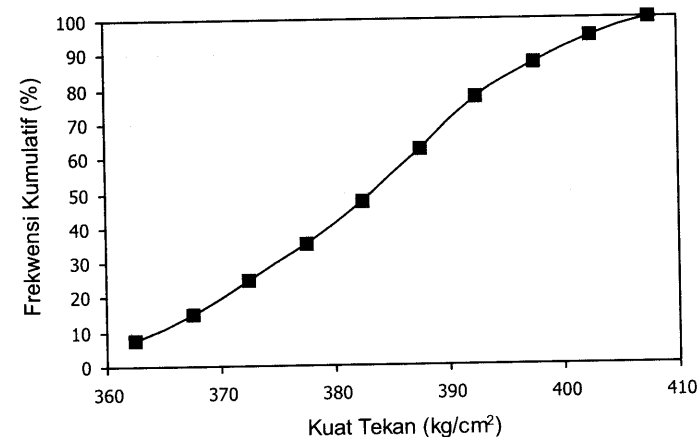
Tabel 3.2 Data Distribusi untuk Membuat Gambar Histogram

Interval Kelas Kg/cm ²	Titik Tengah kg/cm ²	Frekuensi N	Prosentasi (%)	Kumulatif (%)
360-365	362.5	3	7.5	7.5
365-370	367.5	3	7.5	15.0
370-375	372.5	4	10.0	25.0
375-380	377.5	4	10.0	35.0
380-385	382.5	5	12.5	47.5
385-390	387.5	6	15.0	62.5
390-395	392.5	6	15.0	77.5
395-400	397.5	4	10.0	87.5
400-405	402.5	3	7.5	95.0
405-410	407.5	2	5.0	100.0
		40	100.0	

Dari data pada Tabel 3.2, kemudian dibuat gambar histogram dan frekuensi kumulatifnya. Untuk membuat histogram, klik **Chart Wizard** pada toolbar > Chart type **Column** > **Series** > **Add** > **Category (X) axis labels**: pilih range "Titik Tengah" > **Values**: pilih range "Prosentasi" > **Next**: masukkan judul/title pada gambar. Untuk grafik frekuensi kumulatif, pilih grafik tipe **XY (Scatter)** dengan garis > **Series** > **X-Values**: pilih range "Titik Tengah" dan **Y-Values**: pilih range "Kumulatif".



Gambar 3.10 Histogram hasil uji kuat tekan beton



Gambar 3.11 Frekuensi kumulatif kuat tekan beton

3.2.2 Range, Mean, dan Standard Deviation

Hasil penggolongan atau pendistribusian data umumnya didefinisikan ke dalam tiga parameter statistik, yaitu range (jarak), mean (rata-rata), dan standard deviation. Range adalah perbedaan antara nilai yang terkecil dan yang terbesar dari sekumpulan data. Dari hasil uji di atas terdapat range sebesar 48.24, yaitu dari 361.23 hingga 409.47 dalam satuan kg/cm². Mean atau nilai rata-rata adalah hasil penjumlahan nilai data dibagi dengan jumlah total data (n). Dari hasil uji di atas diperoleh mean 384.875 kg/cm². Jika kumpulan data dibagi ke dalam M kelas, rumusan mean adalah:

$$\bar{x} = \frac{\sum_{i=1}^M m_i x_i}{n} \quad (3.17)$$

di mana:

m_i = frekuensi pada interval kelas ke- i

x_i = titik tengah interval kelas ke- i

Mean merupakan nilai yang umum dipakai untuk mewakili nilai-nilai dari sekumpulan data. Konsisten dengan hal tersebut, beberapa ukuran dibuat untuk menyatakan penyimpangan atau deviasi, dan salah satu di antaranya adalah rumusan mean square deviation (MSD). MSD ini juga disebut dengan variance (varians), dan merupakan ukuran yang lazim dipakai. Rumusnya adalah:

$$\frac{\sum_{i=1}^M m_i (x_i - \bar{x})^2}{n} \quad (3.18)$$

Ukuran lain yang dipakai adalah standard deviation atau deviasi standar (S), yang merupakan penarikan akar dari MSD:

$$S = \sqrt{\frac{\sum_{i=1}^M m_i (x_i - \bar{x})^2}{n}} \quad (3.19)$$

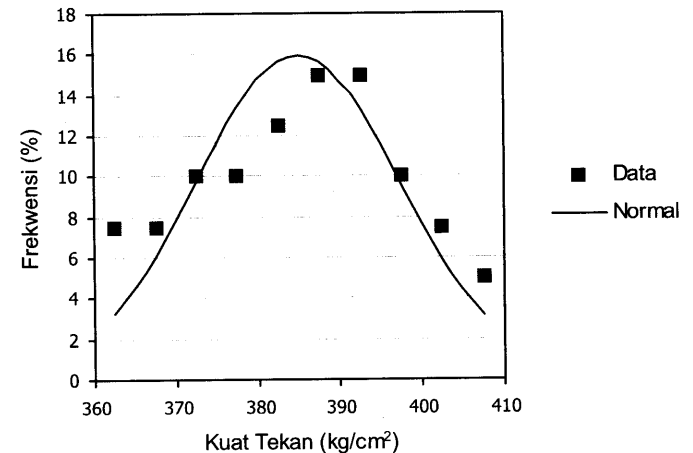
Melalui Persamaan 3.19, deviasi standar untuk data hasil uji pada Tabel 3.2 dapat dihitung sebesar 12.49 kg/cm², dan untuk pembagi = $(n - 1)$ diperoleh 12.65 kg/cm². Telah diketahui bahwa deviasi standar adalah ukuran penyimpangan nilai hasil uji dari rata-ratanya; di sini besar-kecilnya juga dapat menjadi ukuran dari mutu pelaksanaan suatu pekerjaan.

3.2.3 Distribusi Normal

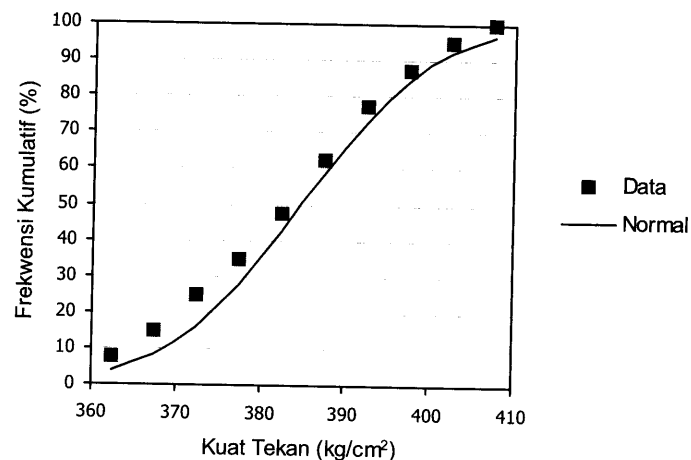
Dari histogram dan pengukuran parameter statistik di atas dapat digambarkan lengkung distribusi normal. Fungsi lengkung distribusi normal adalah:

$$f(x, S, \bar{x}) = \frac{1}{S\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\bar{x}}{S}\right)^2} \quad (3.20)$$

Untuk distribusi log-normal dapat diperoleh dengan mengganti x dengan $\ln(x)$ dalam Persamaan 3.20.



Gambar 3.12 Garis distribusi normal yang fit dengan data Tabel 3.2



Gambar 3.13 Distribusi kumulatif normal yang fit dengan data Tabel 3.2

	A	B	C	D	E	F	G
1	Interval Kelas	Titik Tengah	Frekuensi	Prosen	Kumulatif	Normal	Normal
2	kg/cm ²	Kg/cm ²		(%)	(%)		Kumulatif
3		k	N	f			
4	360-365	362.5	3	7.5	7.5	3.2	3.7
5	365-370	367.5	3	7.5	15.0	6.1	8.2
6	370-375	372.5	4	10.0	25.0	9.8	16.1
7	375-380	377.5	4	10.0	35.0	13.4	27.8
8	380-385	382.5	5	12.5	47.5	15.7	42.5
9	385-390	387.5	6	15.0	62.5	15.6	58.3
10	390-395	392.5	6	15.0	77.5	13.2	72.9
11	395-400	397.5	4	10.0	87.5	9.6	84.4
12	400-405	402.5	3	7.5	95.0	5.9	92.1
13	405-410	407.5	2	5.0	100.0	3.1	96.5
14			40	100.0			
15							
16	Mean = 384.875 kg/cm ²						
17	S = 12.499 kg/cm ²						

	F	G
1	Normal	Normal
2		Kumulatif
3		
4	=500/(S*SQRT(2*PI()))*EXP(-0.5*((k-Mean)/S)^2)	=NORMDIST(k,Mean,S,TRUE)*100
5	=500/(S*SQRT(2*PI()))*EXP(-0.5*((k-Mean)/S)^2)	=NORMDIST(k,Mean,S,TRUE)*100

	D	E	F
15			
16	Mean =	=SUMPRODUCT(k,n)/SUM(n)	kg/cm ²
17	S =	=SQRT(SUMPRODUCT(n,(k-Mean)^2)/SUM(n))	kg/cm ²

Gambar 3.14 Contoh penyajian data hasil uji yang digunakan untuk mencari distribusi normal dan formula-formula yang digunakan

Gambar 3.12 dan 3.13 memperlihatkan garis distribusi normal yang fit dengan prosentasi frekuensi dalam Tabel 3.2. Gambar 3.14 memperlihatkan formula-formula yang digunakan dalam penyajian hasil uji tersebut.

Mean: =SUMPRODUCT(k,n)/SUM(n)

S: =SQRT(SUMPRODUCT(n,(k-Mean)^2)/SUM(n))

Normal: =500/(S*SQRT(2*PI()))*EXP(-0.5*((k-Mean)/S)^2)

Normal Kumulatif: =NORMDIST(k,Mean,S,TRUE)*100

Nilai mean (rata-rata) dan S (deviasi standar) dihitung memakai Persamaan 3.17 dan 3.19. Skala aktual untuk fungsi normal (Persamaan 3.20) ditentukan dari interval yang dipakai, dalam hal ini diambil 5 kg/cm². Kemudian dikalikan dengan 100 untuk penggambaran dalam persen (%).

Dalam Excel juga dikenal beberapa cara lain untuk mencari distribusi, seperti Exponential, Weibull, Gamma, Poisson, dan Student- t .

PROGRAM MATRIKS

Perhitungan dengan cara matriks sangat cocok dan mudah untuk diselesaikan dengan komputer. Oleh karena itu, seiring dengan semakin pesatnya penggunaan komputer, cara ini menjadi sangat populer dan banyak dipakai sebagai ganti dari analisis dengan metode yang manual. Dalam aplikasinya, perhitungan dengan cara matriks ini dipakai untuk analisis struktur atau dalam langkah-langkah penyelesaian dengan metode numerik (numerical method).

4.1 Pengertian Matriks

Suatu matriks adalah suatu array persegi panjang yang di dalamnya terdiri atas komponen-komponen bilangan pembentuknya.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & & & \\ a_{31} & & & \\ - & & & \\ - & & & \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix}$$

Notasi untuk matriks biasanya dipakai [] dan () atau { } untuk matriks baris atau kolom. Matriks A dapat ditulis $[A] = [a_{ij}]$, di mana a_{ij} adalah komponen-komponen matriks, $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$. Subskrip m menunjukkan banyaknya baris dan n untuk kolom, atau biasa disebut orde ($m \times n$). Apabila anggota $m = n$ maka disebut matriks bujur sangkar.

4.1.1 Tipe Matriks

1. Matriks baris,

jika $m = 1$.

Contoh: $\{ 1 \ 2 \ 3 \ 4 \}$

2. Matriks kolom,

jika $n = 1$.

Contoh: $\begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}$

3. Matriks Bujur Sangkar

Matriks Diagonal

Semua komponennya = 0 kecuali pada diagonal utamanya.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Upper Triangular Matrix

Semua komponen di bawah diagonal utamanya = 0.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & a_{24} & \dots & a_{2n} \\ 0 & 0 & a_{33} & a_{34} & \dots & a_{3n} \\ 0 & 0 & 0 & a_{44} & \dots & a_{4n} \\ - & - & - & - & \dots & - \\ 0 & 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

Lower Triangular Matrix

Semua komponen di atas diagonal utamanya = 0.

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & 0 & \dots & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & \dots & 0 \\ - & - & - & - & \dots & 0 \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & \dots & a_{mn} \end{bmatrix}$$

Matriks Skalar

Matriks skalar adalah matriks diagonal di mana komponen diagonalnya merupakan bilangan yang sama.

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Matriks Satuan (unit matriks)

Matriks satuan adalah matriks skalar di mana komponen diagonalnya = 1. Matriks ini juga disebut matriks identitas dan umumnya dinotasikan dengan $[I]$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Band Matrix

Band matriks adalah matriks bujur sangkar ($n \times n$) di mana komponen yang bukan nol dikelompokkan dan membentuk jalur komponen diagonal.

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \dots & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & \dots & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & 0 & \dots & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

Matriks Simetris

Suatu matriks [A] dikatakan simetris, jika

$$[A]^T = [A] \text{ atau } a_{ij} = a_{ji}$$

Contoh:

$$\begin{bmatrix} 1 & 4 & -5 \\ 4 & 2 & 6 \\ -5 & 6 & 3 \end{bmatrix}$$

4.1.2 Operasi Matriks

Penjumlahan dan Pengurangan Matriks

Apabila [A] dan [B] adalah dua matriks berukuran sama, keduanya dapat dijumlahkan. Jika [A] = [a_{ij}] dan [B] = [b_{ij}] menghasilkan [C] = [c_{ij}] maka ditulis:

$$[C] = [A] + [B]$$

$$c_{ij} = a_{ij} + b_{ij} \text{ untuk setiap } i \text{ dan } j.$$

$$\text{Contoh: } [A] = \begin{bmatrix} 2 & 4 & 3 \\ 1 & 5 & 4 \end{bmatrix} \quad [B] = \begin{bmatrix} 0 & -3 & 2 \\ 4 & 2 & -3 \end{bmatrix}$$

$$[A] + [B] = \begin{bmatrix} 2+0 & 4-3 & 3+2 \\ 1+4 & 5+2 & 4-3 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 5 \\ 5 & 7 & 1 \end{bmatrix}$$

Mengurangi [A] dengan [B] adalah sama dengan menjumlahkan [A] dengan [-B] atau ditulis [A] - [B] = [A] + [-B].

Perkalian Matriks

Perkalian [A] (m x n) dengan [B] (n x p) menghasilkan [C] orde (m x p). Jadi syarat perkalian matriks adalah jumlah kolom [A] harus sama dengan jumlah baris [B]. Jika [A] = [a_{ij}], [B] = [b_{ij}] dan [C] = [c_{ij}] maka

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$$

atau ditulis,

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$$

Contoh 1:

$$[A] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \quad [B] = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$[C] = [A] \times [B] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} \end{bmatrix}$$

Contoh 2:

$$[A] = \begin{bmatrix} 2 & 3 \\ 4 & 1 \\ 7 & 4 \end{bmatrix} \quad [B] = \begin{bmatrix} 3 & 3 \\ -1 & 5 \end{bmatrix}$$

$$[C] = [A] \times [B] = \begin{bmatrix} 2 & 3 \\ 4 & 1 \\ 7 & 4 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ -1 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 2.3+3.-1 & 2.3+3.5 \\ 4.3+1.-1 & 4.3+1.5 \\ 7.3+4.-1 & 7.3+4.5 \end{bmatrix} = \begin{bmatrix} 3 & 21 \\ 11 & 17 \\ 17 & 41 \end{bmatrix}$$

Invers Matriks

Pembagian tidak dikenal dalam operasi matriks, untuk itu digunakan invers matriks. Invers matriks hanya dikerjakan pada matriks bujur sangkar dan ditulis dengan notasi $[]^{-1}$. Perkalian dengan invers matriks sama dengan pembagian dalam matriks. Contoh, $[A][B] = [C]$ di mana $[A]$ adalah matriks bujur sangkar, maka persamaan tersebut dapat ditulis menjadi operasi pembagian $[B] = [A]^{-1}[C]$. $[A]^{-1}$ disebut invers dari $[A]$.

Tranpose Matriks

Apabila $[A]$ adalah matriks orde $(m \times n)$, maka tranpose dari $[A]$ adalah matriks dengan orde $(n \times m)$ atau ditulis dengan $[A]^T$. Baris dan kolom dari $[A]$ menjadi kolom dan baris dari $[A]^T$.

$$[B] = [A]^T$$

$$b_{ij} = a_{ji}$$

Contoh:

$$[A] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$[A]^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Beberapa sifat yang berhubungan dengan transpose matriks:

$$([A]^T)^T = [A]$$

$$([A]+[B])^T = [A]^T + [B]^T$$

$$([A][B])^T = [B]^T [A]^T$$

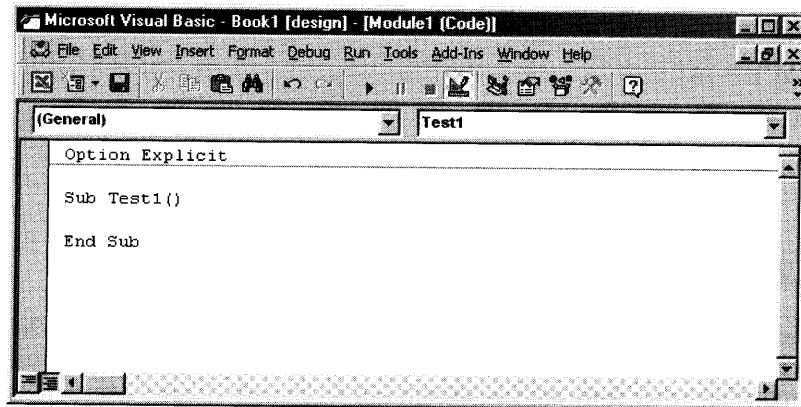
4.2 Macro Program

Macro dengan aplikasi Visual Basic (VB) dapat dikembangkan sedemikian rupa sehingga menjadi sebuah program penyelesaian perhitungan yang kompatibel untuk suatu aplikasi teknik. Hal ini berbeda dengan sebuah fungsi dalam macrosheet atau VB yang terbatas digunakan untuk menyelesaikan tugas-tugas tertentu saja. Macro yang demikian disebut dengan macro program atau dapat disebut dengan program saja. Tahapan-tahapan operasinya adalah:

1. Membaca input data dari worksheet
2. Mengeksekusi input data dalam VB
3. Mencetak hasil ke worksheet

4.2.1 Membuat Macro Program

Langkah-langkah membuat macro program sama seperti membuat fungsi pemakai, namun dengan lebih dulu menuliskan namanya. Langkah-langkahnya dengan memilih **Tools > Macro > Macros** atau **Run Macro**, kemudian tuliskan nama program misalnya *Test1* > **Create** hingga muncul jendela Visual Basic Editor seperti Gambar 4.1. Setelah itu diikuti oleh penulisan perintah-perintah di dalam prosedur Sub, sama seperti membuat perintah untuk sebuah fungsi. Namun pembacaan data input menjadi lebih spesifik dan mengacu pada referensi sel dari worksheet. Judul kolom worksheet dibuat menjadi kode numerik, misalnya untuk sel yang berada pada baris 3 kolom ke-2 (B3) menjadi Cells (3, 2) dan seterusnya.

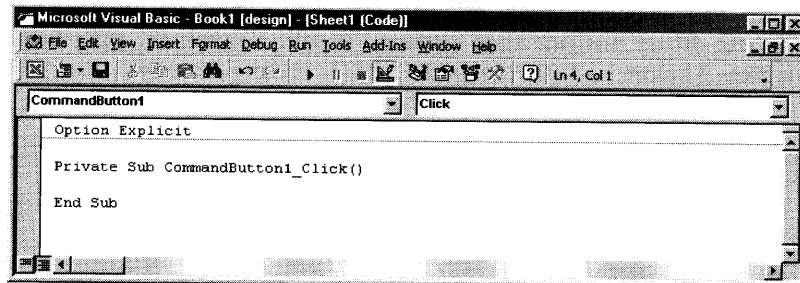


Gambar 4.1 Prosedur program dalam module VB

Untuk mengoperasikan macro, ada dua cara yang dapat dilakukan, yaitu:

1. Dengan **Tools > Macro > Macros > Test1 > Run** atau mengklik **Run Macro** pada toolbar > **Test1 > Run**.
2. Mengklik **Command Button** pada worksheet.

Cara pertama dilakukan apabila kita sudah mendefinisikan nama program. Sedangkan cara kedua di sini dipakai hanya untuk mempersingkat langkah pertama. Untuk mengerjakan cara kedua ini, harus dibuat dulu objek Command Button ke dalam worksheet. Untuk itu aktifkan **Control Toolbox**, lalu pilih dengan mengklik kontrol Command Button pada Control Toolbox dan tempatkan ke dalam worksheet dengan mengklik satu kali pada worksheet. Setelah jadi, klik dua kali Command Button hingga keluar jendela Visual Basic Editor seperti di bawah ini. Program kemudian dapat ditulis di dalam prosedur `Private Sub CommandButton1_Click()`.



Kalimat **Option Explicit** yang muncul sebelum prosedur menyatakan bahwa VB menghendaki pendeklarasian variabel lebih dahulu. Pilihan ini akan muncul jika kita mengaktifkan **Require Variable Declaration** dalam **Tools > Options > Editor**.

Untuk mengedit program, cara pertama dengan **Run Macro > Edit**. Sedangkan cara kedua dengan mengklik dua kali Command Button pada saat Design Mode aktif. Untuk menjalankan program dengan cara kedua adalah dengan mengklik Command Button pada saat Design Mode tidak aktif.

4.2.2 Program Untuk Operasi Matriks

Program perhitungan yang dipakai untuk keperluan analisis dalam rekayasa Teknik Sipil pada umumnya dibuat untuk mengerjakan operasi matriks. Oleh karena itu, perkalian dan invers matriks adalah dua operasi utama dalam program. Berikut ini contoh-contoh penulisan program untuk operasi perkalian dan mencari invers matriks.

Perkalian Matriks

Dalam contoh berikut diberikan sebuah program untuk menghitung perkalian dua matriks. Program diberi nama `PerkalianMatriks1`. Input data dan hasilnya dikerjakan pada `Sheet1`. Tanda apostrophe (') adalah tanda agar program mengabaikan komentar di dalamnya. Tanda ini untuk menyisipkan penjelasan.

```
Sub PerkalianMatriks1()
    Dim MA(3, 2) As Double
    Dim MB(2, 2) As Double
    Dim MC(1 To 3, 1 To 2) As Double

    '=====
    'Baca Input Data
    '=====

    'Input Komponen [A] atau MA
    MA(1, 1) = Sheet1.Cells(6, 2)
    MA(2, 1) = Sheet1.Cells(7, 2)
    MA(3, 1) = Sheet1.Cells(8, 2)
    MA(1, 2) = Sheet1.Cells(6, 3)
    MA(2, 2) = Sheet1.Cells(7, 3)
    MA(3, 2) = Sheet1.Cells(8, 3)

    'Input Komponen [B] atau MB
    MB(1, 1) = Sheet1.Cells(6, 5)
    MB(2, 1) = Sheet1.Cells(7, 5)
    MB(1, 2) = Sheet1.Cells(6, 6)
    MB(2, 2) = Sheet1.Cells(7, 6)
```

```

'=====
'Perkalian Matriks
'[C] = [A].[B]
'=====
MC(1, 1) = MA(1, 1) * MB(1, 1) + MA(1, 2) * MB(2, 1)
MC(2, 1) = MA(2, 1) * MB(1, 1) + MA(2, 2) * MB(2, 1)
MC(3, 1) = MA(3, 1) * MB(1, 1) + MA(3, 2) * MB(2, 1)
MC(1, 2) = MA(1, 1) * MB(1, 2) + MA(1, 2) * MB(2, 2)
MC(2, 2) = MA(2, 1) * MB(1, 2) + MA(2, 2) * MB(2, 2)
MC(3, 2) = MA(3, 1) * MB(1, 2) + MA(3, 2) * MB(2, 2)
'=====
'Cetak Hasil Perhitungan :
'=====
Sheet1.Range(Cells(12, 2), Cells(14, 3)).FormulaArray = MC
End Sub

```

Dalam contoh ini, [A] mempunyai orde (3x2) dan [B] orde (2x2). Data input komponen [A] dibuat dari range sel (6,2) hingga sel (8,3), sedangkan komponen [B] pada range sel (6,5) hingga (7,6). Soal perkalian matriks ini diambil dari Contoh 2 dalam Bab 4.1.2. Hasil soal kemudian dicetak pada range sel (12, 2) hingga (14, 3), memakai penulisan **FormulaArray** = [C]. Untuk menjalankan program, langkah yang digunakan adalah **Tools > Macro > Macros > PerkalianMatriks1 > Run**. Hasilnya seperti diperlihatkan di bawah ini.

	A	B	C	D	E	F	G
1							
2	[A]= 3x2						
3	[B]= 2x2						
4	[A][B]= 3x2						
5							
6	[A] =	2	3	[B] =	3	3	
7		4	1		-1	5	
8		7	4				
9							
10	[C] = [A][B]						
11							
12	[C] =	3	21				
13		11	17				
14		17	41				

Untuk matriks berukuran besar, contoh **PerkalianMatriks1** menjadi tidak efisien, karena semua operasinya dalam contoh itu dituliskan satu per satu. Sebagai gantinya dapat dipakai perintah **For ... Next** untuk membaca input data, untuk mengolah data maupun mencetak hasilnya. Cara ini banyak menghemat penulisan program. Dalam contoh berikut ini kita akan menghitung perkalian matriks (6x6) dengan matriks (6x3). Program diberi nama **PerkalianMatriks2**. Input data dan hasilnya sekarang dikerjakan pada **Sheet2**.

```

Sub PerkalianMatriks2()

Dim m, n, p As Integer
m = Sheet2.Cells(2, 5)
n = Sheet2.Cells(3, 5)
p = Sheet2.Cells(4, 5)
ReDim MA(1 To m, 1 To n) As Double
ReDim MB(1 To n, 1 To p) As Double
ReDim MC(1 To m, 1 To p) As Double

'Baca input [A]

For II = 1 To m
For IJ = 1 To n
MA(II, IJ) = Sheet2.Cells(5 + II, 1 + IJ)
Next IJ: Next II
'Baca input [B]

For II = 1 To n
For IJ = 1 To p
MB(II, IJ) = Sheet2.Cells(5 + II, 9 + IJ)
Next IJ: Next II

'Perkalian Matriks
'[A](mxn) x [B](nxp) menghasilkan
'[C](mxp)
For II = 1 To m
For IJ = 1 To p
MC(II, IJ) = 0
For IK = 1 To n
MC(II, IJ) = MC(II, IJ) + MA(II, IK) * MB(IK, IJ)
Next IK: Next IJ: Next II

'Cetak Hasil
For II = 1 To m
For IJ = 1 To p
Sheet2.Cells(15 + II, 1 + IJ) = MC(II, IJ)
Next IJ
Next II

End Sub

```

Dalam program ini, pencetakan hasil dengan **FormulaArray** telah diganti dengan dua perintah **For ... Next**. Looping pertama dan kedua masing-masing untuk pencetakan baris dan kolom dari [C]. Dengan demikian, kita

tidak perlu menghitung range array yang akan dihasilkan karena sudah diketahui sebesar $m \times p$. Selain itu, ukuran ini dapat diubah-ubah berdasarkan nilai yang akan dimasukkan ke dalam program. Array yang demikian disebut array dinamis, dan didefinisikan dengan statement **ReDim**. Sedangkan array yang statis adalah array yang ukurannya sudah ditetapkan pada awal program dan didefinisikan dengan statement **Dim**.

Perkalian matriks juga dapat dikerjakan di dalam spreadsheet dengan fungsi MMULT(array1,array2). Untuk contoh PerkalianMatriks2, formulanya adalah =MMULT(B6:G11,J6:L11). Untuk mencetak komponen-komponen hasil perkalian digunakan fungsi Indeks. Misalnya untuk mencetak komponen [C₅₃] formulanya adalah =INDEX(MMULT(B6:G11,J6:L11),5,3) dan seterusnya.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	[A] =	mxn		m =	6							
3	[B] =	nxp		n =	6							
4	[C] =	mxp		p =	3							
5												
6	[A] =	20	5	0	0	0	0		[B] =	0.25	0	0
7		5	20	0	0	0	0			0.25	1	0
8		0	0	30	10	0	0			-0.50	1	0
9		0	0	10	30	0	0			-0.50	0	1
10		0	0	0	0	12	6			0.25	0	1
11		0	0	0	0	6	12			0.25	0	0
12												
13												
14	[C] = [A].[B]											
15												
16	[C] =	6.25	5	0								
17		6.25	20	0								
18		-20	30	10								
19		-20	10	30								
20		4.5	0	12								
21		4.5	0	6								
22												

Invers Matriks

Ada beberapa metode untuk mencari invers dari matriks, salah satunya dengan metode Gauss-Jordan. Sebenarnya metode ini lebih umum dikenal sebagai eliminasi Gauss-Jordan dalam menyelesaikan sistem n persamaan linear dengan n bilangan yang tidak diketahui. Secara garis besar langkah-langkahnya dengan menempatkan matriks satuan [I] di sebelah kanan matriks yang akan dicari invers-nya. Misalnya [F] adalah suatu invers dari [A], maka melalui operasi penjumlahan dan perkalian baris pada komponen-komponen [A] dan [I], [A] akan menjadi [I] dan [I] menjadi [F]. Proses ini dapat dinyatakan dengan:

$$[A : I] \rightarrow [I : F] \text{ di mana } [F] = [A]^{-1}$$

Penulisan program untuk mencari invers matriks dapat dilihat di bawah ini. Sebagai contoh kita ambil data [A] dari PerkalianMatriks2. Input data dan hasilnya dikerjakan pada lembar yang sama pada Sheet2.

Sub INVERS()

```
'Baca input data
m = Cells(2, 5)
ReDim MA(1 To m, 1 To m) As Double
For II = 1 To m
For IJ = 1 To m
MA(II, IJ) = Sheet2.Cells(5 + II, 1 + IJ)
Next IJ
Next II



'Mencari Invers Matriks Dengan Metode Gauss-Jordan :
For II = 1 To m
For IJ = 1 To m
If IJ <> II Then MA(II, IJ) = MA(II, IJ) / MA(II, II)
Next IJ
For IK = 1 To m
If IK = II Then GoTo 10
For IJ = 1 To m
If IJ <> II Then MA(IK, IJ) = MA(IK, IJ) - MA(II, IJ) *
MA(IK, II)
Next IJ
10 Next IK
For IK = 1 To m
If IK <> II Then MA(IK, II) = -MA(IK, II) / MA(II, II)
Next IK
MA(II, II) = 1 / MA(II, II)
Next II


'Cetak Hasil
For II = 1 to m
For IJ = 1 to m
Sheet2.Cells(15 + II, 6 + IJ) = MA(II, IJ)
Next IJ
Next II
End Sub
```

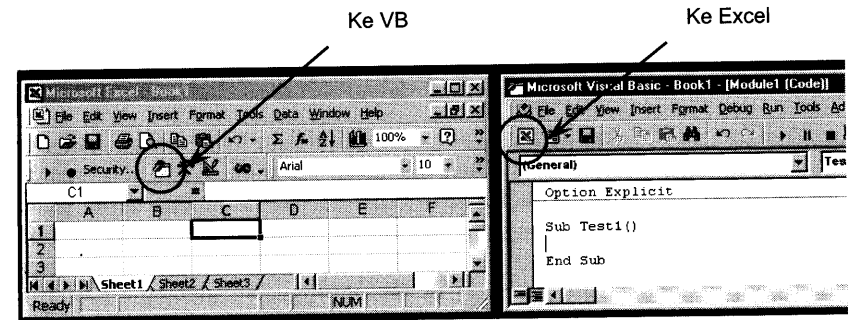
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	[A] =	mxn		m =	6							
3	[B] =	nxp		n =	6							
4	[C] =	mxp		p =	3							
5												
6	[A] =	20	5	0	0	0	0		[B] =	0.25	0	0
7		5	20	0	0	0	0			0.25	1	0
8		0	0	30	10	0	0			-0.50	1	0
9		0	0	10	30	0	0			-0.50	0	1
10		0	0	0	0	12	6			0.25	0	1
11		0	0	0	0	6	12			0.25	0	0
12												
13												
14	[C] = [A].[B]											
15												
16	[C] =	6.25	5	0		[A] ⁻¹ =	0.053	-0.01	0	0	0	0
17		6.25	20	0			-0.01	0.053	0	0	0	0
18		-20	30	10			0	0	0.038	-0.01	0	0
19		-20	10	30			0	0	-0.01	0.038	0	0
20		4.5	0	12			0	0	0	0	0.111	-0.06
21		4.5	0	6			0	0	0	0	-0.06	0.111
22												

Setelah program dijalankan, isi Sheet2 akan tampak seperti di atas. Hasil invers [A] ditempatkan pada range sel (G16 : L21). Untuk membuktikan hasil ini adalah sebuah invers, maka hasil kali $[A]^{-1}$ dengan [C] harus = [B]. Cobalah diperiksa dengan fungsi MMULT. Mencari invers matriks dalam spreadsheet juga dapat dilakukan dengan fungsi MINVERSE. Cara menggunakan fungsi ini sama seperti MMULT.

4.2.3 Tip


Pada operasi macro program cara pertama, Anda dapat berpindah-pindah dari worksheet ke jendela VB atau sebaliknya dengan mengklik icon Excel  atau Visual Basic Editor  seperti diperlihatkan pada Gambar 4.2.

Sedangkan apabila cara kedua yang dipakai, Anda harus mengaktifkan **Design Mode**  kemudian mengklik dua kali pada Command Button untuk ke jendela VB. Untuk menjalankan program, matikan Design Mode (Exit Design Mode) kemudian klik sekali pada Command Button.



Gambar 4.2 Shortcut perpindahan antara worksheet dan jendela VB

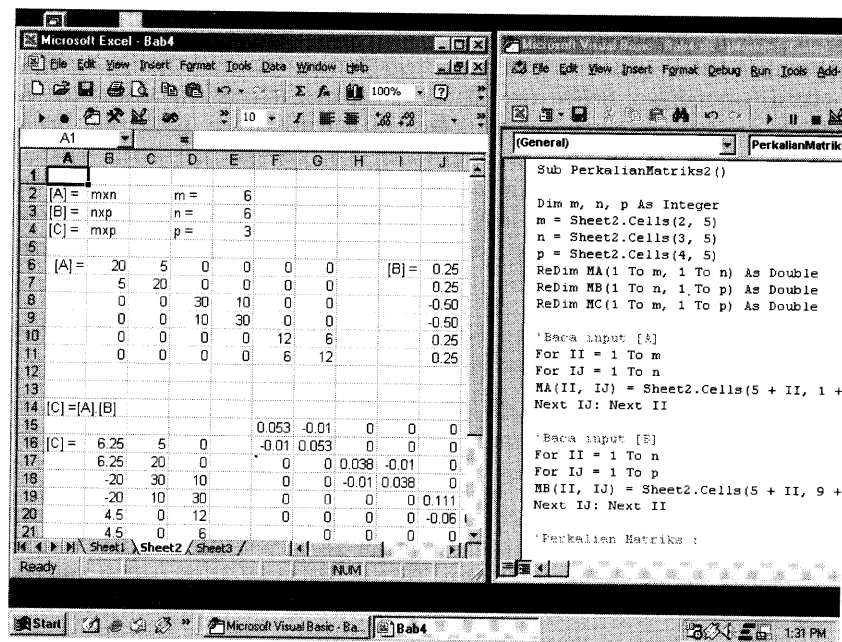
Perpindahan dari worksheet ke jendela VB dan sebaliknya adalah hal yang akan sering dilakukan jika Anda mendesain sebuah program. Hal ini digunakan untuk menelusuri langkah-langkah program, misalnya pada setiap operasi perhitungan. Sebenarnya untuk mempercepat proses perpindahan ini adalah dengan menempatkan worksheet dan jendela VB pada satu layar, seperti tampak di monitor pada Gambar 4.3. Dengan demikian, apabila program dijalankan, interaksinya dengan worksheet dapat dilihat secara bersamaan. Berikut langkah-langkah untuk menempatkan worksheet dan jendela VB dalam satu layar dan sebuah tip yang dapat Anda gunakan sesudahnya.

1. Dalam worksheet klik icon **Restore** , kemudian drag lembar ini ke kiri dan tempatkan kira-kira setengah ukuran layar monitor.
2. Kemudian pindah ke jendela VB, dengan cara yang sama tempatkan ke sebelah kanan layar monitor. Sekarang Anda sudah mendapatkan kedua jendela ini secara bersamaan.
3. Untuk ke baris program yang dikehendaki, tempatkan kursor dan tekan **Ctrl+F8**, maka program akan berpindah ke baris tersebut yang ditandai dengan warna kuning (default). Tekan **F8** untuk meneruskan operasi program.
4. Hasil operasi yang tidak dicetak dapat dilihat nilainya dengan menempelkan pointer pada salah variabel program, misalnya pada satu variabel matriks seperti yang diperlihatkan Gambar 4.4. Tahan pointer

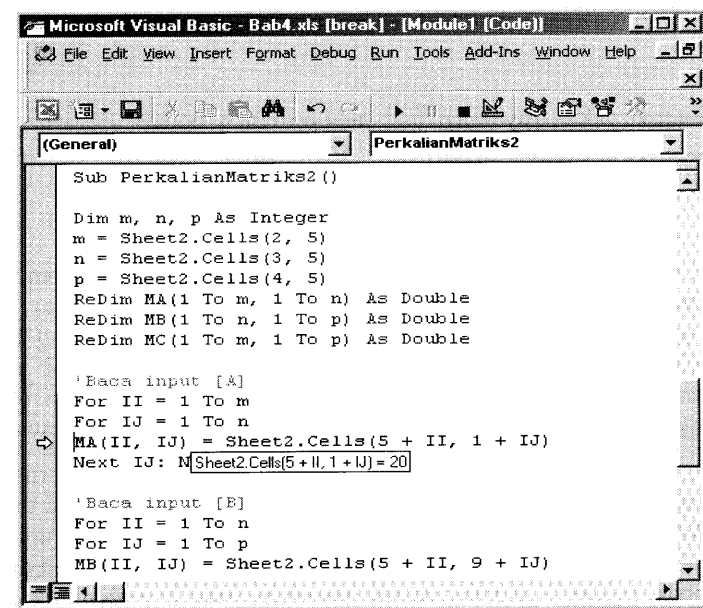
sebentar hingga muncul nilainya. Nilai ini menunjukkan nilai sel dari worksheet di sebelah kiri.

Dengan langkah 1 dan 2, perpindahan antar worksheet – jendela VB cukup dengan mengklik satu lembar tanpa kehilangan lembar lainnya. Ini akan sangat membantu apabila Anda ingin menjalankan program sekaligus melihat hasilnya. Untuk menjalankan program dari awal, tekan F8 kemudian tekan lagi F8 untuk melanjutkan ke baris berikutnya.

Langkah-langkah di atas juga sangat membantu dalam mengembangkan alur program dan memperbaiki kesalahan penulisan program. Salah satu bentuk kesalahan yang mungkin terjadi adalah tidak cocoknya dimensi array suatu variabel dengan statement yang dibuat sehingga waktu program dijalankan, VB mengeluarkan kotak pesan error "subscript out of range". Misalnya dalam suatu pengulangan (looping) di mana prosesnya melebihi subscript yang ditetapkan. Untuk menuju lokasi di mana terjadi kesalahan, klik **Debug** dan **Reset** untuk menghentikan jalannya program.



Gambar 4.3 Penempatan worksheet dan jendela VB dalam satu layar



Gambar 4.4 Melihat nilai variabel program dengan pointer

4.3 Metode Matriks Untuk Analisis Struktur

Struktur pada umumnya merupakan suatu rangkaian elemen struktur yang saling terhubung satu sama lain pada titik-titik diskritnya (nodal-nodalnya). Penyelesaian untuk sistem rangkaian elemen ini dapat dinyatakan dengan susunan persamaan linear simultan yang dibuat dalam bentuk matriks. Dengan bentuk matriks, perhitungan struktur menjadi mudah untuk diselesaikan dengan bantuan komputer.

4.3.1 Struktur Atas

Metode yang akan dipakai untuk analisis struktur adalah dengan metode kekakuan. Sedikit pengenalan mengenai dasar-dasar metode ini akan dijelaskan di sini, pembaca dianjurkan membaca pada referensi lain untuk melengkapi pembahasan pada topik ini. Pada dasarnya, dengan metode kekakuan, kita mencari hubungan gaya-lendutan struktur yang dinyatakan sebagai berikut.

$$\{P\} = [K]\{X\} \quad (4.1)$$

di mana:

$\{P\}$ matriks vektor gaya-gaya luar yang bekerja di nodal

$[K]$ matriks kekakuan dari struktur

$\{X\}$ matriks vektor lendutan di nodal

Dengan menurunkan $[K]$, dapat diperoleh hubungan gaya–lendutan seperti yang dinyatakan oleh Persamaan 4.1. Pada prosesnya di sini adalah menurunkan matriks kekakuan dari satu elemen, yang kemudian dipakai untuk elemen lainnya yang sama dan disatukan membentuk matriks kekakuan struktur. Untuk memenuhi kontinuitas deformasi pada elemen-elemennya adalah dengan mensuperposisikan vektor-vektor yang sesuai.

Selanjutnya untuk menyederhanakan perhitungan, dilakukan partisi atau pemisahan komponen matriks dari hubungan gaya–lendutan, yaitu di nodal yang bebas dan di perletakan. Hal ini karena lendutan di perletakan umumnya = 0. Apabila X_b menyatakan vektor lendutan di perletakan, X_f adalah di nodal yang bebas, P_f dan P_b adalah vektor gaya luar yang koresponding dengan lendutannya, maka Persamaan 4.1 dapat ditulis sebagai berikut:

$$\begin{Bmatrix} P_f \\ P_b \end{Bmatrix} = \begin{bmatrix} K_{ff} & K_{fb} \\ K_{bf} & K_{bb} \end{bmatrix} \begin{Bmatrix} X_f \\ X_b \end{Bmatrix} \quad (4.2)$$

Apabila vektor lendutan nodal perletakan $\{X_b\} = 0$, Persamaan 4.2 dapat ditulis menjadi:

$$\{P_f\} = [K_{ff}]\{X_f\} \quad (4.3)$$

dan,

$$\{P_b\} = [K_{bf}]\{X_f\} \quad (4.4)$$

Vektor lendutan di nodal bebas kemudian diperoleh dari Persamaan 4.3 menjadi $\{X_f\} = [K_{ff}]^{-1}\{P_f\}$. Setelah $\{X_f\}$ diketahui, gaya-gaya dalam elemen dapat dicari. Di sini gaya-gaya dalam elemen ditinjau dalam koordinat lokal, dan diperoleh melalui persamaan berikut ini.

$$\{P\}_i = \{P_o\} + [K]_i\{X\}_i \quad (4.5)$$

di mana:

$\{P_o\}$ adalah reaksi-reaksi di ujung elemen akibat beban yang langsung diterima oleh elemen tersebut

$[K]_i$ matriks kekakuan lokal elemen

$\{X\}_i$ matriks deformasi elemen = $[T]_i\{X_f\}$

$[T]_i$ matriks transformasi elemen

Untuk mencari reaksi perletakan, dapat langsung ditinjau dalam koordinat struktur dengan menggunakan Persamaan 4.4 menjadi:

$$\{R_b\} + \{P_b\} = [K_{bf}]\{X_f\}$$

atau,

$$\{R_b\} = [K_{bf}]\{X_f\} - \{P_b\} \quad (4.6)$$

4.3.2 Struktur Bawah

Analisis struktur di sini adalah mencari hubungan gaya–lendutan pada struktur yang berinteraksi langsung dengan tanah. Pada umumnya dikerjakan dengan metode finite element. Metode ini banyak dipakai karena lebih *efisien* dalam hal memberikan informasi dari hasil analisisnya dibandingkan dengan metode-metode yang klasik, terutama apabila dikerjakan pada struktur-struktur yang *besar* atau pada model struktur yang spesifik. Dalam metode finite element, sistem struktur adalah rangkaian yang dibangun dari sejumlah finite element, di mana satu dengan lainnya terhubung hanya pada nodal-nodalnya.

Beberapa persamaan dibuat untuk mencari hubungan gaya–lendutan di nodal, dimulai dengan mencari hubungan antara gaya-gaya luar di nodal $\{P\}$ dengan gaya-gaya dalam elemen $\{F\}$ berdasarkan prinsip kesetimbangan.

Hubungan P dan F dinyatakan sebagai:

$$\{P\} = [A]\{F\} \quad (4.7)$$

Hubungan lendutan (nodal) X dan deformasi (elemen) d adalah:

$$\{d\} = [B]\{X\}, \text{ dan dapat dibuktikan } \{d\} = [A]^T\{X\} \quad (4.8)$$

PROGRAM EXCEL UNTUK REKAYASA TEKNIK SIPIL

Dari sifat elastis elemen, hubungan F dan d dinyatakan:

$$\{F\} = [S]\{d\} \quad (4.9)$$

Substitusi Persamaan 4.8 ke 4.9 mendapatkan,

$$\{F\} = [S]\{d\} = [S][A]^T\{X\} \quad (4.10)$$

Substitusi Persamaan 4.10 ke 4.7 mendapatkan,

$$\{P\} = [A]\{F\} = [A][S][A]^T\{X\} \quad (4.11)$$

$\{X\}$ adalah lendutan yang hendak dicari, maka:

$$\{X\} = ([A][S][A]^T)^{-1}\{P\} \quad (4.12)$$

Substitusi $\{X\}$ kembali ke Persamaan 4.10 untuk mencari gaya dalam $\{F\}$.

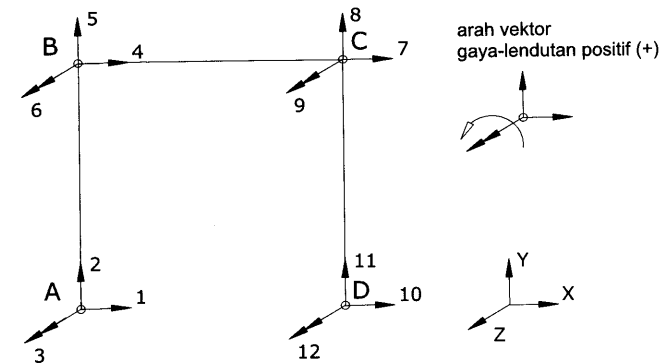
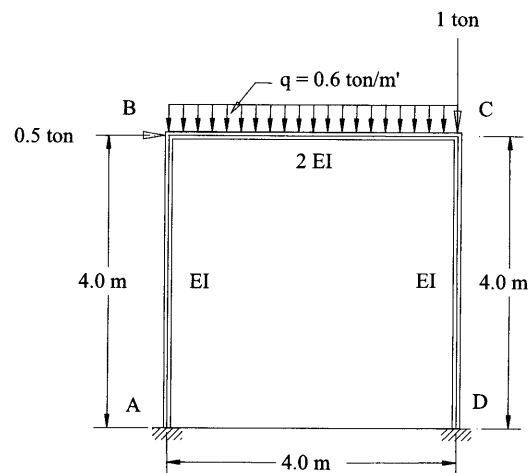
Persamaan 4.10, 4.11 dan 4.12 ini adalah persamaan-persamaan fundamental dalam analisis dengan metode finite element (Bowles, 1982).

Materi program yang diberikan di sini terdiri atas beberapa program untuk analisis struktur atas dan struktur bawah. Juga ditambah satu program untuk penyelesaian konsolidasi tanah. Metode yang dipakai untuk analisis struktur adalah metode seperti yang sudah dijelaskan pada Subbab 4.3. Diharapkan referensi-referensi ini akan saling melengkapi dan dapat dijadikan sebagai pengetahuan dasar bagi pembaca untuk membuat sebuah program penyelesaian. Harap dibedakan tujuan pemrograman di Excel dengan tujuan pemrograman yang membangun suatu program aplikasi sendiri. Dengan Excel berarti program secara keseluruhan menjadi lebih mudah dan praktis, sebab hanya dibuat kasus per kasus. Setiap kasus dikemas sedemikian rupa dengan memanfaatkan fasilitas dalam worksheet kemudian dipresentasikan seperti halnya dalam membuat laporan secara teknis. Ini sesuai dengan hakekat bekerja dengan Excel.

5.1 Struktur Frame Balok

Contoh:

Frame sederhana, dengan panjang elemen masing-masing 4 m. Kekakuan elemen BC adalah 2 kali kekakuan elemen AB atau CD. Titik B dan C adalah nodal bebas, masing-masing terdiri atas 3 komponen lendutan.



Gambar 5.1 Vektor gaya-lendutan di nodal dalam sistem koordinat struktur (global)

Semua elemen balok terbuat dari material elastis linear yang sama dengan modulus elastisitas, $E = 2.1 \times 10^6 \text{ ton/m}^2$.

Dimensi elemen $AB = BC = CD = 30 \text{ cm}$.

- o Beban luar yang diketahui adalah:

Beban H pada nodal B = 0.5 ton.

Beban V pada nodal C = 1 ton.

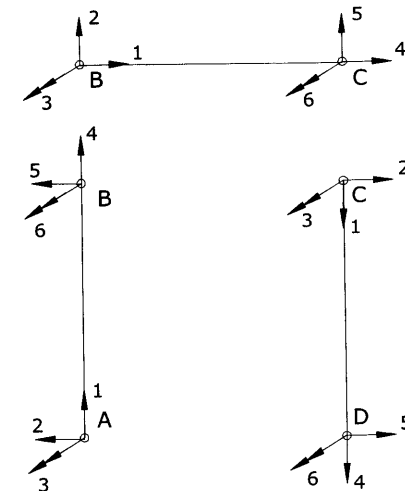
Beban merata q pada elemen BC = 0.6 ton/m'.

- o Beban merata q menjadi vektor gaya-gaya luar yang ekuivalen di nodal B dan C:

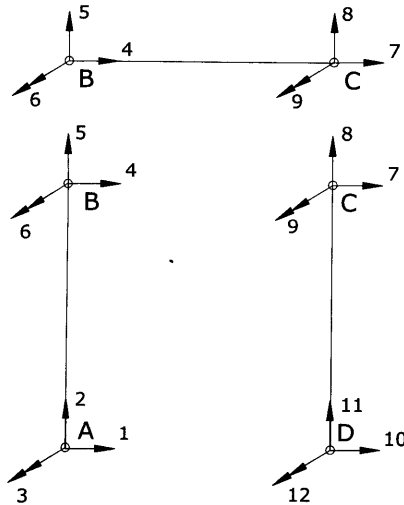
(Lihat Gambar 5.1.)

$$P_{BC(M)} = -P_{CB(M)} = 1/12 \cdot q \cdot L^2 = -0.8 \text{ ton.m.}$$

$$P_{B(V)} = P_{C(V)} = -1/2 \cdot q \cdot L = -1.2 \text{ ton.}$$



Gambar 5.2 Vektor gaya-lendutan elemen dalam sistem koordinat lokal



Gambar 5.3 Vektor gaya-lendutan elemen yang sesuai dengan sistem koordinat global

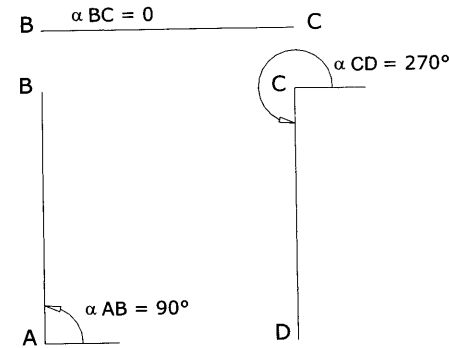
Tahapan Analisis Frame

1. Matriks kekakuan untuk elemen balok diturunkan sebagai berikut.

$$[K]_i = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (5.1)$$

2. Matriks transformasi vektor dari koordinat elemen lokal ke koordinat global adalah:

Untuk 6 komponen vektor gaya-lendutan setiap elemen,



$$[T]_i = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & 0 & 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

3. Matriks kekakuan elemen pada sistem koordinat global,

$$[K]_s = [T]_i^T [K]_i [T]_i \quad (5.3)$$

Matriks kekakuan elemen kemudian disuperposisikan dengan kekakuan elemen lainnya, yang secara keseluruhan membentuk matriks kekakuan $[K]_{\text{global}}$ struktur.

$$\text{Superposisi } [K]_s \rightarrow [K]_{\text{global}} \quad (5.4)$$

Lendutan di nodal bebas dihitung berdasarkan Persamaan 4.3, menjadi:

$$\{X_f\} = [K_{ff}]^{-1} \{P_f\} \quad (5.5)$$

Gaya-gaya dalam setiap elemen dapat dicari dengan mentransformasikan lebih dulu lendutan dari koordinat global kembali ke koordinat elemen (lokal),

$$\{X\}_i = [T]_i \{X_f\}$$

Setelah mendapatkan lendutan elemen $\{X\}_i$, maka gaya dalam elemen dapat diperoleh melalui Persamaan 4.5:

$$\{P\}_i = \{P_o\} + [K]_i \{X\}_i \quad (5.6)$$

4. Reaksi perletakan kemudian dapat dihitung dari Persamaan 4.6,

$$\{R_b\} = [K_{bf}] \{X_f\} - \{P_b\} \quad (5.7)$$

Dari urutan proses di atas kemudian dibuatkan program sederhana untuk analisis struktur frame yang diberi nama FRAME2D.

Pemrograman

Dalam membuat program formula matriks, yang harus dikuasai tentunya cara bagaimana merancang-bangun matriks. Dalam hal ini adalah bagaimana membuat variabel-variabel persamaan seperti pada Persamaan 5.3 sampai 5.7. Ada beberapa hal dalam penulisan program yang sebenarnya memerlukan sedikit imajinasi dan kreatifitas dari penulisnya. Hal pertama misalnya, bagaimana merancang $[K]_{global}$. Di sini diperlukan manipulasi langkah, agar $[K]_{elemen}$ dapat ditempatkan pada posisinya di dalam $[K]_{global}$. Cara yang dipakai adalah dengan membuat sebuah variabel integer, yaitu Indexs untuk menyamakan subskripnya. Indexs membaca penomoran yang dipakai untuk elemen dalam sistem koordinat global dan dijadikan sebagai subskrip $[K]_{global}$ sekaligus kemudian melakukan superposisinya. Programnya sebagai berikut.

```
'=====
'Bangun [K]global
'=====

'Mengosongkan [K]Global
For II = 1 To NP
For IJ = 1 To NP
GK(II, IJ) = 0
Next IJ
Next II
```

```
'Subskrip [K]elemen pada [K]global
For II = 1 To NM
For IJ = 1 To 6
For IK = 1 To 6
Indexs(IJ, II) = Cells(10 + II, 1 + IJ)
Next IJ
Next II

For II = 1 To NM
For IJ = 1 To 6
For IK = 1 To 6
GK(Indexs(IJ, II), Indexs(IK, II)) = GK(Indexs(IJ, II),
Indexs(IK, II)) + P2(IJ, IK, II)
Next IK
Next IJ
Next II
```

Hal yang kedua adalah untuk melakukan partisi (pemisahan) matriks untuk membentuk $[K_{ff}]$ dan $[K_{bf}]$. Langkah-langkah ini pun dapat dibuat dengan cara membuat variabel integer, yaitu Index untuk partisi $[K_{ff}]$ dan SupM untuk partisi $[K_{bf}]$.

```
'Partisi [GK]--->[GKP]

For II = 1 To DOF
Index(II, 1) = Cells(7, 2 + II)
Next II

For II = 1 To DOF
For IJ = 1 To DOF
GKP(Index(II, 1), Index(IJ, 1)) = GK(Index(II, 1), Index(IJ, 1))
GKP(II, IJ) = GKP(Index(II, 1), Index(IJ, 1))
Next IJ
Next II

'Mencari reaksi perletakan
'Melakukan partisi pada perletakan

For II = 1 To 6
SupM(II, 1) = Cells(8, 1 + II)
Next II

For II = 1 To DOF
For IJ = 1 To 6
GKS(IJ, II) = GK(SupM(IJ, 1), Index(II, 1))
Next IJ
Next II
```

Listing program FRAME2D selengkapnya dapat dilihat pada bagian akhir buku ini. Data input dan output untuk soal frame diperlihatkan dalam form FRAME2D pada Gambar 5.4. Berikut ini hasil perhitungan dari setiap tahapan analisis frame.

Elemen AB

[K] (matriks kekakuan elemen)

47250.00	0.00	0.00	-47250.00	0.00	0.00
0.00	265.78	531.56	0.00	-265.78	531.56
0.00	531.56	1417.50	0.00	-531.56	708.75
-47250.00	0.00	0.00	47250.00	0.00	0.00
0.00	-265.78	-531.56	0.00	265.78	-531.56
0.00	531.56	708.75	0.00	-531.56	1417.50

[T] (matriks transformasi vektor dari koordinat lokal ke koordinat global)

0.00	1.00	0.00	0.00	0.00	0.00
-1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	-1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00

$[K]_s = [T]_i^T [K]_i [T]_i$ (matriks kekakuan elemen dalam koordinat global)

265.78	0.00	-531.56	-265.78	0.00	-531.56
0.00	47250.00	0.00	0.00	-47250.00	0.00
-531.56	0.00	1417.50	531.56	0.00	708.75
-265.78	0.00	531.56	265.78	0.00	531.56
0.00	-47250.00	0.00	0.00	47250.00	0.00
-531.56	0.00	708.75	531.56	0.00	1417.50

Elemen BC

[K]

59850.00	0.00	0.00	-59850.00	0.00	0.00
0.00	540.15	1080.29	0.00	-540.15	1080.29
0.00	1080.29	2880.78	0.00	-1080.29	1440.39
-59850.00	0.00	0.00	59850.00	0.00	0.00
0.00	-540.15	-1080.29	0.00	540.15	-1080.29
0.00	1080.29	1440.39	0.00	-1080.29	2880.78

[T]

1.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00

[K]_s

59850.00	0.00	0.00	-59850.00	0.00	0.00
0.00	540.15	1080.29	0.00	-540.15	1080.29
0.00	1080.29	2880.78	0.00	-1080.29	1440.39
-59850.00	0.00	0.00	59850.00	0.00	0.00
0.00	-540.15	-1080.29	0.00	540.15	-1080.29
0.00	1080.29	1440.39	0.00	-1080.29	2880.78

Elemen CD

[K]

47250.00	0.00	0.00	-47250.00	0.00	0.00
0.00	265.78	531.56	0.00	-265.78	531.56
0.00	531.56	1417.50	0.00	-531.56	708.75
-47250.00	0.00	0.00	47250.00	0.00	0.00
0.00	-265.78	-531.56	0.00	265.78	-531.56
0.00	531.56	708.75	0.00	-531.56	1417.50

[T]

0.00	-1.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	-1.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00

[K]_s

265.78	0.00	531.56	-265.78	0.00	531.56
0.00	47250.00	0.00	0.00	-47250.00	0.00
531.56	0.00	1417.50	-531.56	0.00	708.75
-265.78	0.00	-531.56	265.78	0.00	-531.56
0.00	-47250.00	0.00	0.00	47250.00	0.00
531.56	0.00	708.75	-531.56	0.00	1417.50

[K]_{global} Frame

	1	2	3	4	5	6	7	8	9	10	11	12
1	265.78	0.00	-531.56	-265.78	0.00	-531.56	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	47250.00	0.00	0.00	-47250.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	-531.56	0.00	1417.50	531.56	0.00	708.75	0.00	0.00	0.00	0.00	0.00	0.00
4	-265.78	0.00	531.56	60115.78	0.00	531.56	-59850.00	0.00	0.00	0.00	0.00	0.00
5	0.00	-47250.00	0.00	0.00	47790.15	1080.29	0.00	-540.15	1080.29	0.00	0.00	0.00
6	-531.56	0.00	708.75	531.56	1080.29	4298.28	0.00	-1080.29	1440.39	0.00	0.00	0.00
7	0.00	0.00	0.00	-59850.00	0.00	0.00	60115.78	0.00	531.56	-265.78	0.00	531.56
8	0.00	0.00	0.00	0.00	-540.15	-1080.29	0.00	47790.15	-1080.29	0.00	-47250.00	0.00
9	0.00	0.00	0.00	0.00	1080.29	1440.39	531.56	-1080.29	4298.28	-531.56	0.00	708.75
10	0.00	0.00	0.00	0.00	0.00	0.00	-265.78	0.00	-531.56	265.78	0.00	-531.56
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-47250.00	0.00	0.00	47250.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	531.56	0.00	708.75	-531.56	0.00	1417.50

Partisi di nodal bebas $\rightarrow [K_{ff}]$

	4	5	6	7	8	9
4	60115.78	0.00	531.56	-59850.00	0.00	0.00
5	0.00	47790.15	1080.29	0.00	-540.15	1080.29
6	531.56	1080.29	4298.28	0.00	-1080.29	1440.39
7	-59850.00	0.00	0.00	60115.78	0.00	531.56
8	0.00	-540.15	-1080.29	0.00	47790.15	-1080.29
9	0.00	1080.29	1440.39	531.56	-1080.29	4298.28

Partisi di nodal perletakan $\rightarrow [K_{bf}]$

	4	5	6	7	8	9
1	-265.78	0.00	-531.56	0.00	0.00	0.00
2	0.00	-47250.00	0.00	0.00	0.00	0.00
3	531.56	0.00	708.75	0.00	0.00	0.00
10	0.00	0.00	0.00	-265.78	0.00	-531.56
11	0.00	0.00	0.00	0.00	-47250.00	0.00
12	0.00	0.00	0.00	531.56	0.00	708.75

$$\text{Lendutan di nodal bebas: } \{X_f\} = \begin{Bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00117 \\ -0.00002 \\ -0.00039 \\ 0.00117 \\ -0.00005 \\ 0.00017 \\ 0.00000 \\ 0.00000 \\ 0.00000 \end{Bmatrix}$$

Setelah mendapatkan $\{X_f\}$, gaya-gaya dalam elemen dapat dihitung melalui Persamaan (5.6). Reaksi perletakan kemudian dihitung dengan

Persamaan 5.7. Hasil program dipresentasikan dalam form input-ouput FRAME2D pada Gambar 5.4.

	A	B	C	D	E	F	G	H	I	J	K	L
1	FRAMED STRUCTURE											
2												
3	Dim	0.300	Unit TON-METER									
4	NP	12										
5	NM	3										
6	Ec	2.10E+06										
7	DOF	6	4	5	6	7	8	9				
8	Supports	1	2	3	10	11	12					
9												
10	MEMNO	NP1	NP2	NP3	NP4	NP5	NP6	LENGTH	WIDTH	INERTIA	AREA	α
11	1	1	2	3	4	5	6	4.00	0.300	0.00068	0.09	90.0
12	2	4	5	6	7	8	9	4.00	0.380	0.00137	0.11	0.0
13	3	7	8	9	10	11	12	4.00	0.300	0.00068	0.09	270.0
14												
15												
16												
17												
18												
19	NP	Global P-X		Local P-X								
20	1	[P]	[X]	MEMNO	NP1	NP2	NP3	NP4	NP5	NP6		
21	2	0.0	0.00000	1	0.97021	0.10176	0.34336	-0.97021	-0.10176	0.06367		
22	3	0.0	0.00000	2	0.39824	-0.22979	-0.86367	-0.39824	0.22979	-0.05548		
23	4	0.0	0.00000	3	2.42979	0.39824	0.85548	-2.42979	-0.39824	0.73749		
24	5	0.5	0.00117									
25	6	-1.2	-0.00002									
26	7	-0.8	-0.00039									
27	8	0.0	0.00117									
28	9	-2.2	-0.00005									
29	10	0.8	0.00017									
30	11	0.0	0.00000									
31	12	0.0	0.00000									

Gambar 5.4 Form input dan output FRAME2D

Keterangan:

- NP: jumlah komponen vektor gaya-lendutan, NM: jumlah elemen, Ec: modulus elastis bahan, α : sudut transformasi elemen.
- DOF: degree of freedom atau derajat kebebasan struktur. Dalam form penulisannya dibuat berurut, yaitu jumlah DOF (input) adalah 6, yaitu pada komponen vektor 4, 5, 6, 7, 8 dan 9.
- Nodal perletakan (input) adalah pada komponen vektor 1, 2, 3, 10, 11, dan 12.
- Nomor member (MEMNO) 1, 2, dan 3 berturut-turut adalah elemen AB, BC, dan CD.
- Hasil gaya untuk elemen BC (MEMNO 2) dalam FORM2D belum dijumlahkan dengan $\{P_0\}$. Hasil akhir untuk gaya dalam elemen diperoleh menurut Persamaan 5.6:

$$\begin{aligned} \text{NP1} &= 0.39824, \text{NP2} = -0.22979 + (+1.2), \text{NP3} = -0.86367 + (+0.8), \\ \text{NP4} &= -0.39824, \text{NP5} = 0.22979 + (+1.2), \text{NP6} = -0.05548 + (-0.8). \end{aligned}$$

Sebenarnya matriks elemen $\{P_o\}$ dapat dibuat programnya (berdasarkan jenis hubungan elemen, jenis beban, eksentrisitas beban dan sebagainya), namun hal ini akan memperbanyak penulisan data input. Oleh karena itu lebih baik (praktis) untuk dihitung terpisah. Anda dapat memanfaatkan ruang kosong dalam form input untuk menuliskan besarnya $\{P_o\}$, kemudian ditambahkan ke dalam operasi program, sehingga output FRAME2D nantinya sudah merupakan hasil akhir.

Salah satu kemudahan membuat program di dalam Excel adalah untuk hal-hal yang sifatnya administratif, seperti form input dan outputnya dapat dikerjakan di dalam worksheet. Jadi kita tidak menemui kesulitan dalam mengatur tata letak, menambahkan teks, mengatur format tampilan, dan sebagainya. Hal ini juga ditunjang dengan fungsi-fungsi built-in dalam worksheet hingga fasilitas untuk penggambaran grafik. Ada kalanya beberapa operasi perhitungan atau pun subprogram dikerjakan di dalam worksheet, kemudian dipanggil dan diolah oleh VBA.

5.2 Struktur Rangka Batang

Elemen batang dalam rangka batang (truss) dianalisis dengan menganggap pada elemen hanya bekerja gaya aksial saja. Dengan demikian setiap elemen batang mempunyai 2 komponen vektor yang bebas di ujung-ujungnya. Rangka batang ini umumnya terbuat dari bahan baja di mana satu sama lain terhubung sebagai sendi.

Matriks kekakuan elemen batang diturunkan sebagai berikut:

$$[K]_i = \begin{bmatrix} \frac{AE}{L} & 0 & -\frac{AE}{L} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{AE}{L} & 0 & \frac{AE}{L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Matriks transformasi elemen batang ke koordinat global adalah:

$$[T]_i = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \alpha & \sin \alpha \\ 0 & 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

Persamaan matriks kekakuan elemen dalam sistem koordinat global adalah:

$$[K]_s i = [T]_i^T [K]_i [T]_i \quad (5.8)$$

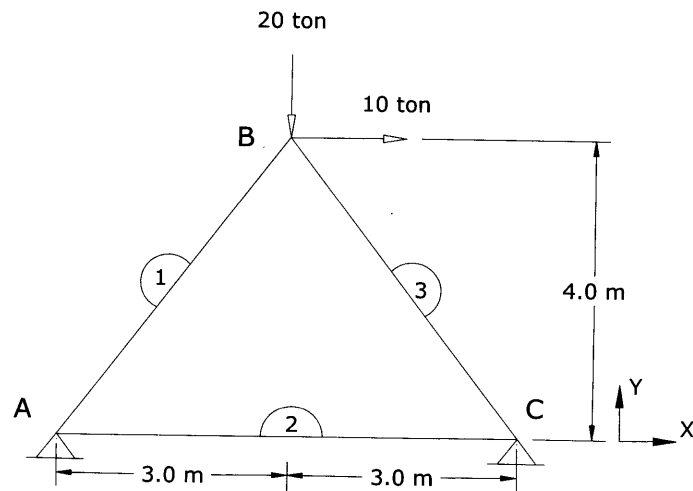
$[K]_s$ kemudian disuperposisikan pada komponen-komponen vektor yang sesuai untuk mendapatkan $[K]_{\text{global}}$. Gaya dalam dan reaksi perletakan dapat dihitung melalui Persamaan 4.2, yaitu dengan mengadakan partisi komponen matriks di nodal bebas dan nodal perletakan.

Contoh:

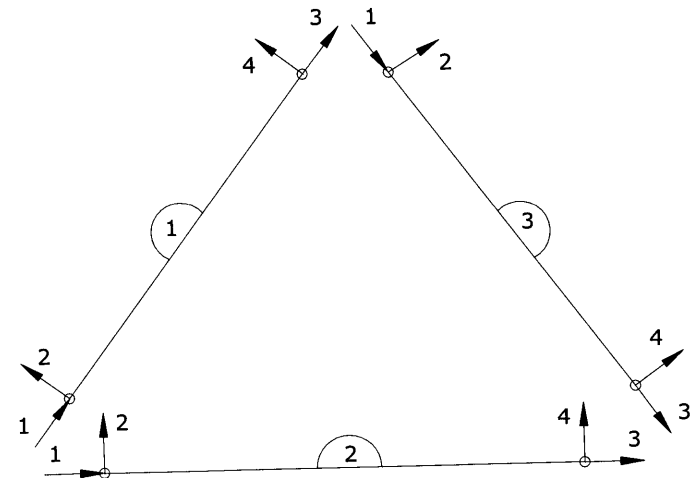
Pada soal berikut ini kita mengerjakan langkah demi langkah program untuk menyelesaikan analisis rangka batang berikut cara-cara yang dipakai. Dengan demikian akan lebih mudah disampaikan hal-hal yang prinsip dalam penulisan program di Excel. Bagian ini juga melengkapi penjelasan pada soal FRAME2D sebelumnya.

Contoh soal diperlihatkan dalam Gambar 5.5, berikut sistem penomorannya dalam Gambar 5.6 sampai Gambar 5.8. Setiap elemen batang terbuat dari material elastis linear yang sama dengan $E = 2100000 \text{ kg/cm}^2$ dan luas penampang $A = 68.4 \text{ cm}^2$.

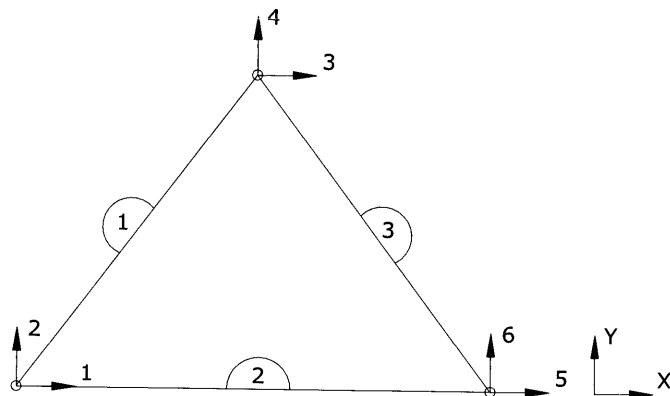
Struktur rangka batang secara keseluruhan hanya mempunyai 2 derajat kebebasan (DOF), pada komponen vektor 3 dan 4. Sedangkan pada kedua perletakannya tidak ada komponen vektor yang bebas, jadi tidak ada translasi sama sekali.



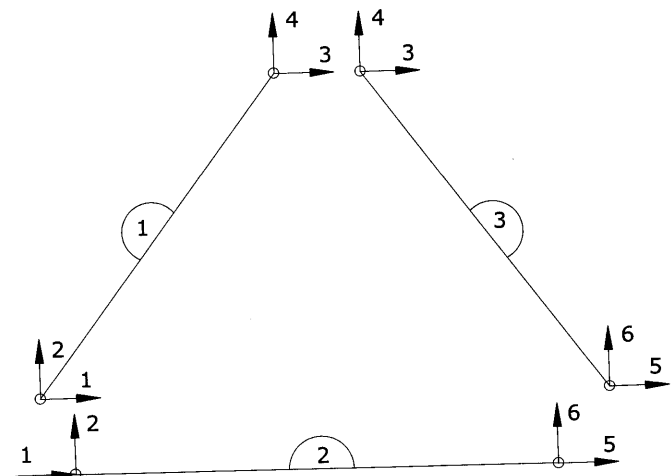
Gambar 5.5 Geometri dan pembebanan pada rangka batang



Gambar 5.7 Vektor gaya-lendutan elemen batang dalam sistem koordinat lokal



Gambar 5.6 Vektor gaya-lendutan di nodal dalam sistem koordinat global



Gambar 5.8 Vektor gaya-lendutan elemen batang yang sesuai dengan sistem koordinat global

Sebelum masuk ke dalam langkah pertama, kita bangun dulu matriks kekakuan elemen dan matriks transformasinya, yaitu [K] dan [T]. Program untuk analisis rangka batang ini dimodifikasi dari program FRAME2D dan dinamakan menjadi TRUSS.

'Bangun [T] Elemen

```
RAD = 180 / 3.14159265358979
For II = 1 To NM
  T(1, 1, II) = Cos(ALPHA(II, 1) / RAD): T(1, 2, II) =
  Sin(ALPHA(II, 1) / RAD): T(1, 3, II) = 0: T(1, 4, II) = 0
  T(2, 1, II) = -Sin(ALPHA(II, 1) / RAD): T(2, 2, II) =
  Cos(ALPHA(II, 1) / RAD): T(2, 3, II) = 0: T(2, 4, II) = 0
  T(3, 1, II) = 0: T(3, 2, II) = 0: T(3, 3, II) = Cos(ALPHA(II, 1)
  / RAD): T(3, 4, II) = Sin(ALPHA(II, 1) / RAD)
  T(4, 1, II) = 0: T(4, 2, II) = 0: T(4, 3, II) = -Sin(ALPHA(II,
  1) / RAD): T(4, 4, II) = Cos(ALPHA(II, 1) / RAD)
Next II
```

'Bangun [K] Elemen

```
For II = 1 To NM
  MK(1, 1, II) = A(II, 1) * Ec / L(II, 1): MK(1, 2, II) = 0: MK(1,
  3, II) = -A(II, 1) *
  Ec / L(II, 1): MK(1, 4, II) = 0
  MK(2, 1, II) = 0: MK(2, 2, II) = 0: MK(2, 3, II) = 0: MK(2, 4,
  II) = 0
  MK(3, 1, II) = -A(II, 1) * Ec / L(II, 1): MK(3, 2, II) = 0:
  MK(3, 3, II) = A(II, 1) *
  Ec / L(II, 1): MK(3, 4, II) = 0
  MK(4, 1, II) = 0: MK(4, 2, II) = 0: MK(4, 3, II) = 0: MK(4, 4,
  II) = 0
Next II
```

Catatan:

Setiap matriks elemen mempunyai tiga indeks di dalamnya; indeks pertama dan kedua menyatakan subskrip baris dan kolom dari matriks, sedangkan indeks ketiga menyatakan matriks untuk elemen ke-i.

Langkah pertama dalam analisis rangka batang adalah mencari matriks kekakuan elemen AB, AC, dan BC dalam koordinat global berdasarkan Persamaan 5.8. Programnya sebagai berikut.

'Transpose [T]--->[TT]elemen

```
For II = 1 To NM
  For IJ = 1 To 4
    For IK = 1 To 4
      TT(IJ, IK, II) = T(IK, IJ, II)
    Next IK
  Next IJ
Next II
```

```
Next IJ
Next II
```

'Bangun [K]elemen = [TT].[K]elemen.[T]

```
For II = 1 To NM
  For IJ = 1 To 4
    For IK = 1 To 4
      P1(IJ, IK, II) = 0
    For IL = 1 To 4
      P1(IJ, IK, II) = P1(IJ, IK, II) + TT(IJ, IL, II) * MK(IL, IK,
      II)
    Next IL
  Next IK
Next IJ
For IJ = 1 To 4
  For IK = 1 To 4
    P2(IJ, IK, II) = 0
  For IL = 1 To 4
    P2(IJ, IK, II) = P2(IJ, IK, II) + P1(IJ, IL, II) * T(IL, IK, II)
  Next IL
Next IK
Next IJ
Next II
```

Catatan:

Membuat matriks transpose adalah dengan membuat subkrip baris menjadi subkrip kolom, dan sebaliknya. Jadi transpose $T(IJ, IK, II) = T(IK, IJ, II)$. Untuk perkalian tiga matriks, perlu dibuat variabel baru, yaitu P1 dan P2. P2 adalah matriks kekakuan elemen yang merupakan hasil akhir perkalian tiga matriks.

Langkah berikutnya melakukan superposisi [K]AB, [K]AC, dan [K]BC pada komponen-komponen vektor yang sesuai. Dalam form input sudah dibuat nomor vektor tersebut yang mengikuti sistem penomoran struktur pada contoh soal. Form input soal diperlihatkan pada Gambar 5.9. Sekarang diperlukan manipulasi langkah program untuk menempatkan komponen-komponen [K]elemen pada [K]global.

MEMNO	NP1	NP2	NP3	NP4
1	1	2	3	4
2	1	2	5	6
3	3	4	5	6

Gambar 5.9 Form input penomoran vektor gaya-lendutan elemen batang dengan sistem koordinat global

```
'Bangun [K]global
```

```
For II = 1 To NP
For IJ = 1 To NP
GK(II, IJ) = 0
Next IJ
Next II
```

```
'Subskrip [K]elemen pada [K]global
For II = 1 To NM
For IJ = 1 To 4
Indexs(IJ, II) = Cells(10 + II, 1 + IJ)
Next IJ
Next II
```

```
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
GK(Indexs(IJ, II), Indexs(IK, II)) = GK(Indexs(IJ, II),
Indexs(IK, II)) + P2(IJ, IK, II)
Next IK
Next IJ
Next II
```

Catatan:

Dalam program, dibuat suatu variabel integer, yaitu Indexs(p,q) yang membaca form input pada Gambar 5.9. Subskrip p adalah nomor vektor elemen batang dalam koordinat global, sedangkan q adalah nomor elemen itu sendiri. Komponen [K]elemen kemudian dapat ditempatkan di dalam [K]global, yaitu dengan menempatkan Indexs(p,q) sebagai subskrip [K]global sekaligus mengerjakan superposisinya melalui operasi berikut.

$$GK(Indexs(IJ, II), Indexs(IK, II)) = GK(Indexs(IJ, II), Indexs(IK, II)) + P2(IJ, IK, II)$$

Dengan cara ini, tidak ada masalah bagi Anda dalam memberikan nomor urut pada elemen, asal penulisan nomor vektor gaya-lendutan setiap elemen sesuai dengan sistem penomoran yang ditinjau secara global struktur.

Sampai langkah ini, akan dikeluarkan dulu setiap tahapan dalam program.

Matriks Kekakuan Elemen:

[K]AB

28728.00	0.00	-28728.00	0.00
0.00	0.00	0.00	0.00
-28728.00	0.00	28728.00	0.00
0.00	0.00	0.00	0.00

[K]AC

23940.00	0.00	-23940.00	0.00
0.00	0.00	0.00	0.00
-23940.00	0.00	23940.00	0.00
0.00	0.00	0.00	0.00

[K]BC

28728.00	0.00	-28728.00	0.00
0.00	0.00	0.00	0.00
-28728.00	0.00	28728.00	0.00
0.00	0.00	0.00	0.00

Matriks Transformasi Elemen:

[T]AB

0.60	0.80	0.00	0.00
-0.80	0.60	0.00	0.00
0.00	0.00	0.60	0.80
0.00	0.00	-0.80	0.60

[T]AC

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

[T]BC

0.60	-0.80	0.00	0.00
0.80	0.60	0.00	0.00
0.00	0.00	0.60	-0.80
0.00	0.00	0.80	0.60

Matriks Kekakuan Elemen dalam Koordinat Global:

$$[K]_s = [T]_i^T [K]_i [T]_i$$

[K]_s AB

10342.08	13789.44	-10342.08	-13789.44
13789.44	18385.92	-13789.44	-18385.92
-10342.08	-13789.44	10342.08	13789.44
-13789.44	-18385.92	13789.44	18385.92

[K]_s AC

23940.00	0.00	-23940.00	0.00
0.00	0.00	0.00	0.00
-23940.00	0.00	23940.00	0.00
0.00	0.00	0.00	0.00

[K]_s BC

10342.08	-13789.44	-10342.08	13789.44
-13789.44	18385.92	13789.44	-18385.92
-10342.08	13789.44	10342.08	-13789.44
13789.44	-18385.92	-13789.44	18385.92

Matriks Kekakuan Global:

	1	2	3	4	5	6
1	34282.08	13789.44	-10342.08	-13789.44	-23940.00	0.00
2	13789.44	18385.92	-13789.44	-18385.92	0.00	0.00

3	-10342.08	-13789.44	20684.16	0.00	-10342.08	13789.44
4	-13789.44	-18385.92	0.00	36771.84	13789.44	-18385.92
5	-23940.00	0.00	-10342.08	13789.44	34282.08	-13789.44
6	0.00	0.00	13789.44	-18385.92	-13789.44	18385.92

Langkah selanjutnya melakukan partisi [K]_{global} di nodal bebas dan nodal perletakan. Manipulasi langkah program untuk membentuk [K]_{ff} sudah dibahas sebelumnya dalam contoh struktur frame balok.

$$[K]_{\text{global}} = \begin{matrix} & \begin{matrix} 3 \\ 4 \\ 1 \\ 2 \\ 5 \\ 6 \end{matrix} \\ \begin{matrix} 3 \\ 4 \\ 1 \\ 2 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} & & & & & \\ & K_{ff} & & & & \\ & & K_{fb} & & & \\ & & & & & \\ & K_{bf} & & & & \\ & & & K_{bb} & & \\ & & & & & \end{bmatrix} \end{matrix}$$

$$[K]_{ff} = \begin{bmatrix} 20684.16 & 0.00 \\ 0.00 & 36771.84 \end{bmatrix}$$

Dari Persamaan 4.3,

$$\{P_f\} = [K]_{ff} \{X_f\}$$

$$\{X_f\} = [K]_{ff}^{-1} \{P_f\}$$

$$\{X_f\} = \begin{Bmatrix} 0.00048 \\ -0.00054 \end{Bmatrix}$$

Setelah lendutan di titik bebas diketahui, dapat dicari gaya dalam setiap elemen melalui persamaan:

$$\{P\}_i = \{P_o\} + [K]_i ([T]_i \{X_f\})$$

$[T]_i \{X_f\}$ adalah matriks lendutan $\{X_f\}$ yang ditransformasikan dari sistem koordinat global ke sistem koordinat lokal.

Di sini tidak ada beban yang langsung diterima oleh elemen batang sehingga $\{P_o\} = 0$. Persamaan untuk mencari gaya dalam elemen menjadi:

$$\{P\}_i = [K]_i ([T]_i \{X_f\})$$

Besarnya gaya dalam setiap elemen adalah:

$\{P\}_{AB} :$

4.167
0.00
-4.167
0.00

$\{P\}_{AC} :$

0.00
0.00
0.00
0.00

$\{P\}_{BC} :$

20.833
0.00
-20.833
0.00

Besarnya reaksi perletakan dapat dicari dari Persamaan 4.6:

$$\{R_b\} = [K_{bf}] \{X_f\} - \{P_b\}$$

Di sini tidak ada reaksi langsung yang diterima pada perletakan dari gaya luar, sehingga $\{P_b\} = 0$. Jadi Persamaan 4.6 menjadi:

$$\{R_b\} = [K_{bf}] \{X_f\}$$

Dalam tahap ini, kita harus membentuk $[K_{bf}]$ lebih dahulu. Cara yang dipakai sama seperti dalam membentuk $[K_{ff}]$. Programnya adalah:

```
For II = 1 To 4
  SupM(II, 1) = Cells(7, 4 + II)
Next II
```

```
For II = 1 To NL
  For IJ = 1 To 4
    GKS(IJ, II) = GK(SupM(IJ, 1), Index(II, 1))
  Next IJ
Next II
```

Catatan:

Di sini ditambah suatu variabel integer SupM(p,1) yang membaca nomor vektor perletakan dalam sistem koordinat global. Jadi SupM(1,1) nilainya adalah = 1, SupM(2,1) = 2, SupM(3,1) = 5 dan SupM(4,1) = 6. Sedangkan variabel Index sudah dipakai sebelumnya, mempunyai nilai-nilai = 3 dan 4 (lihat FRAME2D). $[K_{bf}]$ dalam program diberi nama GKS, dan dengan looping di atas subskrip $[K_{bf}]$ menjadi:

$$[K_{bf}] = \begin{bmatrix} 1,3 & 1,4 \\ 2,3 & 2,4 \\ 5,3 & 5,4 \\ 6,3 & 6,4 \end{bmatrix}$$

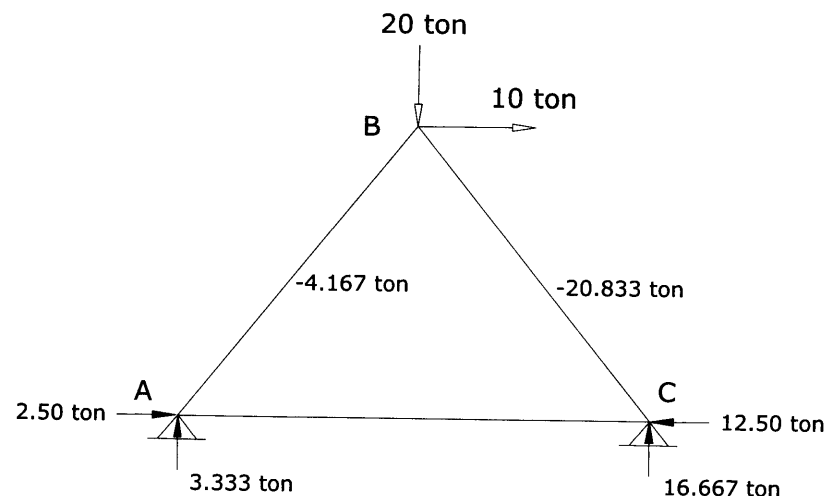
dan komponennya adalah:

$$= \begin{bmatrix} -10342.08 & -13789.44 \\ -13789.44 & -18385.92 \\ -10342.08 & 13789.44 \\ 13789.44 & -18385.92 \end{bmatrix}$$

Besarnya reaksi perletakan kemudian dihitung:

$$\{R_b\} = \begin{Bmatrix} 2.500 \\ 3.333 \\ -12.500 \\ 16.667 \end{Bmatrix} \text{ ton}$$

Hasil perhitungan dari gaya dalam elemen dan reaksi perletakan kemudian digambarkan ke dalam rangka batang seperti yang diperlihatkan dalam Gambar 5.10. Tanda negatif dipakai untuk menyatakan gaya elemen yang bersifat tekan, dan positif untuk gaya elemen yang bersifat tarik.



Gambar 5.10 Penggambaran hasil perhitungan pada rangka batang

Periksa kesetimbangan pada titik B:

$$\Sigma H = 10 + 4.167 \times 3/5 - 20.833 \times 3/5 = 0$$

$$\Sigma V = 20 - 4.167 \times 4/5 - 20.833 \times 4/5 = 0 \quad \dots \text{OK!}$$

Periksa kesetimbangan seluruh struktur:

$$\Sigma F_V = 20 - 3.333 - 16.667 = 0$$

$$\Sigma F_H = 10 + 2.50 - 12.50 = 0 \quad \dots \text{OK!}$$

Form input-output TRUSS untuk perhitungan rangka batang adalah sebagai berikut.

Microsoft Excel - Truss

File Edit View Insert Format Tools Data Window Help

Security... 12

A1 FRAMED STRUCTURE - TRUSS

1	FRAMED STRUCTURE - TRUSS							
2	Unit TON-METER							
3								
4	NP	6						
5	NM	3						
6	Ec	2.10E+07	ton/m ²					
7	DOF	2	3	4				
8	Supports	1	2	5	6			
9								
10	MEMNO	NP1	NP2	NP3	NP4	LENGTH	AREA	α
11	1	1	2	3	4	5.00	0.00684	53.1
12	2	1	2	5	6	6.00	0.00684	0.0
13	3	3	4	5	6	5.00	0.00684	306.9
14								
15								
16		Global P-X			Local P-X			
17	NP	P	X	MEMNO	NP1	NP2	NP3	NP4
18	1	0.00	0.00000	1	4.1667	0.0000	-4.1667	0.0000
19	2	0.00	0.00000	2	0.0000	0.0000	0.0000	0.0000
20	3	10.00	0.00048	3	20.8333	0.0000	-20.8333	0.0000
21	4	-20.00	-0.00054					
22	5	0.00	0.00000					
23	6	0.00	0.00000					
24								
25	Support Reaction							
26	R1	R2	R3	R4				
27	2.5000	3.3333	-12.5000	16.6667				
28								
29								
30								

TRUSS

Draw AutoShapes

Ready NUM

Keterangan:

- NP: jumlah komponen vektor gaya-lendutan, NM: jumlah elemen, Ec: modulus elastis bahan, α : sudut transformasi elemen.
- DOF: degree of freedom atau derajat kebebasan struktur. Dalam form di atas, jumlah DOF adalah 2, yaitu pada komponen vektor 3 dan 4. Pada kasus lain, DOF dapat = 3 apabila perletakan C adalah sebuah rol, sehingga ada translasi mendatar.
- Nodal perletakan pada komponen vektor 1, 2, 5, 6.

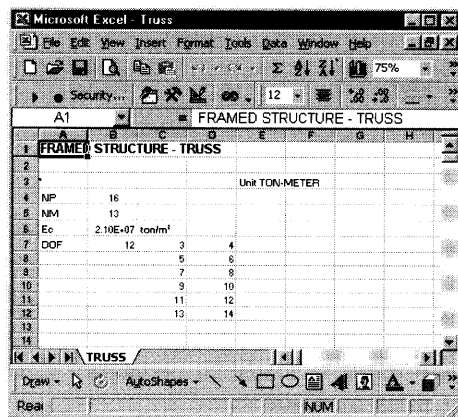
Form input TRUSS seperti di atas tentu akan terlalu terbatas untuk soal-soal struktur yang terdiri atas banyak elemen. Sebenarnya bentuk ideal form adalah setiap data input dan output ditempatkan ke dalam satu kolom sendiri, sehingga seberapa banyaknya data struktur yang ada dapat dituliskan seluruhnya. Namun dari segi kepraktisan, adakalanya perlu dilakukan pembatasan untuk setiap soal.

Sekarang form TRUSS akan dibuat seperti pada Gambar 5.11 karena akan dipakai untuk analisis struktur dengan jumlah elemen yang lebih banyak. Untuk itu, data input DOF dan nomor-nomor vektornya perlu dibuat menjadi beberapa kolom dengan sejumlah baris. Setiap perubahan ini tentu akan ikut mengubah perintah program yang membaca data input. Untuk dua kolom pada C7:D7 dan sejumlah baris, dapat dibuat perintah berikut:

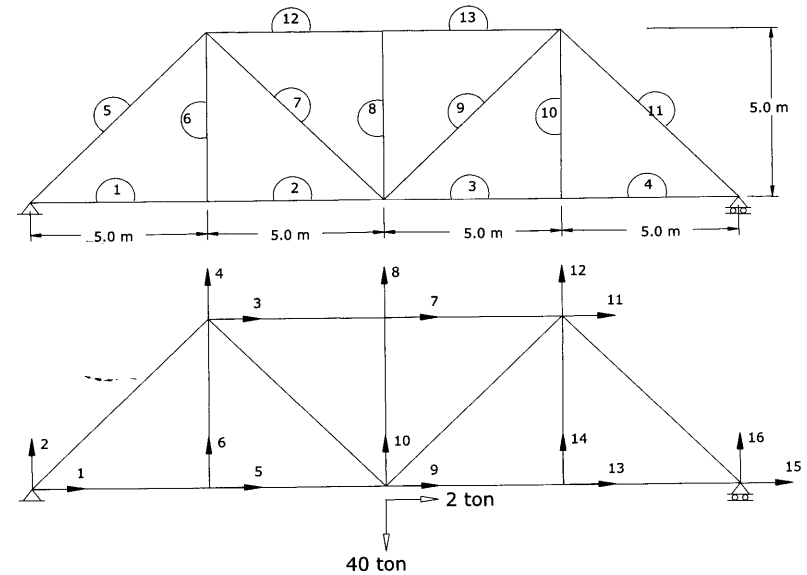
```
c = -1
For II = 1 To DOF
For IJ = 1 To 2
If II + IJ + c > DOF Then GoTo 5 Else Index(II + IJ + c, 1) =
Cells(6 + II, 2 + IJ)
Next IJ
c = c + 1
Next II
```

Catatan:

Dalam membuat langkah-langkah program, tidak ada suatu rumusan tertentu yang harus digunakan. Seperti juga pada seni penulisan lainnya, di sini hanyalah berdasarkan kreasi dari penulisnya. Hal ini tidak baku, Anda dapat memakai cara penulisan lain asal memang dapat mengerjakan perintah yang dimaksud.



Gambar 5.11 Contoh form input TRUSS



Gambar 5.12 Struktur rangka batang

Dalam contoh berikut, program TRUSS akan dipakai untuk menganalisis struktur rangka batang dengan geometri, sistem penomoran struktur, pembebanan dan kondisi perletakan seperti Gambar 5.12.

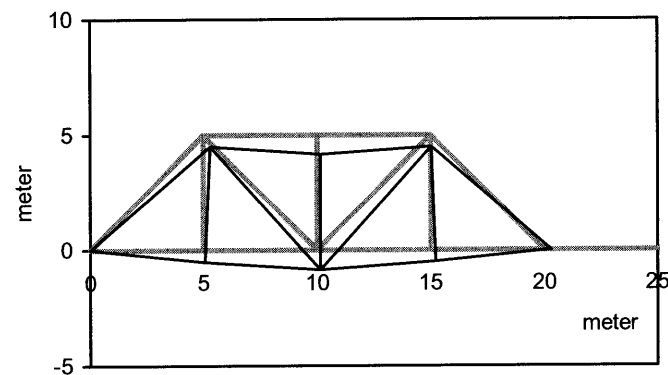
Data struktur pada Gambar 5.12 kemudian diimplementasikan ke dalam form input TRUSS sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	FRAMED	STRUCTURE - TRUSS												
2														
3						Unit TON-METER								
4	NP	16												
5	NM	13												
6	Eo	2.10E+07	ton/m ²											
7	DOF	13	3	4										
8			5	6										
9			7	8										
10			9	10										
11			11	12										
12			13	14										
13			15											
14														
15														
16	Supports	1	2	15	16									
17														
18	MEMNO	NP1	NP2	NP3	NP4			LENGTH	AREA					
19	1	1	2	5	6			5.00	0.013200					
20	2	5	6	9	10			5.00	0.013200					
21	3	9	10	13	14			5.00	0.013200					
22	4	13	14	15	16			5.00	0.013200					
23	5	1	2	3	4			7.07	0.013200					
24	6	5	6	3	4			5.00	0.013200					
25	7	9	10	3	4			7.07	0.013200					
26	8	9	10	7	8			5.00	0.013200					
27	9	9	10	11	12			7.07	0.013200					
28	10	13	14	11	12			5.00	0.013200					
29	11	15	16	11	12			7.07	0.013200					
30	12	3	4	7	8			5.00	0.013200					
31	13	7	8	11	12			5.00	0.013200					
32								5.00	0.013200					
33														

Output program dalam form TRUSS:

	Global P-X			Local P-X				Support Reaction				
	NP	P1	P2	MEMNO	NP1	NP2	NP3	NP4	R1	R2	R3	R4
34	1	0.00	0.000		-22.00	0.00	22.00	0.00	-2.00	20.00	0.00	20.00
35	2	0.00	0.000	2	-22.00	0.00	22.00	0.00				
36	3	0.00	0.001	3	-20.00	0.00	20.00	0.00				
39	4	0.00	-0.003	4	-20.00	0.00	20.00	0.00				
40	5	0.00	0.000	5	28.28	0.00	-28.28	0.00				
41	6	0.00	-0.003	6	0.00	0.00	0.00	0.00				
42	7	0.00	0.001	7	-28.28	0.00	28.28	0.00				
43	8	0.00	-0.004	8	0.00	0.00	0.00	0.00				
44	9	2.00	0.001	9	-28.28	0.00	28.28	0.00				
45	10	-40.00	-0.004	10	0.00	0.00	0.00	0.00				
46	11	0.00	0.000	11	28.28	0.00	-28.28	0.00				
47	12	0.00	-0.002	12	40.00	0.00	-40.00	0.00				
48	13	0.00	0.001	13	40.00	0.00	-40.00	0.00				
49	14	0.00	-0.002									
50	15	0.00	0.002									
51	16	0.00	0.000									
52												

Hasil lendutan {X} kemudian diplot ke dalam grafik dan ditunjukkan pada Gambar 5.13. Grafik dibuat di Excel dengan tipe XY(scatter), terdiri atas beberapa serie data dari koordinat struktur sebelum dan sesudah dibebani. Juga diberikan faktor skala agar hasil lendutan tampak dalam gambar. Pembuatan seri data dan formula yang digunakan dapat dilihat di dalam file program.

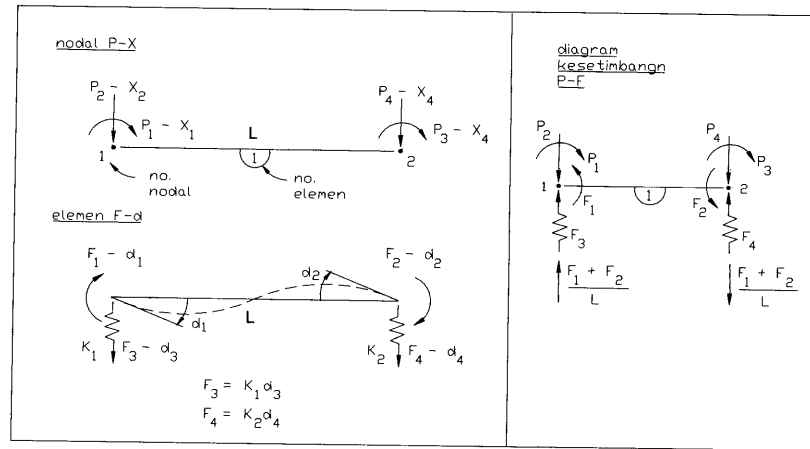


Gambar 5.13 Lendutan rangka batang

5.3 Balok di Atas Medium Elastis

Balok yang dibebani di atas medium elastis adalah teori yang sering dipakai dalam prediksi beban-deformasi untuk analisis struktur yang berinteraksi langsung dengan tanah. Medium elastis adalah pendekatan terhadap perilaku tanah sebagai kumpulan pegas yang elastis linear, independent dan satu sama lain tidak saling berhubungan (asumsi tanah Winkler). Dengan semakin pesatnya penggunaan komputer, kemudian teori tersebut lebih banyak diselesaikan dengan metode numerik, khususnya metode finite element.

Dalam metode finite element, sistem struktur dianggap sebagai rangkaian terpasang dari sejumlah elemen dengan jumlah dan ukuran yang terbatas (finite-element). Medium tanah pada setiap elemen diwakili oleh sepasang pegas pada kedua nodalnya. Dasar-dasar yang digunakan di sini adalah Persamaan 4.7 sampai 4.12, dan notasi-notasi yang dipakai menyatakan hubungan komponen gaya-gaya eksternal (nodal) dan internal finite element (member) pada Gambar 5.14.



Gambar 5.14 Komponen gaya-gaya eksternal (nodal) dan internal (member) dari finite element

Komponen $[A]$, $[S]$ elemen adalah:

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{L} & \frac{1}{L} & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{L} & -\frac{1}{L} & 0 & 1 \end{bmatrix}$$

$$[S] = \begin{bmatrix} \frac{4EI}{L} & \frac{2EI}{L} & 0 & 0 \\ \frac{2EI}{L} & \frac{4EI}{L} & 0 & 0 \\ 0 & 0 & K_1 & 0 \\ 0 & 0 & 0 & K_2 \end{bmatrix}$$

Keterangan:

- E modulus elastisitas elemen (balok)
- I momen inersia penampang elemen
- K_1, K_2 konstanta pegas (spring) tanah = $(L/2).B.K_s$
- B lebar balok
- K_s modulus subgrade reaction

Lendutan pondasi $\{X\}$ dan gaya elemen $\{F\}$ dicari melalui Persamaan 4.12 dan 4.10:

$$\begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix} = \{[A][S][A]^T\}^{-1} \begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{Bmatrix} \begin{Bmatrix} M \\ V \\ M \\ V \end{Bmatrix}$$

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} \begin{Bmatrix} M \\ M \\ V \\ V \end{Bmatrix} = [S][A]^T \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix}$$

$\{[A][S][A]^T\}^{-1}$ invers dari matriks global struktur

$[S][A]^T$ matriks elemen

M vektor gaya momen

V vektor gaya vertikal

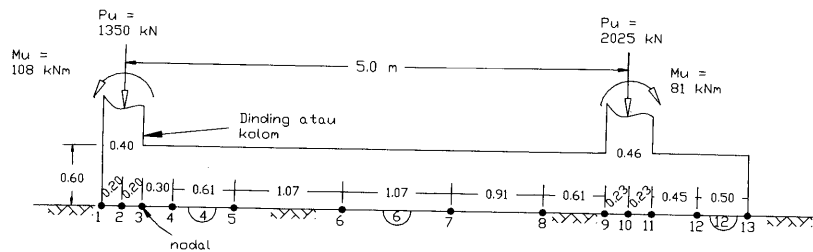
Gaya elemen F_3 dan F_4 di sini adalah reaksi pegas tanah pada elemen atau disebut dengan reaksi tanah (soil reaction).

Keuntungan metode ini adalah bahwa setiap faktor dari pengaruh karakteristik elemen yang tidak konstan dan medium yang tidak seragam dapat dengan mudah dimasukkan ke dalam perhitungan. Namun demikian, penting untuk membagi struktur menjadi lebih banyak elemen pada sekitar nodal pembebanannya agar hasil gaya-gaya dalam yang timbul dapat terekspos dengan baik.

Contoh:

Program perhitungan untuk balok di atas medium elastis diberi nama BOF. Aplikasi program ini sekarang dipakai untuk analisis pondasi di atas medium tanah dengan modulus tanah K_s . Data pondasi untuk soal diambil dari buku J.E Bowles, Foundation Analysis and Design. Adaptasi ini untuk membandingkan output program yang dibuat di Excel dengan Program For Beams (PFB) yang sudah dibuat dalam bahasa Fortran (versi terdahulu). Hasilnya kemudian dibuat pada Tabel 5.1.

Pondasi dibagi 12 elemen/13 nodal, dengan panjang elemen seperti yang diperlihatkan pada Gambar 5.15.



Data dan model dari buku J.E Bowles, Foundation Analysis and Design, 1982.

Gambar 5.15 Analisis pondasi dengan metode finite element

Load Factor, $LF = 1.571$

Modulus reaksi tanah, $K_s = 14000 \text{ kN/m}^3$

Modulus elastis pondasi, $E_b = 21700000 \text{ kPa}$

Dimensi pondasi = 0.60 m

Lebar pondasi, $B = 2.64 \text{ m}$

Ambil, Input $K_s = LF.K_s = 22000 \text{ kN/m}^3$

Analisis balok dengan metode finite elemen memakai sistem penomoran yang menerus, sehingga mudah diterka bahwa tipe matriks global ASA^T adalah sebuah band matriks, di mana komponen-komponen semua matriks elemen ASA^T akan membentuk jalur diagonal. Dalam merancang matriks global ini, dapat dilakukan dengan melihat penempatan komponen matriks elemen seperti berikut.

[ASA^T]elemen1(1,1) menempati [ASA^T]global(1,1), dan seterusnya.
 [ASA^T]elemen2(1,1) menempati [ASA^T]global(3,3), dan seterusnya.
 [ASA^T]elemen3(1,1) menempati [ASA^T]global(5,5), dan seterusnya,
 dan seterusnya hingga NM.

Dengan pola penempatan seperti di atas, dapat dibangun matriks global sebagai berikut.

```

n = 0
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
GASAT(IJ + n, IK + n) = ASAT(IJ, IK, II)
Next IK
Next IJ
n = n + 2
Next II

```

Selanjutnya dengan melihat penempatan komponen matriks elemen dan sekaligus mengerjakan superposisinya pada komponen-komponen vektor yang sesuai:

$$\begin{aligned} [\text{ASA}^T]_{\text{global}(3,3),(3,4)} &= [\text{ASA}^T]_{\text{elemen1}\{(3,3),(3,4)\}} + \\ [\text{ASA}^T]_{\text{elemen2}\{(1,1),(1,2)\}} \\ [\text{ASA}^T]_{\text{global}(4,3),(4,4)} &= [\text{ASA}^T]_{\text{elemen1}\{(4,3),(4,4)\}} + \\ [\text{ASA}^T]_{\text{elemen2}\{(2,1),(2,2)\}} \end{aligned}$$

$$\begin{aligned} [\text{ASA}^T]_{\text{global}(5,5),(5,6)} &= [\text{ASA}^T]_{\text{elemen2}\{(3,3),(3,4)\}} + \\ [\text{ASA}^T]_{\text{elemen3}\{(1,1),(1,2)\}} &= [\text{ASA}^T]_{\text{elemen2}\{(4,3),(4,4)\}} + \\ [\text{ASA}^T]_{\text{global}(6,5),(6,6)} &= [\text{ASA}^T]_{\text{elemen2}\{(4,3),(4,4)\}} + \\ [\text{ASA}^T]_{\text{elemen3}\{(2,1),(2,2)\}} &= [\text{ASA}^T]_{\text{elemen2}\{(4,3),(4,4)\}} + \end{aligned}$$

Hubungan di atas dapat ditulis ke dalam perintah program berikut:

```

n = 0
For IK = 1 To NM - 1
    GASAT(3 + n, 3 + n) = ASAT(3, 3, IK) + ASAT(1, 1, IK + 1)
    GASAT(3 + n, 4 + n) = ASAT(3, 4, IK) + ASAT(1, 2, IK + 1)
    GASAT(4 + n, 3 + n) = ASAT(4, 3, IK) + ASAT(2, 1, IK + 1)
    GASAT(4 + n, 4 + n) = ASAT(4, 4, IK) + ASAT(2, 2, IK + 1)
n = n + 2
Next IK

```

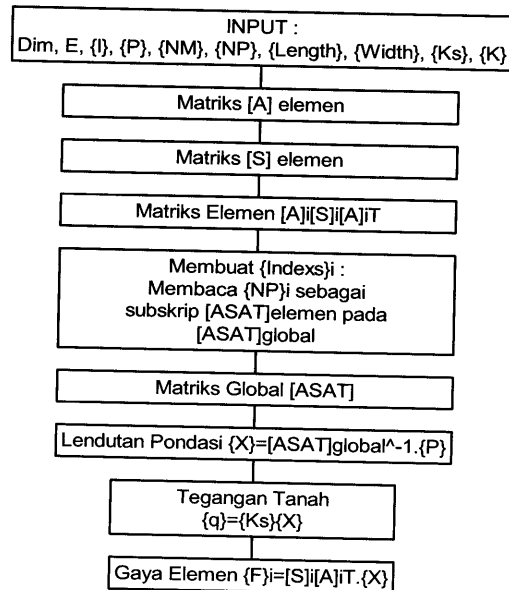
Dalam pemrograman di Excel, cara penulisan seperti ini sebenarnya dapat juga digantikan dengan memanfaatkan data input dalam form. Perbedaan

yang tampak adalah bahwa pada listing ini tidak terdapat rujukan pada alamat sel. Point yang akan ditekankan dalam pemrograman di Excel adalah bahwa kita bekerja dengan suatu form yang aktif, di mana di dalamnya sudah tersedia semua variabel program yang dibutuhkan. Cara penulisan seperti di atas dapat digantikan dengan cara seperti dalam FRAME2D dan TRUSS sebelumnya:

```
For II = 1 To NM
For IJ = 1 To 4
Indexs(IJ, II) = Cells(11 + II, 1 + IJ)
Next IJ
Next II

For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
GASAT(Indexs(IJ, II), Indexs(IK, II)) = GASAT(Indexs(IJ, II),
Indexs(IK, II)) + ASAT(IJ, IK, II)
Next IK
Next IJ
Next II
```

Proses perhitungan selengkapanya dapat digambarkan ke dalam bagan alir (flowchart) di bawah ini.



Data input dan output program untuk contoh soal dipresentasikan sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	BOF													
2		26												
3	NP													
4	NM	12												
5														
6	E(beam)	2.17E+07	KPa											
7	Dim	0.60												
8														
9														
10														
11	MEMNO	NP1	NP2	NP3	NP4	LENGTH	WIDTH	INERTIA	Ke (kN/m ³)	K (kN/m)				
12	1	1	2	3	4	0.200	2.640	0.047520	22,000.00	11,816.00	5,808.00			
13	2	3	4	5	6	0.200	2.640	0.047520	22,000.00	5,808.00	5,808.00			
14	3	5	6	7	8	0.300	2.640	0.047520	22,000.00	8,712.00	8,712.00			
15	4	7	8	9	10	0.510	2.640	0.047520	22,000.00	17,714.40	17,714.40			
16	5	9	10	11	12	1.070	2.640	0.047520	22,000.00	31,072.80	31,072.80			
17	6	11	12	13	14	1.070	2.640	0.047520	22,000.00	31,072.80	31,072.80			
18	7	13	14	15	16	0.910	2.640	0.047520	22,000.00	26,426.40	26,426.40			
19	8	15	16	17	18	0.510	2.640	0.047520	22,000.00	17,714.40	17,714.40			
20	9	17	18	19	20	0.230	2.640	0.047520	22,000.00	6,679.20	6,679.20			
21	10	19	20	21	22	0.230	2.640	0.047520	22,000.00	6,679.20	6,679.20			
22	11	21	22	23	24	0.450	2.640	0.047520	22,000.00	13,068.00	13,068.00			
23	12	23	24	25	26	0.500	2.640	0.047520	22,000.00	14,520.00	29,040.00			
24														
25	LOAD MATRIX	DISPLACEMENT OF THE STRUCTURE: (X)=[G-ASAT]^-1.(P)							ELEMENT FORCES: (F)=[SAT].(X)					
26	NP	Load	No	Rotation (r)	No	Translation	No	Soil Pressure	No	Moments (kN.m)	Soil Reactions (kN)			
27	1	0.00	1	-0.00251	2	0.01179	1	259.419	1	0.000	-27.395	136.973	65.574	
28	2	0.00	3	-0.00251	4	0.01129	2	248.388	2	-80.605	296.981	65.574	62.677	
29	3	-108.00	5	-0.00247	6	0.01079	3	237.414	3	-296.981	574.536	94.016	87.702	
30	4	1350.00	7	-0.00235	8	0.01007	4	221.471	4	-574.536	976.621	178.328	155.248	
31	5	0.00	9	-0.00189	10	0.00876	5	192.866	5	-976.621	1224.419	272.320	227.861	
32	6	0.00	11	-0.00074	12	0.00733	6	161.329	6	-1224.419	984.594	227.861	222.841	
33	7	0.00	13	0.00040	14	0.00717	7	157.775	7	-984.594	405.382	169.519	207.569	
34	8	0.00	15	0.00101	16	0.00785	8	172.801	8	-405.382	-194.373	139.140	150.760	
35	9	0.00	17	0.00108	18	0.00851	9	187.233	9	194.373	-468.259	58.844	58.450	
36	10	0.00	19	0.00100	20	0.00875	10	192.822	10	549.259	-384.282	58.450	59.906	
37	11	0.00	21	0.00090	22	0.00897	11	197.319	11	384.282	-141.201	117.207	122.105	
38	12	0.00	23	0.00078	24	0.00934	12	205.564	12	141.201	0.000	135.672	282.402	
39	13	0.00	25	0.00075	26	0.00972	13	213.941						
40	14	0.00												
41	15	0.00												
42	16	0.00												
43	17	0.00												
44	18	0.00												
45	19	81.00												
46	20	2025.00												
47	21	0.00												
48	22	0.00												
49	23	0.00												
50	24	0.00												
51	25	0.00												
52	26	0.00												
53														

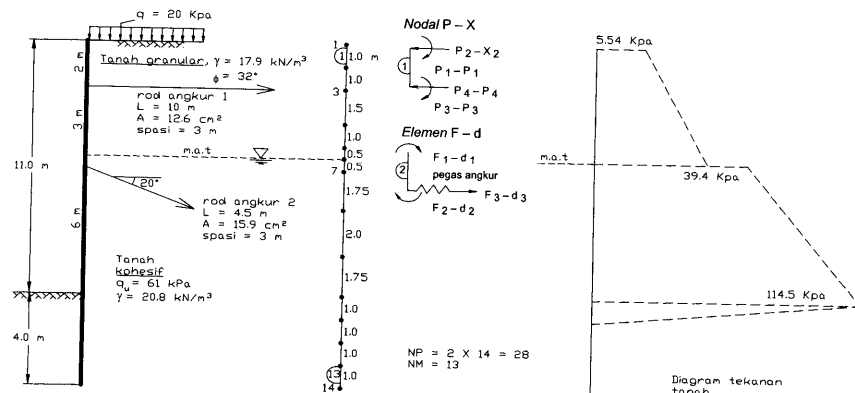
Tabel 5.1 Beberapa Perbandingan Output BOF dan PFB

	Hasil yang seharusnya	BOF	PFB
$\Sigma F_v = 0$	3375.0	3375.0	3371.92
ΣM ujung kanan elemen 1 dan kiri elemen 2 = P_3	-108.0	-108.0	-108.0
ΣM ujung kanan elemen 9 dan kiri elemen 10 = P_{19}	81.0	81.0	81.26
Gaya momen di ujung kiri elemen 1	0.0	0.0	0.0
Gaya momen di ujung kanan elemen 12	0.0	0.0	-0.063

Hasil output tegangan tanah (soil pressure) digunakan untuk memeriksa apakah pemberian beban pada pondasi tidak melebihi daya dukung izin tanah (q_a). Nilai yang maksimum (dalam soal ini dibagi dengan LF) tidak boleh melebihi q_a .

5.4 Dinding Penahan Tanah

Aplikasi program BOF sekarang akan dipakai untuk analisis dinding penahan tanah. Program sedikit dimodifikasi agar cocok untuk pemodelan di mana pada bagian yang bebas (di muka dinding) mempunyai $K = 0$. Program analisis dinding penahan tanah diberi nama WALL. Data dan model untuk contoh soal ini diadaptasi dari buku J.E Bowles, Foundation Analysis and Design.



Data dan model dari buku J.E Bowles, Foundation Analysis and Design, 1982.

Gambar 5.16 Analisis dinding penahan tanah dengan metode finite element

Diagram tekanan tanah terhadap dinding diperlihatkan dalam Gambar 5.16 dan gaya pada setiap nodal dapat dicari dengan interpolasi linear. Nilai pegas tanah kohesif di muka dinding diambil = 3660 kN/m. Sedangkan rod angkur terbuat dari bahan baja, dengan spesifikasi yang diperlihatkan dalam gambar. Nilai pegas rod angkur per meter lebar dinding diperoleh melalui rumus:

$$K = \frac{AE}{sL}$$

$$\text{Untuk rod 1} = \frac{12.6(2.0 \times 10^8)}{3(10)/10^4} = 8400 \text{ kN/m.}$$

$$\text{Untuk rod 2} = \frac{15.9(2.0 \times 10^8) \cos 20^\circ}{3(4.5)/10^4} \approx 22140 \text{ kN/m.}$$

Form input-output WALL

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	WALL														
2															
3	NP	28													
4	NM	13													
5															
6	E(wall)	2.00E+08	kN/m ²												
7															
8															
9															
10															
11	MEMNO	NP1	NP2	NP3	NP4	LENGTH	WIDTH	INERTIA							
12	1	1	2	3	4	1.000	1.000	0.00038							
13	2	3	4	5	6	1.000	1.000	0.00038							
14	3	5	6	7	8	1.500	1.000	0.00038			8,400.00				
15	4	7	8	9	10	1.000	1.000	0.00038							
16	5	9	10	11	12	0.500	1.000	0.00038							
17	6	11	12	13	14	0.500	1.000	0.00038							
18	7	13	14	15	16	1.750	1.000	0.00038			22,140.00				
19	8	15	16	17	18	2.000	1.000	0.00038							
20	9	17	18	19	20	1.750	1.000	0.00038							
21	10	19	20	21	22	1.000	1.000	0.00038			7,320.00	3,660.00	3,660.00		
22	11	21	22	23	24	1.000	1.000	0.00038			7,320.00	3,660.00	3,660.00		
23	12	23	24	25	26	1.000	1.000	0.00038			7,320.00	3,660.00	3,660.00		
24	13	25	26	27	28	1.000	1.000	0.00038			7,320.00	3,660.00	3,660.00		
25	LOAD MATRIX														
26	NP	Load	No	Rotation (rad)	Translation (m)	Soil Pressur	No	Moments (kN.m)							
27	1	0.00	1	0.00281	2	-0.00349	1	0.000	3.597	0.000	0.000				
28	2	3.60	3	0.00264	4	-0.00067	2	-3.597	17.693	0.000	0.000				
29	3	0.00	5	0.00298	6	0.00223	3	-17.693	42.060	18.708	0.000				
30	4	10.50	7	0.00357	8	0.00707	4	-42.060	85.886	0.000	0.000				
31	5	0.00	9	0.00441	10	0.01101	5	-85.886	118.310	0.000	0.000				
32	6	20.86	11	0.00508	12	0.01337	6	-118.310	160.712	0.000	0.000				
33	7	0.00	13	0.00599	14	0.01613	7	-160.712	-201.777	357.082	0.000				
34	8	27.58	15	0.00552	16	0.02741	8	-201.777	-337.995	0.000	0.000				
35	9	0.00	17	-0.00156	18	0.03196	9	-337.995	-147.753	0.000	0.000				
36	10	-21.02	19	-0.00714	20	0.02371	10	-147.753	6.879	86.772	58.320				
37	11	0.00	21	-0.00807	22	0.01593	11	116.640	-6.879	44.871	58.320	29.265			
38	12	19.95	23	-0.00773	24	0.00800	12	58.531	-44.871	24.333	29.265	1.897			
39	13	0.00	25	-0.00727	26	0.00052	13	3.794	-24.333	0.000	1.897	-24.333			
40	14	65.14	27	-0.00711	28	-0.00665	14	-48.686							
41	15	0.00													
42	16	139.03													
43	17	0.00													
44	18	176.82													
45	19	0.00													
46	20	132.69													
47	21	0.00													
48	22	0.00													
49	23	0.00													
50	24	0.00													
51	25	0.00													
52	26	0.00													
53															

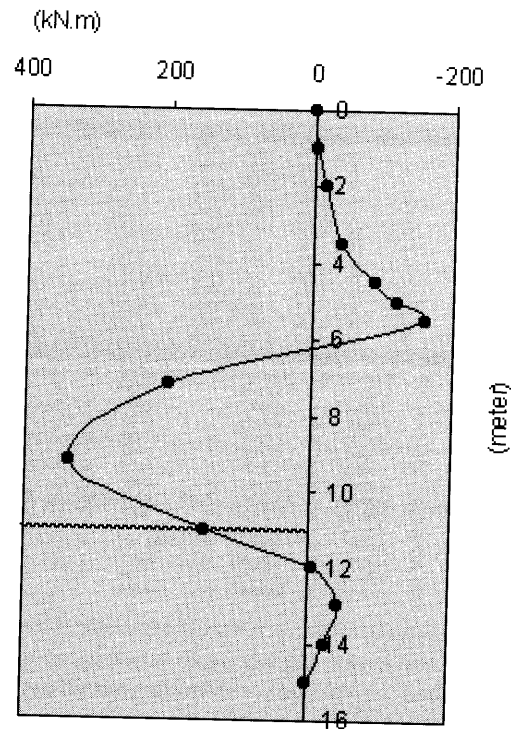
Hasil gaya momen kemudian dibuat ke dalam grafik seperti yang diperlihatkan pada Gambar 5.17. Untuk keperluan desain, diambil momen lentur positif maksimum yang bekerja di bawah angkur, sedangkan momen negatifnya berada di sekitar angkur dan sekitar ujung tancapan. Gaya dan tegangan pada rod angkur diperoleh sebagai berikut:

$$\text{Rod 1: } P_{\text{aksial}} = 3 \times 18.708 = 56.124 \text{ kN}$$

$$f_s = \frac{P_{\text{aksial}}}{A} = \frac{56.124}{12.6} = 4.454 \text{ kN/cm}^2$$

$$\text{Rod 2: } P_{\text{aksial}} = 3 \times 357.082 / \cos 20^\circ = 1139.996 \text{ kN}$$

$$f_s = \frac{P_{\text{aksial}}}{A} = \frac{1139.996}{15.9} = 71.698 \text{ kN/cm}^2$$



Gambar 5.17 Distribusi momen lentur sepanjang dinding

5.4 Pondasi Pelat

Program komputer untuk analisis pondasi pelat dapat menggunakan metode finite grid (FG) seperti yang diajukan oleh Bowles (1974). Metode ini adalah suatu pendekatan terhadap analisis finite element. Analisis dengan metode FG memakai persamaan-persamaan yang sama dengan analisis balok di atas medium elastis pada Bab 5.3. Dengan menjumlahkan gaya-gaya pada setiap nodal dalam Gambar 5.18 dan dari sifat elastis elemen, [A] dan [S] elemen dibangun sebagai berikut.

$$[A] = \begin{bmatrix} -\sin\alpha & 0 & -\cos\alpha & 0 & 0 \\ \cos\alpha & 0 & -\sin\alpha & 0 & 0 \\ \frac{1}{L} & \frac{1}{L} & 0 & -1 & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 & 0 \\ -\frac{1}{L} & -\frac{1}{L} & 0 & 0 & -1 \end{bmatrix}$$

$$[S] = \begin{bmatrix} \frac{4EI}{L} & \frac{2EI}{L} & 0 & 0 & 0 \\ \frac{2EI}{L} & \frac{4EI}{L} & 0 & 0 & 0 \\ 0 & 0 & \frac{\Omega GJ}{L} & 0 & 0 \\ 0 & 0 & 0 & K_1 & 0 \\ 0 & 0 & 0 & 0 & K_2 \end{bmatrix}$$

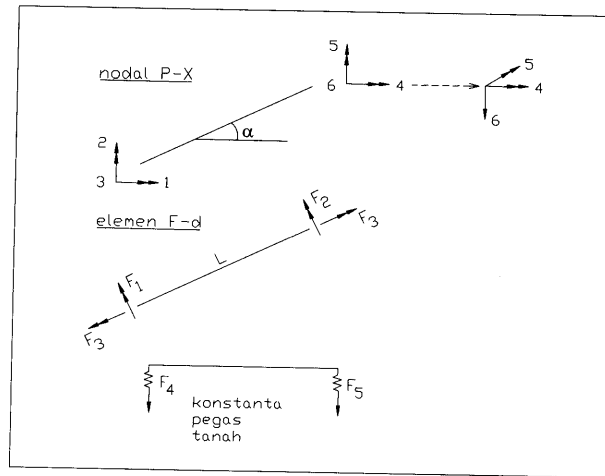
Keterangan:

G, Ω, J komponen torsi
 (G : modulus geser bahan pondasi, J : momen inersia polar,
 Ω : faktor torsi, diambil = 1 untuk pondasi pelat)
 K_1, K_2 konstanta pegas tanah

Proses perhitungan dalam program sama seperti bagan alir BOF. Program analisis untuk pondasi pelat (mat foundation) diberi nama MATF.

Contoh:

Pondasi pelat di atas tanah yang uniform dengan $K_s = 2000 \text{ ton/m}^3$, menerima beban vertikal dan momen pada kolomnya. Data input selengkapnya diperlihatkan dalam form input MATF berdasarkan geometri dan sistem penomoran grid pada Gambar 5.19.

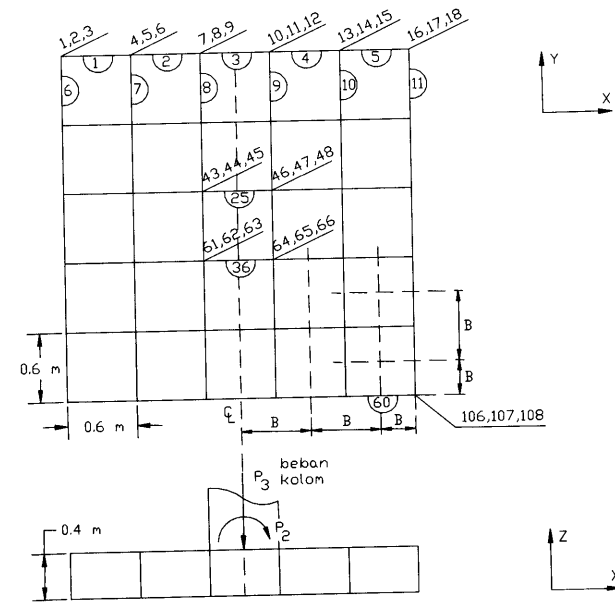


Gambar 5.18 Komponen gaya-gaya finite element grid untuk analisis pondasi pelat

Data Input:

Beban vertikal kolom diketahui = 40 ton. Dalam program, besar beban ini menjadi terbagi rata sebesar 10 ton kepada 4 nodal kolom yaitu pada nodal 15, 16, 21 dan 22. Demikian pula dengan beban momennya.

Data input untuk K dibedakan untuk nodal sebelah dalam dan luar. Pada nodal sebelah dalam harus diperhitungkan pengaruh akumulasi pegas tanah dari elemen-elemen yang bersebelahan karena superposisinya. Dengan demikian, secara global kontribusinya akan mencerminkan kondisi tanah yang uniform.



Gambar 5.19 Geometri struktur dan sistem penomoran grid

Dari sistem penomoran grid pada Gambar 5.19, Anda akan mendapatkan data input seperti dalam form. Tampak bahwa data yang dibutuhkan untuk membangun grid ini sangat banyak, oleh karena itu diperlukan suatu cara yang tepat untuk mengisinya. Caranya dengan membatasi bagian-bagian yang tipikal untuk kemudian di-copy ke tempat lain. Grid pondasi yang simetris ini menghasilkan banyak bagian-bagian data yang tipikal, baik nilainya maupun formula yang digunakan (dalam form dibatasi dengan garis).

Form input MATF

Form output MATF

MATF		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Grid Size	5	5	(K vs. V)	Q	1.0	Unitless-meter													
2	NP	100			μ	0.15														
3	NM	60			G	913,0435	ton/m ²													
4	NJ	36																		
5	Dim	0.40 m																		
6	Ec	2.10E+06	ton/m ²																	
7	Script	100	a																	
MEMNO	NP1	NP2	NP3	NP4	NP5	NP6	LENGTH	WIDTH	INERTIA	INERTIA	α	Ks (ton/m ²)	K (ton/m)	Node	P1	P2	P3			
1	1	2	3	4	5	6	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	1	0.00	0.00	0.00		
2	4	5	6	7	8	9	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	2	0.00	0.00	0.00		
3	7	8	9	10	11	12	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	3	0.00	0.00	0.00		
4	10	11	12	13	14	15	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	4	0.00	0.00	0.00		
5	13	14	15	16	17	18	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	5	0.00	0.00	0.00		
6	16	17	18	19	20	21	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	6	0.00	0.00	0.00		
7	19	20	21	22	23	24	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	7	0.00	0.00	0.00		
8	22	23	24	25	26	27	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	8	0.00	0.00	0.00		
9	25	26	27	28	29	30	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	9	0.00	0.00	0.00		
10	28	29	30	31	32	33	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	10	0.00	0.00	0.00		
11	31	32	33	34	35	36	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	11	0.00	0.00	0.00		
12	34	35	36	37	38	39	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	12	0.00	0.00	0.00		
13	37	38	39	40	41	42	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	13	0.00	0.00	0.00		
14	40	41	42	43	44	45	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	14	0.00	0.00	0.00		
15	43	44	45	46	47	48	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	15	0.00	0.00	0.00		
16	46	47	48	49	50	51	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	16	0.00	0.00	0.00		
17	49	50	51	52	53	54	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	17	0.00	0.00	0.00		
18	52	53	54	55	56	57	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	18	0.00	0.00	0.00		
19	55	56	57	58	59	60	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	19	0.00	0.00	0.00		
20	58	59	60	61	62	63	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	20	0.00	0.00	0.00		
21	61	62	63	64	65	66	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	21	0.00	0.00	0.00		
22	64	65	66	67	68	69	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	22	0.00	0.00	0.00		
23	67	68	69	70	71	72	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	23	0.00	0.00	0.00		
24	70	71	72	73	74	75	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	24	0.00	0.00	0.00		
25	73	74	75	76	77	78	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	25	0.00	0.00	0.00		
26	76	77	78	79	80	81	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	26	0.00	0.00	0.00		
27	79	80	81	82	83	84	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	27	0.00	0.00	0.00		
28	82	83	84	85	86	87	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	28	0.00	0.00	0.00		
29	85	86	87	88	89	90	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	29	0.00	0.00	0.00		
30	88	89	90	91	92	93	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	30	0.00	0.00	0.00		
31	91	92	93	94	95	96	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	31	0.00	0.00	0.00		
32	94	95	96	97	98	99	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	32	0.00	0.00	0.00		
33	97	98	99	100	101	102	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	33	0.00	0.00	0.00		
34	100	101	102	103	104	105	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	34	0.00	0.00	0.00		
35	103	104	105	106	107	108	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	35	0.00	0.00	0.00		
36	106	107	108	109	110	111	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	36	0.00	0.00	0.00		
37	109	110	111	112	113	114	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	37	0.00	0.00	0.00		
38	112	113	114	115	116	117	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	38	0.00	0.00	0.00		
39	115	116	117	118	119	120	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	39	0.00	0.00	0.00		
40	118	119	120	121	122	123	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	40	0.00	0.00	0.00		
41	121	122	123	124	125	126	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	41	0.00	0.00	0.00		
42	124	125	126	127	128	129	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	42	0.00	0.00	0.00		
43	127	128	129	130	131	132	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	43	0.00	0.00	0.00		
44	130	131	132	133	134	135	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	44	0.00	0.00	0.00		
45	133	134	135	136	137	138	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	45	0.00	0.00	0.00		
46	136	137	138	139	140	141	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	46	0.00	0.00	0.00		
47	139	140	141	142	143	144	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	47	0.00	0.00	0.00		
48	142	143	144	145	146	147	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	48	0.00	0.00	0.00		
49	145	146	147	148	149	150	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	49	0.00	0.00	0.00		
50	148	149	150	151	152	153	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	50	0.00	0.00	0.00		
51	151	152	153	154	155	156	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	51	0.00	0.00	0.00		
52	154	155	156	157	158	159	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	52	0.00	0.00	0.00		
53	157	158	159	160	161	162	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	53	0.00	0.00	0.00		
54	160	161	162	163	164	165	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	54	0.00	0.00	0.00		
55	163	164	165	166	167	168	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	55	0.00	0.00	0.00		
56	166	167	168	169	170	171	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	56	0.00	0.00	0.00		
57	169	170	171	172	173	174	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	57	0.00	0.00	0.00		
58	172	173	174	175	176	177	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	58	0.00	0.00	0.00		
59	175	176	177	178	179	180	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	59	0.00	0.00	0.00		
60	178	179	180	181	182	183	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	60	0.00	0.00	0.00		
61	181	182	183	184	185	186	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	61	0.00	0.00	0.00		
62	184	185	186	187	188	189	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	62	0.00	0.00	0.00		
63	187	188	189	190	191	192	0.600	0.300	0.0016	0.0025	0	2.0000	180.00	180.00	63	0.00				

- $\Sigma MNJ_2 = 0.29189 + 0.40634 + (-0.69823) = 0,$
- $\Sigma MNJ_{10} = -2.73479 + 4.26324 - (-1.38817) + (-2.91662) = 0 \dots \text{OK! dst.}$

Catatan:

1. * dalam menyamakan ΣF_v di nodal, analisis FG mengambil nilai negatif untuk gaya pegas elemen. Jadi gaya yang negatif menyatakan gaya ini pada translasi yang positif (searah gravitasi) atau sebaliknya (gaya positif menyatakan kondisi tarik pada pegas).
2. Dalam contoh soal ini terjadi translasi yang negatif (tarik pada pegas), di mana sebagian pondasi bergerak menjauhi tanah. Oleh karena itu matriks ASA^T perlu dibangun kembali dengan $K = 0$ pada nodal-nodal tersebut. Di sini berat pondasi juga perlu dimasukkan ke dalam perhitungan. Hasilnya kemudian diperiksa dan jika perlu ulangi langkah ini hingga tercapai konvergensi dalam penyelesaiannya.

Lendutan pondasi kemudian diplot dan hasilnya dipresentasikan dalam Gambar 5.20. Mengeluarkan hasil lendutan ke dalam penggambaran grafik sebenarnya merupakan pengujian yang paling mudah dilakukan untuk memeriksa apakah semua data input sudah dituliskan dengan benar. Grafik 3D Excel di sini banyak sekali membantu, terutama untuk memeriksa data input yang memang relatif banyak seperti dalam contoh soal di atas. Dapat juga ditambahkan bahwa setiap kali selesai membangun atau merancang matriks, Anda dapat memeriksa setiap komponennya dengan cara mencetak ke worksheet. Cari dalam form tempat yang kosong untuk menempatkannya dan jangan lupa memperhitungkan ukuran matriks yang akan dicetak agar tidak menghapus isi data yang lain.

Skema matriks global ASA^T diperlihatkan pada Gambar 5.21. Sebenarnya setiap konstruksi akan mempunyai skema matriks globalnya sendiri yang spesifik, umumnya merupakan band matriks di mana komponen-komponen yang bukan nol akan membentuk jalur komponen diagonal, sedangkan komponen yang nilainya nol berada di luar tepi diagonal. Pada umumnya juga merupakan matriks simetris. Di sini ada suatu rumusan untuk mencari bandwidth, yaitu jarak maksimum komponen matriks elemen terhadap diagonal matriks global pada setiap baris. Untuk grid seperti dalam contoh soal ini, rumusan bandwidth (NB) adalah:

$$NB = NP_{\max} - NP_{\min} + 1$$

Di mana nilai NP_{\max} dan NP_{\min} ditentukan dari dua nodal yang bersebelahan dalam satu elemen grid. Dari Gambar 5.19, NB adalah $= 21 - 1 + 1 = 21$ dan skemanya seperti yang diperlihatkan dalam Gambar 5.21.

Dalam pemrograman di Excel, rumusan seperti bandwidth ini tidak perlu digunakan dalam merancang matriks. Anda dapat memanfaatkan form untuk membuat variabel yang dibutuhkan dalam memanipulasi langkah-langkah program. Contohnya sudah diperlihatkan dalam teknik-teknik penulisan FRAME2D dan TRUSS.

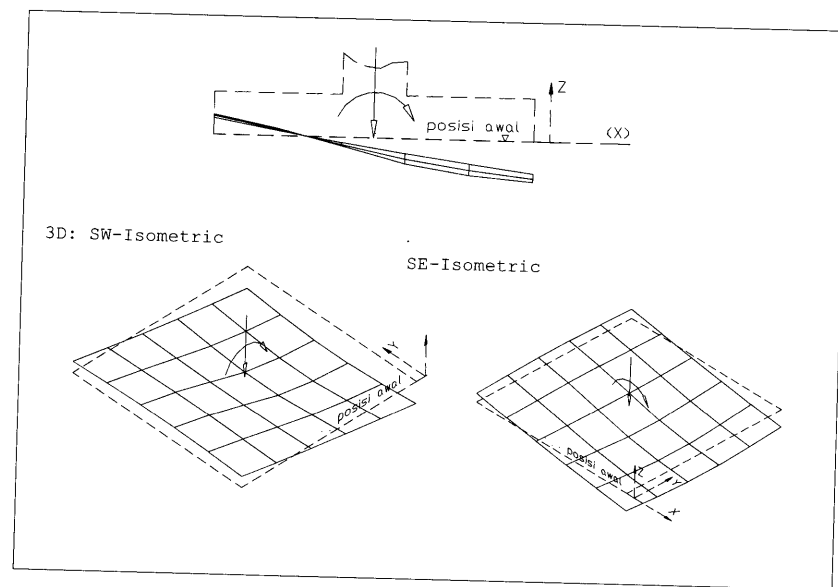
Selain teknik penulisan untuk merancang-bangun matriks, teknik penulisan untuk mencetak hasil program ke worksheet sama pentingnya untuk dikuasai. Anda harus dapat memilah-milah setiap hasil output program dan sekaligus mengurutkannya. Dalam MATF, X_1 dan X_2 menyatakan vektor rotasi di nodal, dan X_3 untuk translasinya. F_1, F_2, F_3 menyatakan vektor gaya momen di elemen, serta F_4 dan F_5 untuk gaya vertikalnya. Berikut ini penulisan untuk pencetakan lendutan $\{X\}$ ke dalam worksheet:

'Cetak $\{X\}$:

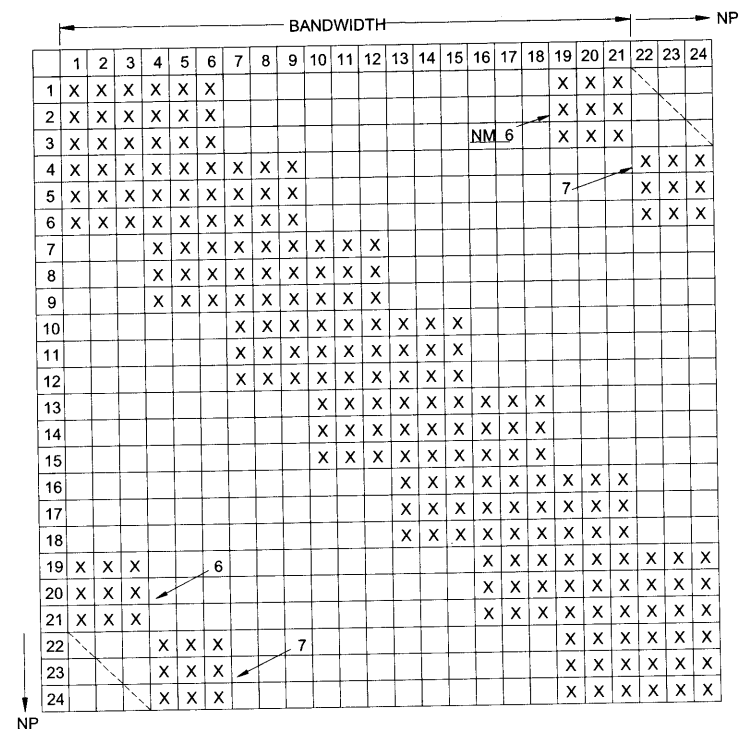
```
a = 1 ' untuk rotasi akibat torsi
b = 2 ' untuk rotasi akibat momen
c = 3 ' untuk translasi
d = 1 'soil pressure = KS * X(3)
For II = 1 To NJ
Cells(13 + II, 18) = a
Cells(13 + II, 19) = X(a, 1)
Cells(13 + II, 20) = b
Cells(13 + II, 21) = X(b, 1)
Cells(13 + II, 22) = c
Cells(13 + II, 23) = X(c, 1)
Cells(13 + II, 24) = d
Cells(13 + II, 25) = KS(II, 1) * X(c, 1)
a = a + 3
b = b + 3
c = c + 3
d = d + 1
Next II
```

Dengan perintah di atas, program akan mencetak baris demi baris yang dimulai dari baris ke-14 pada worksheet. Variabel a, b, c dibuat untuk memanipulasi langkah program agar hasil lendutan tercetak dengan suatu kelipatan. Nilai 3 ditentukan berdasarkan jumlah vektor $P-X$ di setiap nodal. Di sini program juga menyisipkan indeks lendutan sebelum mencetak hasil lendutannya.

Perintah pencetakan untuk gaya elemen $\{F\}$ sedikit lebih bervariasi. Untuk mempelajarinya, lihat pada form di mana hasil $\{F\}$ ditempatkan, kemudian buka listing program MATF. Anda dapat memahami hubungan perintah program dan pencetakannya dengan menjalankan program langkah per langkah dengan menekan **Ctrl+F8 > F8**.



Gambar 5.20 Lendutan pondasi pelat



Gambar 5.21 Skema matriks global pelat

5.5 Konsolidasi Satu Dimensi

Konsolidasi satu dimensi dapat diselesaikan dengan cara numerik, yaitu dengan metode finite difference (FD). Metode ini berdasarkan pada suatu grid kedalaman dan waktu seperti yang diperlihatkan Gambar 5.22. Kedalaman lapisan tanah dibagi m bagian yang sama dengan tebal Δz , sedangkan periode waktu dibagi n bagian yang sama dengan selang Δt . Penyelesaian dengan metode finite difference diambil dari teorema Taylor, di mana untuk persamaan konsolidasi satu dimensi menjadi,

$$u_{i,j+1} = u_{i,j} + \beta (u_{i-1,j} + u_{i+1,j} - 2u_{i,j}) \quad (5.9)$$

di mana, $\beta = \frac{c_v \Delta t}{(\Delta z)^2}$

c_v koefisien konsolidasi

$u_{i,j}$ tekanan air pori berlebih pada kedalaman i dan waktu t .

β disebut sebagai operator dari Persamaan 5.9, dan supaya konvergen nilai ini tidak boleh melebihi $\frac{1}{2}$. Nilai yang dipakai berada dalam kisaran $\frac{1}{6}$ hingga $\frac{1}{2}$.

Untuk suatu periode waktu t pada lapisan terbuka:

$$T_v = \frac{c_v (n \Delta t)}{(\frac{1}{2} m \Delta z)^2} = 4 \beta n / m^2 \quad (5.10)$$

Pada lapisan setengah tertutup, penyebutnya menjadi $(m \Delta z)^2$:

$$T_v = \beta n / m^2 \quad (5.11)$$

T_v disebut faktor waktu.

Untuk titik-titik pada batas impermeabel di mana tidak ada aliran air yang melewati batas ini, Persamaan 5.9 menjadi:

$$u_{i,j+1} = u_{i,j} + \beta (2u_{i-1,j} - 2u_{i,j}) \quad (5.12)$$

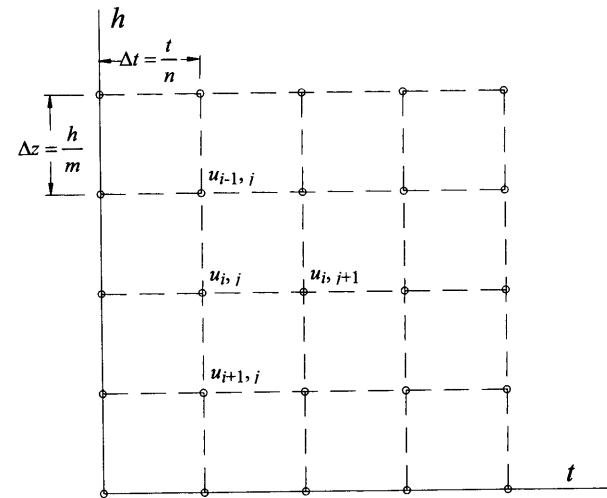
Dalam pemrograman, baik Persamaan 5.9 maupun Persamaan 5.12 tidaklah sulit untuk dibuatkan programnya, karena analisisnya berdasarkan pada suatu grid yang sama sisi. Hal ini dapat dibuat seperti membangun sebuah matriks dengan ukuran $m \times n$ (lihat Gambar 5.22). Metode FD ini sebenarnya juga dapat diselesaikan dengan formula spreadsheet. Oleh karena bentuk penyelesaiannya adalah sebuah grid $m \times n$, maka dapat dibuat dalam worksheet menjadi baris dan kolom. Namun di sini banyak ditemukan keterbatasan-keterbatasan yang paling mendasar dan penyelesaiannya sudah menjadi tidak praktis. Dalam pemrograman VB, penyelesaian grid FD dikerjakan dengan membuatnya ke dalam perintah pengulangan. Dengan demikian untuk mencari hasil tekanan air pori pada periode waktu tertentu dapat diketahui dengan memberikan nilai yang ditetapkan pada pengulangan tersebut. Untuk lapisan setengah tertutup dapat ditulis:

```

5  For j = 0 To n 'half closed layered
    u(0, j + 1) = 0
  Next j
  For i = 1 To m
    u(i, 0) = Cells(12 + i, 2)
  Next i

  For j = 0 To n
    For i = 1 To m
      If i = m Then u(i, j + 1) = u(i, j) + Beta * (2 * u(i - 1, j)
        - 2 * u(i, j)): GoTo 10
      u(i, j + 1) = u(i, j) + Beta * (u(i - 1, j) + u(i + 1, j) - 2
        * u(i, j))
    Next i
  Next j: GoTo 20
10
20

```



Gambar 5.22 Grid kedalaman dan waktu ($h-t$)

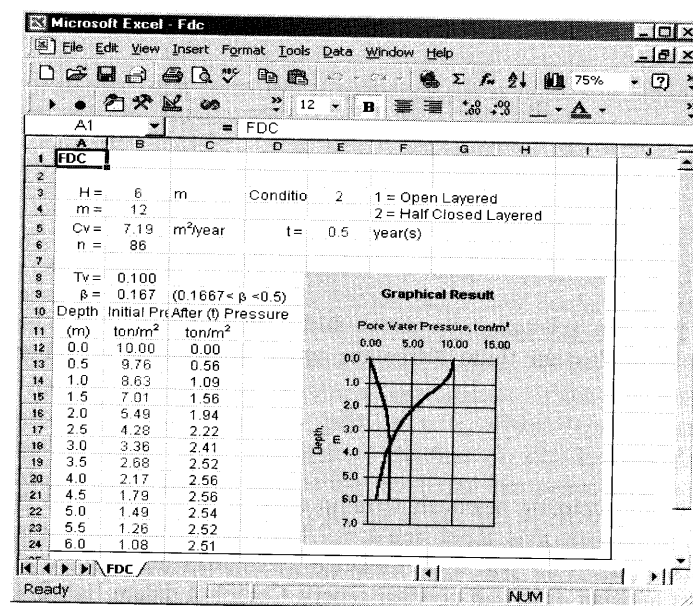
Listing program selengkapnya dapat dilihat pada bagian akhir buku ini. Program penyelesaian finite difference untuk konsolidasi ini diberi nama FDC.

Contoh 1:

Suatu lapisan lempung setengah tertutup (half closed layered) di mana air mengalir pada batas atas. Koefisien $c_v = 7.19 \text{ m}^2/\text{tahun}$ dan tebal lapisan 6 m. Distribusi tekanan air pori awal terhadap kedalaman diambil sama dengan distribusi tekanan di bawah pondasi seperti Contoh 4 dalam Bab 2. Diminta mencari besarnya distribusi tekanan air pori setelah 6 bulan.

Kedalaman (m)	Tekanan air pori (ton/m ²)
0.0	10.00
0.5	9.76
1.0	8.63
1.5	7.01
2.0	5.49
2.5	4.28
3.0	3.36
3.5	2.68
4.0	2.17
4.5	1.79
5.0	1.49
5.5	1.26
6.0	1.08

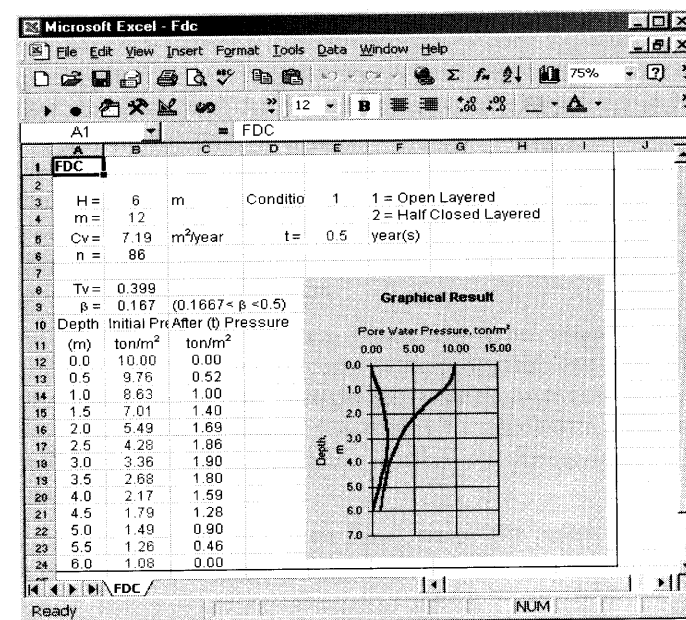
Form input-output FDC



Contoh 2:

Sekarang kondisi lapisan lempung berada di atas suatu lapisan pasir, sehingga dapat diasumsikan sebagai lapisan yang terbuka (open layered) di mana air juga mengalir pada batas bawah. Data yang diberikan sama seperti soal 1.

Form input-output FDC



Hasil di atas memperlihatkan bahwa konsolidasi lebih cepat selesai pada lapisan terbuka dibanding lapisan setengah tertutup. Hal ini memang disebabkan lintasan airnya yang lebih pendek seperti juga yang dinyatakan oleh faktor waktu (T_v). Hasil akhir dari tekanan air pori akan berangsur menjadi simetris terhadap tengah-tengah lapisan lempung. Pada batas atas dan bawah yang permeabel, tekanan air pori akan segera nol sesaat setelah konsolidasi dimulai hingga periode waktu t . Di sini diperlukan suatu *judgement* yang tepat untuk memodelkan kondisi batas bawah ke dalam FDC, terutama untuk lapisan lempung yang relatif tipis.

Keuntungan dengan metode finite difference, yaitu setiap pola dari tekanan air pori berlebih dapat dimasukkan ke dalam perhitungan. Data input dapat berdasarkan keadaan awal di lapangan atau model beban yang bekerja secara bertahap, misalnya dalam penyelesaian konsolidasi tanah di bawah bendungan atau tanah timbunan.

BAB 6

SCRIPT FILE AUTOCAD

Dalam menampilkan output program secara grafis, dapat digunakan alternatif lain dengan membuatnya di AutoCAD. Anda dapat bekerja lebih jauh dengan grafik atau gambar struktur. Kelebihan AutoCAD adalah software ini dibuat hanya khusus untuk menyelesaikan pekerjaan gambar. Pada awalnya, cukup diperlukan penguasaan pada tingkat dasar untuk dapat membuat tampilan grafik, mengedit, mengatur halaman dan mencetaknya ke atas kertas. Namun minat yang semakin besar untuk membuat program, akan seiring dengan kecakapan menggunakan software ini.

Script file di sini adalah file yang berisi teks urutan perintah yang dijalankan oleh AutoCAD untuk proses suatu penggambaran. Urut-urutan ini dibuat otomasinya oleh VBA sebagai ganti langkah-langkah manual yang memerlukan waktu lama. Script file dibuat menggunakan pengolah kata, seperti MS.WIN Notepad atau MS. Word yang disimpan dalam format ASCII dan diberi extension .scr.

Gambar 5.20 menunjukkan contoh hasil script file untuk penggambaran lendutan (translasi) pondasi pelat. Di dalamnya berisi teks berikut.

```
_3dmesh
6
6
0,0,0.224472720870677
0,0.6,0.220394494686006
0,1.2,0.218264578883796
0,1.8,0.218264578883788
0,2.4,0.220394494685983
0,3,0.224472720870639
0.6,0,7.03540050240806E-02
0.6,0.6,6.43051243562464E-02
0.6,1.2,6.18111921247522E-02
```


0.6,1.8,6.18111921247445E-02
 0.6,2.4,0.064305124356223
 0.6,3,7.03540050240418E-02
 1.2,0,-8.40146054744982E-02
 1.2,0.6,-9.59696803024441E-02
 1.2,1.2,-0.103838921406436
 1.2,1.8,-0.103838921406445
 1.2,2.4,-9.59696803024683E-02
 1.2,3,-0.084014605474538
 1.8,0,-0.233739891927716
 1.8,0.6,-0.253231609112864
 1.8,1.2,-0.268952870823938
 1.8,1.8,-0.268952870823946
 1.8,2.4,-0.253231609112889
 1.8,3,-0.233739891927757
 2.4,0,-0.371850787539531
 2.4,0.6,-0.394176821686788
 2.4,1.2,-0.411012481540078
 2.4,1.8,-0.411012481540086
 2.4,2.4,-0.394176821686812
 2.4,3,-0.371850787539572
 3,0,-0.500891019972049
 3,0.6,-0.521891277201477
 3,1.2,-0.536019076286047
 3,1.8,-0.536019076286054
 3,2.4,-0.521891277201499
 3,3,-0.500891019972088

Keterangan:

1. `_3dmesh` adalah perintah yang dijalankan pertama kali oleh AutoCAD untuk memulai penggambaran mesh atau grid 3 dimensi.
2. Perintah berikut untuk menentukan ukuran mesh secara berurutan dalam arah aksis-X dan Y. Pondasi pelat pada contoh Bab 5.5 mempunyai ukuran 6 x 6 sesuai dengan jumlah nodal pada kedua sisinya. Setiap ukuran ini ditulis pada satu baris sendiri, berisi angka tanpa boleh disertai oleh karakter lain termasuk spasi.
3. Menentukan koordinat masing-masing nodal. AutoCAD menggunakan titik vertex, yaitu lokasi dari setiap sudut elemen mesh. Data input dibaca mulai dari vertex (0,0), pada $x = 0$, $x = 0.6$ dan seterusnya seperti dalam script di atas. Penting untuk dicatat bahwa hasil output MATF untuk

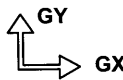
lendutan tidak mempunyai urutan yang sama dengan yang dibaca oleh AutoCAD. Oleh karena itu diperlukan penyesuaian dan dibuatkan urutan-urutannya.

Bentuk perintah untuk penulisan script file adalah:

```
Open "c:\nama file" For Output As #1
Print #1, "teks"
Write #1, variabel
Print #1, numerik & "karakter " & numerik
Close #1
```

Dengan kode program di atas, secara otomatis script file akan diberi nama "nama file" dan disimpan di direktori C. Tanda "&" setelah statement `Print #1` untuk menggabungkan data karakter dan numerik dalam membuat vertex atau nodal (x,y,z). Untuk maksud yang sama, Anda juga dapat menggunakan statement `Write #1, var x, var y, var z`.

Masalah yang dihadapi di sini adalah bagaimana menyamakan pembacaan output program dengan urutan perintah dalam AutoCAD. Seperti diketahui, hasil translasi pelat dicetak ke dalam worksheet dengan mengurutkan dari nomor subskrip yang terkecil hingga terbesar, yaitu X_3 , X_6 hingga X_{108} . Penempatannya seperti yang diperlihatkan pada Gambar 6.1a. Sedangkan dari pembacaan vertex AutoCAD dimulai dari X_{93} , X_{75} , ... X_{96} , X_{78} , ... dan seterusnya hingga yang terakhir, yaitu X_{18} .



3	6	9	12	15	18
21	24	27	30	33	36
39	42	45	48	51	54
57	60	63	66	69	72
75	78	81	84	87	90
93	96	99	102	105	108

Gambar 6.1a Penomoran untuk translasi pada pelat

Di sini kita perlu membangun suatu matriks baris (NJ x 1) yang berisi urutan nomor translasi yang dibaca AutoCAD berdasarkan Gambar 6.1a. Sebagai variabel bantu, program sudah mempunyai $X(NJ \times 1)$ yaitu berisi data translasi dengan urutan 3, 6, 9, hingga 108. Jadi Anda diminta membangun matriks baru (NJ x 1) dari $X(NJ \times 1)$. Matriks baris yang akan dicari, misalnya dinamakan $\{X_r\}$. Untuk mendapatkan $\{X_r\}$, langkah pertama adalah membuat dulu suatu matriks (6 x 6) atau (GY+1 x GX+1) yaitu untuk menggambarkan model penomoran pada Gambar 6.1a, dimulai dari nodal di sebelah kiri atas kemudian ke arah kanan. Perintahnya adalah:

```
a = 3
For II = 1 To GY + 1
For IJ = 1 To GX + 1
Mat(II, IJ) = X(a, 1)
a = a + 3
Next IJ
Next II
```

Dengan demikian, isi komponen [Mat] adalah $Mat(1,1) = X_3$, $Mat(1,2) = X_6$, hingga $Mat(6,6) = X_{108}$. Langkah kedua adalah membuat skema urutan pembacaan AutoCAD di dalam [Mat] itu sendiri:

$$[Mat] = \begin{bmatrix} 1,1 & 1,2 & 1,3 & 1,4 & 1,5 & 1,6 \\ \uparrow & \nearrow & \uparrow & \searrow & & \uparrow \\ \uparrow & \searrow & \uparrow & \searrow & & \uparrow \\ 6,1 & 6,2 & 6,3 & 6,4 & 6,5 & 6,6 \end{bmatrix}$$

Dari skema tersebut dapat dibuat perintah berikut:

```
a = 1
For II = 1 To GX + 1
For IJ = GY + 1 To 1 Step -1
Xr(a, 1) = Mat(IJ, II) * Sc 'Sc adalah skala agar lendutan
tampak dalam gambar
a = a + 1
Next IJ
Next II
```

Setelah membuat urutan pembacaan untuk lendutan translasinya (aksis-Z), sekarang yang harus dibuat adalah koordinat grid pada aksis-X dan Y. Perhatikan sebuah grid pelat pada Gambar 6.1b. Kuncinya di sini adalah, setiap panjang elemen dalam grid diwakili oleh setiap panjang elemen pada sisi atas dan samping pelat. Koordinat grid kemudian diperoleh dengan

menjumlahkan panjang-panjang elemen tersebut. Setelah itu diurutkan sesuai pembacaan di AutoCAD, sama seperti dalam mengerjakan translasi di atas.

Elemen no:	1	2	3	4	5
6					
17					
28					
39					

Gambar 6.1b Grid pondasi pelat

Perintah selengkapnya untuk penggambaran AutoCAD adalah:

```
'=====
'Membuat AutoCAD ScriptFile: MatF1.scr
'untuk plot Lendutan
'=====
```

'Membuat urutan lendutan:

```
a = 3
For II = 1 To GY + 1
For IJ = 1 To GX + 1
Mat(II, IJ) = X(a, 1)
a = a + 3
Next IJ
Next II
```

```
a = 1
For II = 1 To GX + 1
For IJ = GY + 1 To 1 Step -1
Xr(a, 1) = Mat(IJ, II) * Sc
a = a + 1
Next IJ
Next II
```

'Membuat koordinat grid:

```
a = 0
For II = 1 To GY + 1
For IJ = 1 To GX + 1
LX(IJ, 1, II) = a
Next IJ
a = a + L(II, 1)
Next II
For II = 1 To GY + 1
a = 0
b = (GX + 1) + (GY - 1) * (GX + GX + 1)
For IJ = 1 To GX + 1
```

```

LY(IJ, 1, II) = a
If IJ = GX + 1 Then GoTo 30 Else a = a + L(b, 1)
b = b - (GX + GX + 1)
Next IJ
30 Next II

Open "c:\MatF1.scr" For Output As #1
Print #1, "_3dmesh"
Write #1, GY + 1
Write #1, GX + 1
d = 0
c = 1
For II = 1 To GX + 1
a = 0
For IJ = 1 To GY + 1
Print #1, LX(IJ, 1, II) & ", " & LY(IJ, 1, II) & ", " & -Xr(c, 1)
c = c + 1
a = a + 1
Next IJ
d = d + 1
Next II
Close #1

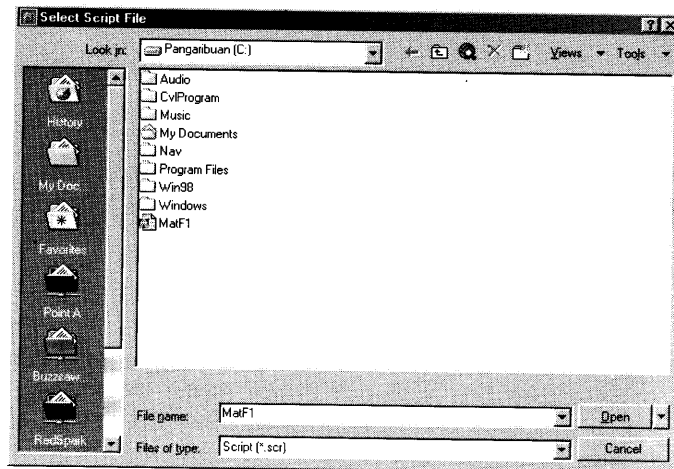
```

```

'=====
'=====


```

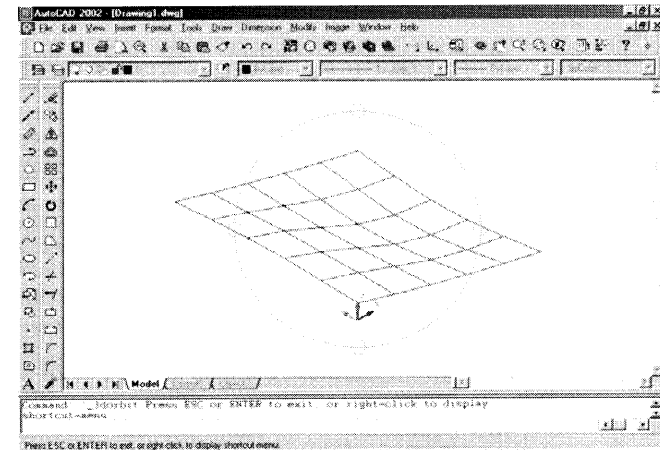
Sekarang jalankan program MATF. Setelah itu buka AutoCAD Anda. Ketikkan **scr** pada baris **Command** dan tekan Enter hingga keluar kotak dialog **Select Script File**. Pilih file **MatF1.scr** pada directory **C:**, kemudian klik **Open** atau tekan **Enter**. Isi script kemudian diplot pada layar. Gambar di sini dikerjakan pada AutoCAD versi 2002.



Gambar 6.2 Kotak dialog Select Script File

Untuk menampilkan gambar secara 3 dimensi, lakukan langkah berikut:

1. Dari menu **View**, klik **3D Views > Top**.
2. Klik icon **3D Orbit**  pada toolbar atau dari menu **View > 3D Orbit**.
3. Drag mouse pada kutub-kutub lingkaran orbit, hingga mendapatkan tampilan yang Anda inginkan.



Gambar 6.3 Tampilan AutoCAD 2002 dengan 3D Orbit

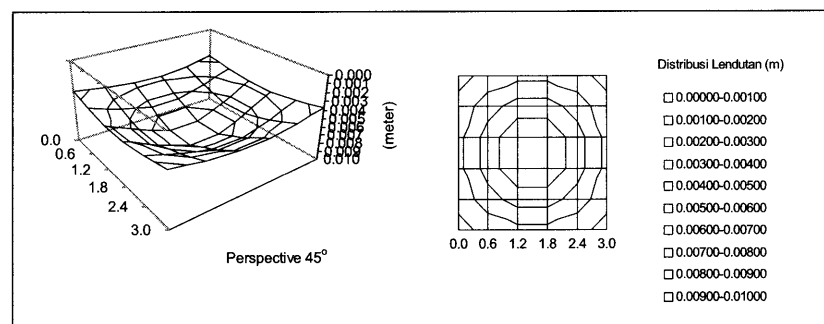
Untuk melihat urutan-urutan perintah penggambaran yang sudah dikerjakan AutoCAD, tekan **F2**.

Sebenarnya ada alternatif lain untuk membuat script file (namun tidak praktis), yaitu dengan membuat sebagian perintahnya di dalam worksheet, karena output lendutan sudah tercetak di sana. Kemudian hasilnya dipanggil dan diurutkan oleh VBA. Atau alternatif lainnya dengan membuat semuanya secara manual dalam worksheet menggunakan fungsi **CONCATENATE**. Setelah selesai kemudian di-copy dan di-paste di **WIN Notepad**. File disimpan dengan ekstension **.scr**. Di sini nilai positif (+) untuk penggambaran translasi yang berada di atas titik 0 (posisi awal) dan nilai negatif (-) untuk sebaliknya. Oleh karena itu, kalikan hasilnya dengan -1 agar arah translasi sesuai dengan arah pembebanannya.

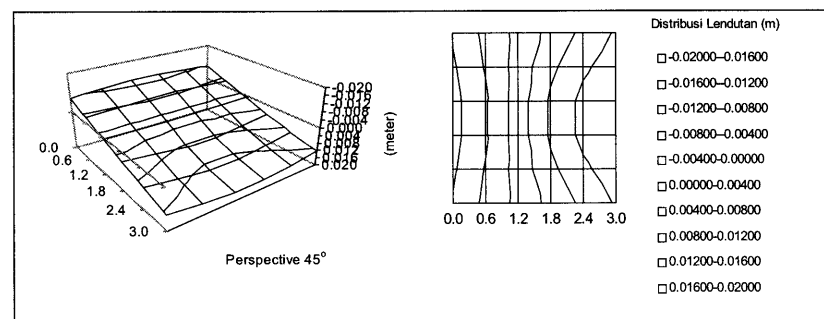
Membuat script file pada AutoCAD tidak hanya digunakan untuk penggambaran grafik dari hasil program atau membuat gambar struktur. Pada aplikasi lain juga dapat digunakan untuk pekerjaan topografi-survey,

pekerjaan cut and fill, membuat drilling log, dan sebagainya. Pada tingkat ini tentunya diperlukan penguasaan software yang lebih baik.

Tampilan gambar 3D dalam Excel dapat juga diberikan di sini, namun hal ini tentu tidak dapat diperbandingkan dengan tampilan AutoCAD. Gambar 6.4 dan 6.5 menunjukkan hasil contoh penggambaran Excel untuk lendutan pondasi pelat. Ada beberapa hal di sini yang dapat dicatat.



Gambar 6.4 Chart Excel untuk lendutan pondasi akibat beban vertikal



Gambar 6.5 Chart Excel untuk lendutan pondasi pelat akibat beban vertikal dan momen

Penggambaran grafik 3D di Excel dibuat dengan menggunakan **Chart type > Surface**. Satu serie data berisi sepasang data aksis-X dan Z, dan setiap serie data mempunyai pengukuran yang sama pada aksis-Z. Format aksis-Z kemudian dibuat dalam **reverse order** agar arah lendutan searah

pembebanannya. Namun grafik di sini tidak mempunyai opsi untuk data pada aksis-Y sehingga penggambaran lendutan hanya tepat untuk elemen grid yang sama panjang.

Secara default Excel akan mengelompokkan data pada aksis-Z yang mempunyai range sama. Setiap range dapat dibatasi alur-alur garis yang memotong elemen grid dan dikelompokkan ke dalam warna tertentu, apabila Anda memilih surface yang berwarna. Keuntungan pembagian ini adalah memudahkan melihat daerah range nilai tertentu yang mungkin tidak kasat mata dalam tampilan 3D.

Gambar 6.4 dan 6.5 memperlihatkan pondasi pelat dengan range lendutannya. Untuk beban vertikal yang terpusat seperti dalam Gambar 6.4, range lendutan akan simetris terhadap titik pusat pelat dan jika pada pelat juga dikerjakan momen, hasilnya seperti Gambar 6.5. Jadi, jika translasi berbanding lurus dengan tegangan tanah (soil pressure), lokasi tegangan yang mempunyai nilai-nilai maksimum dengan mudah dapat diperiksa dalam sistem koordinat pelat. Pada daerah ini tegangan tanahnya tidak boleh melebihi daya dukung tanah pada daerah tersebut. Penggambaran ini akan sangat membantu jika dikerjakan pada pelat-pelat yang relatif besar seperti pondasi raft di atas permukaan tanah yang mempunyai deskripsi tanah yang tidak seragam.


Penggambaran grafik tipe surface ini menjadi lebih maksimal jika kita juga mengeluarkan hasil distribusi gaya geser dan momen dari elemen struktur. Dan juga menjadi mudah untuk diperiksa arah distribusi gaya-gaya tersebut secara keseluruhan (global struktur).

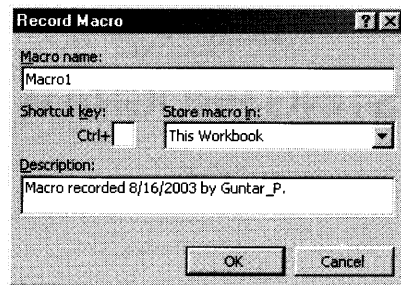
LISTING PROGRAM


7.1 Visual Basic for Application

Visual Basic for Application (VBA) adalah bahasa pemrograman yang diturunkan dari Visual Basic (VB) untuk pengembangan aplikasi macro. Selain Microsoft Office, banyak software non-Microsoft yang sekarang ini juga menggunakan VBA, sebut saja di antaranya AutoCAD, CorelDRAW, Visio, Norton, dan sebagainya. Jadi VB tidak mempunyai kaitan khusus dan bukan merupakan bagian internal dari Excel. VB membaca dan menganalisis objek yang diekspos oleh Excel melalui perpustakaan objek (object library) yang terdapat di dalam Excel. Referensi untuk objek-objek tersebut disimpan di Microsoft Excel 9.0 Object Library (EXCEL9.OLB). Dalam struktur pemrograman untuk suatu aplikasi perhitungan, Excel mengekspos objek berupa referensi sel-sel data input untuk diproses oleh VB dan hasilnya kemudian dikembalikan ke dalam worksheet. Jadi dalam penulisan input program, yang dilakukan sebenarnya adalah menyimpan nilai objek dalam variabel, atau nilai variabel = nilai referensi sel dari objek worksheet.

Cara terbaik untuk mengetahui bagaimana VB berkomunikasi dengan objek Excel adalah dengan merekam macro. Tool ini memberi pengenalan yang cepat tentang semua objek Excel sekaligus cara menuliskan kode program dalam VB. Sebagai contoh, kita akan merekam macro untuk menghapus data pada suatu range sel. Anggap saja kita perlu membuat suatu otomatisasi langkah untuk membersihkan data output yang lama, karena program akan mencetak yang baru. Misalkan proses ini akan dikerjakan pada program BOF. Di sini data output ditempatkan pada range sel D27:N39. Langkah-langkah untuk merekam macro adalah sebagai berikut.

1. Klik **Record Macro**  pada toolbar, hingga muncul kotak dialog berikut ini.



2. Ganti nama defaultnya menjadi **HapusOutput**, dan klik OK.
3. Sekarang memulai proses merekam macro. Pilih range sel D27:N50 sebagai data yang akan dibersihkan. Meskipun data output dalam contoh analisis balok hanya pada range sel D27:N39, ada baiknya diambil lebih dari itu untuk soal lain dengan jumlah elemen yang lebih banyak. Jadi N50 diambil sementara sebagai range yang maksimum, atau setara dengan data 23 finite element balok.
4. Tekan **Delete** untuk menghapus range sel D27:N50 kemudian tempatkan pointer pada sel D27.
5. Akhiri proses ini dengan mengklik stop recording .

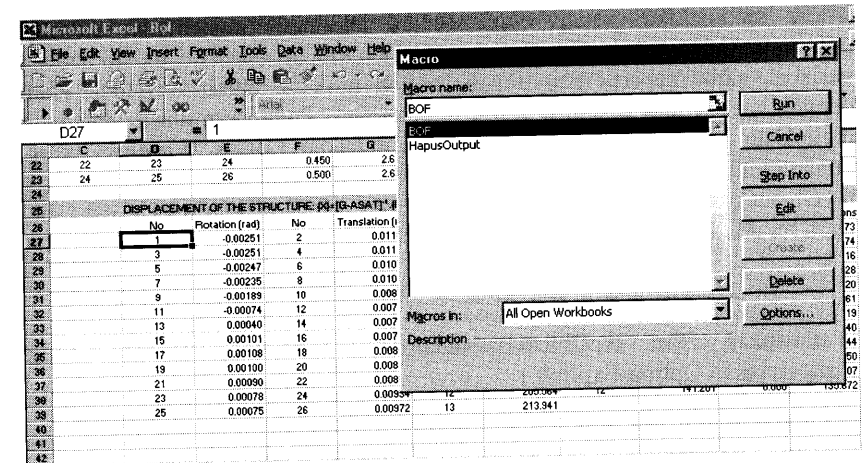
Macro yang dibuat disimpan di dalam workbook, dan untuk melihatnya, Anda harus membuka jendela Visual Basic Editor dengan **Run Macro** ► > **HapusOutput** > **Edit**. Sebenarnya tidaklah sulit untuk memahami bahasa macro yang ditampilkan di sini, karena sudah mencerminkan langkah-langkah yang sudah kita lakukan.

```
Option Explicit
Sub HapusOutput()
    HapusOutput Macro
    Macro recorded 8/16/2003 by Guntar_P.

    Range("D27:N50").Select
    Selection.ClearContents
    Range("D27").Select
End Sub
```

Sekarang lihat listing program BOF. Macro *HapusOutput* di sini ditempatkan pada awal langkah program. Sekarang kembali ke worksheet. Jalankan BOF agar data yang tadi dihapus dicetak kembali. Kemudian tempatkan pointer

pada sembarang sel. Setelah siap, jalankan BOF sekali lagi, dan perhatikan pada layar. Sesaat setelah program dijalankan, hasil output yang ada segera terhapus dan pointer berada pada sel D27, dan secara hampir bersamaan pula hasil output yang baru dari VB dicetak ke dalam worksheet.



Dalam prosedur macro yang dibuat di atas terdapat kata-kata perintah atau statement seperti `Range()`, `Select`, `Selection`, dan `ClearContents`. Statement dapat dibedakan maknanya menurut istilah-istilah yang digunakan dalam VB. Dalam bagian ini, `Select` dan `ClearContents` disebut dengan **metode** yaitu tindakan yang dilakukan saat program dijalankan statement. Sedangkan `Range()` dan `Selection` adalah statement yang merujuk pada **objek**, dalam hal ini adalah objek `Range`. Bagian dari objek adalah **propertinya**, yaitu semua atribut yang didefinisikan untuk objek tersebut.

Kode program di sini dibuat secara berurutan, yaitu nama objek, tanda titik dan diikuti properti atau metodenya. Properti selalu diikuti dengan sama dengan (=) untuk memasukkan nilainya. Misalnya akan mendefinisikan lebar kolom lebih besar dari ukuran standarnya, maka statement-nya adalah, `Columns("C:C").ColumnWidth = 10.0`, artinya properti `ColumnWidth` dari kolom C ditetapkan nilainya menjadi 10.0. Pada contoh lain, misalnya akan mendefinisikan nama suatu sel atau range sel, statement-nya adalah, `Range().Name = "NamaRange"` atau `Cells().Name = "NamaSel"`. Perintahnya sama kalau dikerjakan melalui menu **Insert** > **Name** > **Define** atau **Create**. Hal ini sudah pernah dijelaskan pada bab awal buku ini. Supaya macro dapat digunakan untuk setiap objek yang diseleksi, digunakan kata **Selection**.

Contoh:

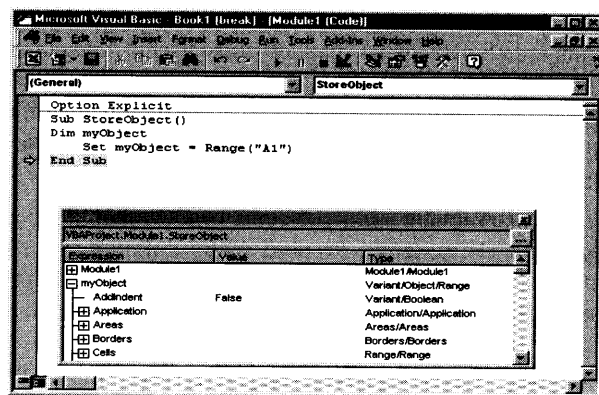
```
...
Selection.Font.ColorIndex = 3
Selection.CreateNames Top:=True, Left:=False, Bottom:=False,
Right:=False
```

Untuk mempelajari setiap objek secara lengkap, Anda harus menampilkannya secara fokus. Caranya dengan menyimpan objek ke dalam sebuah variabel, sama seperti menyimpan sebuah nilai ke dalam variabel, yaitu dengan menggunakan tanda sama dengan (=). Pada permulaan statement digunakan kata kunci **Set**. Setelah itu digunakan **Locals Windows** untuk menampilkan nilai dan tipe variabel tersebut. Misalkan, ambil variabel **myObject** untuk menyimpan objek **Range** sel **A1**. Ketikkan teks "Test Cell" pada sel **A1**. Kemudian buat macro-nya sebagai berikut:

```
Sub StoreObject()
    Set myObject = Range("A1")
End Sub
```


(Tambahkan **Dim myObject** sesudah **Sub StoreObject()** jika Anda menggunakan **Option Explicit**.)


Dari menu **VBA** klik **View > Locals Window** dan jalankan macro dengan menekan **F8**. Tekan tiga kali untuk menetapkan **Range("A1")** ke dalam variabel. **Locals Window** akan tampak seperti berikut.

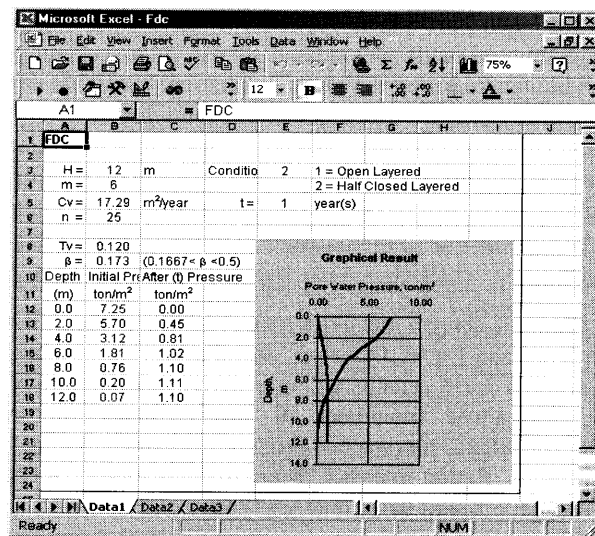


Variabel **myObject** tampak memiliki tanda plus di sebelah kiri dan tipenya adalah **Variant/Object/Range**. Klik tanda plus pada variabel **myObject** tersebut. Sebuah daftar dengan semua properti objek **Range** sel **A1** muncul. Setiap properti menampilkan nilai atau mempunyai tanda plus di dekatnya.

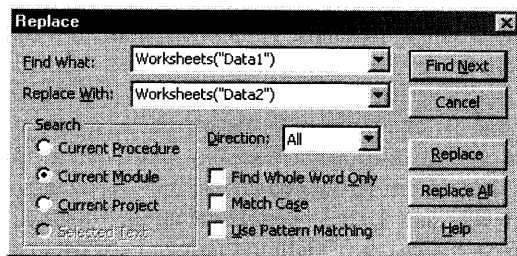
Properti dengan tanda plus adalah referensi untuk sebuah objek, sedangkan yang tidak mempunyai tanda plus adalah sebuah nilai. Sekarang drag scroll bar ke bawah hingga ke properti **Value**. Terlihat bahwa **Value** dari **myObject** adalah string "Test Cell". Anda dapat mengganti properti ini dari **Locals Windows** dengan mengklik satu kali pada teks tersebut. Ganti nilainya misalnya menjadi string "New" dan tekan **Enter**. Sekarang terlihat dalam worksheet nilainya sudah berubah menjadi yang baru. Beberapa properti dari **Range A1** adalah read-only sehingga Anda tidak dapat mengubahnya.

Untuk melihat semua objek yang terdapat di Excel, Anda dapat mengakses dari **Object Browser** dengan mengklik icon  pada toolbar **VBA** atau tekan **F2**. Pilih perpustakaan Excel. Objek di sini digolongkan ke dalam kelasnya (classes di sebelah kiri) sedangkan anggotanya adalah properti, metode, fungsi dan sebagainya (members di sebelah kanan). **Name**, **Cells**, **FormulaArray**, misalnya, merupakan properti dari objek **Range**. Sedangkan dalam hierarkinya, **Range** menjadi properti dari objek **Worksheet**, dan **Worksheet** menjadi properti dari objek **Workbook**.

Membuat program untuk aplikasi perhitungan sebenarnya sama dengan membuat suatu prosedur macro untuk otomasi langkah-langkah pekerjaan di dalam worksheet. Namun di sini **VB** lebih banyak mengerjakan prosedur matematikanya dan hanya sedikit berhubungan dengan objek. Oleh karena itu, nilai objek menjadi sangat strategis. Pada pemrograman mandiri, data input-output ditempatkan dalam file-file sendiri (*.dat, *.txt, *.out, dan sebagainya). Sedangkan di Excel kita menempatkan data input-output di dalam worksheet. Supaya data tersebut dapat disimpan-pakai, worksheet perlu diberi nama seperti layaknya sebuah file data. Misalnya untuk suatu program di mana terdapat 3 sheet data seperti terlihat pada Gambar 7.1, masing-masing dengan nama **Data1**, **Data2**, dan **Data3**. Penulisan untuk referensi selnya adalah **Worksheets("Data1").Cells()** yaitu untuk membaca data pada sheet **Data1**, atau **Worksheets("Data2").Cells()** untuk membaca sheet **Data2**, dan seterusnya. Anda tidak perlu lagi menggunakan **Worksheets()** apabila data input disimpan dalam nama defaultnya, seperti **Sheet1**, **Sheet2**, dan seterusnya. Untuk mempercepat proses penggantian nama ini, Anda dapat melakukan dengan tool **Find**  kemudian klik **Replace** seperti diperlihatkan pada Gambar 7.2. Jika tidak ada nama worksheet yang ditunjuk, program secara otomatis akan memproses referensi sel dari worksheet yang aktif (terbuka).



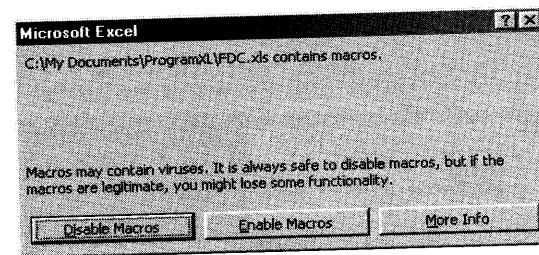
Gambar 7.1 Contoh program dengan tiga sheet data



Gambar 7.2 Tool Find and Replace untuk mengganti nama sheet data

7.2 Petunjuk Pemakaian Program

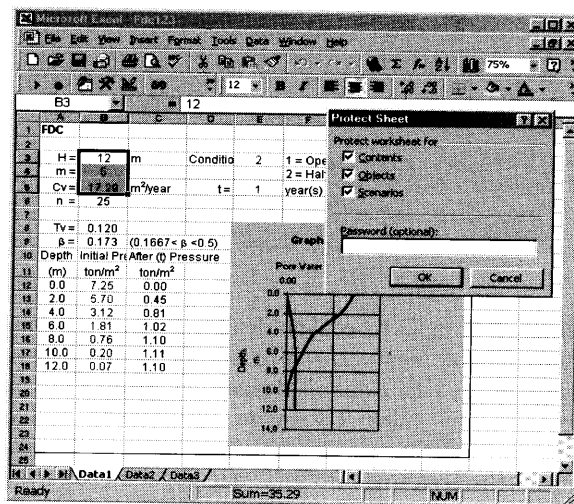
Pada waktu Anda membuka file program, akan keluar kotak pesan seperti di bawah ini. Pastikan Anda mengklik **Enable Macros** agar macro dapat dijalankan. Pesan ini sebenarnya peringatan dari Excel mengenai macro yang berisi virus komputer. Misalnya kalau Anda tidak yakin dengan sumber macro tersebut, Anda dapat mengklik Disable Macro agar macro tidak aktif. Jika perlu, segera keluar dari Excel dan periksa dengan program antivirus yang ada.



Dalam worksheet, data input dibedakan dengan diberi warna biru dari warna defaultnya (hitam) untuk memudahkan proses memasukkan data. Pemberian warna ini juga hampir selalu diikuti dengan melakukan proteksi pada sel, supaya form program tidak terganggu (corrupted) akibat kesalahan pemencetan tombol keyboard atau gangguan lainnya. Dengan demikian, hanya sel dengan teks berwarna biru saja yang dapat diubah isinya. Membuat proteksi sel dalam form berarti membebaskan sel input data dari proteksi worksheet agar dapat menerima perubahan. Sedangkan bagian sel yang lain akan diproteksi dan tidak dapat menerima perubahan, misalnya untuk sel-sel yang menyimpan rumus-rumus atau yang berisi keterangan-keterangan program.

Langkah-langkah untuk memproteksi sel adalah sebagai berikut.

1. Pilih sel atau range sel yang berisi data input (berwarna biru).
2. Klik **Format > Cells** hingga keluar kotak dialog **Format Cells**.
3. Klik tab **Protection** kemudian klik option **Locked** sehingga tanda centangnya hilang.
4. Klik **OK**.
5. Ulangi langkah 1 s.d. 4 untuk data input yang lain.
6. Kemudian proteksi worksheet dengan **Tools > Protection > Protect Worksheet**. Tambahkan password jika memang dibutuhkan dan klik **OK** untuk semua pilihan.
7. Untuk mengembalikan sheet lagi tanpa proteksi, dilakukan dengan **Tools > Protection > Unprotect Sheet**.



Untuk menjalankan program, klik **Run Macro** ► > *program* > **Run** atau menggunakan **Command Button** yang sudah ditempatkan pada form. Waktu yang diperlukan untuk memproses setiap soal bergantung pada data input yang mempengaruhi besarnya ukuran array dalam formula matriks. Semakin besar ukuran matriks, semakin banyak pula jumlah persamaan yang harus diselesaikan, dan prosesnya memerlukan waktu pula.

Peringatan:

Program Excel di sini dibuat hanya untuk membaca data input pada worksheet yang terbuka. Oleh karena itu, pastikan bukan sheet atau workbook lain yang dibuka pada saat program dijalankan, karena dapat menghapus data pada sheet tersebut. Supaya program hanya membaca data pada file-nya sendiri, Anda harus menuliskan referensi sel dari worksheet yang dimaksud. Caranya seperti yang sudah dijelaskan sebelumnya.

7.3 Listing Program

Semua listing program yang dilampirkan di bawah ini merupakan bagian dari masing-masing form data input dan output yang dikerjakan di dalam worksheet. Penulisan program dibuat berdasarkan format susunan data yang ada dalam form, yang pada intinya dibuat seinteraktif mungkin dengan pemakainya, sebagaimana apabila bekerja dengan Excel. Soal-soal yang ada di sini adalah rekaan semata untuk keperluan contoh perhitungan.

Sub FRAME2D()

```
'=====
'Program Frame Balok 2 Dimensi - FRAME2D
'dibuat oleh: Guntar Pangaribuan
'berdasarkan materi dalam buku
'Aplikasi Excel Untuk Rekayasa Teknik Sipil
'Rev.1 /2002
'=====
```

Dim NP, NM, DOF As Integer

```
Dm = Cells(3, 2)
NP = Cells(4, 2)
NM = Cells(5, 2)
Ec = Cells(6, 2)
DOF = Cells(7, 2)
```

```
ReDim A(NM, 1) As Double
ReDim T(6, 6, NM) As Double
ReDim TT(6, 6, NM) As Double
ReDim MK(6, 6, NM) As Double
ReDim GK(NP, NP) As Double
ReDim Index(DOF, 1) As Integer
ReDim Indexs(NP, NM) As Integer
ReDim GKP(NP, NP) As Double
ReDim GKS(6, DOF) As Double
ReDim P(NP, 1) As Double
ReDim P1(6, 6, NM) As Double
ReDim P2(6, 6, NM) As Double
ReDim L(NM, 1) As Double
ReDim EI(NM, 1) As Double
ReDim X(NP, 1) As Double
ReDim Xs(6, 1, NM) As Double
ReDim Xi(6, 1, NM) As Double
ReDim Pi(6, 1, NM) As Double
ReDim Rs(6, 1) As Double
ReDim ALPHA(NM, 1) As Double
ReDim SupM(6, 1) As Integer
```

```
'Baca Input L, EI, A elemen
For II = 1 To NM
L(II, 1) = Cells(10 + II, 8)
Next II
```

```
For II = 1 To NM
EI(II, 1) = Ec * Cells(10 + II, 10)
Next II
```

```
For II = 1 To NM
A(II, 1) = Cells(10 + II, 11)
Next II
```

```
'Baca sudut transformasi elemen
For II = 1 To NM
ALPHA(II, 1) = Cells(10 + II, 12)
Next II
```

```
'Baca Input Beban P
For II = 1 To NP
```

```
P(II, 1) = Cells(18 + II, 2)
Next II
```

```
'Bangun [T]elemen
```

```
RAD = 180 / 3.14159265358979
```

```
For II = 1 To NM
```

```
T(1, 1, II) = Cos(ALPHA(II, 1) / RAD): T(1, 2, II) =
Sin(ALPHA(II, 1) / RAD): T(1, 3, II) = 0: T(1, 4, II) = 0: T(1,
5, II) = 0: T(1, 6, II) = 0
T(2, 1, II) = -Sin(ALPHA(II, 1) / RAD): T(2, 2, II) =
Cos(ALPHA(II, 1) / RAD): T(2, 3, II) = 0: T(2, 4, II) = 0: T(2,
5, II) = 0: T(2, 6, II) = 0
T(3, 1, II) = 0: T(3, 2, II) = 0: T(3, 3, II) = 1: T(3, 4, II) =
0: T(3, 5, II) = 0: T(3, 6, II) = 0
T(4, 1, II) = 0: T(4, 2, II) = 0: T(4, 3, II) = 0: T(4, 4, II) =
Cos(ALPHA(II, 1) / RAD): T(4, 5, II) = Sin(ALPHA(II, 1) / RAD):
T(4, 6, II) = 0
T(5, 1, II) = 0: T(5, 2, II) = 0: T(5, 3, II) = 0: T(5, 4, II) =
-Sin(ALPHA(II, 1) / RAD): T(5, 5, II) = Cos(ALPHA(II, 1) / RAD):
T(5, 6, II) = 0
T(6, 1, II) = 0: T(6, 2, II) = 0: T(6, 3, II) = 0: T(6, 4, II) =
0: T(6, 5, II) = 0: T(6, 6, II) = 1
Next II
```

```
'Bangun [K]elemen
```

```
For II = 1 To NM
```

```
MK(1, 1, II) = A(II, 1) * Ec / L(II, 1): MK(1, 2, II) = 0: MK(1,
3, II) = 0: MK(1, 4, II) = -A(II, 1) * Ec / L(II, 1): MK(1, 5,
II) = 0: MK(1, 6, II) = 0
MK(2, 1, II) = 0: MK(2, 2, II) = 12 * EI(II, 1) / L(II, 1) ^ 3:
MK(2, 3, II) = 6 * EI(II, 1) / L(II, 1) ^ 2: MK(2, 4, II) = 0:
MK(2, 5, II) = -12 * EI(II, 1) / L(II, 1) ^ 3: MK(2, 6, II) = 6
* EI(II, 1) / L(II, 1) ^ 2
MK(3, 1, II) = 0: MK(3, 2, II) = 6 * EI(II, 1) / L(II, 1) ^ 2:
MK(3, 3, II) = 4 * EI(II, 1) / L(II, 1): MK(3, 4, II) = 0: MK(3,
5, II) = -6 * EI(II, 1) / L(II, 1) ^ 2: MK(3, 6, II) = 2 *
EI(II, 1) / L(II, 1)
MK(4, 1, II) = -A(II, 1) * Ec / L(II, 1): MK(4, 2, II) = 0:
MK(4, 3, II) = 0: MK(4, 4, II) = A(II, 1) * Ec / L(II, 1): MK(4,
5, II) = 0: MK(4, 6, II) = 0
MK(5, 1, II) = 0: MK(5, 2, II) = -12 * EI(II, 1) / L(II, 1) ^ 3:
MK(5, 3, II) = -6 * EI(II, 1) / L(II, 1) ^ 2: MK(5, 4, II) = 0:
MK(5, 5, II) = 12 * EI(II, 1) / L(II, 1) ^ 3: MK(5, 6, II) = -6
* EI(II, 1) / L(II, 1) ^ 2
MK(6, 1, II) = 0: MK(6, 2, II) = 6 * EI(II, 1) / L(II, 1) ^ 2:
MK(6, 3, II) = 2 * EI(II, 1) / L(II, 1): MK(6, 4, II) = 0: MK(6,
5, II) = -6 * EI(II, 1) / L(II, 1) ^ 2: MK(6, 6, II) = 4 *
EI(II, 1) / L(II, 1)
Next II
```

```
'Transpose [T]--->[TT]elemen
```

```
For II = 1 To NM
```

```
For IJ = 1 To 6
```

```
For IK = 1 To 6
```

```
TT(IJ, IK, II) = T(IK, IJ, II)
```

```
Next IK
```

```
Next IJ
```

```
Next II
```

```
'Bangun [K]elemen = [TT].[K]elemen.[T]
```

```
For II = 1 To NM
```

```
For IJ = 1 To 6
```

```
For IK = 1 To 6
```

```
P1(IJ, IK, II) = 0
```

```
For IL = 1 To 6
```

```
P1(IJ, IK, II) = P1(IJ, IK, II) + TT(IJ, IL, II) * MK(IL, IK,
II)
```

```
Next IL
```

```
Next IK
```

```
Next IJ
```

```
For IJ = 1 To 6
```

```
For IK = 1 To 6
```

```
P2(IJ, IK, II) = 0
```

```
For IL = 1 To 6
```

```
P2(IJ, IK, II) = P2(IJ, IK, II) + P1(IJ, IL, II) * T(IL, IK, II)
```

```
Next IL
```

```
Next IK
```

```
Next IJ
```

```
Next II
```

```
'Bangun [K]global
```

```
For II = 1 To NP
```

```
For IJ = 1 To NP
```

```
GK(II, IJ) = 0
```

```
Next IJ
```

```
Next II
```

```
'Subskrip [K]elemen pada [K]global
```

```
For II = 1 To NM
```

```
For IJ = 1 To 6
```

```
Indexs(IJ, II) = Cells(10 + II, 1 + IJ)
```

```
Next IJ
```

```
Next II
```

```
For II = 1 To NM
```

```
For IJ = 1 To 6
```

```
For IK = 1 To 6
```

```
GK(Indexs(IJ, II), Indexs(IK, II)) = GK(Indexs(IJ, II),
Indexs(IK, II)) + P2(IJ, IK, II)
```

```
Next IK
```

```
Next IJ
```

```
Next II
```

```
'Partisi [GK]--->[GKP]
```

```
For II = 1 To DOF
```

```
Index(II, 1) = Cells(7, 2 + II)
```

```
Next II
```

```
For II = 1 To DOF
```

```
For IJ = 1 To DOF
```

```
GKP(Index(II, 1), Index(IJ, 1)) = GK(Index(II, 1), Index(IJ, 1))
```

```
GKP(II, IJ) = GKP(Index(II, 1), Index(IJ, 1))
```

```
Next IJ
```

```
Next II
```

```
'Invers [GKP]
```

```
For II = 1 To DOF
```

```
For IJ = 1 To DOF
```

```
If IJ <> II Then GKP(II, IJ) = GKP(II, IJ) / GKP(II, II)
```

```

Next IJ
For IK = 1 To DOF
  If IK = II Then GoTo 10
For IJ = 1 To DOF
  If IJ <> II Then GKP(IK, IJ) = GKP(IK, IJ) - GKP(II, IJ)
* GKP(IK, II)
Next IJ
10 Next IK
For IK = 1 To DOF
  If IK <> II Then GKP(IK, II) = -GKP(IK, II) / GKP(II,
II)
Next IK
GKP(II, II) = 1 / GKP(II, II)
Next II

```

```

'Lendutan struktur {X} = [GKP]^-1.{P}
For II = 1 To DOF
  X(Index(II, 1), 1) = 0
For IJ = 1 To DOF
  X(Index(II, 1), 1) = X(Index(II, 1), 1) + GKP(II, IJ) *
P(Index(IJ, 1), 1)
Next IJ
Next II

```

```

'Cetak {X}
For II = 1 To NP
  Cells(18 + II, 3) = X(II, 1)
Next II

```

```

'Penomoran {X}(NP,1) menjadi {X}s(6,1)
For II = 1 To NM
  For IJ = 1 To 6
  Xs(IJ, 1, II) = X(Indexs(IJ, II), 1)
Next IJ
Next II

```

```

'=====
'Transformasi {X}s ke koordinat elemen
'{X}i=[T].{X}s
'=====

```

```

For II = 1 To NM
  For IJ = 1 To 6
  Xi(IJ, 1, II) = 0
For IK = 1 To 6
  Xi(IJ, 1, II) = Xi(IJ, 1, II) + T(IJ, IK, II) * Xs(IK, 1, II)
Next IK
Next IJ
Next II

```

```

'{P}i=[K]i.{X}i
For II = 1 To NM
  For IJ = 1 To 6
  Pi(IJ, 1, II) = 0
For IK = 1 To 6
  Pi(IJ, 1, II) = Pi(IJ, 1, II) + MK(IJ, IK, II) * Xi(IK, 1, II)
Next IK
Next IJ
Next II

```

```

'Cetak {P}i
For II = 1 To NM
  For IJ = 1 To 6
  Cells(18 + II, 5 + IJ) = Pi(IJ, 1, II)
Next IJ
Next II

```

```

'Mencari reaksi perletakan
'Melakukan partisi pada perletakan

```

```

For II = 1 To 6
  SupM(II, 1) = Cells(8, 1 + II)
Next II

```

```

For II = 1 To DOF
  For IJ = 1 To 6
  GKS(IJ, II) = GK(SupM(IJ, 1), Index(II, 1))
Next IJ
Next II

```

```

'Reaksi di nodal perletakan

```

```

For II = 1 To 6
  For IJ = 1 To DOF
  Rs(II, 1) = Rs(II, 1) + GKS(II, IJ) * X(Index(IJ, 1), 1)
Next IJ
Next II

```

```

For II = 1 To 6
  Cells(26, 4 + II) = Rs(II, 1) - P(SupM(II, 1), 1)
Next II

```

```

End Sub

```

```

Sub TRUSS()

```

```

'=====
'Program Analisa Rangka Batang - TRUSS
'dibuat oleh: Guntar Pangaribuan
'berdasarkan materi dalam buku
'Aplikasi Excel Untuk Rekayasa Teknik Sipil
'Rev.1 /2002
'=====

```

```

Dim NP, NM, DOF As Integer

```

```

Dm = Cells(3, 2)
NP = Cells(4, 2)
NM = Cells(5, 2)
Ec = Cells(6, 2)
DOF = Cells(7, 2)

```

```

ReDim A(NM, 1) As Double
ReDim T(4, 4, NM) As Double
ReDim TT(4, 4, NM) As Double
ReDim MK(4, 4, NM) As Double
ReDim GK(NP, NP) As Double
ReDim Index(DOF, 1) As Integer
ReDim Indexs(NP, NM) As Integer
ReDim GKP(NP, NP) As Double
ReDim GKS(4, DOF) As Double
ReDim P(NP, 1) As Double
ReDim P1(4, 4, NM) As Double
ReDim P2(4, 4, NM) As Double
ReDim L(NM, 1) As Double
ReDim EI(NM, 1) As Double
ReDim X(NP, 1) As Double
ReDim Xs(4, 1, NM) As Double
ReDim Ps(6, 1, NM) As Double
ReDim Xi(6, 1, NM) As Double
ReDim Pe(6, 1, NM) As Double
ReDim Pi(6, 1, NM) As Double
ReDim Rs(4, 1) As Double
ReDim ALPHA(NM, 1) As Double
ReDim SupM(4, 1) As Integer

```

```

'Baca Input L, EI, A elemen
For II = 1 To NM
L(II, 1) = Cells(18 + II, 8)
Next II

```

```

For II = 1 To NM
A(II, 1) = Cells(18 + II, 9)
Next II

```

```

'Baca sudut transformasi elemen
For II = 1 To NM
ALPHA(II, 1) = Cells(18 + II, 11)
Next II

```

```

'Baca Input Beban P
For II = 1 To NP
P(II, 1) = Cells(35 + II, 2)
Next II

```

```

'Bangun [T]elemen
RAD = 180 / 3.14159265358979
For II = 1 To NM
T(1, 1, II) = Cos(ALPHA(II, 1) / RAD): T(1, 2, II) =
Sin(ALPHA(II, 1) / RAD): T(1, 3, II) = 0: T(1, 4, II) = 0
T(2, 1, II) = -Sin(ALPHA(II, 1) / RAD): T(2, 2, II) =
Cos(ALPHA(II, 1) / RAD): T(2, 3, II) = 0: T(2, 4, II) = 0
T(3, 1, II) = 0: T(3, 2, II) = 0: T(3, 3, II) = Cos(ALPHA(II, 1)
/ RAD): T(3, 4, II) = Sin(ALPHA(II, 1) / RAD)
T(4, 1, II) = 0: T(4, 2, II) = 0: T(4, 3, II) = -Sin(ALPHA(II,
1) / RAD): T(4, 4, II) = Cos(ALPHA(II, 1) / RAD)
Next II

```

```

'Bangun [K]elemen
For II = 1 To NM

```

```

MK(1, 1, II) = A(II, 1) * Ec / L(II, 1): MK(1, 2, II) = 0: MK(1,
3, II) = -A(II, 1) * Ec / L(II, 1): MK(1, 4, II) = 0
MK(2, 1, II) = 0: MK(2, 2, II) = 0: MK(2, 3, II) = 0: MK(2, 4,
II) = 0
MK(3, 1, II) = -A(II, 1) * Ec / L(II, 1): MK(3, 2, II) = 0:
MK(3, 3, II) = A(II, 1) * Ec / L(II, 1): MK(3, 4, II) = 0
MK(4, 1, II) = 0: MK(4, 2, II) = 0: MK(4, 3, II) = 0: MK(4, 4,
II) = 0
Next II

```

```

'Transpose [T]--->[TT]elemen
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
TT(IJ, IK, II) = T(IK, IJ, II)
Next IK
Next IJ
Next II

```

```

'Bangun [K]elemen = [TT].[K]elemen.[T]
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
P1(IJ, IK, II) = 0
For IL = 1 To 4
P1(IJ, IK, II) = P1(IJ, IK, II) + TT(IJ, IL, II) * MK(IL, IK,
II)
Next IL
Next IK
Next IJ
For IJ = 1 To 4
For IK = 1 To 4
P2(IJ, IK, II) = 0
For IL = 1 To 4
P2(IJ, IK, II) = P2(IJ, IK, II) + P1(IJ, IL, II) * T(IL, IK, II)
Next IL
Next IK
Next IJ
Next II

```

```

'Bangun [K]global
For II = 1 To NP
For IJ = 1 To NP
GK(II, IJ) = 0
Next IJ
Next II

```

```

'Subskrip [K]elemen pada [K]global
For II = 1 To NM
For IJ = 1 To 4
Indexs(IJ, II) = Cells(18 + II, 1 + IJ)
Next IJ
Next II

```

```

For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
GK(Indexs(IJ, II), Indexs(IK, II)) = GK(Indexs(IJ, II),
Indexs(IK, II)) + P2(IJ, IK, II)
Next IK

```

```

Next IJ
Next II

'Partisi [GK]--->[GKP]
c = -1
For II = 1 To DOF
For IJ = 1 To 2
If II + IJ + c > DOF Then GoTo 5 Else Index(II + IJ + c, 1) =
Cells(6 + II, 2 + IJ)
Next IJ
c = c + 1
Next II

5 For II = 1 To DOF
For IJ = 1 To DOF
GKP(Index(II, 1), Index(IJ, 1)) = GK(Index(II, 1), Index(IJ,
1))
GKP(II, IJ) = GKP(Index(II, 1), Index(IJ, 1))
Next IJ
Next II

'Invers [GKP]
For II = 1 To DOF
For IJ = 1 To DOF
If IJ <> II Then GKP(II, IJ) = GKP(II, IJ) / GKP(II, II)
Next IJ
For IK = 1 To DOF
If IK = II Then GoTo 10
For IJ = 1 To DOF
If IJ <> II Then GKP(IK, IJ) = GKP(IK, IJ) - GKP(II, IJ)
* GKP(IK, II)
Next IJ
10 Next IK
For IK = 1 To DOF
If IK <> II Then GKP(IK, II) = -GKP(IK, II) / GKP(II,
II)
Next IK
GKP(II, II) = 1 / GKP(II, II)
Next II

'Lendutan struktur {X} = [GKP]^-1.{P}
For II = 1 To DOF
X(Index(II, 1), 1) = 0
For IJ = 1 To DOF
X(Index(II, 1), 1) = X(Index(II, 1), 1) + GKP(II, IJ) *
P(Index(IJ, 1), 1)
Next IJ
Next II

'Cetak {X}
For II = 1 To NP
Cells(35 + II, 3) = X(II, 1)
Next II

'Penomoran {X}(NP,1) menjadi {X}s(4,1)
For II = 1 To NM
For IJ = 1 To 4
Xs(IJ, 1, II) = X(Indexs(IJ, II), 1)
Next IJ
Next II

```

```

'=====
'Transformasi {X}s ke koordinat elemen
'{X}i=[T].{X}s
'=====

For II = 1 To NM
For IJ = 1 To 4
Xi(IJ, 1, II) = 0
For IK = 1 To 4
Xi(IJ, 1, II) = Xi(IJ, 1, II) + T(IJ, IK, II) * Xs(IK, 1, II)
Next IK
Next IJ
Next II
'{P}i=[K]i.{X}i
For II = 1 To NM
For IJ = 1 To 4
Pi(IJ, 1, II) = 0
For IK = 1 To 4
Pi(IJ, 1, II) = Pi(IJ, 1, II) + MK(IJ, IK, II) * Xi(IK, 1, II)
Next IK
Next IJ
Next II

'Cetak {P}i
For II = 1 To NM
For IJ = 1 To 4
Cells(35 + II, 5 + IJ) = Pi(IJ, 1, II)
Next IJ
Next II

'Mencari reaksi perletakan
'Melakukan partisi pada perletakan

For II = 1 To 4
SupM(II, 1) = Cells(16, 1 + II)
Next II

For II = 1 To DOF
For IJ = 1 To 4
GKS(IJ, II) = GK(SupM(IJ, 1), Index(II, 1))
Next IJ
Next II

'Reaksi di nodal perletakan

For II = 1 To 4
For IJ = 1 To DOF
Rs(II, 1) = Rs(II, 1) + GKS(II, IJ) * X(Index(IJ, 1), 1)
Next IJ
Next II

For II = 1 To 4
Cells(36, 10 + II) = Rs(II, 1) - P(SupM(II, 1), 1)
Next II

End Sub

```

Sub BOF()

```
'=====
'Program Analisa Balok di Atas Medium Elastis - BOF
'dibuat oleh: Guntar Pangaribuan
'berdasarkan materi dalam buku
'Aplikasi Excel Untuk Rekayasa Teknik Sipil
'Rev.1 /2002
'=====
```

```
Range("D27:N50").Select
Selection.ClearContents
Range("D27").Select
```

```
NP = Cells(3, 2)
NM = Cells(4, 2)
Ec = Cells(6, 2)
```

```
ReDim MA(4, 4, NM) As Double
ReDim TMA(4, 4, NM) As Double
ReDim MS(4, 4, NM) As Double
ReDim MAS(4, 4, NM) As Double
ReDim MSAT(4, 4, NM) As Double
ReDim ASAT(4, 4, NM) As Double
ReDim GASAT(NP, NP) As Double
ReDim Indexs(NP, NM) As Double
ReDim L(NM, 1) As Double
ReDim K(2, 1, NM) As Double
ReDim KS(NP / 2, 1) As Double
ReDim EI(NM, 1) As Double
ReDim P(NP, 1) As Double
ReDim X(NP, 1) As Double
ReDim Xi(4, 1, NM) As Double
ReDim F(4, 1, NM) As Double
```

```
'Baca Input L, EI, K elemen
For II = 1 To NM
L(II, 1) = Cells(11 + II, 6)
Next II
```

```
For II = 1 To NM
K(1, 1, II) = Cells(11 + II, 10)
K(2, 1, II) = Cells(11 + II, 11)
Next II
```

```
For II = 1 To NP / 2 - 1
KS(II, 1) = Cells(11 + II, 9)
Next II
KS(NP / 2, 1) = Cells(11 + (NP / 2 - 1), 9) 'asumsi = KS (NP/2 - 1)
```

```
For II = 1 To NM
EI(II, 1) = Cells(11 + II, 8) * Ec
Next II
```

```
'Baca Input Beban P
For II = 1 To NP
P(II, 1) = Cells(26 + II, 2)
Next II
```

```
'Bangun Matriks [A]elemen
For II = 1 To NM
MA(1, 1, II) = 1: MA(1, 2, II) = 0: MA(1, 3, II) = 0: MA(1, 4, II) = 0
MA(2, 1, II) = 1 / L(II, 1): MA(2, 2, II) = 1 / L(II, 1): MA(2, 3, II) = 1: MA(2, 4, II) = 0
MA(3, 1, II) = 0: MA(3, 2, II) = 1: MA(3, 3, II) = 0: MA(3, 4, II) = 0
MA(4, 1, II) = -1 / L(II, 1): MA(4, 2, II) = -1 / L(II, 1): MA(4, 3, II) = 0: MA(4, 4, II) = 1
Next II
```

```
'Bangun Matriks [S]elemen
For II = 1 To NM
MS(1, 1, II) = 4 * EI(II, 1) / L(II, 1): MS(1, 2, II) = 2 * EI(II, 1) / L(II, 1): MS(1, 3, II) = 0: MS(1, 4, II) = 0
MS(2, 1, II) = 2 * EI(II, 1) / L(II, 1): MS(2, 2, II) = 4 * EI(II, 1) / L(II, 1): MS(2, 3, II) = 0: MS(2, 4, II) = 0
MS(3, 1, II) = 0: MS(3, 2, II) = 0: MS(3, 3, II) = K(1, 1, II): MS(3, 4, II) = 0
MS(4, 1, II) = 0: MS(4, 2, II) = 0: MS(4, 3, II) = 0: MS(4, 4, II) = K(2, 1, II)
Next II
```

```
'Transpose [A]
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
TMA(IJ, IK, II) = MA(IK, IJ, II)
Next IK
Next IJ
Next II
```

```
'Bangun ASAT elemen
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
MAS(IJ, IK, II) = 0
For IL = 1 To 4
MAS(IJ, IK, II) = MAS(IJ, IK, II) + MA(IJ, IL, II) * MS(IL, IK, II)
Next IL
Next IK
Next IJ
For IJ = 1 To 4
For IK = 1 To 4
ASAT(IJ, IK, II) = 0
```

```
For IL = 1 To 4
ASAT(IJ, IK, II) = ASAT(IJ, IK, II) + MAS(IJ, IL, II) * TMA(IL, IK, II)
Next IL
Next IK
Next IJ
Next II
```

```
'Bangun ASAT global
For II = 1 To NP
For IJ = 1 To NP
```

```

GASAT(II, IJ) = 0
Next IJ
Next II

'Subskrip ASAT elemen pada ASAT global
For II = 1 To NM
For IJ = 1 To 4
Indexs(IJ, II) = Cells(11 + II, 1 + IJ)
Next IJ
Next II

For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
GASAT(Indexs(IJ, II), Indexs(IK, II)) = GASAT(Indexs(IJ, II),
Indexs(IK, II)) + ASAT(IJ, IK, II)
Next IK
Next IJ
Next II

'Invers ASAT global
For II = 1 To NP
For IJ = 1 To NP
If IJ <> II Then GASAT(II, IJ) = GASAT(II, IJ) / GASAT(II, II)
Next IJ
For IK = 1 To NP
If IK = II Then GoTo 20
For IJ = 1 To NP
If IJ <> II Then GASAT(IK, IJ) = GASAT(IK, IJ) - GASAT(II, IJ) *
GASAT(IK, II)
Next IJ
20 Next IK
For IK = 1 To NP
If IK <> II Then GASAT(IK, II) = -GASAT(IK, II) / GASAT(II, II)
Next IK
GASAT(II, II) = 1 / GASAT(II, II)
Next II

'Lendutan pondasi {X} = [GASAT]^-1.{P}
For II = 1 To NP
X(II, 1) = 0
For IJ = 1 To NP
X(II, 1) = X(II, 1) + GASAT(II, IJ) * P(IJ, 1)
Next IJ
Next II

'Cetak {X}:
a = 1 ' untuk rotasi akibat momen
b = 2 ' untuk translasi akibat gaya vertikal
c = 1 ' untuk tegangan tanah akibat diberikan translasi
For II = 1 To NP / 2
Cells(26 + II, 4) = a
Cells(26 + II, 5) = X(a, 1)
Cells(26 + II, 6) = b
Cells(26 + II, 7) = X(b, 1)
Cells(26 + II, 8) = c
Cells(26 + II, 9) = KS(II, 1) * X(b, 1)
a = a + 2
b = b + 2
c = c + 1

```

```

Next II

'Penomoran {x}NP x 1 menjadi {xi}4 x 1
n = 1
For II = 1 To NM
For IJ = 1 To 4
Xi(IJ, 1, II) = X(n, 1)
n = n + 1
Next IJ
n = n - 2
Next II

'Gaya dalam elemen {F}i = [S].[A]T.{Xi}
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
MSAT(IJ, IK, II) = 0
For IL = 1 To 4
MSAT(IJ, IK, II) = MSAT(IJ, IK, II) + MS(IJ, IL, II) * TMA(IL,
IK, II)
Next IL
Next IK
Next IJ
For IJ = 1 To 4
F(IJ, 1, II) = 0
For IL = 1 To 4
F(IJ, 1, II) = F(IJ, 1, II) + MSAT(IJ, IL, II) * Xi(IL, 1, II)
Next IL
Next IJ
Next II

'Cetak {F}i
a = 1
For II = 1 To NM
Cells(26 + II, 10) = a
For IJ = 1 To 4
If IJ >= 3 Then Cells(26 + II, 10 + IJ) = F(IJ, 1, II): GoTo 40
Cells(26 + II, 10 + IJ) = F(IJ, 1, II)
40 Next IJ
a = a + 1
Next II

End Sub

```

Sub WALL()

```
'=====
'Program Analisa Dinding Penahan Tanah - WALL
'dibuat oleh: Guntar Pangaribuan
'berdasarkan materi dalam buku
'Aplikasi Excel Untuk Rekayasa Teknik Sipil
'Rev.1 /2002
'=====
```

```
Range("D27:O50").Select
Selection.ClearContents
Range("D27").Select
```

```
NM = Cells(4, 2)
NP = Cells(3, 2)
Ec = Cells(6, 2)
```

```
ReDim MA(4, 4, NM) As Double
ReDim TMA(4, 4, NM) As Double
ReDim MS(4, 4, NM) As Double
ReDim MAS(4, 4, NM) As Double
ReDim MSAT(4, 4, NM) As Double
ReDim ASAT(4, 4, NM) As Double
ReDim GASAT(NP, NP) As Double
ReDim L(NM, 1) As Double
ReDim K(2, 1, NM) As Double
ReDim KS(NP / 2, 1) As Double
ReDim EI(NM, 1) As Double
ReDim P(NP, 1) As Double
ReDim X(NP, 1) As Double
ReDim Xi(4, 1, NM) As Double
ReDim F(4, 1, NM) As Double
ReDim Indexs(4, NM) As Double
```

```
'Baca Input L, EI, K elemen
For II = 1 To NM
L(II, 1) = Cells(11 + II, 6)
Next II
```

```
For II = 1 To NM
K(1, 1, II) = Cells(11 + II, 11)
K(2, 1, II) = Cells(11 + II, 12)
Next II
```

```
For II = 1 To ((NP / 2) - 1)
KS(II, 1) = Cells(11 + II, 10)
Next II
KS(NP / 2, 1) = Cells(11 + (NP / 2 - 1), 10) 'asumsi = KS (NP/2 - 1)
```

```
For II = 1 To NM
EI(II, 1) = Cells(11 + II, 8) * Ec
Next II
```

```
'Baca Input Beban P
For II = 1 To NP
P(II, 1) = Cells(26 + II, 2)
```

Next II

```
'Bangun Matriks [A]elemen
For II = 1 To NM
MA(1, 1, II) = 1: MA(1, 2, II) = 0: MA(1, 3, II) = 0: MA(1, 4, II) = 0
MA(2, 1, II) = 1 / L(II, 1): MA(2, 2, II) = 1 / L(II, 1): MA(2, 3, II) = 1: MA(2, 4, II) = 0
MA(3, 1, II) = 0: MA(3, 2, II) = 1: MA(3, 3, II) = 0: MA(3, 4, II) = 0
MA(4, 1, II) = -1 / L(II, 1): MA(4, 2, II) = -1 / L(II, 1): MA(4, 3, II) = 0: MA(4, 4, II) = 1
Next II
```

```
'Bangun Matriks [S]elemen
For II = 1 To NM
MS(1, 1, II) = 4 * EI(II, 1) / L(II, 1): MS(1, 2, II) = 2 * EI(II, 1) / L(II, 1): MS(1, 3, II) = 0: MS(1, 4, II) = 0
MS(2, 1, II) = 2 * EI(II, 1) / L(II, 1): MS(2, 2, II) = 4 * EI(II, 1) / L(II, 1): MS(2, 3, II) = 0: MS(2, 4, II) = 0
MS(3, 1, II) = 0: MS(3, 2, II) = 0: MS(3, 3, II) = K(1, 1, II): MS(3, 4, II) = 0
MS(4, 1, II) = 0: MS(4, 2, II) = 0: MS(4, 3, II) = 0: MS(4, 4, II) = K(2, 1, II)
Next II
```

```
'Transpose [A]elemen
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
TMA(IJ, IK, II) = MA(IK, IJ, II)
Next IK
Next IJ
Next II
```

```
'Bangun ASAT elemen
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
MAS(IJ, IK, II) = 0
For IL = 1 To 4
MAS(IJ, IK, II) = MAS(IJ, IK, II) + MA(IJ, IL, II) * MS(IL, IK, II)
Next IL
Next IK
Next IJ
For IJ = 1 To 4
For IK = 1 To 4
ASAT(IJ, IK, II) = 0
For IL = 1 To 4
ASAT(IJ, IK, II) = ASAT(IJ, IK, II) + MAS(IJ, IL, II) * TMA(IL, IK, II)
Next IL
Next IK
Next IJ
Next II
```

```
'Bangun ASAT global
For II = 1 To NP
For IJ = 1 To NP
```



```

GASAT(II, IJ) = 0
Next IJ
Next II

'Subskrip [K]elemen pada [K]global
For II = 1 To NM
For IJ = 1 To 4
Indexs(IJ, II) = Cells(11 + II, 1 + IJ)
Next IJ
Next II

For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
GASAT(Indexs(IJ, II), Indexs(IK, II)) = GASAT(Indexs(IJ, II),
Indexs(IK, II)) + ASAT(IJ, IK, II)
Next IK
Next IJ
Next II

'Invers ASAT global
For II = 1 To NP
For IJ = 1 To NP
If IJ <> II Then GASAT(II, IJ) = GASAT(II, IJ) / GASAT(II, II)
Next IJ
For IK = 1 To NP
If IK = II Then GoTo 20
For IJ = 1 To NP
If IJ <> II Then GASAT(IK, IJ) = GASAT(IK, IJ) - GASAT(II, IJ) *
GASAT(IK, II)
Next IJ
20 Next IK
For IK = 1 To NP
If IK <> II Then GASAT(IK, II) = -GASAT(IK, II) / GASAT(II, II)
Next IK
GASAT(II, II) = 1 / GASAT(II, II)
Next II

'Lendutan dinding {x} = [GASAT]^-1.{P}
For II = 1 To NP
X(II, 1) = 0
For IJ = 1 To NP
X(II, 1) = X(II, 1) + GASAT(II, IJ) * P(IJ, 1)
Next IJ
Next II

'Cetak {x}:
a = 1 ' untuk rotasi akibat momen
b = 2 ' untuk translasi akibat gaya vertikal
c = 1 ' untuk tegangan tanah akibat diberikan translasi
For II = 1 To NP / 2
Cells(26 + II, 4) = a
Cells(26 + II, 5) = X(a, 1)
Cells(26 + II, 6) = b
Cells(26 + II, 7) = X(b, 1)
Cells(26 + II, 8) = c
Cells(26 + II, 9) = KS(II, 1) * X(b, 1)
a = a + 2
b = b + 2

```

```

c = c + 1
Next II

'Penomoran {x}NP x 1 menjadi {xi}4 x 1
n = 1
For II = 1 To NM
For IJ = 1 To 4
Xi(IJ, 1, II) = X(n, 1)
n = n + 1
Next IJ
n = n - 2
Next II

'Gaya dalam elemen {F}i = [S].[A]T.{xi}
For II = 1 To NM
For IJ = 1 To 4
For IK = 1 To 4
MSAT(IJ, IK, II) = 0
For IL = 1 To 4
MSAT(IJ, IK, II) = MSAT(IJ, IK, II) + MS(IJ, IL, II) * TMA(IL,
IK, II)
Next IL
Next IK
Next IJ
For IJ = 1 To 4
F(IJ, 1, II) = 0
For IL = 1 To 4
F(IJ, 1, II) = F(IJ, 1, II) + MSAT(IJ, IL, II) * Xi(IL, 1, II)
Next IL
Next IJ
Next II

'Cetak {F}i
a = 1
For II = 1 To NM
Cells(26 + II, 10) = a
For IJ = 1 To 4
If IJ >= 3 Then Cells(26 + II, 11 + IJ) = F(IJ, 1, II): GoTo 40
Cells(26 + II, 10 + IJ) = F(IJ, 1, II)
40 Next IJ
a = a + 1
Next II

End Sub

```

Sub MATF()

```
'=====
'Program Analisa Pondasi Pelat - MATF
'dibuat oleh: Guntar Pangaribuan
'berdasarkan materi dalam buku
'Aplikasi Excel Untuk Rekayasa Teknik Sipil
'Rev.1 /2002
'=====
```

```
Range("U14:AI73").Select
Selection.ClearContents
Range("U14").Select
```

```
GX = Cells(3, 2)
GY = Cells(3, 3)
NP = Cells(4, 2)
NM = Cells(5, 2)
NJ = Cells(6, 2)
Ec = Cells(9, 2)
G = Cells(5, 7)
w = Cells(3, 7)
Sc = Cells(10, 2)
```

```
ReDim MA(6, 5, NM) As Double
ReDim TMA(5, 6, NM) As Double
ReDim MS(5, 5, NM) As Double
ReDim MAS(6, 5, NM) As Double
ReDim MSAT(5, 6, NM) As Double
ReDim ASAT(6, 6, NM) As Double
ReDim GASAT(NP, NP) As Double
ReDim L(NM, 1) As Double
ReDim LX(GX + 1, 1, GY + 1) As Double
ReDim LY(GY + 1, 1, GX + 1) As Double
ReDim K(2, 1, NM) As Double
ReDim Ks(NJ, 1) As Double
ReDim EI(NM, 1) As Double
ReDim J(NM, 1) As Double
ReDim P(NP, 1) As Double
ReDim X(NP, 1) As Double
ReDim Xi(6, 1, NM) As Double
ReDim Xr(NJ, 1) As Double
ReDim F(5, 1, NM) As Double
ReDim ALPHA(NM, 1) As Double
ReDim Indexs(6, NM) As Integer
ReDim Mat(6, 6) As Double
```

```
'Baca Input L, EI, K elemen
For II = 1 To NM
L(II, 1) = Cells(13 + II, 8)
Next II
```

```
For II = 1 To NM
K(1, 1, II) = Cells(13 + II, 14)
K(2, 1, II) = Cells(13 + II, 15)
Next II
```

```
For II = 1 To NJ
Ks(II, 1) = Cells(13 + II, 13)
```

Next II

```
For II = 1 To NM
EI(II, 1) = Cells(13 + II, 10) * Ec
J(II, 1) = Cells(13 + II, 11)
Next II
```

```
'Baca sudut transformasi elemen
For II = 1 To NM
ALPHA(II, 1) = Cells(13 + II, 12)
Next II
```

```
'Baca Input Beban P
a = 0
For II = 1 To NJ
For IJ = 1 To 3
P(II + a, 1) = Cells(13 + II, 16 + IJ)
If IJ >= 3 Then GoTo 5 Else a = a + 1
Next IJ
5 Next II
```

```
'Bangun [A]elemen
rad = 180 / 3.14159265358979
For II = 1 To NM
MA(1, 1, II) = -Sin(ALPHA(II, 1) / rad): MA(1, 2, II) = 0: MA(1, 3, II) = -Cos(ALPHA(II, 1) / rad): MA(1, 4, II) = 0: MA(1, 5, II) = 0
MA(2, 1, II) = Cos(ALPHA(II, 1) / rad): MA(2, 2, II) = 0: MA(2, 3, II) = -Sin(ALPHA(II, 1) / rad): MA(2, 4, II) = 0: MA(2, 5, II) = 0
MA(3, 1, II) = 1 / L(II, 1): MA(3, 2, II) = 1 / L(II, 1): MA(3, 3, II) = 0: MA(3, 4, II) = -1: MA(3, 5, II) = 0
MA(4, 1, II) = 0: MA(4, 2, II) = -Sin(ALPHA(II, 1) / rad): MA(4, 3, II) = Cos(ALPHA(II, 1) / rad): MA(4, 4, II) = 0: MA(4, 5, II) = 0
MA(5, 1, II) = 0: MA(5, 2, II) = Cos(ALPHA(II, 1) / rad): MA(5, 3, II) = Sin(ALPHA(II, 1) / rad): MA(5, 4, II) = 0: MA(5, 5, II) = 0
MA(6, 1, II) = -1 / L(II, 1): MA(6, 2, II) = -1 / L(II, 1): MA(6, 3, II) = 0: MA(6, 4, II) = 0: MA(6, 5, II) = -1
Next II
```

```
'Bangun [S]elemen
For II = 1 To NM
MS(1, 1, II) = 4 * EI(II, 1) / L(II, 1): MS(1, 2, II) = 2 * EI(II, 1) / L(II, 1): MS(1, 3, II) = 0: MS(1, 4, II) = 0: MS(1, 5, II) = 0
MS(2, 1, II) = 2 * EI(II, 1) / L(II, 1): MS(2, 2, II) = 4 * EI(II, 1) / L(II, 1): MS(2, 3, II) = 0: MS(2, 4, II) = 0: MS(2, 5, II) = 0
MS(3, 1, II) = 0: MS(3, 2, II) = 0: MS(3, 3, II) = w * G * J(II, 1) / L(II, 1): MS(3, 4, II) = 0: MS(3, 5, II) = 0
MS(4, 1, II) = 0: MS(4, 2, II) = 0: MS(4, 3, II) = 0: MS(4, 4, II) = K(1, 1, II): MS(4, 5, II) = 0
MS(5, 1, II) = 0: MS(5, 2, II) = 0: MS(5, 3, II) = 0: MS(5, 4, II) = 0: MS(5, 5, II) = K(2, 1, II)
```

```

Next II

'Transpose [A]
For II = 1 To NM
For IJ = 1 To 5
For IK = 1 To 6
TMA(IJ, IK, II) = MA(IK, IJ, II)
Next IK
Next IJ
Next II

'Bangun ASAT elemen
For II = 1 To NM
For IJ = 1 To 6
For IK = 1 To 5
MAS(IJ, IK, II) = 0
For IL = 1 To 5
MAS(IJ, IK, II) = MAS(IJ, IK, II) + MA(IJ, IL, II) * MS(IL, IK, II)
Next IL
Next IK
Next IJ
For IJ = 1 To 6
For IK = 1 To 6
ASAT(IJ, IK, II) = 0
For IL = 1 To 5
ASAT(IJ, IK, II) = ASAT(IJ, IK, II) + MAS(IJ, IL, II) * TMA(IL, IK, II)
Next IL
Next IK
Next IJ
Next II

'Bangun ASAT global
For II = 1 To NP
For IJ = 1 To NP
GASAT(II, IJ) = 0
Next IJ
Next II
For II = 1 To NM
For IJ = 1 To 6
Indexs(IJ, II) = Cells(13 + II, 1 + IJ)
Next IJ
Next II

For II = 1 To NM
For IJ = 1 To 6
For IK = 1 To 6
GASAT(Indexs(IJ, II), Indexs(IK, II)) = GASAT(Indexs(IJ, II), Indexs(IK, II)) + ASAT(IJ, IK, II)
Next IK
Next IJ
Next II

'Invers ASAT global
For II = 1 To NP

```

```

For IJ = 1 To NP
If IJ <> II Then GASAT(II, IJ) = GASAT(II, IJ) / GASAT(II, II)
Next IJ
For IK = 1 To NP
If IK = II Then GoTo 20
For IJ = 1 To NP
If IJ <> II Then GASAT(IK, IJ) = GASAT(IK, IJ) - GASAT(II, IJ) * GASAT(IK, II)
Next IJ
20 Next IK
For IK = 1 To NP
If IK <> II Then GASAT(IK, II) = -GASAT(IK, II) / GASAT(II, II)
Next IK
GASAT(II, II) = 1 / GASAT(II, II)
Next II

'{X} = [GASAT]^-1.{P}
For II = 1 To NP
X(II, 1) = 0
For IJ = 1 To NP
X(II, 1) = X(II, 1) + GASAT(II, IJ) * P(IJ, 1)
Next IJ
Next II

'Cetak {X}:
a = 1 ' untuk rotasi akibat torsi
b = 2 ' untuk rotasi akibat momen
c = 3 ' untuk translasi
d = 1 'soil pressure = KS*X(3)
For II = 1 To NJ
Cells(13 + II, 21) = a
Cells(13 + II, 22) = X(a, 1)
Cells(13 + II, 23) = b
Cells(13 + II, 24) = X(b, 1)
Cells(13 + II, 25) = c
Cells(13 + II, 26) = X(c, 1)
Cells(13 + II, 27) = d
Cells(13 + II, 28) = KS(II, 1) * X(c, 1)
a = a + 3
b = b + 3
c = c + 3
d = d + 1
Next II

'=====
'Membuat AutoCAD ScriptFile: MatFl.scr
'untuk plot Lendutan
'=====

'Membuat urutan lendutan:
a = 3
For II = 1 To GY + 1
For IJ = 1 To GX + 1
Mat(II, IJ) = X(a, 1)
a = a + 3
Next IJ
Next II

a = 1
For II = 1 To GX + 1
For IJ = GY + 1 To 1 Step -1

```

```

Xr(a, 1) = Mat(IJ, II) * Sc
a = a + 1
Next IJ
Next II

'Membuat koordinat grid:
a = 0
For II = 1 To GY + 1
For IJ = 1 To GX + 1
LX(IJ, 1, II) = a
Next IJ
a = a + L(II, 1)
Next II

For II = 1 To GY + 1
a = 0
b = (GX + 1) + (GY - 1) * (GX + GX + 1)
For IJ = 1 To GX + 1
LY(IJ, 1, II) = a
If IJ = GX + 1 Then GoTo 30 Else a = a + L(b, 1)
b = b - (GX + GX + 1)
Next IJ
30 Next II

Open "c:\MatF1.scr" For Output As #1
Print #1, "_3dmesh"
Write #1, GY + 1
Write #1, GX + 1
d = 0
c = 1
For II = 1 To GX + 1
a = 0
For IJ = 1 To GY + 1
Print #1, LX(IJ, 1, II) & ", " & LY(IJ, 1, II) & ", " & -Xr(c, 1)
c = c + 1
a = a + 1
Next IJ
d = d + 1
Next II
Close #1

'=====
'=====

'Penomoran {X} NP x NP menjadi {X}i 6 x 1
For II = 1 To NM
For IJ = 1 To 6
Xi(IJ, 1, II) = X(Cells(13 + II, 1 + IJ), 1)
Next IJ
Next II

'{F}i = {S}.[A]T.{X}i
For II = 1 To NM
For IJ = 1 To 5
For IK = 1 To 6
MSAT(IJ, IK, II) = 0
For IL = 1 To 5
MSAT(IJ, IK, II) = MSAT(IJ, IK, II) + MS(IJ, IL, II) * TMA(IL, IK, II)

```

```

Next IL
Next IK
Next IJ
For IJ = 1 To 5
F(IJ, 1, II) = 0
For IL = 1 To 6
F(IJ, 1, II) = F(IJ, 1, II) + MSAT(IJ, IL, II) * Xi(IL, 1, II)
Next IL
Next IJ
Next II

'Cetak {F}i
a = 1
For II = 1 To NM
Cells(13 + II, 30) = a
For IJ = 1 To 5
If IJ >= 3 Then Cells(13 + II, 30 + IJ) = F(IJ, 1, II): GoTo 40
Cells(13 + II, 30 + IJ) = F(IJ, 1, II)
40 Next IJ
a = a + 1
Next II

End Sub

```

Sub FDC()

```

'=====
'Program Metode Finite Difference untuk Konsolidasi 1 Dimensi
'- FDC
'dibuat oleh: Guntar Pangaribuan
'berdasarkan materi dalam buku
'Aplikasi Excel Untuk Rekayasa Teknik Sipil
'Rev.1 /2002
'=====

Range("C12:C24").Select
Selection.ClearContents
Range("C12").Select

h = Cells(3, 2) 'tebal total lapisan
t = Cells(5, 5) 'total waktu
m = Cells(4, 2) 'jumlah lapisan
n = Cells(6, 2) 'jumlah interval waktu
Beta = Cells(9, 2)

ReDim u(0 To m + 1, 0 To n + 1) As Double
Dim i, j As Integer

u(0, 0) = 0
If Cells(3, 5) = 1 Then GoTo 15 Else GoTo 5

```

```

5  For j = 0 To n 'half closed layered
   u(0, j + 1) = 0
   Next j

   For i = 1 To m
   u(i, 0) = Cells(12 + i, 2)
   Next i

   For j = 0 To n
   For i = 1 To m
   If i = m Then u(i, j + 1) = u(i, j) + Beta * (2 * u(i - 1,
j) - 2 * u(i, j)): GoTo 10
   u(i, j + 1) = u(i, j) + Beta * (u(i - 1, j) +
u(i + 1, j) - 2 * u(i, j))
   Next i
10 Next j: GoTo 20

15 For j = 0 To n 'open layered
   u(0, j + 1) = 0
   u(m, j + 1) = 0
   Next j

   For i = 1 To m - 1
   u(i, 0) = Cells(12 + i, 2)
   Next i

   For j = 0 To n
   For i = 1 To m - 1
   u(i, j + 1) = u(i, j) + Beta * (u(i - 1, j) + u(i + 1, j) -
2 * u(i, j))
   Next i
   Next j

   'Cetak Hasil
20 For i = 0 To m
   Cells(12 + i, 3) = u(i, n + 1)
   Next i

   End Sub

```

DAFTAR PUSTAKA

- Armenakas, Anthony E, *Modern Structural Analysis, The Matrix Method Approach*, Mc.Graw-Hill, 1991.
- Bardet, Jean-Pierre, *Experimental Soil Mechanics*, Prentice-Hall Inc, 1997.
- Bowles, J.E, *Foundation Analysis and Design*, Third Edition, International Student Edition, 1982.
- Bowles, J. E, *Sifat-Sifat Fisis dan Geoteknis Tanah*, Edisi Kedua, Penerbit Erlangga, Jakarta, 1989.
- Craig, John Clark, *The Microsoft Visual Basic for MS-DOS Workshop*, Microsoft Press, Washington, 1993.
- Craig, R.F, *Mekanika Tanah*, Edisi Keempat, Penerbit Erlangga, Jakarta, 1987.
- Hutahaean, E. Drs., *Kalkulus Diferensial dan Integral*, Penerbit PT Gramedia, Jakarta, 1983.
- Jacobson, Reed, *Microsoft Excel 97 Visual Basic*, Penerbit PT Elex Media Komputindo, Jakarta, 1998.
- Pandia, Henry, *Visual Basic 6 Tingkat Lanjut*, Penerbit ANDI, Yogyakarta, 2002.
- Supartono, F.X Ir. dan Ir. Teddy Boen, *Analisa Struktur Dengan Metode Matrix*, Penerbit Universitas Indonesia, 1984.
- Wade, Ellis Jr and Ed Lodi, *Structured Programming Using Turbo Basic*, Academic Press Inc., 1988.