

BAB II

TINJAUAN PUSTAKA

2.1 Analisa Struktur

2.1.1 Aljabar Matriks

Pengertian Matriks adalah rangkaian bilangan yang disusun dalam baris dan kolom. Bila susunan itu terdiri dari m baris dan n kolom, disebut matriks $m \times n$. Bila m dan n sama besar, disebut matriks bujur sangkar ("Square Matrix") secara umum matriks $m \times n$ dapat ditulis sebagai berikut :

$$[A] = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \quad (2.1)$$

Suatu unsur matriks dengan simbol a_{ij} berarti unsur tersebut berada pada baris i dan kolom j .

2.1.2 Type Matriks

1. Matriks Baris

Suatu matriks yang hanya terdiri atas satu baris disebut matriks baris ("Row Matrix"). Cara penulisan biasanya digunakan sepasang kurung kait, sebagai contoh :

$$[A] = [a_1 \quad a_2 \quad a_3] \quad (2.2)$$

2. Matriks Kolom

Matriks yang hanya terdiri dari satu kolom disebut matriks kolom ("Column Matrix"). Cara penulisan pada umumnya sebagai berikut :

$$[A] = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (2.3)$$

3. Matriks Bujur Sangkar

Matriks Bujur sangkar adalah matriks yang besar baris dan kolomnya sama :

Contoh :

$$[A] = \begin{bmatrix} 9 & 8 & 12 \\ 15 & 4 & 7 \\ 5 & 10 & 1 \end{bmatrix} \quad (2.4)$$

4. Matriks Simetri

Matriks Simetri adalah matriks bujur sangkar yang unsur-unsur secara diagonalnya memiliki nilai yang sama, contoh :

$$[A] = \begin{bmatrix} 9 & 15 & 12 \\ 15 & 4 & 7 \\ 12 & 7 & 1 \end{bmatrix} \quad (2.5)$$

5. Matriks Diagonal

Matriks Diagonal adalah matriks yang unsur-unsur selain diagonalnya bernilai nol, contoh :

nol, contoh :

$$[A] = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad (2.6)$$

6. Matriks Satuan

Matriks Satuan adalah matriks yang semua unsur diagonalnya bernilai satu.

Contoh :

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

7. "Band Matrix"

"Band Matrix" adalah matriks dengan unsur-unsur didekat diagonalnya tidak

sama dengan nol. Contoh :

$$[A] = \begin{bmatrix} 2 & 3 & 5 & 0 & 0 & 0 \\ 3 & 1 & 3 & 3 & 0 & 0 \\ 0 & 2 & 6 & 5 & 7 & 0 \\ 0 & 0 & 8 & 7 & 6 & 1 \\ 0 & 0 & 0 & 9 & 6 & 8 \end{bmatrix} \quad (2.8)$$

2.1.3 Operasi Aljabar Matriks

1. Penambahan dan Pengurangan Matriks

Penambahan dan pengurangan matriks hanya dapat dilakukan jika ukuran matriks tersebut sama besar. Penjumlahan/pengurangan matriks dilakukan dengan menambah/mengurangi unsur-unsur matriks yang sesuai.

Contoh :

$$\begin{bmatrix} 9 & 8 \\ 15 & 4 \\ 5 & 10 \end{bmatrix} + \begin{bmatrix} 2 & 5 \\ 1 & 3 \\ 5 & 4 \end{bmatrix} = \begin{bmatrix} 11 & 13 \\ 16 & 7 \\ 10 & 14 \end{bmatrix}$$

Sifat penjumlahan/pengurangan adalah kedua matriks yang dijumlahkan/dikurangkan dapat ditukar letaknya.

$$\text{Contoh : } [A] + [B] = [B] + [A] \quad (2.9)$$

2. Perkalian Matriks dengan Sebuah Bilangan

Perkalian matriks dengan sebuah bilangan (misal x) adalah sama dengan perkalian atas unsur-unsur matriks tersebut dengan bilangan itu. Dan letak keduanya bisa ditukar. Contoh :

$$x [A] = [A] x \quad (2.10)$$

$$\text{Jika } [A] = \begin{bmatrix} 9 & 8 \\ 15 & 4 \\ 5 & 10 \end{bmatrix}$$

$$\text{maka } x [A] = \begin{bmatrix} x.9 & x.8 \\ x.15 & x.4 \\ x.5 & x.10 \end{bmatrix} \quad \text{Bila } x = 2, \text{ maka } x [A] = \begin{bmatrix} 18 & 16 \\ 30 & 8 \\ 10 & 20 \end{bmatrix}$$

3. Perkalian Matriks dengan Matriks

Suatu matriks dapat dikalikan dengan matriks yang lain dengan sifat dan syarat perkalian sebagai berikut :

- a. Perkalian matriks A dengan matriks B tidak sama dengan perkalian matriks B dengan matriks A.

$$[A] \cdot [B] \neq [B] \cdot [A] \quad (2.11)$$

- b. Dua matriks A dan matriks B hanya dapat dikalikan dengan cara $[A] [B]$ apabila jumlah kolom matriks $[A]$ sama dengan jumlah baris matriks $[B]$. Adapun matriks hasil perkaliannya mempunyai jumlah baris matriks $[A]$, dan jumlah kolom sama dengan jumlah kolom matriks B. Contoh :

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} ; \quad [B] = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix}$$

$$[A] \times [B] = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix}$$

$$[B] \times [A] = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

$$= \begin{bmatrix} b_{11}a_{11} + b_{12}a_{21} & b_{11}a_{12} + b_{12}a_{22} & b_{11}a_{13} + b_{12}a_{23} \\ b_{21}a_{11} + b_{22}a_{21} & b_{21}a_{12} + b_{22}a_{22} & b_{21}a_{13} + b_{22}a_{23} \\ b_{31}a_{11} + b_{32}a_{21} & b_{31}a_{12} + b_{32}a_{22} & b_{31}a_{13} + b_{32}a_{23} \end{bmatrix}$$

$$\begin{matrix} [A] & \times & [B] = & [C] & \text{p harus sama dengan q} \\ (m \times p) & & (q \times n) & (m \times n) & \end{matrix} \quad (2.12)$$

$$\begin{matrix} [A] & \times & [B] \times & [C] & = & [D] \\ (m \times p) & & (q \times r) & (s \times n) & & (m \times n) \end{matrix} \quad (2.13)$$

Syarat : $p = q$ dan $r = s$

Pada perkalian 3 buah matriks seperti diatas dapat dilakukan dengan mengalikan [A] dan [B] terlebih dahulu, kemudian hasilnya dikalikan dengan [C]. Atau [B] dikalikan terlebih dahulu dengan [C], kemudian matriks [A] dikalikan dengan matriks hasil perkalian [B] dan [C].

$$\text{Contoh : } ([A] [B]) [C] = [A] ([B] [C]) \quad (2.14)$$

4. Transpose Matriks

Misal terdapat dua buah matriks [C] dan [D] seperti dibawah ini :

$$[C] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ dan } [D] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Matriks [D] dapat diperoleh dengan cara menukar baris dan kolom matriks [C].

Dalam hal ini matriks [D] dikatakan transpose matriks [C] dan dituliskan sebagai :

$$[D] = [C]^T \quad (2.15)$$

Beberapa sifat transpose matriks :

- a. Bila suatu matriks ditranspose dua kali, maka akan diperoleh matriks semula, yaitu

$$([A]^T)^T = [A] \quad (2.16)$$

- b. Bila transpose dari dua matriks dijumlahkan hasilnya sama dengan transpose hasil penjumlahan kedua matriksnya.

$$[A]^T + [B]^T = ([A] + [B])^T \quad (2.17)$$

- c. Transpose dari suatu perkalian matriks sama dengan perkalian dari transpose masing- masing matriks dengan urutan terbalik.

$$([A] [B])^T = [B]^T [A]^T \quad (2.18)$$

d. Transpose dari suatu matriks simetri sama dengan matriks itu sendiri. Jika matriks itu simetri, maka

$$[A] = [A]^T \quad (2.19)$$

Bila $[A]$ matriks *anti simetri* (yaitu suatu matriks dengan unsur-unsur diagonalnya nol dan unsur-unsur terhadap diagonal sama besar tetapi berlainan tanda), maka

$$[A] = -[A]^T \quad (2.20)$$

e. Hasil kali suatu matriks terhadap matriks transposenya selalu berupa matriks simetri.

$$[A] [A]^T = [C] \quad (2.21)$$

$[C]$ matriks simetri

f. Suatu matriks bujur sangkar selalu dapat diuraikan menjadi penjumlahan dari matriks simetri dan matriks anti simetri. Contoh :

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2,5 & 1,5 \\ 2,5 & 1 & 1,5 \\ 1,5 & 1,5 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -0,5 & 1,5 \\ 0,5 & 0 & 0,5 \\ -1,5 & -0,5 & 0 \end{bmatrix}$$

Simetri

Antisimetri

$$[a_{ij}] = [b_{ij}] + [c_{ij}] \quad (2.22)$$

dengan : $b_{ij} = (1/2) (a_{ij} + a_{ji}) \Rightarrow$ simetri

$c_{ij} = (1/2) (a_{ij} - a_{ji}) \Rightarrow$ anti simetri

5. Invers Matriks

Suatu matriks $[A]$ disebut sebagai inversnya matriks $[B]$ bila hasil kali keduanya merupakan matriks satuan.

$$[A][B] = [I] \quad (2.23)$$

Langkah-langkah proses invers matriks adalah :

- Gantikan masing-masing unsur matriks dengan masing-masing nilai kofaktornya.
- Transposkan matriks yang diperoleh tersebut. Matriks kemudian disebut sebagai matriks "adjoint".
- Hitung nilai determinan (matriks) aslinya.
- Bagilah unsur-unsur matriks "Adjoint" dengan nilai determinan matriks aslinya.

Contoh Inverskanlah matriks berikut ini :

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

Penjelasan :

$$a. A_{11} = 2 \quad A_{12} = -3 \quad A_{21} = -2 \quad A_{22} = 1$$

$$b. \begin{bmatrix} 2 & -3 \\ -2 & 1 \end{bmatrix}$$

$$c. A_{adj} = \begin{bmatrix} 2 & -2 \\ -3 & 1 \end{bmatrix}$$

$$d. \text{Determinan } |A| = 2 - 6 = -4$$

e. Matriks invers

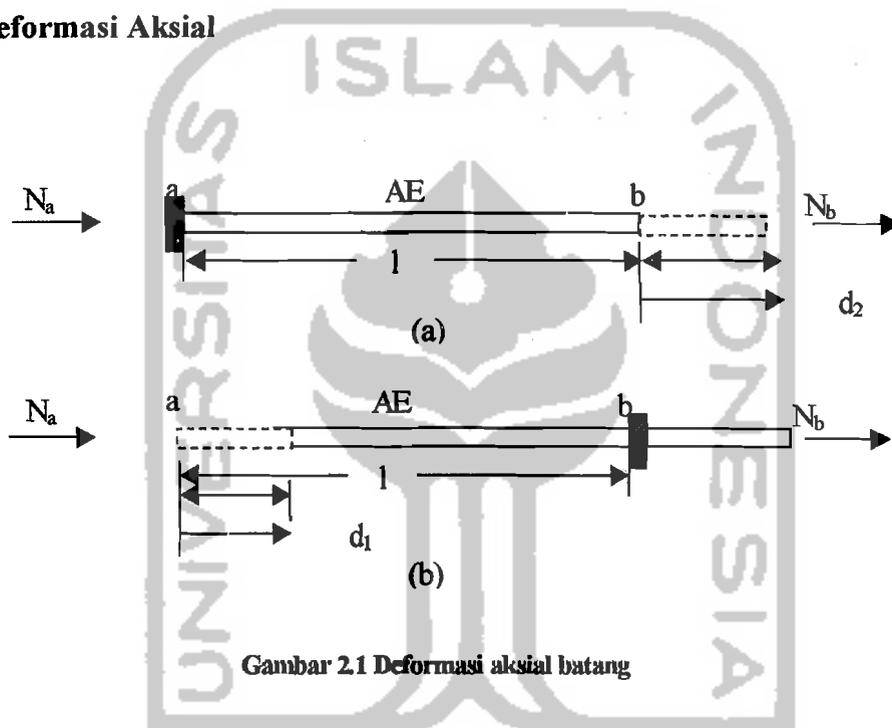
$$f. [A]^{-1} = -\frac{1}{4} \begin{bmatrix} 2 & -2 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} -0,5 & 0,5 \\ 0,75 & -0,25 \end{bmatrix}$$

g. Kontrol :

$$\begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} -0,5 & 0,5 \\ 0,75 & -0,25 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Suatu matriks yang matriks inversnya nol disebut matriks “singular”.

2.1.4 Deformasi Aksial



Gambar 2.1 Deformasi aksial batang

Suatu konstruksi bangunan apabila menerima beban luar, baik itu beban pada batang atau beban pada titik buhul, maka konstruksi tersebut akan mengalami deformasi. Dengan memperhatikan gambar 2.1, sebuah batang dibebani N_a dan N_b pada ujung-ujungnya, maka :

Syarat keseimbangan pada gambar 2.1.(a)

$$N_a = -\frac{AE}{l} \cdot d_2 \quad (2.24)$$

$$N_b = \frac{AE}{I} \cdot d_2 \quad (2.24)$$

Dan keseimbangan pada gambar 2.1.(b)

$$N_a = \frac{AE}{I} \cdot d_1 \quad (2.25)$$

$$N_b = -\frac{AE}{I} \cdot d_1 \quad (2.25)$$

Dengan menggabungkan persamaan (2.24) dan (2.25), maka akan diperoleh

$$N_a = \frac{AE}{I} \cdot d_1 - \frac{AE}{I} \cdot d_2 \quad (2.26)$$

$$N_b = -\frac{AE}{I} \cdot d_1 + \frac{AE}{I} \cdot d_2 \quad (2.26)$$

Persamaan (2.26) dapat ditulis dalam bentuk matriks sebagai berikut :

$$\begin{bmatrix} N_a \\ N_b \end{bmatrix} = \begin{bmatrix} \frac{AE}{I} & -\frac{AE}{I} \\ -\frac{AE}{I} & \frac{AE}{I} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (2.26.a)$$

$$\text{atau : } [N] = [k] [d] \quad (2.26.b)$$

dengan = [N] = matriks beban luar

[k] = matriks kekakuan batang

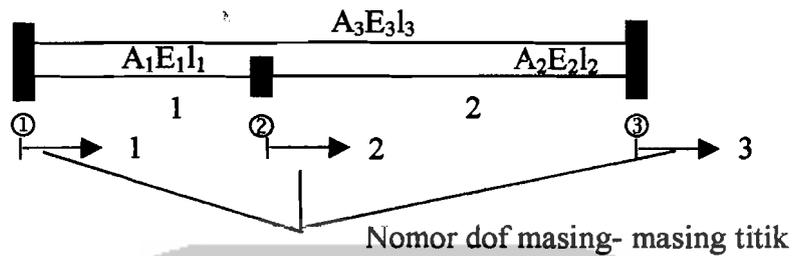
[d] = matriks displsemen

“Joint Code” (JCODE) dan “Member Code” (MCODE)

JCODE adalah satu set angka yang terdiri atas nomor-nomor derajat kebebasan pada suatu titik.

MCODE adalah satu set angka yang terdiri atas nomor-nomor derajat kebebasan pada ujung-ujung suatu batang.

JCODE dan MCODE merupakan alat bantu untuk menyusun matriks kekakuan, matriks beban luar dan untuk keperluan lain.



Gambar 2.2 Derajat Kebebasan (dof)

Dari gambar 2.2 dapat disusun :

JCODE(1) = 1 artinya Joint Code titik 1 = nomor derajat kebebasan 1.

JCODE(2) = 2 artinya Joint Code titik 2 = nomor derajat kebebasan 2.

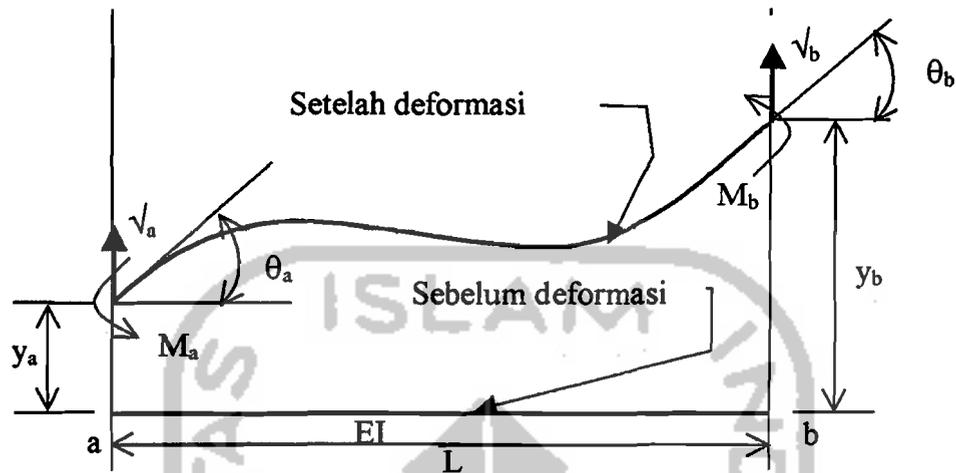
dan seterusnya.

MCODE(1) = [1 2] artinya ujung-ujung batang 1 mempunyai nomor derajat kebebasan (no dof) 1 dan 2.

MCODE(2) = [2 3] artinya ujung-ujung batang 2 mempunyai nomor derajat kebebasan (no dof) 2 dan 3.

MCODE(3) = [1 3] artinya ujung-ujung batang 3 mempunyai nomor derajat kebebasan (no dof) 1 dan 3.

2.1.5 Deformasi Lentur



Gambar 2.3 Batang Lentur

Untuk memperoleh persamaan dasar batang lentur dapat diturunkan dari persamaan "Slope Deflection".

Persamaan "slope deflection" :

$$M_a = \frac{2EI}{L}(2\theta_a + \theta_b - 3\psi_{ab}) \quad (2.27)$$

$$M_b = \frac{2EI}{L}(\theta_a + 2\theta_b - 3\psi_{ab})$$

Dengan

$$\psi_{ab} = \frac{1}{L}(Y_b - Y_a) \quad (2.28)$$

Agar memenuhi syarat keseimbangan, maka :

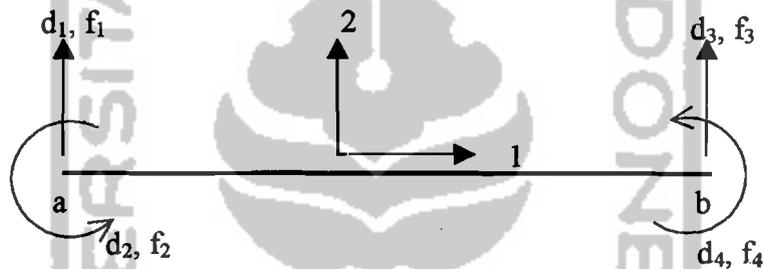
$$V_a = \frac{1}{L}(M_a + M_b) \quad (2.29)$$

$$V_b = -V_a$$

Dengan mengkombinasikan persamaan (2.27), (2.28) dan (2.29), maka akan dapat diperoleh :

$$\begin{aligned} V_a &= \alpha (12 Y_a + 6L \theta_a - 12 Y_b + 6L \theta_b) \\ M_a &= \alpha (6L Y_a + 4L^2 \theta_a - 6L Y_b + 2L^2 \theta_b) \\ V_b &= \alpha (-12 Y_a - 6L \theta_a + 12 Y_b - 6L \theta_b) \\ M_b &= \alpha (6L Y_a + 2L^2 \theta_a - 6L Y_b + 4L^2 \theta_b) \end{aligned} \quad (2.30)$$

Dengan $\alpha = \frac{EI}{L^3}$



Gambar 2.4 Batang Lentur (bentuk matriks)

Untuk memudahkan proses hitungan dengan metode matriks, maka indeks persamaan diganti dengan nomor urut. Begitu pula notasi yang lain diganti sesuai dengan gambar 2.4.

Arah gaya maupun deformasi pada gambar 2.4 adalah arah-arah positif.

Dengan demikian dapat disusun persamaan berikut :

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \alpha \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 12 & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \quad (2.31)$$

$$\text{dengan } \alpha = \frac{EI}{L^3}$$

Persamaan (2.31) disebut *persamaan dasar* batang lentur.

Persamaan (2.31) secara simbolis dapat dituliskan sebagai berikut :

$$f = k d \quad (2.32)$$

Dengan

$$K = \alpha \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 12 & 2L^2 & -6L & 4L^2 \end{bmatrix}, \alpha = \frac{EI}{L^3} \quad (2.32a)$$

2.1.6 Rangka Batang Bidang (“Truss Element”)

Rangka batang bidang didefinisikan sebagai konstruksi rangka dengan titik-titik buhulnya berupa sendi (dianggap sendi). Sehingga deformasi yang terjadi pada batang-batangnya akibat beban luar adalah deformasi aksial.

Pada rangka bidang persamaan dasarnya sama dengan persamaan pada deformasi aksial, hanya saja arah kedudukan batang-batangnya sembarang. Dengan kata lain sistem kordinat masing-masing batangnya tidak selalu sama dengan sistem kordinat strukturnya. Dengan demikian untuk menganalisis konstruksi ini diperlukan transformasi kordinat.

1. Persamaan dasar

Arah gaya dan displenemen yang tampak pada gambar 2.5 adalah arah positif.



Gambar 2.5 "Element" Rangka

Sesuai dengan persamaan (2.26), maka :

$$f = k d \quad (2.33)$$

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \gamma \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \gamma = \frac{AE}{L} \quad (2.34)$$

$$\text{dengan : } k = \gamma \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \gamma = \frac{AE}{L}$$

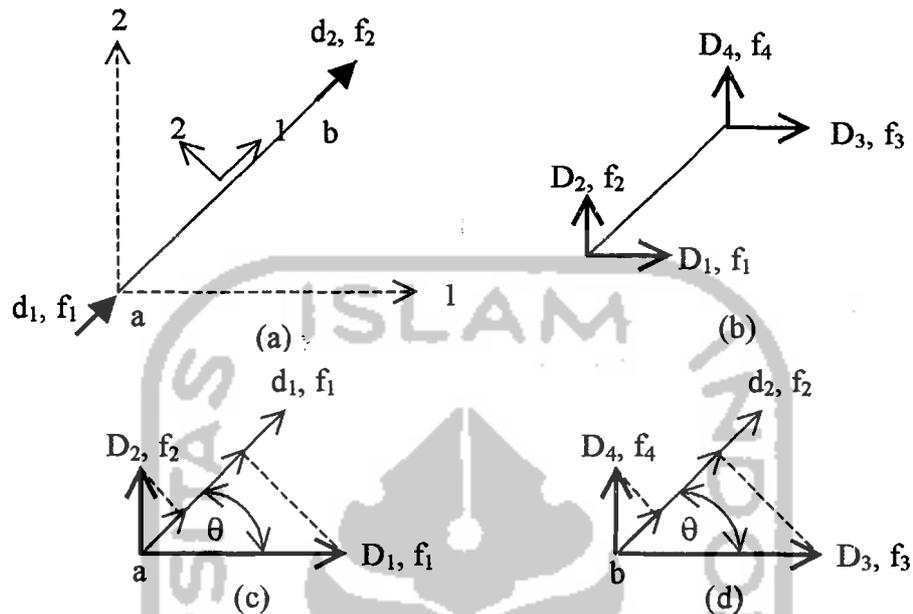
$$\text{atau : } k = \begin{bmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{bmatrix} \quad (2.35)$$

2. Matriks Transformasi

Telah disebutkan diatas bahwa arah kedudukan batang-batang Rangka Batang Bidang adalah sembarang, sehingga kedudukan Rangka batang Bidang secara umum dapat dilihat pada gambar 2.6.

Garis putus-putus pada gambar 2.6 menunjukkan Sistem Kordinat Global (sistem kordinat struktur). Biasanya sumbu 1/sumbu X ditetapkan horisontal dan sumbu2/sumbu Y ditetapkan vertikal. Sedangkan pada Kordinat Lokal digambarkan

dengan garis penuh, sumbu/sumbu X ditetapkan sumbu tiap batang dan sumbu 2/sumbu Y ditetapkan tegak lurus sumbu batang.



Gambar 2.6 a) Kondisi Lokal
 b) Kondisi Global
 c) Transformasi ujung a
 d) Tranformasi ujung b

Gambar 2.6a menunjukkan gaya dan deformasi pada sistem kordinat lokal, sedang gambar 2.6b pada sistem kordinat global. Dari gambar 2.6c akan diperoleh persamaan berikut :

$$d_1 = D_1 \cos \theta + D_2 \sin \theta \quad (2.36)$$

dalam bentuk matriks :

$$d_1 = \begin{bmatrix} \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \quad (2.37)$$

Dari gambar 2.6c dan analog dengan persamaan (2.37) dapat diperoleh :

$$d_2 = [\text{Cos}\theta \quad \text{Sin}\theta] \begin{bmatrix} D_3 \\ D_4 \end{bmatrix} \quad (2.38)$$

Untuk memudahkan dalam penjelasan lebih lanjut ujung-ujung batang biasa disebut dengan ujung a dan ujung b seperti terlihat pada gambar 2.6a. Dengan demikian persamaan (2.37) dan (2.38) dapat dituliskan sebagai berikut :

$$d_a = [c \quad s] D_a \quad (2.39)$$

$$d_b = [c \quad s] D_b$$

dengan

$$: c = \text{Cos } \theta, s = \text{Sin } \theta$$

$$D_a = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix}, D_b = \begin{bmatrix} D_3 \\ D_4 \end{bmatrix}$$

Jika diambil $\lambda = [c \quad s]$, maka persamaan (2.39) dapat dituliskan :

$$\begin{bmatrix} d_a \\ d_b \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} D_a \\ D_b \end{bmatrix} \quad (2.40)$$

Untuk menyederhanakan penulisan dan penjabaran lebih lanjut persamaan (2.40) sering dituliskan sebagai berikut :

$$[d] = [\Lambda] [D] \quad (2.41)$$

$$\text{dengan : } [d] = \begin{bmatrix} d_a \\ d_b \end{bmatrix}, [\Lambda] = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}, [D] = \begin{bmatrix} D_a \\ D_b \end{bmatrix}$$

Analog dengan persamaan (2.40), akan disusun persamaan sebagai berikut :

$$\begin{bmatrix} f_a \\ f_b \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} F_a \\ F_b \end{bmatrix} \quad (2.42)$$

atau :

$$[f] = [\Lambda][F] \quad (2.43)$$

Dari persamaan (2.48) dan (2.50) dapat diperoleh :

$$\begin{bmatrix} D_a \\ D_b \end{bmatrix} = \begin{bmatrix} \lambda^T & 0 \\ 0 & \lambda^T \end{bmatrix} \begin{bmatrix} d_a \\ d_b \end{bmatrix} \quad (2.44)$$

atau :

$$[D] = [\Lambda]^T [d] \quad (2.45)$$

dan

$$\begin{bmatrix} F_a \\ F_b \end{bmatrix} = \begin{bmatrix} \lambda^T & 0 \\ 0 & \lambda^T \end{bmatrix} \begin{bmatrix} f_a \\ f_b \end{bmatrix} \quad (2.46)$$

atau ;

$$[F] = [\Lambda]^T [f] \quad (2.47)$$

Dari persamaan (2.26) dan (2.55) akan diperoleh :

$$[F] = \Lambda^T k d \quad (2.48)$$

Dari persamaan (2.56) dan (2.49), dapat diperoleh :

$$F = \Lambda^T \Lambda D \quad (2.49)$$

Persamaan (2.57) identik dengan rumus $F = K D$, sehingga dapat diperoleh :

$$K = \Lambda^T k \Lambda \quad (2.50)$$

Sehingga :

$$\begin{aligned} K &= \begin{bmatrix} \lambda^T & 0 \\ 0 & \lambda^T \end{bmatrix} \begin{bmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \\ &= \begin{bmatrix} \lambda^T k_{aa} \lambda & \lambda^T k_{ab} \lambda \\ \lambda^T k_{ba} \lambda & \lambda^T k_{bb} \lambda \end{bmatrix} \begin{bmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{bmatrix} \end{aligned} \quad (2.51)$$

dengan : $K_{aa} = \lambda^T k_{aa} \lambda$

$$K_{ab} = \lambda^T k_{ab} \lambda \quad (2.52)$$

$$K_{ba} = \lambda^T k_{ba} \lambda$$

$$K_{bb} = \lambda^T k_{bb} \lambda$$

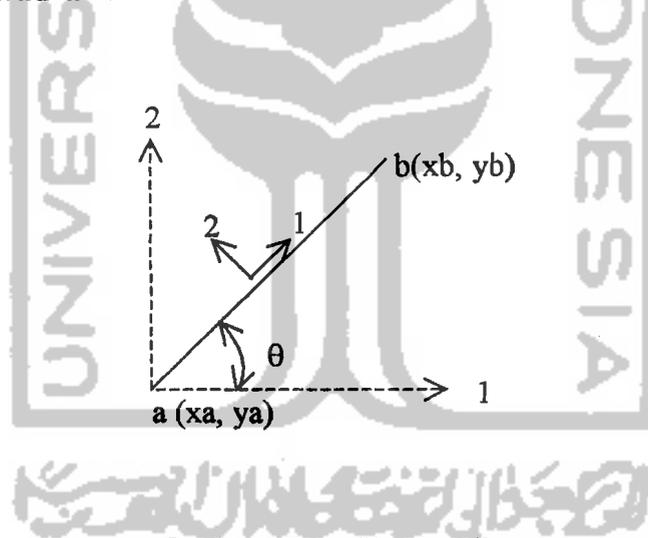
Dengan $\lambda = [c \ s]$ dan dari persamaan (2.43) maka akan diperoleh :

$$[K] = \gamma \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix}, \gamma = \frac{AE}{L} \quad (2.53)$$

[K] adalah matriks kekakuan “element” (batang) pada sistem global.

Cara menghitung matriks transformasi :

a. θ pada kwadran I

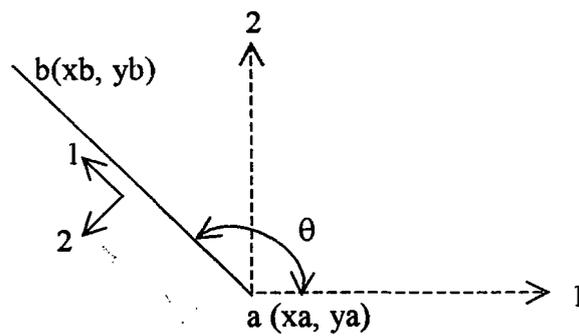


Gambar 2.7 Kwadran I

$$\cos \theta = \frac{X_b - X_a}{L} \quad (\text{positif})$$

$$\sin \theta = \frac{Y_b - Y_a}{L} \quad (\text{positif}) \quad (2.54)$$

b. θ pada kwadran II



Gambar 2.8 Kwadran II

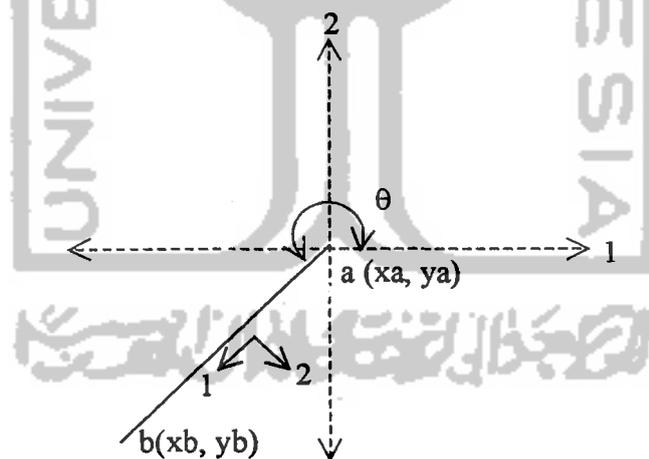
$$\cos \theta = \frac{X_b - X_a}{L}$$

(negatif)

$$\sin \theta = \frac{Y_b - Y_a}{L}$$

(positif)

(2.55)

c. θ pada kwadran III

Gambar 2.9 Kwadran III

$$\cos \theta = \frac{X_b - X_a}{L}$$

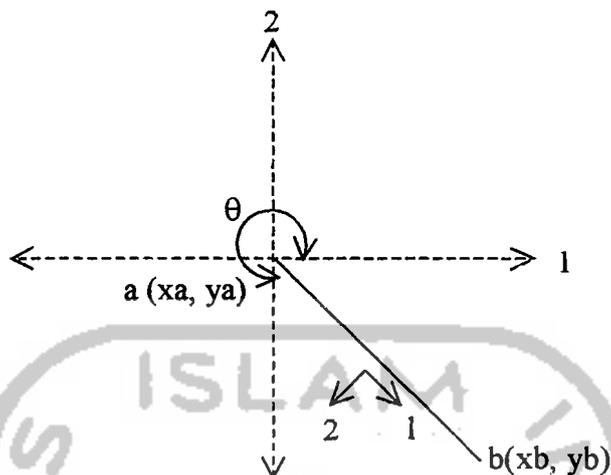
(negatif)

$$\sin \theta = \frac{Y_b - Y_a}{L}$$

(negatif)

(2.56)

d. θ pada kwadran IV



Gambar 2.10 Kwadran IV

$$\begin{aligned} \cos \theta &= \frac{X_b - X_a}{L} && \text{(positif)} \\ \sin \theta &= \frac{Y_b - Y_a}{L} && \text{(negatif)} \end{aligned} \quad (2.57)$$

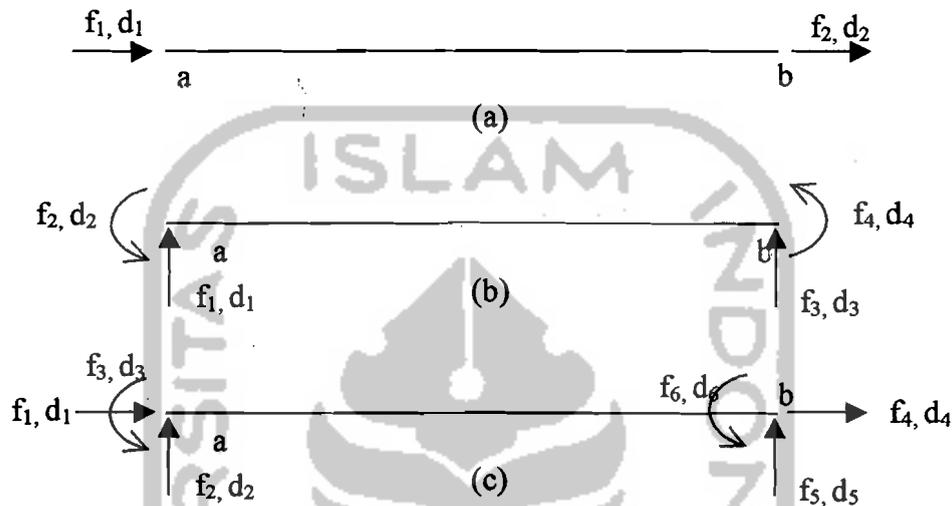
Berdasarkan analisis diatas, dapat diambil kesimpulan bahwa dapat dibentuk suatu persamaan yang berlaku umum :

$$\begin{aligned} \cos \theta &= \frac{X.\text{nomor.titik.besar} - X.\text{nomor.titik.kecil}}{L} \\ \sin \theta &= \frac{Y.\text{nomor.titik.besar} - Y.\text{nomor.titik.kecil}}{L} \end{aligned} \quad (2.58)$$

2.1.7 Portal Bidang (“Frames Structures”)

Portal adalah suatu konstruksi rangka yang batang-batangnya dihubungkan secara kaku sempurna, sehingga sudut antara batang-batangnya sebelum dan sesudah pembebanan tetap tidak berubah.

Karena adanya beban luar, baik itu beban pada batang (“element load”) atau beban pada titik buhul (“joint load”), batang akan mengalami deformasi aksial dan lentur. Dengan demikian persamaan dasarnya merupakan gabungan antara persamaan dasar batang yang berdeformasi aksial dan persamaan dasar batang lentur.



Gambar 2.11 Batang (a) deformasi aksial
(b) deformasi lentur
(c) kombinasi (a) dan (b)

Untuk lebih jelasnya kedua persamaan tersebut diatas akan ditulis kembali di-bawah ini.

1. Batang Yang Mengalami Deformasi Aksial

Lihat gambar 2.11(a)

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = j \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \quad j = \frac{AE}{L} \quad (2.59)$$

2. Batang Yang Mengalami Deformasi Lentur

Lihat gambar 2.11(b)

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \alpha \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 12 & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}, \alpha = \frac{EI}{L^3} \quad (2.60)$$

3. Batang Yang Mengalami Deformasi Aksial dan Lentur (Portal)

Lihat gambar 2.11(c)

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \alpha \begin{bmatrix} \beta & 0 & 0 & -\beta & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ -\beta & 0 & 0 & \beta & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} \quad (2.61)$$

$$\text{dimana } \alpha = \frac{EI}{L^3}, \beta = \frac{AL^2}{I}$$

Persamaan (2.61) disebut persamaan dasar struktur portal bidang. Persamaan (2.61) tersebut diturunkan berdasarkan sumbu batang sejajar dengan sumbu X lokal. Dengan demikian agar dapat dilakukan penyusunan matriks kekakuan seluruh sistem struktur diperlukan matriks transformasi.

4. Matriks Transformasi

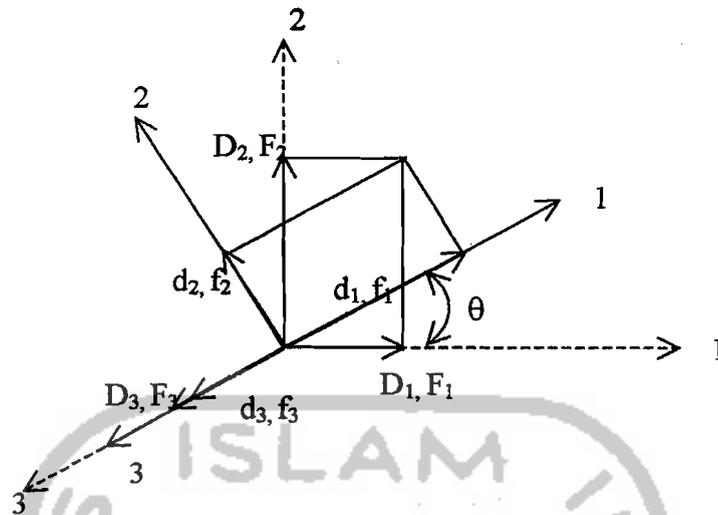
$$d_1 = D_1 \cos \theta + D_2 \sin \theta$$

$$d_2 = -D_1 \sin \theta + D_2 \cos \theta \quad (2.62)$$

$$d_3 = D_3$$

Persamaan (2.62) dapat dituliskan dalam bentuk matriks sebagai berikut :

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \quad (2.63)$$



Gambar 2.12 Transformasi Kordinat

$$\text{atau } [d_a] = [\lambda] [D_a] \quad (2.64)$$

Jika $\text{Cos } \theta = c$ dan $\text{Sin } \theta = s$ maka

$$[\lambda] = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.65)$$

Analog dengan persamaan (2.63) akan dapat diperoleh

$$\begin{bmatrix} d_4 \\ d_5 \\ d_6 \end{bmatrix} = \begin{bmatrix} \text{Cos } \theta & \text{Sin } \theta & 0 \\ -\text{Sin } \theta & \text{Cos } \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_4 \\ D_5 \\ D_6 \end{bmatrix} \quad (2.66)$$

$$\text{atau : } [d_a] = [\lambda] [D_b] \quad (2.67)$$

Dengan menggabungkan persamaan (2.64) dan persamaan (2.67) akan didapat

$$\begin{bmatrix} d_a \\ d_b \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} D_a \\ D_b \end{bmatrix} \quad (2.68)$$

$$\text{atau } d = [\Lambda] D \quad (2.69)$$

Analog dengan penjabaran persamaan (2.41) s/d persamaan (2.52), maka akan diperoleh :

$$[K] = \begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix} \quad (2.70)$$

$$\text{dengan : } K_{aa} = \lambda^T k_{aa} \lambda$$

$$K_{ab} = \lambda^T k_{ab} \lambda$$

$$(2.71)$$

$$K_{ba} = \lambda^T k_{ba} \lambda$$

$$K_{bb} = \lambda^T k_{bb} \lambda$$

Nilai- nilai k_{aa} , k_{ab} , k_{ba} dan k_{bb} adalah merupakan matriks kekakuan batang pada sistem kordinat lokal, seperti nampak pada persamaan dibawah ini.

$$[k] = \begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix} = \alpha \begin{bmatrix} \beta & 0 & 0 & -\beta & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ -\beta & 0 & 0 & \beta & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} \quad (2.72)$$

sehingga diperoleh :

$$K_{aa} = \alpha \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \alpha \begin{bmatrix} \beta & 0 & 0 \\ 0 & 12 & 6L \\ 0 & 6L & 4L^2 \end{bmatrix} \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.73)$$

Dengan cara yang sama akan dapat dihitung nilai- nilai K_{ab} , K_{ba} dan K_{bb} . Sehingga akan dapat diperoleh matriks kekakuan batang/element pada sistem kordinat global, seperti nampak dibawah ini :

$$[K] = \begin{bmatrix} g_1 & g_2 & g_4 & -g_1 & -g_2 & g_4 \\ & g_3 & g_5 & -g_2 & -g_3 & g_5 \\ & & g_6 & -g_4 & -g_5 & g_7 \\ & & & g_1 & g_2 & -g_4 \\ \text{Simetri} & & & & g_3 & -g_5 \\ & & & & & g_6 \end{bmatrix} \quad (2.74)$$

dengan : $g_1 = \alpha (\beta c^2 + 12 s^2)$

$$g_2 = \alpha cs (\beta - 12)$$

$$g_3 = \alpha (\beta s^2 + 12 c^2)$$

$$g_4 = -\alpha 6 L s$$

$$g_5 = \alpha 6 L c$$

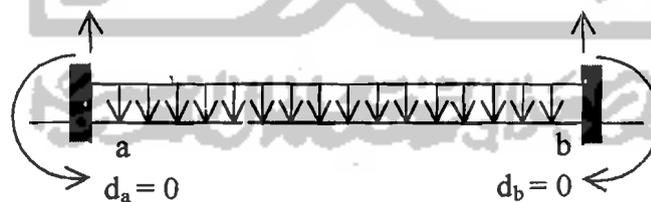
$$g_6 = \alpha 4 L^2$$

$$g_7 = \alpha 2 L^2$$

$$\alpha = \frac{EI}{L^3}, \beta = \frac{AL^2}{I}$$

(2.74 a)

5. "Element Actions"



Gambar 2.13 "Element Actions"

"Element Actions" adalah suatu kondisi luar yang menyebabkan batang tetap diam pada tempatnya jika tidak ada displacements pada titik buhulnya.

"Element Actions" dapat terdiri atas :

- a. pembebanan
- b. perubahan temperatur
- c. ketidaksempurnaan pembuatan, dan lain-lain.

Secara khusus respon dari suatu sistem struktur dapat ditransformasikan kedalam beban titik ekuivalen dengan persamaan :

$$\overline{FS} = F\hat{S} + FS \quad (2.75)$$

\overline{FS} = gaya-gaya dalam sesungguhnya, termasuk didalamnya gaya-gaya primer dan gaya-gaya titik buhul.

$F\hat{S}$ = gaya-gaya primer yang terdiri dari respons batang-batanganya dan gaya pada titik buhul yang dibutuhkan untuk menahan dispersemen pada titik buhul.

FS = sistem gaya ekuivalen tanpa gaya primer.

$$FS = \overline{FS} - F\hat{S} \quad (2.76)$$

FS hanya terdiri dari beban pada titik buhul sebab "element actions" $F\hat{S}$ dan dengan persamaan (2.84), dikurangkan terhadap \overline{FS} menghasilkan FS . Sehingga FS dapat dihitung dengan metode matriks dispersemen. Sedang $F\hat{S}$ dengan jelas dapat dihitung secara sendiri-sendiri tiap batang karena tiap-tiap titik buhulnya ditahan sehingga tidak ada interaksi dengan batang lainnya.

Dalam bentuk matriks beban titik buhul ekuivalen dapat dinyatakan sebagai berikut :

$$F = \overline{F} - \hat{F} \quad (2.77)$$

dengan :

\overline{F} = Vektor beban pada titik buhul

\hat{F} = Vektor beban pada titik buhul tetap (“Fixed and Forces”)

F = Vektor beban ekivalen

\bar{F} sudah dijelaskan terlebih dahulu dan \hat{F} dapat dihitung sebagai berikut :

- a). Untuk tiap-tiap batang/eleman yang menderita beban pada batang, dihitung vektor “fixed and forces “ (f^i) pada kordinat lokal dan kemudian ditransformasikan ke vektor “fixed and forces “ pada kordinat global(\hat{F}^i).
- b). Dengan bantuan MCODE, \hat{F}^i ditransformasikan sesuai dengan nomor derajat kebebasan yang ditunjukkan oleh MCODE sehingga diperoleh $\hat{F}^{(i)}$.

$$\hat{F}^i \xrightarrow{\text{MCODE}} \hat{F}^{(i)} \quad (2.78)$$

- c). Untuk seluruh sistem struktur dengan n derajat kebebasan yang terdiri dari NE(“Number of Element”), diperoleh kondisi keseimbangan .

$$\hat{F} = \sum_{i=1}^{NE} \hat{F}^{(i)} \quad (2.79)$$

Vektor gaya dalam tiap-tiap batang dapat dihitung dengan rumus dibawah ini,

$$\bar{f}^i = f^i + \hat{f}^i \quad (2.80)$$

dengan $f^i = k^i d^i$

6. “Fixed and Forces”

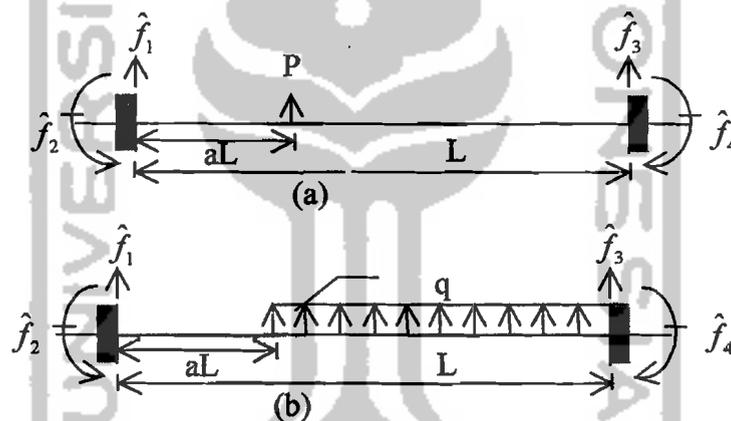
Dibawah ini akan disajikan rumus “fixed and forces” yang paling banyak digunakan pada sistem pembebanan di Indonesia.

Beban terpusat :

$$f = P \begin{bmatrix} -1 - a^2(2a - 3) \\ -La(1 - a)^3 \\ a^2(2a - 3) \\ La^2(1 - a) \end{bmatrix} \quad (2.81)$$

Beban terbagi rata :

$$f = qL \begin{bmatrix} -\frac{1}{2}(1 - a^4 + 2a^3 - 2a) \\ -\frac{L}{12}(1 - 3a^4 + 8a^3 - 6a^2) \\ -\frac{1}{2}(1 + a^4 - 2a^3) \\ \frac{L}{12}(1 + 3a^4 - 4a^3) \end{bmatrix} \quad (2.82)$$

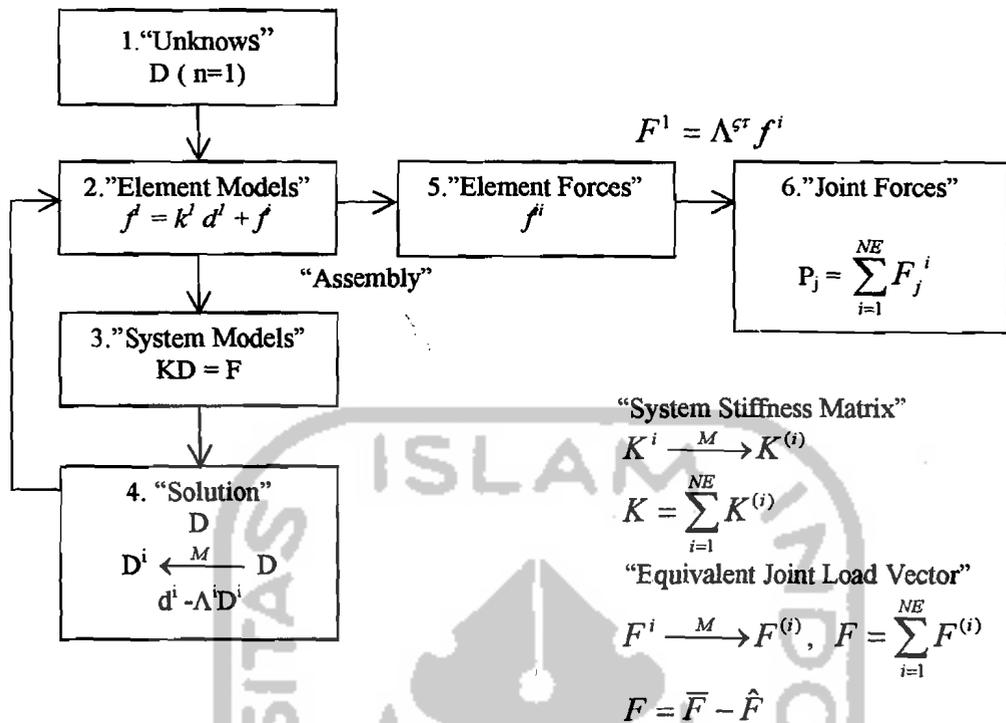


Gambar 2.14 "Fixed and Forces"

- (a) Beban Terpusat
(b) Beban Terbagi rata

7. Prosedur Hitungan

Penjelasan dari diagram alir di bawah :



Gambar 2.15 Metode "Matrix" Displacement" dengan beban pada batang

(n = dof, NE = jumlah batang)

a. "Unknowns"

Menentukan berapa jumlah displacements yang mungkin terjadi pada sistem tersebut yaitu sesuai dengan berapa nilai derajat kebebasan sistem struktur tersebut (=n). Cara menghitung n adalah dengan melihat satu persatu kemungkinan displacements masing-masing titik buhul.

b. "Element Models"

Menentukan bentuk persamaan dasar : $\bar{f}^i = k^i d^i + \hat{f}^i$

\bar{f}^i = gaya dalam

k^i = matriks kekakuan batang pada kordinat lokal

d^i = displesemen ujung tiap- tiap batang pada kordinat lokal

\hat{f}^i = gaya-gaya primer("fixed and forces")

Sehingga dalam hal ini gaya-gaya dalam final adalah merupakan penjumlahan gaya dalam karena displesemen titik buhul dan gaya-gaya primer.

c. "System Models"

Menyusun persamaan simultan $KD = F$, matriks kekakuan sistem struktur (K) dan matriks beban luar (F) disusun. Sehingga dari persamaan $KD = F$ diperoleh matriks displesemen seluruh sistem struktur (D) pada sistem kordinat global.

Menyusun matriks [K] :

$K^i \xrightarrow{M} K^{(i)}$, matriks kekakuan elemen pada kordinat global K^i dengan bantuan "MCOE"(M), indeks ("subscribs") unsur-unsurnya disesuaikan dengan nomor dof pada masing-masing "MCOE"nya. Sehingga diperoleh matriks $K^{(i)}$ dengan indeks sudah sesuai dengan nomor dof . Dengan kata lain $K^{(i)}$ merupakan sumbangan kekakuan batang ke i terhadap kekakuan sistem struktur.

$K = \sum_{i=1}^{NE} K^{(i)}$, matriks kekakuan sistem struktur (K) merupakan penjumlahan matriks kekakuan batang-batangnya $K^{(i)}$.

Menyusun matriks :

$$[F] = [\bar{F}] - [\hat{F}]$$

Matriks beban luar $[F]$ adalah merupakan pengurangan dari matriks beban titik

$[\bar{F}]$ dan matriks gaya-gaya dalam primer $[\hat{F}]$ ("fixed and forces").

$F^i \xrightarrow{JCODE} \bar{F}^{(i)}$, matriks beban luar pada masing-masing titik buhul F^i disusun kembali dengan indeks sesuai dof pada JCODE, sehingga diperoleh $\bar{F}^{(i)}$.

$\bar{F} = \sum_{i=1}^{NJ} \bar{F}^{(i)}$, matriks beban luar titik buhul merupakan penjumlahan dari beban luar masing-masing titik buhul (NJ = "Number of Joint"/jumlah titik buhul).

$\hat{F}^i \xrightarrow{M} \hat{F}^{(i)}$, gaya ujung batang/gaya-gaya primer (sering juga dikatakan *moment primer*) atau "fixed and forces" $[\hat{F}^i]$, letak unsur-unsurnya disusun kembali sesuai dengan nomor dof pada MCODE masing-masing batangnya. Sehingga diperoleh matriks "fixed and forces" $[\hat{F}^{(i)}]$ dengan indeks unsur-unsurnya sesuai dengan nomor dof.

$\hat{F} = \sum_{i=1}^{NE} \hat{F}^{(i)}$, "fixed and forces" total adalah merupakan penjumlahan "fixed and forces" masing-masing batangnya.

d. "Solution"

Dari langkah butir c di atas diperoleh persamaan simultan $KD = F$. Solusi dari persamaan tersebut akan diperoleh matriks displesemen seluruh titik buhul sistem struktur tersebut ($[D]$).

e. "Element Forces" (gaya dalam batang)

Agar supaya dapat dihitung gaya dalam masing-masing batang ($\bar{f} = k^i d^i + f^i$), maka perlu dihitung terlebih dahulu matriks displesemen masing-masing batang (d^i) pada sistem kordinat lokal. Adapun langkah untuk menghitung d^i adalah :

$D \xrightarrow{M} D^i$, matriks dispersemen masing-masing batang pada kordinat global (D^i), unsur-unsurnya diambilkan dari matriks D dengan bantuan nomor indeks sesuai MCODE masing-masing batang.

$d^i = \Lambda^i D^i$, matriks dispersemen batang pada sistem kordinat global D^i dikalikan dengan matriks transformasi D^i , sehingga diperoleh matriks dispersemen masing-masing batang pada sistem kordinat lokal d^i . Setelah diperoleh matriks d^i , maka dengan persamaan (2.88) dapat dihitung gaya dalam masing-masing batang f^i .

f. "Joint Forces" (gaya pada titik buhul)

Gaya-gaya titik buhul pada umumnya dinyatakan pada sistem kordinat global.

Gaya-gaya titik buhul dihitung dengan cara mentransformasikan kembali gaya

dalam masing-masing batang ($P_j = \sum_{i=1}^{NE} \Lambda^{iT} f^i$).

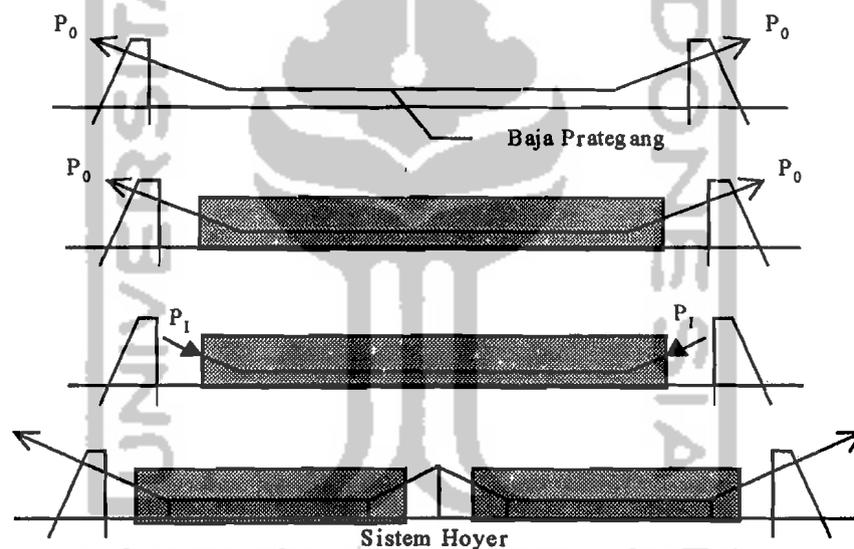
2.2 Struktur Beton Prategang

Beton Prategang menurut ACI adalah beton yang mengalami tegangan internal dengan besar dan distribusi sedemikian, sehingga dapat mengimbangi sampai batas tertentu tegangan yang terjadi akibat beban eksternal.

Tegangan internal diperoleh dengan memberikan tekanan (gaya prategang) pada beton sebelum beban luar bekerja.

Cara pemberian tekanan :

1. Pratarik ("Pretensioning")

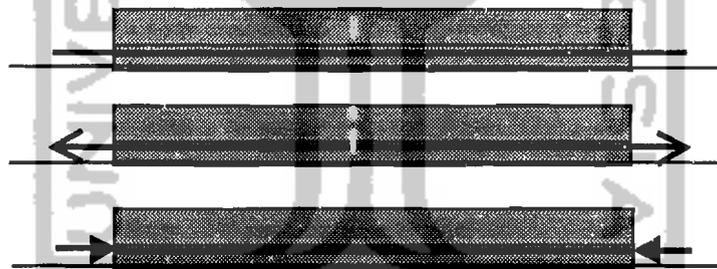


Gambar 2.16 Metode Pratarik (Pretensioning)

Baja prategang ditarik/ditegangkan terlebih dahulu dan dijangkarkan pada dinding penahan, kemudian beton dicor. Setelah beton mencapai umur/kekuatan tertentu, baja prategang dipotong dimuka ujung komponen, sehingga akan menekan komponen. Transfer gaya prategang melalui ikatan/lekatan antara baja prategang dengan beton.

2. Pasca Tarik (“Post Tensioning”)

Sistem Pasca tarik merupakan kebalikan sistem pratarik. Pada cara ini, pertama-tama beton dicor terlebih dahulu, dengan baja prategang yang diletakkan dalam selongsong (“duck”) ikut tercor. Setelah beton mencapai umur/kekuatan tertentu, baja prategang ditarik dan kemudian dijangkarkan/diangkurkan pada ujung komponen. Transfer gaya prategang pada sistem ini melalui angkur. Bila rongga antara selongsong dan baja prategang diisi/di “grout” dengan adukan semen (semen + air + bahan tambah atau semen + air + pasir halus + bahan tambah) maka disebut tendon terikat (“Bounded”), sedang bila tidak di “grouting” disebut tendon tak terikat (“Unbounded”). Pada tendon tak terikat umumnya ruang antara selubung dan baja prategang diberi oli atau bahan kimia lain untuk mencegah terjadinya karat pada tendon.



Gambar 2.17 Metode Pasca Tarik (“Post Tensioning”)

Pemberian gaya prategang (penarikan tendon) dilakukan dengan menarik tendon ke arah longitudinal, berbagai peralatan untuk menarik tendon dapat dikelompokkan menjadi :

a. Alat Mekanis, umumnya dipakai pada pekerjaan dalam jumlah besar yaitu di pabrik, berupa timbangan dengan atau tanpa transmisi, dongkrak skrup atau mesin penggulung kawat.

b. Dongkrak Hidraulis, merupakan sarana paling banyak dipakai. Peralatan paten yang umum dipakai adalah dari Freyssinet, Magnel, dll. Kapasitas dongkrak hidraulis mencapai 1000 ton lebih.

c. Metoda Termal, yaitu dengan memanaskan baja prategang dengan menggunakan listrik yang diangkurkan sebelum beton di cor. Metoda ini disebut juga Prategang Termolistrik.

d. Metoda Kimia, yaitu dengan menggunakan semen yang mengembang, disini derajat pengembangan dikontrol dengan mengubah-ubah kondisi perawatan beton ("Curing"). Aksi pengembangan beton dan pengikatan awal menimbulkan gaya tarik pada tendon dan tekan pada beton, hal ini dikarenakan pengembangan beton dikekang.

Untuk menjelaskan/menganalisis sifat dasar beton prategang, dikenal 3 konsep yaitu :

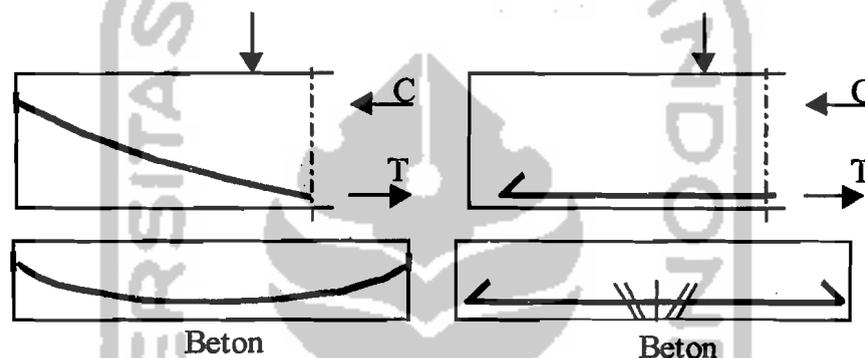
a. Sistem Prategang untuk mengubah beton menjadi bahan yang elastis.

Menurut konsep ini, beton bersifat getas ditransformasikan menjadi bahan yang bersifat elastis, dengan cara memberi tekanan/ desakan terlebih dahulu (pratekan). Dari konsep ini timbul kriteria tidak terjadi tegangan tarik pada beton ("Full Prestress"), karena bagian beton yang akan mengalami tarik akibat beban kerja, mengalami desak terlebih dahulu karena gaya prategang, berarti beton tidak mengalami retak pada beban kerja. Maka pada pandangan ini,

beton digambarkan sebagai benda yang mengalami dua sistem pembebanan, yaitu gaya internal (prategang) dan beban eksternal (berat sendiri + beban mati + beban hidup).

b. Sistem Prategang sebagai kombinasi baja mutu tinggi dengan beton

Seperti pada beton bertulang, beton prategang merupakan kombinasi baja prategang (menahan tarik) dan beton (menahan tekan), kedua bahan membentuk kopel penahan untuk melawan kopel eksternal.



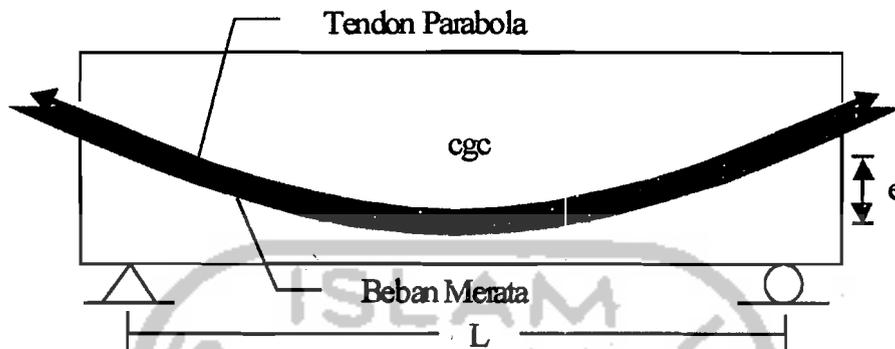
Gambar 2.18 Perbedaan Beton Prategang dan Beton Bertulang

Jika baja mutu tinggi digunakan sebagai tulangan pada beton bertulang, maka akan terjadi retak-retak pada bagian tarik sebelum seluruh kekuatan baja dikerahkan. Karena itu, pada beton prategang, untuk memanfaatkan seluruh kekuatan baja mutu tinggi, baja ditarik terlebih dahulu dan menjangkarkannya ke beton. Dengan demikian diperoleh tegangan dan regangan yang diinginkan untuk kedua bahan.

c. Sistem Prategang untuk mencapai keseimbangan beban

Dalam konsep ini, pemberian gaya prategang dipandang sebagai suatu usaha membuat seimbang gaya-gaya, sehingga komponen struktur yang mengalami

lentur, tidak akan mengalami tegangan lentur pada kondisi pembebanan tertentu.



Gambar 2.19 Sistem Prategang untuk mencapai kesimbangan

Bila tendon dengan profil parabola ditarik, untuk dapat tetap mempertahankan posisinya diperlukan gaya vertikal kebawah. Jika tendon terbungkus beton, maka akan timbul gaya keatas menekan beton, yang berlawanan arah dengan gaya untuk mempertahankan posisi tendon.

Maka akibat gaya prategang, tendon memberikan gaya aksial tekan P dan beban terbagi rata keatas terhadap beton pembungkusnya sebesar :

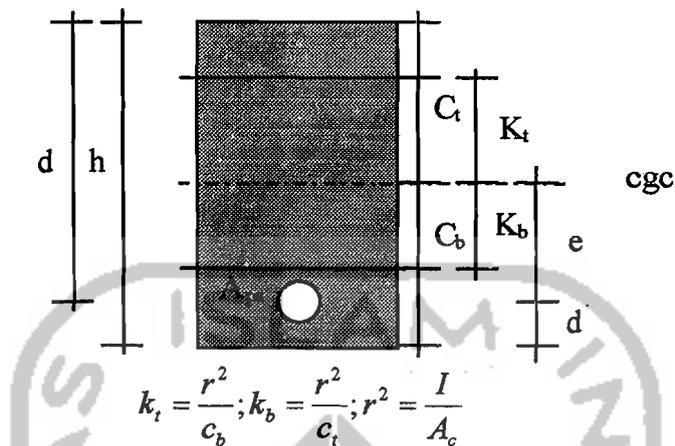
$$w_b = \frac{8.P.e}{L^2} \quad (2.83)$$

Sehingga untuk beban w yang terdistribusi merata, maka beban tidak seimbang adalah $(w-w_b)$.

Pola Tegangan Balok Beton Prategang

Karena hampir semua kelebihan beton adalah pada tingkat beban kerja, dan besarnya gaya prategang umumnya ditentukan oleh tegangan ijin beton, maka bagian utama hitungan analisis dan perencanaan menggunakan beban kerja, dengan

tegangan ijin (metoda elastis) dan anggapan- anggapan dasar sesuai SK-SNI pasal 3.11.



Gambar 2.20 Tampang Beton Prategang

Disamping itu harus juga memenuhi syarat kekuatan yang ditentukan, karena itu pada beberapa bagian komponen struktur juga dilakukan analisis dengan menggunakan beban berfaktor dan faktor reduksi yang sesuai (metoda kekuatan batas).

Pola tegangan yang terjadi pada penampang beton prategang, dilihat dengan meninjau tegangan akibat gaya prategang dan beban-beban yang bekerja pada tahap-tahap tertentu.

Dengan menganggap penampang bebas retak pada tingkat beban kerja, maka seluruh penampang memikul tegangan yang terjadi, sehingga seluruh luas beton diperhitungkan dalam menentukan kedudukan pusat berat dan momen inersia penampang.

Pada dasarnya, baik pada sistem Pratarik maupun sistem Pascatarik, pola tegangan ditinjau pada dua saat yang berbeda, yaitu :

1. Saat awal ("transfer")

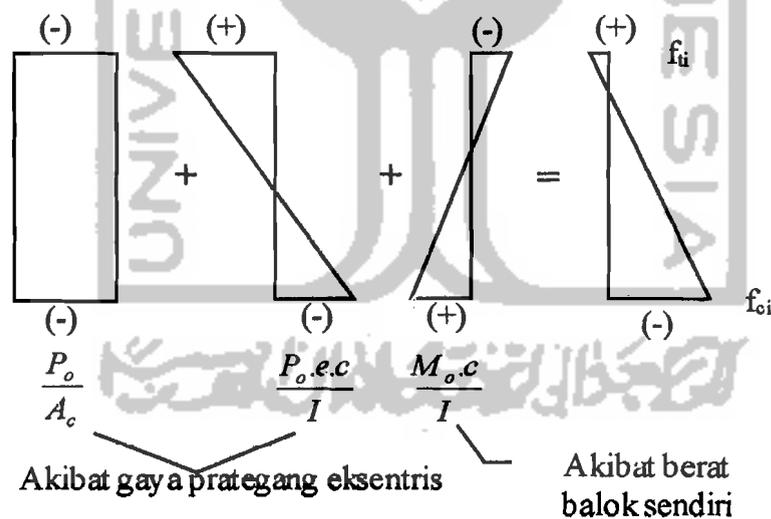
Yaitu saat pelimpahan gaya prategang (penarikan tendon pada sistem Pascatarik, pemotongan tendon pada sistem Pratarik).

Beban-beban yang diperhitungkan adalah :

- gaya prategang awal (sebelum terjadi kehilangan tegangan).
- beban mati berat sendiri.
- pada sistem Pascatarik, mungkin sebagian beban mati telah bekerja.

Sedangkan penampang yang digunakan untuk perhitungan pusat berat dan inersia :

- Pratarik : penampang transformasi atau penampang bruto.
- Pascatarik : penampang netto/bersih.



Gambar 2.21 Tegangan pada saat awal

Maka tegangan-tegangan yang terjadi pada serat atas dan serat bawah pada saat transfer :

- Pada serat tepi atas

$$f_t = -\frac{P_o}{A_c} + \frac{P_o \cdot e \cdot c_t}{I} - \frac{M_o \cdot c_t}{I} \text{ karena } s_t = \frac{I}{c_t} \text{ maka}$$

$$f_t = -\frac{P_o}{A_c} + \frac{P_o \cdot e \cdot A_c}{s_t \cdot A_c} - \frac{M_o}{s_t} \quad (2.83)$$

$$f_t = -\frac{P_o}{A_c} \left[1 - \frac{e \cdot A_c}{s_t} \right] - \frac{M_o}{s_t} \quad (2.84)$$

$$\text{atau : } f_t = -\frac{P_o}{A_c} + \frac{P_o \cdot e - M_o \cdot A_c}{I/c_t \cdot A_c} \quad (2.85)$$

$$f_t = -\frac{P_o}{A_c} + \frac{P_o}{A_c} \left[\frac{e - (M_o/P_o)}{I/c_t} \right] A_c \quad (2.86)$$

$$\text{karena } k_b = \frac{r^2}{c_t} = \frac{I/A_c}{c_t} = \frac{I}{A_c \cdot c_t} \text{ maka}$$

$$f_t = -\frac{P_o}{A_c} \left[1 - \frac{e - (M_o/P_o)}{k_b} \right] \quad (2.87)$$

b. Pada serat tepi bawah

$$f_b = -\frac{P_o}{A_c} - \frac{P_o \cdot e \cdot c_b}{I} + \frac{M_o \cdot c_b}{I} \text{ karena } s_b = \frac{I}{c_b} \text{ maka}$$

$$f_b = -\frac{P_o}{A_c} - \frac{P_o \cdot e \cdot A_c}{s_b \cdot A_c} + \frac{M_o}{s_b} \quad (2.88)$$

$$f_b = -\frac{P_o}{A_c} \left[1 + \frac{e \cdot A_c}{s_b} \right] + \frac{M_o}{s_b} \quad (2.89)$$

$$\text{atau : } f_b = -\frac{P_o}{A_c} - \frac{P_o}{A_c} \left[\frac{e \cdot (M_o/P_o)}{I/c_b} \right] A_c \quad (2.90)$$

$$\text{karena } k_t = \frac{r^2}{c_b} = \frac{I/A_c}{c_b} = \frac{I}{A_c \cdot c_b} \text{ maka}$$

$$f_b = -\frac{P_o}{A_c} \left[1 + \frac{e - (M_o/P_o)}{k_t} \right] \quad (2.91)$$

2. Saat akhir (masa layan/“service”)

Pada saat akhir telah terjadi seluruh kehilangan tegangan (kehilangan gaya prategang). Kehilangan gaya prategang disebabkan :

a. kehilangan sesaat:

- 1). perpendekan elastis beton (pratarik)
- 2). gesekan (pascatarik)
- 3). “draw in”/penggelinciran ankur (pascatarik)

b. kehilangan tergantung waktu :

- 1). rangkai beton
- 2). susut beton
- 3). relaksasi baja

Pada saat akhir, beban-beban yang bekerja/diperhitungkan adalah :

a. gaya prategang efektif (setelah terjadi seluruh kehilangan tegangan)

$$P_e = R.P_o$$

R = rasio kehilangan gaya prategang : Pratarik : ± 25%

Pascatarik : ± 20%

b. seluruh beban eksternal :

- 1). beban berat sendiri : M_o
- 2). beban mati : M_D $M_T = M_o + M_D + M_L$
- 3). beban hidup : M_L

Sedang penampang yang digunakan untuk perhitungan pusat berat inersia :

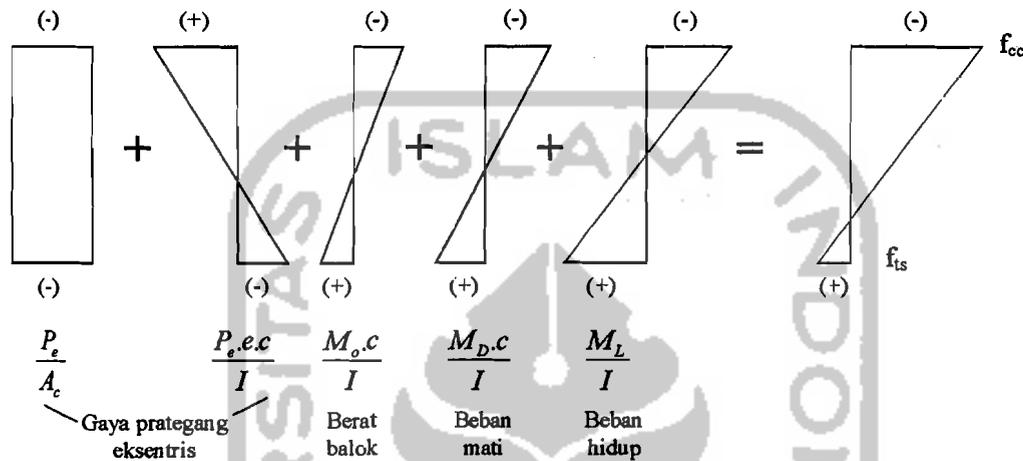


a. pratarik : penampang transformasi atau penampang bruto.

b. pascatarik :

1). tendon “bounded” : penampang transformasi

2). tendon “unbounded” : penampang netto/bersih



Gambar 2.22 Tegangan pada saat akhir

Sehingga tegangan-tegangan yang terjadi :

a. Pada serat tepi atas

$$f_t = -\frac{P_e}{A_c} + \frac{P_e \cdot e \cdot c_t}{I} - \frac{M_o \cdot c_t}{I} - \frac{M_L \cdot c_t}{I} \quad (2.92)$$

$$f_t = -\frac{P_e}{A_c} + \frac{P_e \cdot e \cdot A_c}{s_t \cdot A_c} - \frac{M_o}{s_t} - \frac{M_L}{s} \quad (2.93)$$

$$f_t = -\frac{P_e}{A_c} \left[1 - \frac{e \cdot A_c}{s_t} \right] - \frac{M_T}{s_t} \quad (2.94)$$

b. Pada serat tepi bawah

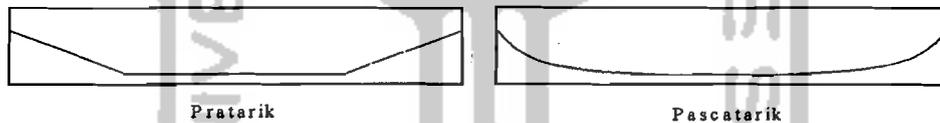
$$f_b = -\frac{P_e}{A_c} - \frac{P_e \cdot e \cdot c_b}{I} + \frac{M_o \cdot c_b}{I} + \frac{M_D \cdot c_b}{I} + \frac{M_L \cdot c_b}{I} \quad (2.95)$$

$$f_b = -\frac{P_e}{A_c} - \frac{P_e \cdot e \cdot A_c}{s_b \cdot A_c} + \frac{M_O}{s_b} + \frac{M_D}{s_b} + \frac{M_L}{s_b} \quad (2.96)$$

$$f_t = -\frac{P_e}{A_c} \left[1 + \frac{e \cdot A_c}{s_b} \right] + \frac{M_T}{s_b} \quad (2.97)$$

Telah diketahui bahwa makin mendekati tumpuan pada ujung komponen, nilai momen makin kecil, maka pola tegangan dengan sendirinya akan berubah menyesuaikan.

Bila tendon dipasang lurus eksentris (eksentrisitas tendon disepanjang balok tetap), diujung balok yang ditumpu sederhana, cenderung terjadi tegangan tarik karena $M_T = 0$. Karena tidak dikehendaki terjadi tegangan tarik, maka kedudukan tendon diujung balok digeser sedemikian, sehingga eksentrisitas terhadap garis netral kecil.



Gambar 2.23 Perbedaan Pratarik dan Pascatarik

Menurut SK-SNI, tegangan ijin beton dan tendon sebagai berikut :

Tegangan ijin beton untuk struktur lentur :

1. Saat transfer

a. Serat terluar mengalami tegangan tekan $0,60 \cdot f_{ci}'$

b. Serat terluar mengalami tegangan tarik kecuali yang diijinkan berikut

dibawah ini $0,25 \sqrt{f_{ci}'}$

c. serat terluar pada ujung komponen struktur yang didukung sederhana

mengalami tarik $0,50\sqrt{f'_{ci}}$

2. Saat akhir

a. serat terluar mengalami tegangan tekan $0,45\sqrt{f'_c}$

b. serat terluar mengalami tegangan tarik yang pada awalnya mengalami

tegangan tekan $0,50\sqrt{f'_c}$

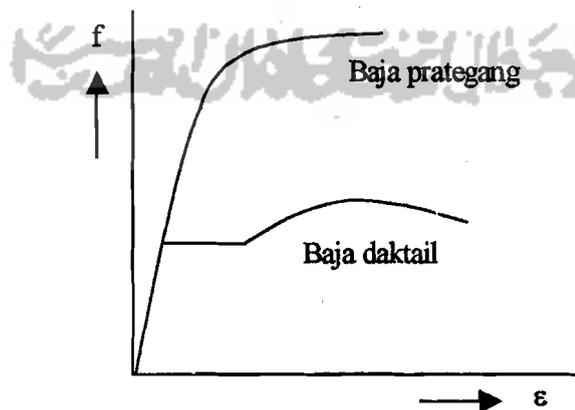
Tegangan ijin tendon pratekan :

1. akibat gaya penjangkaran tendon tetapi tidak lebih besar dari $0,85.f_{pu}$ atau nilai yang direkomendasi pabrik $0,94.f_{py}$

2. sesaat setelah pemindahan gaya pratekan tetapi tidak lebih besar dari $0,74.f_{pu}$ $0,82.f_{py}$

3. tendon pascatarik, pada daerah jangkar dan sambungan, sesaat setelah penjangkaran tendon $0,70.f_{pu}$

2.1.1 Kekuatan Lentur



Gambar 2.24 Grafik Regangan-Tegangan

Beton prategang menggunakan mutu beton tinggi ($f'_c \geq 30$ Mpa) dan baja dengan $f_{pu} > 1700$ Mpa.

Menurut teori kekuatan batas, penampang mencapai keseimbangan regangan bila pada serat tekan terluar beton mencapai regangan hancur $\epsilon_y = 0,003$ dan pada saat yang bersamaan baja mencapai regangan leleh.

Baja prategang tak memperlihatkan titik luluh secara jelas, maka konsep ini berubah, yaitu dengan menggunakan f_{ps} sebagai pengganti f_y .

SK-SNI memberikan batasan rasio tulangan prategang :

$$w_p = \frac{\rho_p \cdot f_{ps}}{f'_c} \leq 0,36 \cdot \beta_1 \quad (2.98)$$

β_1 : konstanta yang tergantung pada mutu beton

$$f'_c \leq 30 \text{ Mpa} \rightarrow \beta_1 = 0,85$$

$$f'_c > 30 \text{ Mpa} \rightarrow \beta_1 = 0,85 - 0,008(f'_c - 30) \geq 0,65$$

$$\rho_p = \frac{A_{ps}}{b \cdot d_p} \rightarrow \text{rasio tulangan prategang}$$

f_{ps} : tegangan dalam tulangan prategang pada saat M_n dicapai (sebagai pengganti f_y).

Batasan rasio tulangan prategang pada dasarnya untuk menjamin terjadinya penampang bertulangan liat/daktail, serta memberi batas maksimum letak garis netral sebesar $0,432 d_p$ dari serat tekan (setara $0,75 \cdot x_b$ pada beton bertulang) pada saat M_n dicapai.

Keseimbangan gaya- gaya dalam :

$$C = T \quad (2.99)$$

$$\text{Gaya tekan} : C = 0,85 \cdot f'_c \cdot b \cdot \beta_1 \cdot x \quad (2.100)$$

$$\text{Gaya tarik} : T = A_{ps} \cdot f_{ps} = \rho_b \cdot b \cdot d_b \cdot f_{ps} \quad (2.101)$$

Karena : $a = \beta_1 \cdot x$ maka :

$$a = \frac{A_{ps} \cdot f_{ps}}{0,85 \cdot f'_c \cdot b} = \frac{\rho_p \cdot b \cdot d_p \cdot f_{ps}}{0,85 \cdot f'_c \cdot b} = \frac{\rho_p \cdot f_{ps} \cdot d_p}{0,85 \cdot f'_c} \quad (2.112)$$

$$\text{Momen Nominal} : M_n = T \left[d_p - \frac{a}{2} \right] \quad (2.113)$$

Sedang f_{ps} menurut SK-SNI dihitung dengan rumus :

1. Tendon terekat (pratarik dan pascatarik dengan “grouting”)

$$f_{ps} = f_{pu} \left[1 - \frac{\gamma_p \cdot \rho_p \cdot f_{pu}}{\beta_1 \cdot f'_c} \right] \quad (2.114)$$

Bila menyertakan tulangan non prategang :

$$f_{ps} = f_{pu} \left[1 - \frac{\gamma_p}{\beta_1} \left(\frac{\rho_p \cdot f_{pu}}{f'_c} + \frac{d(w - w')}{d_p} \right) \right] \quad (2.115)$$

$$\text{dan} \left(\frac{\rho_p \cdot f_{pu}}{f'_c} + \frac{d(w - w')}{d_p} \right) \geq 0,17 \text{ serta nilai } d' \leq 0,15 \cdot d_p$$

dimana :

$$\gamma_p = 0,28 \text{ untuk } \frac{f_{py}}{f_{pu}} \geq 0,90 \text{ kabel bebas tegangan}$$

$$\gamma_p = 0,40 \text{ untuk } \frac{f_{py}}{f_{pu}} \geq 0,85 \text{ kabel dengan relaksasi rendah}$$

$$w = \frac{\rho \cdot f_y}{f'_c} \text{ dan } w' = \frac{\rho' \cdot f_y}{f'_c} \rightarrow \text{tulangan non prategang}$$

d_p : jarak serat terluar ketitik berat tulangan pratekan

d : jarak serat tekan terluar ketitik berat tulangan tarik

d' : jarak serat tekan terluar ketitik berat tulangan tekan

2. Tendon tak terekat

a. perbandingan antara bentang terhadap tinggi komponen ≤ 35

$$f_{ps} = f_{se} + 70 + \frac{f'_c}{100 \cdot \rho_p} \quad (2.116)$$

dimana : $f_{ps} \leq f_{py}$ dan $f_{ps} \leq f_{se} + 400$

b. perbandingan antara bentang terhadap tinggi komponen > 35

$$f_{ps} = f_{se} + 70 + \frac{f'_c}{300 \cdot \rho_p} \quad (2.117)$$

dimana : $f_{ps} \leq f_{py}$ dan $f_{ps} \leq f_{se} + 200$

dengan : f_{se} adalah tegangan efektif tulangan prategang (setelah terjadi seluruh kehilangan tegangan) dalam Mpa.

$$f_{se} \geq 0,05 \cdot f_{pu} ; f_{se} = \frac{P_e}{A_{ps}}$$

Untuk komponen yang mempunyai flens (T dan I), bila garis netral terletak diluar flens, maka luas baja prategang total dibagi menjadi dua bagian. Bagian pertama untuk mengimbangi tekan pada flens, bagian kedua untuk mengimbangi tekan pada badan, dengan tegangan f_{ps} untuk kedua bagian tersebut.

Maka:

$$M_n = A_{pw} \cdot f_{ps} \cdot \left(d_p - \frac{a}{2} \right) + 0,85 \cdot f'_c \cdot (b_f - b_w) h_f \left(d_p - \frac{h_f}{2} \right) \quad (2.118)$$

$$a = \frac{A_{pw} \cdot f_{ps}}{0,85 \cdot f'_c \cdot b_w} \quad (2.119)$$

$$A_{pf} = 0,85 \cdot \frac{f'_c}{f_{ps}} (b_f - b_w) h_f \rightarrow A_{pw} = A_{ps} - A_{pf} \quad (2.120)$$

Bila rasio tulangan $\frac{\rho_p \cdot f_{ps}}{f'_c} > 0,36 \cdot \beta_1$ maka kekuatan momen rencana harus

dihitung berdasarkan kekuatan bagian tekan dari momen kopel.

Untuk memenuhi syarat ini, pada komponen berflens, rasio tulangan hanya diambil bagian yang mengimbangi tekan pada badan (rasio tulangan badan) :

$$\rho_{pw} = \frac{A_{pw}}{b_w \cdot d_p} \quad (2.121)$$

Bila digunakan tulangan non prategang dari baja tulangan deform BJTD bersama-sama tulangan prategang, boleh dianggap menyumbangkan gaya tarik dan boleh dimasukkan dalam perhitungan kuat momen dengan tegangan sebesar tegangan leleh f_y .

2.1.2 Perencanaan

Perencanaan balok beton prategang terdiri atas beberapa tahapan, yaitu :

1. Pemilihan bentuk dan ukuran penampang
2. Peninjauan besar gaya prategang dan eksentrisitas tendon
3. Penentuan tata letak/"lay out" tendon disepanjang balok
4. Pemeriksaan terhadap :
 - a. tegangan-tegangan yang terjadi

- b. lendutan
- c. geser
- d. kuat/kapasitas penampang
- e. dan sebagainya

Jika untuk persamaan-persamaan tegangan diberikan nilai-nilai batas, yaitu tegangan ijin, maka diperoleh pertidaksamaan yang dapat dipergunakan untuk perencanaan.

1. Pada keadaan awal/"transfer" :

$$f_t = -\frac{P_o}{A_c} \left[1 - \frac{e.A_c}{s_t} \right] - \frac{M_o}{s_t} \leq f_{ti} \quad (2.122)$$

$$-\frac{P_o}{A_c} \left[1 - \frac{e.c_t}{r^2} \right] - \frac{M_o}{s_t} \leq f_{ti} \quad (2.123)$$

$$f_b = -\frac{P_o}{A_c} \left[1 - \frac{e.A_c}{s_b} \right] - \frac{M_o}{s_b} \leq f_{ci} \quad (2.124)$$

$$-\frac{P_o}{A_c} \left[1 - \frac{e.c_b}{r^2} \right] - \frac{M_o}{s_b} \leq f_{ci} \quad (2.125)$$

2. Pada keadaan akhir/"service" :

$$f_t = -\frac{P_e}{A_c} \left[1 - \frac{e.A_c}{s_t} \right] - \frac{M_T}{s_t} \leq f_{ts} \quad (2.126)$$

$$-\frac{P_e}{A_c} \left[1 - \frac{e.c_t}{r^2} \right] - \frac{M_T}{s_t} \leq f_{ts} \quad (2.127)$$

$$f_b = -\frac{P_e}{A_c} \left[1 - \frac{e.A_c}{s_b} \right] - \frac{M_T}{s_b} \leq f_{bs} \quad (2.128)$$

$$-\frac{P_e}{A_c} \left[1 - \frac{e.c_b}{r^2} \right] - \frac{M_T}{s_b} \leq f_{bs} \quad (2.129)$$

Dengan :

$$s = \frac{I}{c} ; r^2 = \frac{I}{A_c} \text{ sehingga : } \frac{A_c}{s} = \frac{A_c}{I/c} = \frac{A_c \cdot c}{I} = \frac{c}{r^2}$$

f_{ti} : tegangan ijin beton serat tarik saat “transfer”

f_{ci} : tegangan ijin beton serat tekan saat “transfer”

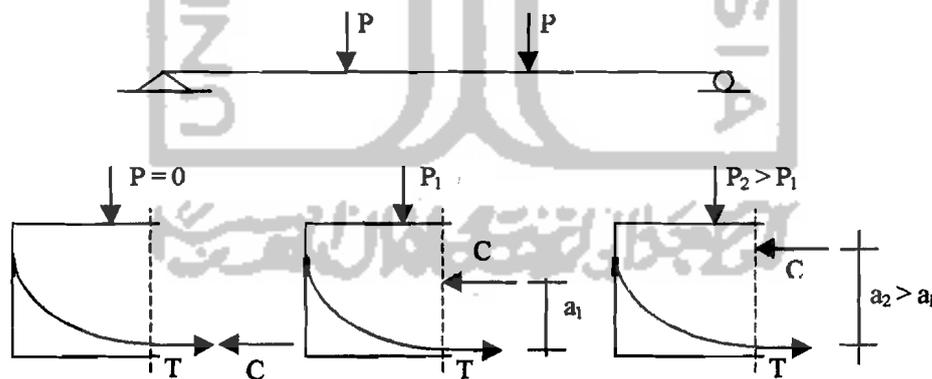
f_{ts} : tegangan ijin beton serat tarik saat akhir

f_{cs} : tegangan ijin beton serat tekan saat akhir

Menurut Statika, momen penahan (momen yang dibentuk oleh kopel C-T) suatu penampang, harus sama atau lebih besar dari momen eksternal yang bekerja pada penampang tersebut.

$$M_{\text{internal}} = M_{\text{eksternal}} \quad (2.130)$$

Pada penampang balok prategang, $T = P$ (gaya prategang) dan letaknya tetap, sedang letak C akan berubah-ubah sesuai dengan perubahan besar momen lentur.



Gambar 2.25 Perubahan letak C sesuai dengan perubahan besar Momen Lentur

Letak kopel C mempengaruhi distribusi tegangan pada penampang yang ditinjau.

Terdapat beberapa kemungkinan sebagai berikut :

Jika C terletak pada batas “kern” bawah, atau $e = k_b$, maka tegangan beton ditepi atas $f_t = 0$, sehingga :

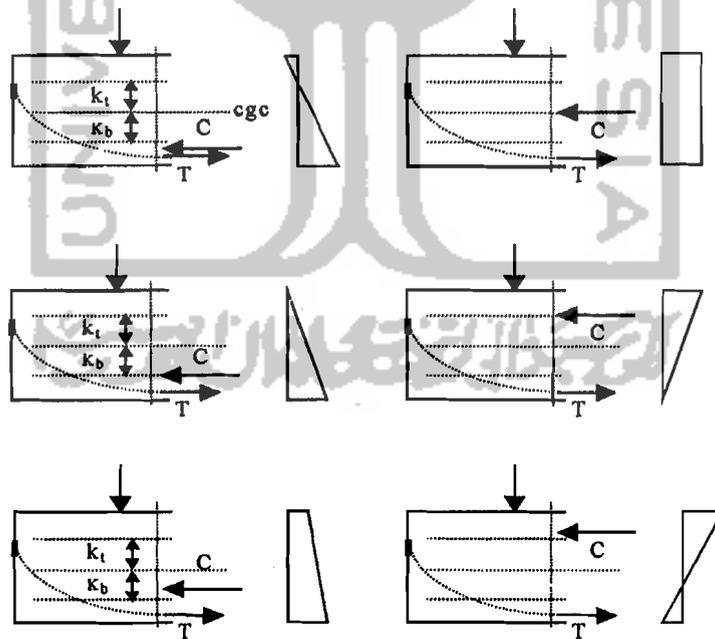
$$f_t = -\frac{P_o}{A_c} \left[1 - \frac{e \cdot c_t}{r^2} \right] = 0$$

$$\left[1 - \frac{e \cdot c_t}{r^2} \right] = 0 \rightarrow \left[1 - \frac{k_b \cdot c_t}{r^2} \right] = 0 \text{ sehingga}$$

$$k_b = \frac{r^2}{c_t} \quad (2.131)$$

Bila C terletak pada batas “kern”, maka tegangan beton ditepi bawah $f_b = 0$, dengan cara yang sama diperoleh :

$$k_t = \frac{r^2}{c_b} \quad (2.132)$$

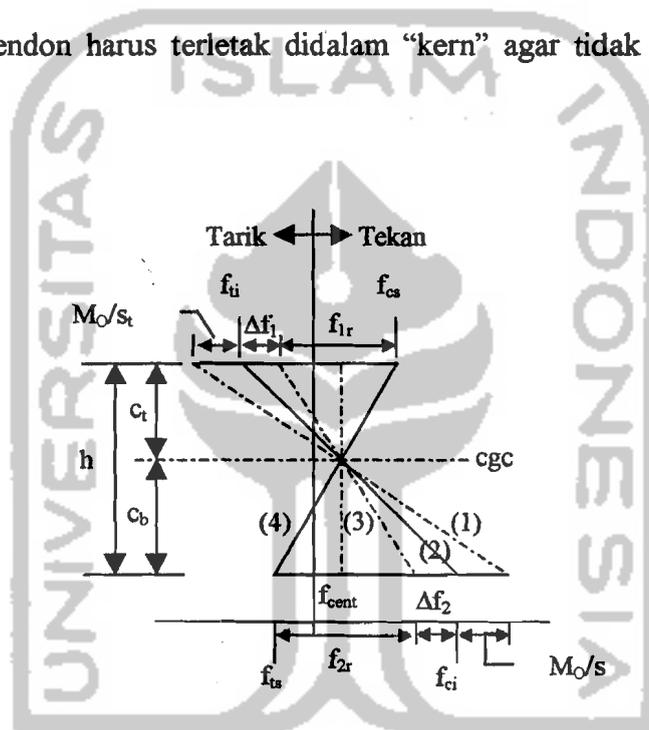


Gambar 2.26 Perubahan tegangan sesuai dengan perubahan letak C

Pada balok yang didukung sederhana, terdapat dua penampang kritis, yaitu pada tengah bentang dan diujung bentang/pada tumpuan. Sebab itu, eksentrisitas tendon harus sedemikian agar tidak terjadi tegangan yang melampaui tegangan ijin atau tegangan tarik yang berlebihan.

Ditengah bentang, dimana terjadi momen maksimum, tendon diletakkan diluar “kern”.

Pada perletakan, tendon harus terletak didalam “kern” agar tidak terjadi tegangan tarik.



Gambar 2.27 Diagram Tegangan

Keterangan : (1) $\rightarrow P_o$
 (2) $\rightarrow P_o + M_o$
 (3) $\rightarrow P_e + M_o$
 (4) $\rightarrow P_e + M_o + M_D + M_L = P_e + M_T$

Dari diagram tegangan diatas, maka persyaratan statis momen gelagar :

$$f_{1r} = \frac{M_D + M_L}{s_t} \rightarrow \text{sehingga } s_t = \frac{M_D + M_L}{f_{1r}} \quad (2.133)$$

$$f_{2r} = \frac{M_D + M_L}{s_b} \rightarrow \text{sehingga } s_b = \frac{M_D + M_L}{f_{2r}} \quad (2.133)$$

Tegangan-tegangan Δf_1 dan Δf_2 adalah perubahan tegangan yang disebabkan oleh kehilangan gaya prategang.

Karena $R = \frac{P_e}{P_o}$ dan kehilangan gaya prategang $(1-R) P_o$ maka

$$\Delta f_1 = (1-R) \left[f_{ti} + \frac{M_o}{s_t} \right] \quad (2.134)$$

$$\Delta f_2 = (1-R) \left[f_{ci} + \frac{M_o}{s_b} \right] \quad (2.134)$$

dan $f_{1r} = f_{ti} - \Delta f_1 - f_{cs}$

$$= f_{ti} - (1-R) \left[f_{ti} + \frac{M_o}{s_t} \right] - f_{cs} = R.f_{ti} - (1-R) \frac{M_o}{s_t} \quad (2.135)$$

$f_{2r} = f_{ts} - \Delta f_2 - f_{ci}$

$$= f_{ts} - f_{ci} - (1-R) \left[f_{ci} + \frac{M_o}{s_b} \right] = f_{ts} - R.f_{ci} - (1-R) \frac{M_o}{s_b} \quad (2.135)$$

sehingga :

$$s_t \geq \frac{M_D + M_L}{R.f_{ti} - (1-R) \frac{M_o}{s_t} - f_{cs}} \quad \text{atau} \quad s_t \geq \frac{(1-R)M_o + M_D + M_L}{R.f_{ti} - f_{cs}} \quad (2.136)$$

$$s_b \geq \frac{M_D + M_L}{f_{ts} - R.f_{ci} - (1-R) \frac{M_o}{s_b}} \quad \text{atau} \quad s_b \geq \frac{(1-R)M_o + M_D + M_L}{f_{ts} - R.f_{ci}} \quad (2.136)$$

Karena $I = s_t c_t = s_b c_b$, maka letak pusat penampang cgc sedemikian, sehingga

$$\frac{c_t}{c_b} = \frac{s_t}{s_b} \quad \text{dimana } h = c_t + c_b, \text{ maka}$$

$$\frac{c_t}{h} = \frac{s_b}{s_t + s_b} \quad (2.137)$$

Memperhatikan diagram tegangan, maka tegangan beton dipusat penampang pada kondisi awal :

$$f_{cent} = f_{ti} - \frac{c_t}{h}(f_{ti} - f_{ci}) \quad (2.138)$$

dan besar gaya prategang yang diperlukan :

$$P_o = A_c \cdot |f_{cent}| \rightarrow A_c = \text{luas penampang beton} \quad (2.139)$$

Gaya prategang P_o bekerja dengan eksentrisitas e , akan memberikan momen lentur $P_o \cdot e$ yang besarnya sedemikian, sehingga :

1. Berdasarkan kondisi serat atas, berlaku :

$$\frac{P_o \cdot e}{s_t} = f_{ti} - f_{cent} + \frac{M_o}{s_t} \text{ atau eksentrisitas tendon}$$

$$e = (f_{ti} - f_{cent}) \frac{s_t}{P_o} + \frac{M_o}{P_o} \quad (2.140)$$

2. Berdasarkan kondisi serat bawah, eksentrisitas tendon :

$$e = (f_{cent} - f_{ci}) \frac{s_b}{P_o} + \frac{M_o}{P_o} \quad (2.141)$$

Rumus-rumus diatas berdasarkan :

1. momen maksimum
2. e berubah/ e variabel
3. Jika e tetap

Harga e maksimum ditetapkan berdasarkan kondisi dipertakan/diujung balok, dimana $M_o = 0$.

$$\Delta f_1 = (1 - R) \cdot f_{ti} \quad (2.142)$$

$$\Delta f_2 = (1 - R) \cdot f_{ci} \quad (2.142)$$

dan $f_{1r} = R \cdot f_{ti} - f_{cs}$ (2.143)

$$f_{2r} = f_{ts} - R \cdot f_{ci} \quad (2.143)$$

Modulus penampang yang diperlukan :

$$s_t \geq \frac{M_O + M_D + M_L}{R \cdot f_{ti} - f_{cs}} \quad (2.144)$$

$$s_b \geq \frac{M_O + M_D + M_L}{f_{ts} - R \cdot f_{ci}} \quad (2.144)$$

Sedangkan tegangan pada pusat penampang beton (f_{cent}) dan besar gaya prategang P_o dihitung dengan menggunakan rumus :

$$f_{cent} = f_{ti} - \frac{e_t}{h} (f_{ti} - f_{ci}) \quad (2.145)$$

$$P_o = A_c \cdot |f_{cent}| \rightarrow A_c = \text{luas penampang beton} \quad (2.146)$$

Karena $M_O = 0$, maka nilai e (eksentrisitas) dihitung berdasarkan kondisi :

1. Serat atas $: e = (f_{ti} - f_{cent}) \frac{s_t}{P_o}$ (2.147)

2. Serat bawah $: e = (f_{cent} - f_{ci}) \frac{s_b}{P_o}$ (2.148)

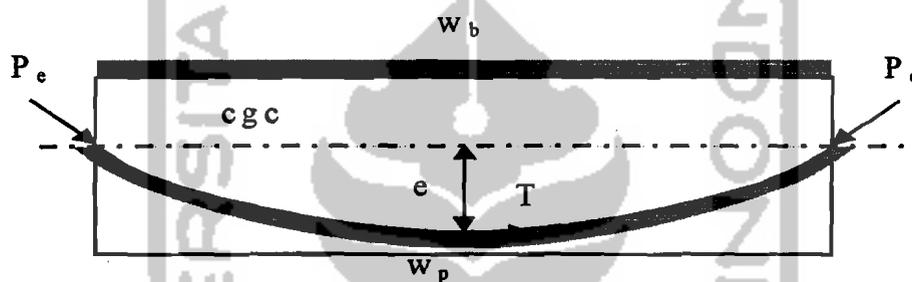
2.1.3 Metoda Beban Berimbang

Bila kabel/tendon prategang yang dipasang dalam posisi melengkung/ membentuk parabola (e variabel) ditarik/dilakukan penegangan, maka timbul gaya-

gaya keatas menekan beton, yang berlawanan arah dengan gaya untuk mempertahankan posisi tendon.

Berarti akibat gaya prategang, akan memberikan beban aksial P pada ujung-ujung komponen dan beban terbagi rata keatas disepanjang komponen.

Dari kenyataan ini, timbul konsep yang memandang upaya prategangan kabel sebagai usaha pengimbangan beban dalam komponen struktur.



Gambar 2.28 "Lay out" tendon bentuk parabola

Momen akibat gaya prategang ditengah bentang, dapat disamakan dengan momen akibat merata ekuivalen, maka :

$$M = T.e \text{ dengan } T = P_e \quad (2.149)$$

$$M = \frac{1}{8} . w_l . L^2 \quad (2.150)$$

maka beban merata ekuivalen (keatas) akibat gaya prategang :

$$w_p = \frac{8.P_e.e}{L^2} \quad (2.151)$$

dan beban merata imbang/ "balance" : $w_b = w_p$ yang dapat berupa

$$w_b = w_o + \lambda.w_D \text{ atau} \quad (2.152)$$

$$w_b = w_o + w_D + \lambda.w_L \text{ atau} \quad (2.153)$$

$$w_b = \lambda \cdot w_T \quad (2.154)$$

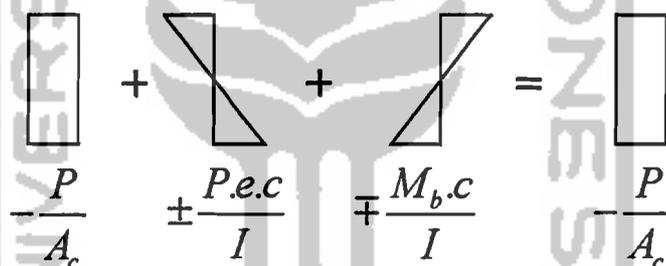
dan gaya prategang efektif: $P_e = \frac{M}{e} = \frac{w_b \cdot L^2}{8 \cdot e}$ (2.155)

beban tidak imbang: $w_{ub} = w_T - w_b$ (2.156)

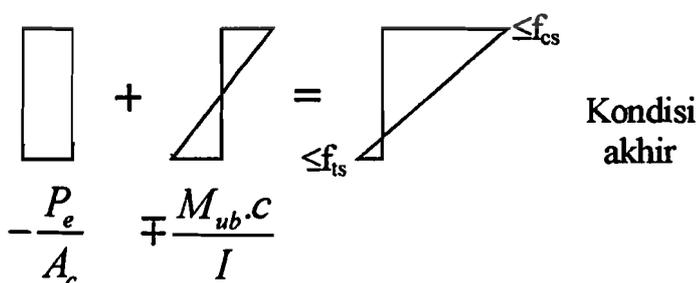
$$M_{ub} = \frac{1}{8} \cdot w_{ub} \cdot L^2 \quad (2.157)$$

sehingga tegangan-tegangan yang terjadi pada beton :

$$f = -\frac{P}{A_c} \pm \frac{M_{ub} \cdot c}{I} \quad (2.158)$$

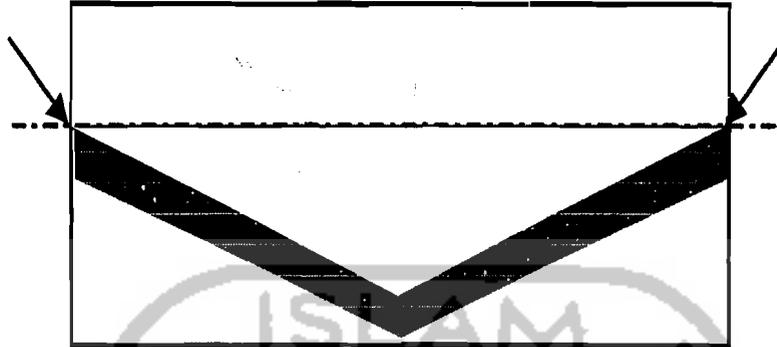


Karena $P \cdot e = M_b$



Gambar 2.29 Diagram tegangan

Bila "lay out" tendon berbentuk segitiga :



Gambar 2.29 "Lay out" tendon berbentuk segitiga

Beban imbang : $w_b = w_p = \frac{2.P.e}{L^2}$ (2.159)

2.3 Borland Delphi 4

2.3.1 Sejarah Borland Delphi 4

Delphi adalah bahasa pemrograman yang dikembangkan dari bahasa pemrograman terstruktur Pascal. Bahasa Pascal ini diciptakan oleh ilmuwan dari Swiss, yaitu Niklaus Wirth pada tahun 1971. Nama Pascal diambil dari ahli matematika dan filsafat dari Perancis, yaitu Blaise Pascal (1623 – 1662).

Salah satu versi Pascal dari beberapa versi yang muncul adalah Turbo Pascal, yang dirilis oleh Borland International Incorporation pada tahun 1983. Pada saat itu Turbo Pascal hanya dapat berjalan pada sistem operasi DOS. Kemudian dalam perkembangan selanjutnya Borland International juga merilis Turbo Pascal yang berjalan pada sistem operasi Windows 3.X, yaitu Turbo Pascal For Windows.

Pada tahun 1992 Borland International menggabungkan Turbo Pascal For DOS dan Turbo Pascal For Windows menjadi satu paket yang diberi nama Borland Pascal Versi 7 yang dilengkapi dengan paket penunjang untuk meningkatkan kemampuan OOP (“Object Oriented Programming”/Bahasa Pemrograman Berorientasi Obyek) yaitu Turbo Vision 2.0.

Pada tahun 1995 Borland International merilis lagi Borland Delphi 1.0 yang berbasis Windows 3.x

Pada tahun 1996 Borland International merilis Borland Delphi 2.0 yang dikhususkan untuk sistem operasi Windows 95 dan Windows NT.

Pada bulan Maret tahun 1997, Borland International kembali merilis Borland Delphi 3.0.

Akhirnya pada tahun 1998, Borland International merilis Borland Delphi 4.0.

Dalam pembuatan sebuah program, Delphi menggunakan sistem RAD (“Rapid Application Development”/Pengembangan Aplikasi secara Cepat). Sistem ini menggunakan pendekatan visual, sehingga seorang programmer bisa lebih mudah mendisain tampilan program (“User interface”). Pada pemrograman biasa (non visual), waktu seorang programmer akan dihabiskan untuk merancang atau memperindah tampilan program dari pada menulis program utamanya.

2.3.2 Menjalankan Borland Delphi 4

Borland Delphi 4 dapat dijalankan dengan berbagai cara, seperti terlihat pada gambar dibawah ini :



Gambar 2.31 Menjalankan Borland Delphi 4

Keterangan gambar 2.31 :

Cara 1, dengan menekan tombol “Start” , Pilih “Program”, “Borland Delphi 4” dan “Delphi 4”.

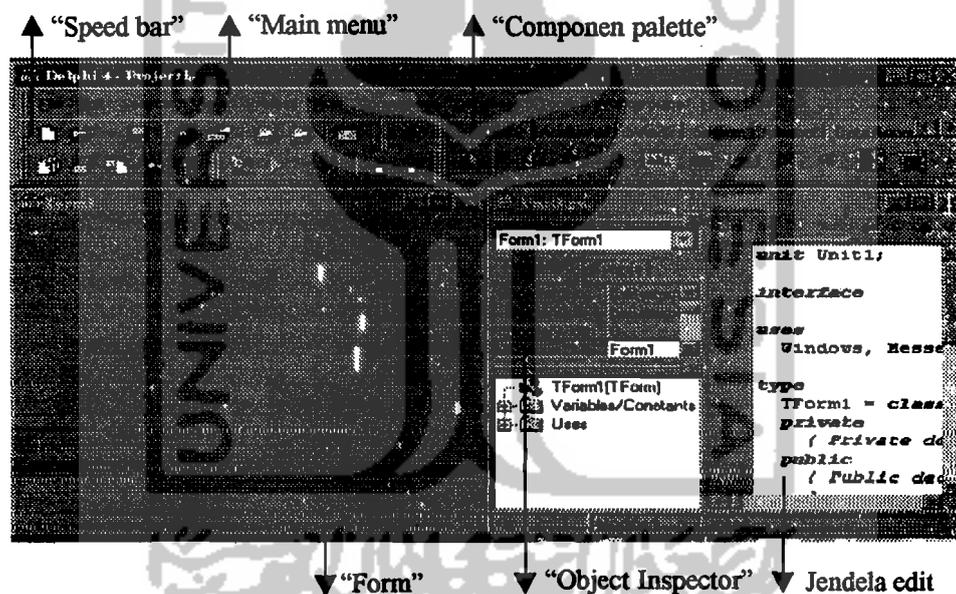
Cara 2, dengan mengklik ganda “shortcut” pada “desktop” Windows 95 atau ikon Delphi yang telah dibuat pada “toolbar” Microsoft Office 97.

Cara 3, dengan memilih dokumen yang pernah dibuka pada menu “Document” pada menu “Start”.

Cara 4, dengan mengklik ganda file proyek Delphi (berekstensi *.dpr) dari “Windows Explorer”.

2.3.4 Istilah pada Layar

Setelah Delphi 4 aktif, pada layar monitor akan terlihat seperti dibawah ini :



Gambar 2.32 Istilah pada layar

1. “Main Menu”

“Main menu” terdiri dari dua komponen yaitu “menu bar” dan “tittle bar”.

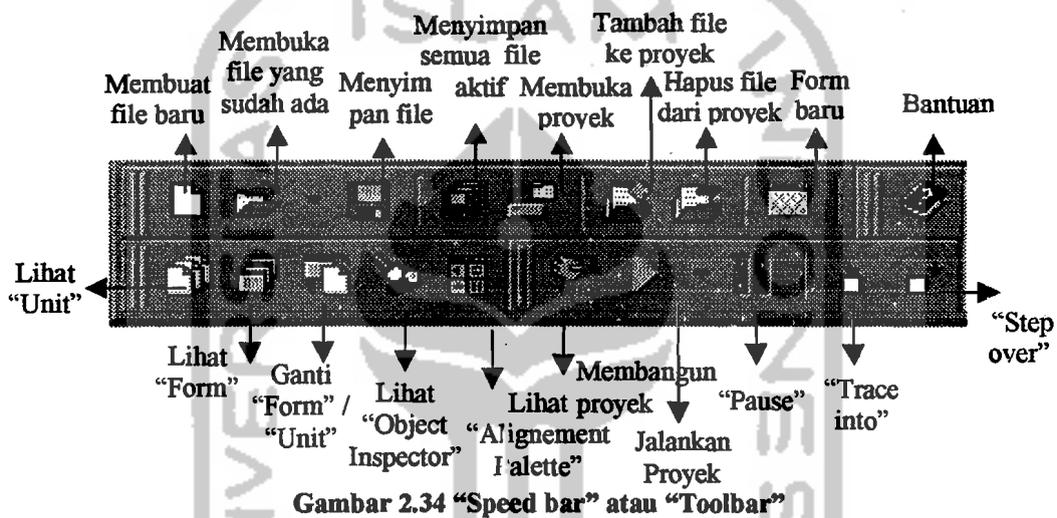
“Menu bar” menampilkan menu berisi perintah-perintah pada Borland Delphi 4, sedangkan “tittle bar” menampilkan judul proyek Delphi yang sedang dikerjakan.



Gambar 2.33 "Main menu"

2. "Speed Bar"

"Speed bar" atau "Toolbar" berguna untuk menampung "icon" untuk mengakses perintah dengan mudah dan cepat.



Gambar 2.34 "Speed bar" atau "Toolbar"

3. "Component Palette"

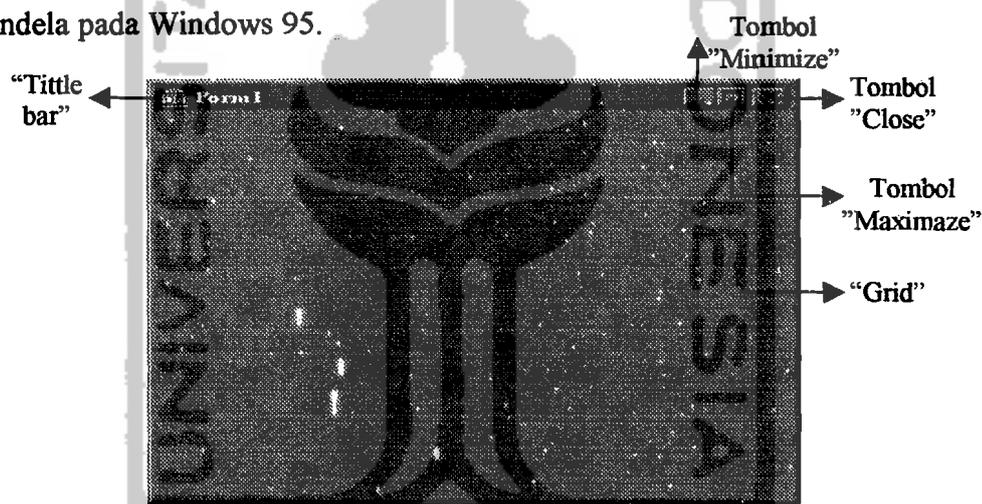
"Component Palette" menampung kumpulan komponen yang siap pakai untuk membangun sebuah proyek, yang biasa disebut sebagai "VCL" ("Visual Component Library"/Pustaka Komponen Visual). Komponen visual tersebut dikelompokkan menurut kegunaan masing-masing. Dalam Delphi 4 ini komponen visual tersebut dibagi kedalam 14 kelompok, yaitu : "Standard", "Additional", "Win32", "System", "Internet", "Data Access", "Data Controls", "Midas", "Decision Cube", "Qreport", "Dialogs", "Win31", "Sample" dan "ActiveX".



Gambar 2.35 "Component Palette"

4. "Form"

"Form" adalah lembar disain tampilan program ("User interface") dari aplikasi yang akan dibuat. Pada hakekatnya "Form" ini merupakan tempat menampung komponen visual Delphi yang dirancang sedemikian rupa, baik letak maupun ukurannya sesuai selera pemrogram. Bentuk dan penggunaan "Form" Delphi ini sesuai dengan standar jendela pada Windows 95.

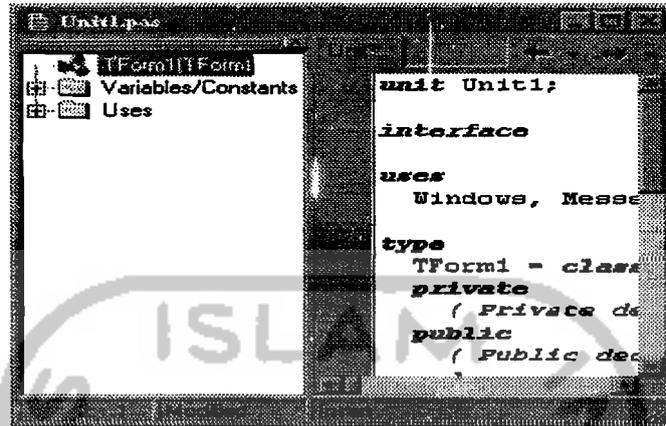


Gambar 2.36 "Form"

5. Jendela Edit

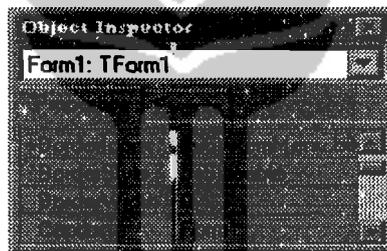
Jendela edit merupakan tempat menuliskan rutin program aplikasi yang dibuat dengan Delphi. Bagian bawah jendela ada 3 buah kotak keterangan. Kotak paling kiri menunjukkan posisi kursor pada jendela edit, kotak tengah berisi kata "modified" yang menunjukkan rutin program sudah diubah tapi belum disimpan. Setelah perubahan tersebut disimpan ke disket atau harddisk maka kata "modified" tersebut

akan hilang. Sedangkan kotak paling kanan menunjukkan mati atau hidupnya tombol Insert.



Gambar 2.37 Jendela Edit

6. “Object Inspector”



Gambar 2.38 “Object Inspector”

“Object Inspector” adalah kotak yang berisi “properti” dan “event” dari sebuah obyek atau komponen. Bagian “properti” untuk mengatur properti sebuah obyek, sedangkan bagian “event” untuk mengisi rutin program sebagai respon obyek tersebut terhadap suatu kejadian (“event”).

2.3.5 Hal-hal Baru pada Delphi 4

Hal-hal baru pada Delphi 4 ini mencakup perbaikan dari versi terdahulu, maupun hal yang memang baru ada pada versi 4 ini. Dibawah ini hal baru tersebut antara lain :

1. Perluasan Kemampuan Bahasa Pemrograman

Delphi 4 memperluas obyek bahasa Pascal antara lain : array yang dinamis, metoda “overloading” dan parameter standar.

2. “Project Manager”

“Project Manager” memungkinkan untuk mengorganisir beberapa proyek terpisah menjadi satu kesatuan yang terpadu.

3. “Code Explorer” dengan “Class Completion”, “Module Navigation” dan “Code Browser”

- a. “Code Explorer” mempermudah pembuatan kelas dengan otomatisasi banyak langkah.
- b. “Class Completion” akan membuatkan rangka kode pada seksi “implementation”.
- c. “Module Navigation” akan memandu perpindahan yang cepat didalam unit antara seksi “interface” dan “implementation”.
- d. “Code Browser” akan membantu melompat ke bagian deklarasi variabel/”identifier” yang dituju.

4. Jendela Alat/”Tool Window” yang bisa digabung.

IDE (“Integrated Development Environment”/Lingkungan Pengembangan Terpadu) dari Delphi lebih mudah di konfigurasi dengan dapat dilepas dan

digabungnya Jendela alat misalnya "Project Manager" dan "Code Explorer" dengan "Code Editor"/ Jendela Edit.

5. "Debugging" yang lebih dikembangkan.

"Debugger" terpadu memiliki banyak wajah baru, termasuk debugging multi proses dan terpisah, tampilan CPU, inspektor, peningkatan "BreakPoint", submenu spesifik "debugger" dan jendela yang bisa dilepas-gabung ("dockable").

6. Peningkatan "ActiveX".

Delphi dilengkapi dengan dukungan terhadap "ActiveX" yang diperluas.

7. Peningkatan VCL ("Visual Component Library").

Hirarki obyek Delphi telah ditingkatkan termasuk komponen baru untuk pembuatan aplikasi pada Windows NT. Sebuah komponen baru yaitu "Action list" memberi kemungkinan mengorganisir secara terpusat semua perintah pada menu dan tombol. Peningkatan VCL pada kemampuan "drag-and-drop" dalam rangka melengkapi kontrol tambahan pada "lay-out" jendela.

8. Dukungan RTL untuk mengatasi masalah tahun 2000.

Variabel global "TwoDigitYearCenturyWindow" akan digunakan oleh fungsi "StrToDate" dan "StrToDateTime" untuk mengontrol interpretasi dua digit tahun ketika mengkonversi tanggal.

9. Peningkatan Database.

Komponen "Data Access" telah ditingkatkan untuk dapat diubah pada saat pembuatan "form" atau modul data. Perubahan ini juga memungkinkan

untuk mengakses data dari tipe server database baru seperti MS Access 97 atau Oracle 8.

2.3.6 Istilah Pada Pemrograman

Cukup banyak istilah baru dalam pemrograman visual, antara lain adalah :

1. "Event"

"Event" adalah aktivitas yang terjadi akibat adanya tindakan dari pemakai, pengaturan oleh timer, atau oleh aplikasi itu sendiri ketika sebuah aplikasi digunakan. Contoh "event" adalah penekanan "mouse" atau penekanan tombol.

2. File eksekusi

File eksekusi adalah file program yang telah dikompilasi sehingga menjadi bahasa mesin. File ini dapat berdiri sendiri dan biasanya berakhiran .EXE.

3. Prosedur dan Fungsi

Prosedur adalah rutin program yang ditulis dalam suatu blok, yang mewakili tugas tertentu. Sedangkan Fungsi adalah prosedur yang lebih spesifik, yaitu khusus untuk perhitungan matematik karena bersifat mengembalikan nilai.

4. "Icon"

"Icon" adalah gambar kecil biasanya berukuran 32 x 32 piksel yang mewakili aplikasi atau perintah tertentu .

5. Komponen Visual

Komponen visual adalah sebuah obyek yang bisa diletakkan pada “form” untuk membentuk tampilan program aplikasi.

6. Obyek

Obyek adalah bagian terkecil dari sebuah aplikasi. Contoh obyek adalah “Form” dan komponen visual.

7. Properti

Properti adalah bagian yang membangun sebuah obyek. Sebagai contoh : Sepeda motor adalah sebuah obyek. Sepeda motor tersebut mempunyai properti jenis, tahun pembuatan, besar cc, warna dan sebagainya.

8. Proyek

Proyek adalah kumpulan file yang membentuk sebuah program aplikasi.

9. “Unit”

“Unit” adalah sekelompok prosedur dan fungsi yang sejenis.

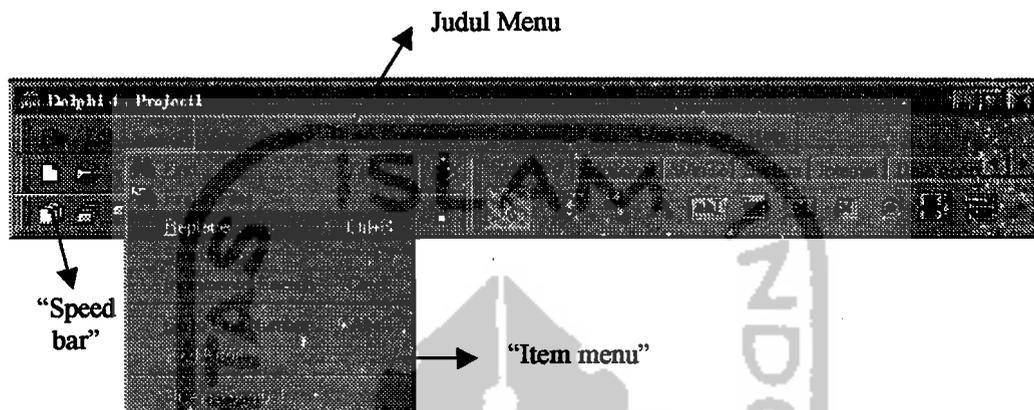
10. “User Interface”

“User Interface” adalah tampilan program yang berhubungan langsung dengan pemakai. Biasanya terdiri dari komponen visual yang disediakan bagi pemakai, misalnya menu, tombol, grafik dan sebagainya.

11. Jendela/ “Window”

Jendela adalah sebuah kotak persegi empat yang berisi tampilan aplikasi atau bagian dari aplikasi. Jendela ini bisa digeser, dibesar-kecil kan bahkan ditutup.

2.3.7 Menu Pada Borland Delphi 4

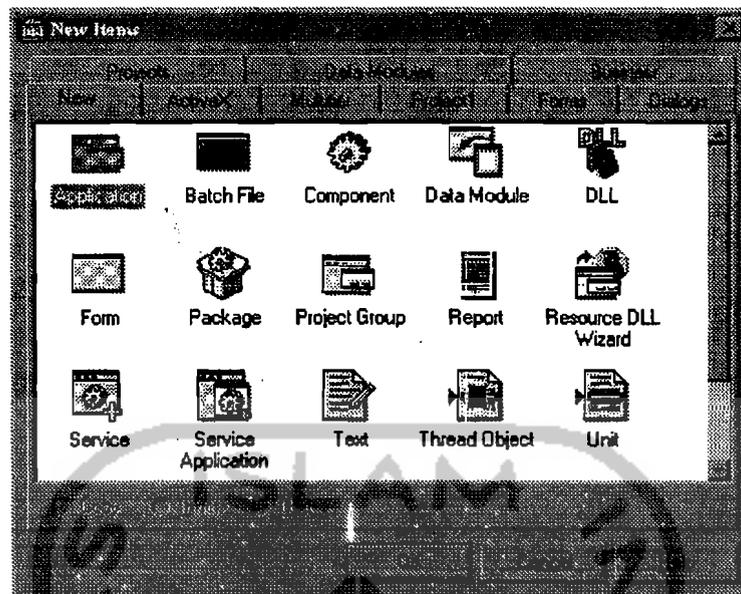


Gambar 2.39 Menu pada Borland Delphi 4

Menu pada Borland Delphi 4 ada 11, yaitu “File”, “Edit”, “Search”, “View”, “Project”, “Run”, “Component”, “Database”, “Tools”, “WorkGroups” dan “Help”. Untuk lebih jelasnya bisa dilihat pada gambar 2.39.

1. Menu “File”

a. “New”, digunakan untuk membuat file baru, sesuai dengan keperluan aplikasi. Delphi menyediakan banyak pilihan untuk membuat file baru ini, seperti terlihat pada gambar dibawah ini :



Gambar 2.40 Kotak dialog "New Items"

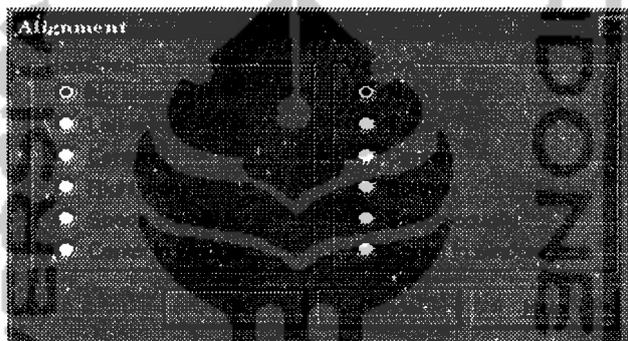
- b. "New Application", item menu ini berfungsi untuk membuat aplikasi/proyek yang baru. Apabila ketika menu ini diklik ada proyek yang sedang dikerjakan, maka Delphi akan meminta konfirmasi untuk menyimpan proyek tersebut.
- c. "New Form", item menu ini berguna untuk membuat "form" baru, untuk ditambahkan pada aplikasi yang sedang dikerjakan.
- d. "Open", item menu ini dipakai untuk membuka file yang sudah ada, baik itu file proyek maupun file "unit".
- e. "Open Project", digunakan untuk membuka file proyek yang sudah ada. Biasanya mempunyai ekstensi *.dpr.
- f. "Reopen", untuk membuka secara cepat file proyek atau file "unit" yang sudah pernah dibuka, pada daftar file yang tersedia.
- g. "Save", untuk menyimpan file/modul yang sedang dikerjakan.

- h. "Save As", menyimpan file/modul yang sedang dikerjakan dengan nama lain/ baru.
- i. "Save Project As", menyimpan proyek yang sedang dikerjakan dengan nama proyek yang baru.
- j. "Save All", menyimpan seluruh file yang terbuka saat itu.
- k. "Close", menutup file "unit" atau modul yang sedang dikerjakan.
- l. "Close All", menutup semua file yang terbuka saat itu termasuk file proyek.
- m. "Use Unit", untuk menambahkan sebuah "unit" pada proyek yang sedang dikerjakan, tanpa harus mengetik nama "unit" tersebut secara manual.
- n. "Print", untuk mencetak "form" atau item yang telah dipilih dari "code editor"
- o. "Exit", keluar dari Delphi, apabila ada file yang terbuka, maka Delphi akan meminta konfirmasi agar file tersebut disimpan.

2. Menu "Edit"

- a. "Undo", untuk membatalkan perintah yang terakhir di eksekusi.
- b. "Redo", kebalikan dari "Undo", yaitu mengembalikan keadaan satu langkah kedepan.
- c. "Cut", memotong obyek atau teks yang telah dipilih kedalam "Clipboard".
- d. "Copy", menggandakan obyek atau teks yang telah dipilih.
- e. "Paste", memindahkan isi "clipboard" pada posisi kursor.
- f. "Delete", menghapus obyek atau teks yang telah dipilih.
- g. "Select All", memilih semua komponen pada "form" atau seluruh teks pada jendela edit.

- h. "Align To Grid", untuk meletakkan tepi terluar dari komponen visual pada grid "form" terdekat.
- i. "Bring To Front", untuk meletakkan obyek pada lapisan layar paling depan.
- j. "Send To Back", untuk meletakkan obyek pada lapisan layar paling belakang.
- k. "Align", untuk meluruskan letak komponen visual yang ada pada "form" baik secara vertikal maupun horisontal dengan pilihan yang tersedia, seperti pada gambar berikut ini :



Gambar 2.41 Kotak dialog "Alignment"

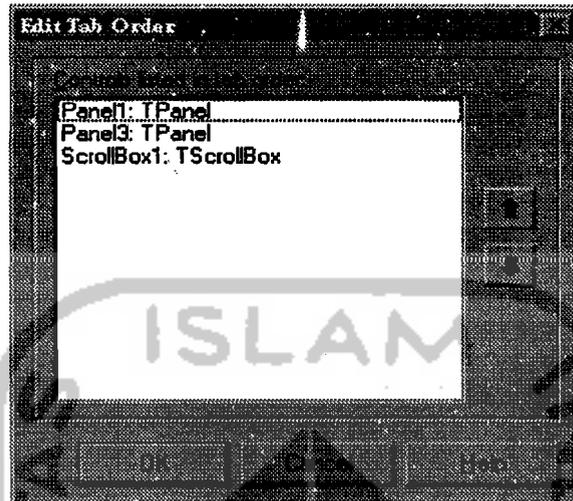
- l. "Size", untuk mengubah ukuran komponen visual sejenis yang dipilih, dengan mengisi dialog seperti dibawah ini :



Gambar 2.42 Kotak dialog "Size"

- m. "Scale", untuk mengubah ukuran komponen visual secara proporsional dengan mengisi prosentase besar terhadap ukuran semula.

- n. “Tab Order”, untuk menampilkan kotak dialog “Tab Order” yang berfungsi untuk mengubah urutan komponen.

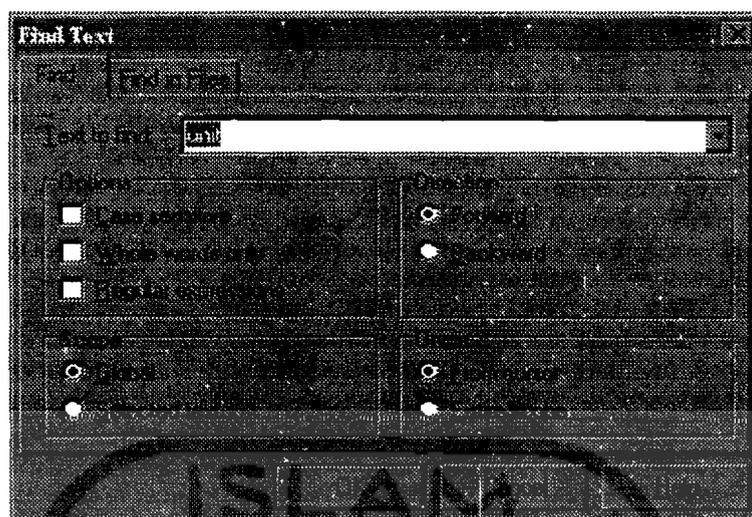


Gambar 2.43 Kotak dialog “Edit Tab Order”

- o. “Creation Order”, untuk mengatur urutan komponen non visual yang telah dibuat.
- p. “Flip Children”, untuk mengubah susunan komponen visual yang terpilih pada form dengan mengubah letaknya secara pencerminan dari kiri kekanan atau sebaliknya.
- q. “Lock Controls”, untuk mengunci posisi obyek pada “form” sehingga tidak bisa diubah lagi.
- r. “Add To Interface”, untuk mendefinisikan sebuah prosedur, fungsi atau properti baru yang akan ditambahkan pada tampilan (“interface”) komponen “ActiveX”.

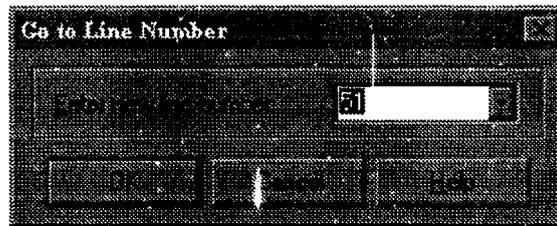
3. Menu “Search”

- a. “Find” untuk mencari sebuah kata pada jendela edit dengan mengisi kotak dialog seperti gambar dibawah ini :



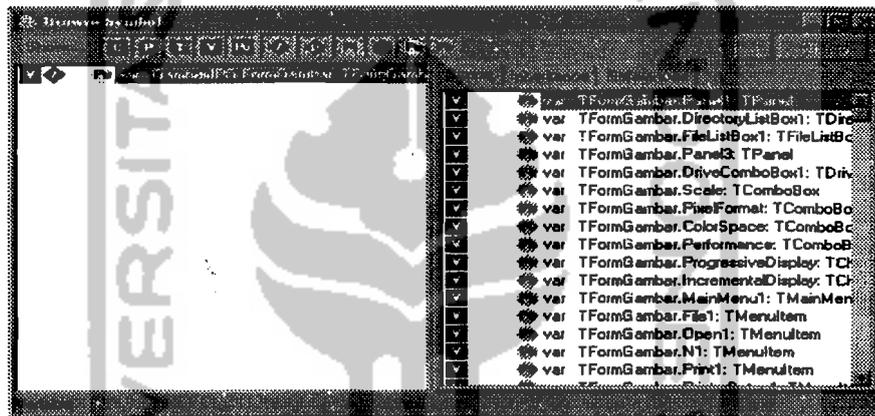
Gambar 2.44 Kotak Dialog "Find Text"

- b. "Find In Files", untuk menampilkan seluruh kata/ "string" yang dimaksud pada file yang dicari. Untuk berpindah pada lokasi "string" tersebut, bisa dilakukan dengan mengklik kata yang dimaksud yang terdapat pada bagian bawah jendela edit.
- c. "Replace", untuk mencari sebuah kata pada jendela edit, kemudian diganti dengan kata yang lain.
- d. "Search Again", untuk mencari lagi kata yang sudah ditentukan oleh pilihan "Find".
- e. "Incremental search", untuk mencari sebuah kata pada jendela edit tanpa mengisi kotak dialog, tapi dengan mengetik kata tersebut huruf demi huruf langsung pada jendela edit.
- f. "Goto Line Number", untuk membawa kursor pada jendela edit ke baris yang diisikan pada kotak dialog "Go To Line Number".



Gambar 2.45 Kotak Dialog “Go To Line Number”

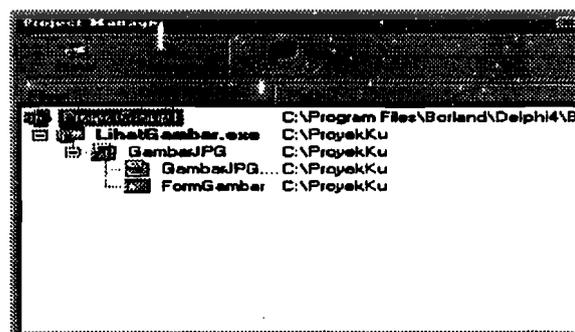
- g. “Find error”, digunakan untuk mencari kesalahan “run-time error”.
- h. “Browse Symbol”, untuk mencari simbol pada aplikasi yang telah dikompilasi atau dijalankan.



Gambar 2.46 Kotak “Browse Symbol”

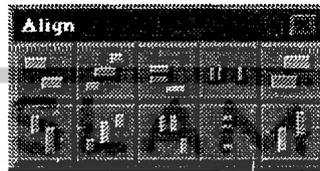
4. Menu “View”

- a. “Project Manager”, untuk menampilkan seluruh modul yang menyusun sebuah proyek Delphi.



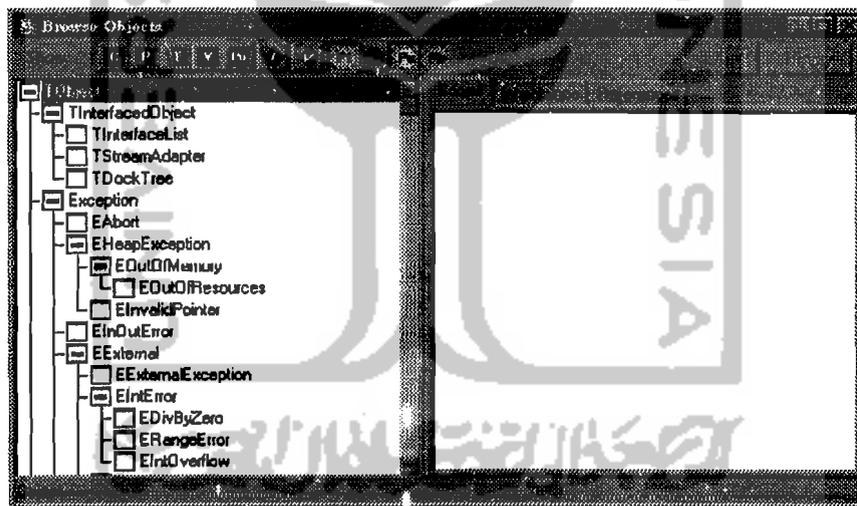
Gambar 2.47 “Project Manager”

- b. "Object Inspector", untuk menampilkan "Object Inspector" yang berfungsi mengatur properti dan "event" pada sebuah komponen visual.
- c. "Alignment Palette", untuk menampilkan kotak dialog "Align" yang berguna untuk mengatur perataan komponen Visual pada "form". Ini adalah versi visual dari pilihan menu "Edit", "Align".



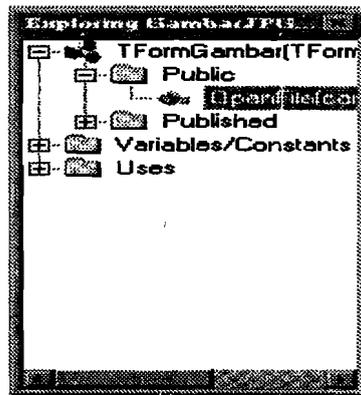
2.48 Kotak "Align"

- d. "Browser", untuk menampilkan hubungan antar obyek yang dipakai untuk menyusun sebuah aplikasi.



Gambar 2.49 Kotak "Browse Object"

- e. "Code Explorer", untuk mempermudah navigasi dalam file unit dan otomatisasi pembuatan kelas. Biasanya tergabung disebelah kiri jendela edit.



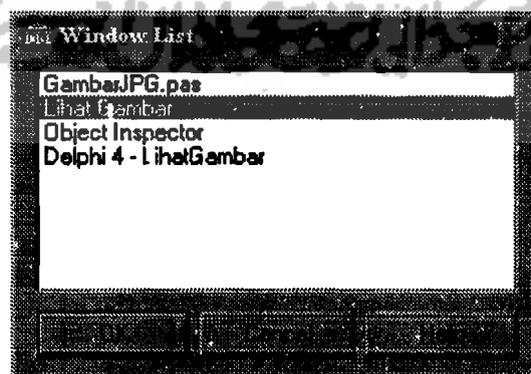
Gambar 2.50 "Code Explorer"

- f. "Component List", untuk melihat daftar komponen yang disediakan Delphi untuk membangun sebuah proyek atau aplikasi.



Gambar 2.51 Kotak "Component"

- g. "Windows List", untuk menampilkan daftar jendela yang terbuka.



Gambar 2.52 "Windows List"

- h. "Debug Window", untuk menampilkan informasi yang berguna untuk melacak kesalahan atau "error" yang terjadi pada saat sebuah aplikasi atau program dijalankan. Item menu ini mempunyai 8 buah sub menu yaitu : "Break Points", "Call Stack", "Watches", "Local Variable", "Thread", "Modules", "CPU" dan "Event Log".
- i. "Toggle Form Unit", untuk berpindah dari tampilan "Form" ke jendela edit dan sebaliknya.
- j. "Units", untuk menampilkan daftar "unit" dari proyek yang sedang dikerjakan.
- k. "Forms", untuk menampilkan daftar "Form" yang terdapat pada proyek yang sedang dikerjakan.
- l. "Type Library", untuk menampilkan jendela "Type Library Editor" yang berguna untuk menguji dan membuat tipe informasi kontrol ActiveX, Otomasi server, obyek MTS ("Microsoft Transaction Server") dan obyek COM ("Component Object Model") lainnya.
- m. "New Edit Window", untuk menampilkan jendela edit yang baru tanpa menutup jendela edit yang lama.
- n. "Toobars", untuk memilih "toolbar" yang akan terlihat atau tersembunyi.

5. Menu "Project"

- a. "Add To Project", untuk menambahkan "unit" baru kedalam proyek.
- b. "Remove To Project", untuk membuang "unit" yang dipilih dari sebuah proyek, dan secara otomatis menghapus nama unit dibawah "uses".

- c. "Import Type Library", untuk melihat daftar pustaka/"library" yang telah terdaftar pada sistem, untuk dapat ditambahkan pada proyek yang sedang dibuat.
- d. "Add To Repository", untuk menambahkan "form" yang telah dibuat pada "Object Repository", agar bisa digunakan kembali pada proyek yang lain.
- e. "View Source", untuk menampilkan kode program dari file proyek yang aktif pada "Code Editor".
- f. "Add New Project", untuk menambahkan item baru seperti aplikasi baru, DLL ("Dynamic Link Library"), atau "Packages" dalam kelompok proyek yang sedang dikerjakan.
- g. "Add Existing Project", untuk membuka proyek yang sudah jadi untuk ditambahkan pada kelompok proyek yang sedang dikerjakan.
- h. "Compile Project", untuk mengkompilasi semua unit dan modul yang telah diubah sejak saat terakhir kali program dijalankan.
- i. "Build Project", untuk mengkompilasi semua file pada kelompok proyek, kecuali yang tidak mengalami perubahan sejak terakhir kali dijalankan.
- j. "Syntax Check Project", untuk memeriksa tata penulisan rutin pada proyek yang sedang dikerjakan tanpa harus menjalankan proyek tersebut.
- k. "Information For [None]", untuk membuka kotak dialog informasi yang berisi informasi kompilasi dan status kompilasi dari proyek yang telah dijalankan.
- l. "Compile All Project", untuk mengkompilasi semua file yang terdapat pada kelompok proyek, yang telah berubah sejak terakhir kali dieksekusi.

- m. "Build All Project", untuk membangun kembali semua komponen pada proyek yang telah mengalami perubahan, sejak eksekusi terakhir.
- n. "Web Deployment Options", untuk mengkonfigurasi sebuah komponen "ActiveX" yang sudah jadi atau form aktif untuk dikirim ke "server Web".
- o. "Web Deploy", untuk mengirim "ActiveX" yang sudah jadi dibuat pada form aktif ke "server Web".
- p. "Options", untuk menampilkan kotak dialog "Project Options" yang berguna untuk mengatur pilihan kompiler, "linker", "Packages", "Version Info" dan direktorinya.

6. Menu "Run"

- a. "Run", untuk menjalankan program atau aplikasi yang sudah selesai dibuat.
- b. "Parameters", untuk memasukkan parameter yang disediakan aplikasi, ketika dijalankan.
- c. "Register ActiveX Server", untuk mendaftarkan proyek ke registri windows. Menu ini bisa dilakukan hanya untuk proyek "ActiveX".
- d. "Unregister ActiveX Server", untuk menghapus proyek dari registri. Item menu ini juga hanya bisa dilakukan hanya untuk proyek "ActiveX".
- e. "Install MTS object", untuk menginstal proyek yang sedang aktif kedalam "package" MTS. Item menu ini hanya bisa dieksekusi untuk proyek MTS.
- f. "Step Over", untuk mengeksekusi program baris demi baris, tetapi tidak akan menjalankan prosedur atau fungsi yang dipanggil baris tersebut.
- g. "Trace Into", untuk mengeksekusi perintah dalam program baris demi baris, termasuk prosedur dan fungsi yang dipanggil dari baris tersebut.

- h. "Trace To Next Source Line", untuk melihat baris mana yang akan dieksekusi kemudian.
- i. "Run To Cursor", untuk menjalankan proyek sampai dengan posisi kursor pada jendela edit.
- j. "Show Execution Point", digunakan untuk kembali ke jendela edit dan melihat baris selanjutnya yang akan dijalankan pada waktu "debugging".
- k. "Program Pause", untuk menghentikan eksekusi program sampai tombol "Run" ditekan kembali.
- l. "Program Reset", untuk menghentikan eksekusi program, dan menjalankannya dari awal lagi.
- m. "Inspect", untuk membuka jendela inspeksi dan memasukkan item yang akan diperiksa atau diselidiki.
- n. "Evaluate/Modify", untuk memodifikasi nilai sebuah variabel pada saat eksekusi program, dan akan langsung dievaluasi setelah tombol "evaluate" pada kotak dialog ditekan.
- o. "Add watch", membuka kotak dialog "Watch Properties" yang berguna untuk membuat dan memodifikasi pengawasan ("Watches").
- p. "Add Breakpoint", untuk membuka kotak dialog "Edit Break Points", dan membuat atau memodifikasi "break point".

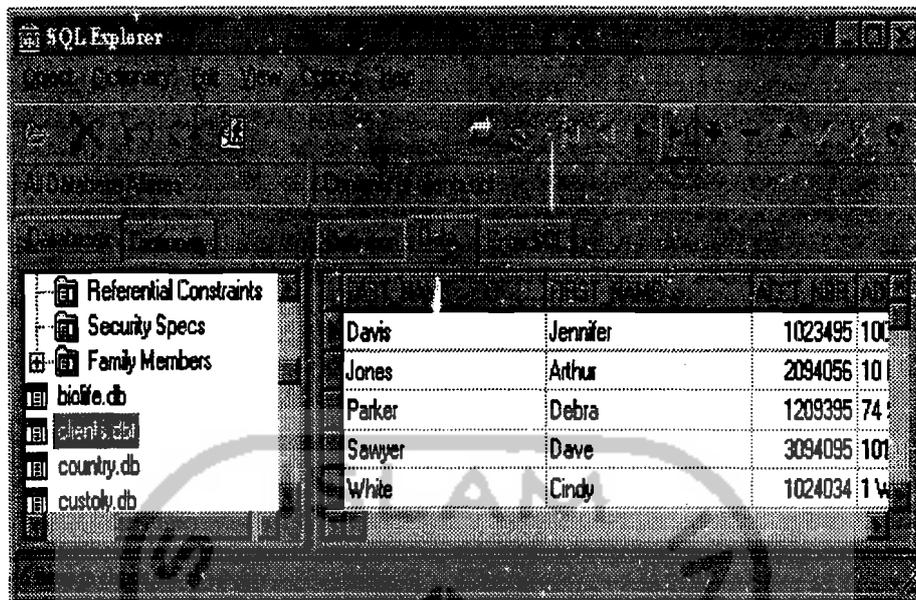
7. Menu "Component"

- a. "New Component", untuk membuat komponen baru bagi proyek yang sedang dikerjakan.

- b. "Install Component", untuk menambahkan komponen pada paket komponen yang baru atau yang sudah ada.
- c. "Import ActiveX Control", untuk menambahkan kontrol "ActiveX" dari luar Delphi sebagai komponen baru. Biasanya file "ActiveX" ini berekstensi OCX.
- d. "Create Component Template", digunakan untuk membuat komponen baru dari sebuah komponen visual standar.
- e. "Install Packages", untuk menentukan paket mana yang akan ditampilkan pada IDE ("Integrated Development Environment") pada waktu mendisain program dan paket "run-time" yang tersedia waktu program tersebut dijalankan. Biasanya mempunyai ekstensi BPL.
- f. "Configure Palette", membuka kotak dialog "Environment Options" yang berguna untuk menyusun (menambah atau mengurangi) kelompok VCL pada komponen palet yang terlihat.

8. Menu "Database"

- a. "Explore", untuk membuka jendela "SQL Explorer" yang berguna untuk mengamati, membuat atau mengedit data dari file database yang ada didalam komputer.

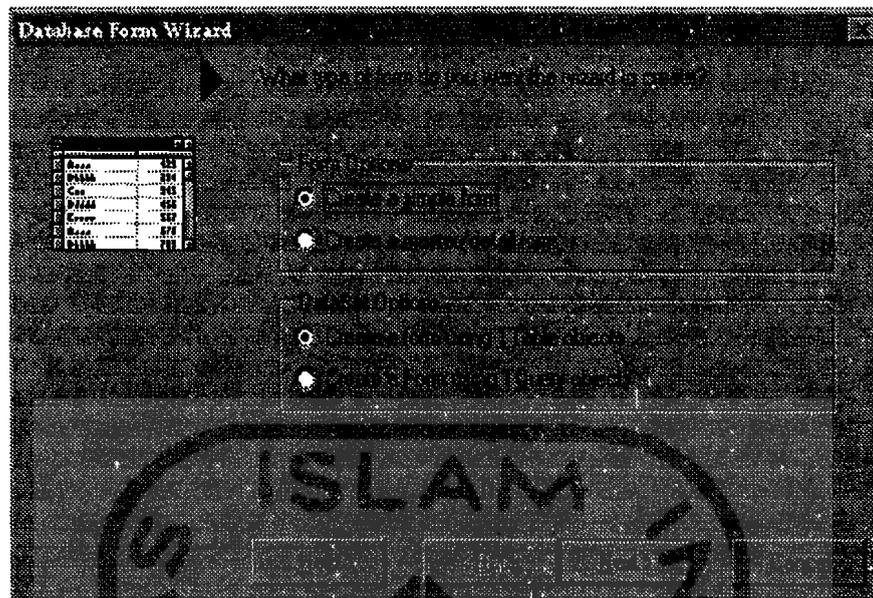


The screenshot shows the SQL Explorer interface. On the left, there is a tree view with the following items: Referential Constraints, Security Specs, Family Members, biofile.db, clients.db, country.db, and custody.db. The main window displays a table with the following data:

Davis	Jennifer	1023495	100
Jones	Arthur	2094056	101
Parker	Debra	1209395	74
Sawyer	Dave	3094095	101
White	Cindy	1024034	14

Gambar 2.53 “SQL Explorer”

- b. “SQL Monitor”, untuk memonitor penempatan sumberdaya SQL (“Structured Query Language”) dan mengakses informasi SQL lainnya untuk sebuah aplikasi.
- c. “Form Wizard”, untuk membuat “Form” yang berisi data dari database lokal atau yang jauh. Proses pembuatan “Form” akan dituntun oleh “Wizard” langkah demi langkah.



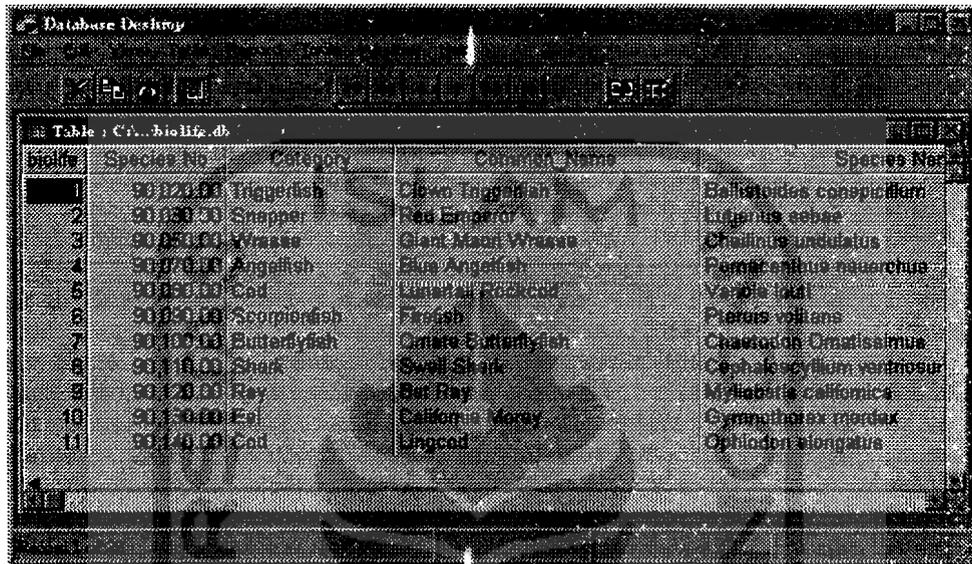
Gambar 2.54 “Database Form Wizard”

9. Menu “Tools”

- a. “Environment Options”, untuk mengatur setting editor, browser dan konfigurasi yang disukai serta menyusun tampilan kelompok komponen pada palet komponen.
- b. “Debugger Options”, menampilkan kotak dialog “Debugger Options” yang berisi 4 buah halaman yaitu : “General” , “Event Log”, “Language Exceptions” dan “OS Exceptions”.
- c. “Repository”, menampilkan kotak dialog “Object Repository” untuk menambah, menghapus atau mengganti nama komponen siap pakai yang terdapat dalam kotak dialog “New Items”.
- d. “Configure Tools”, untuk menambah, mengedit atau menghapus program bantu yang muncul pada menu “Tools”. Sebagai standar Delphi akan

menampilkan 3 buah program bantu yaitu : “ Database Desktop” , “Package Collection Editor” dan “Image Editor”.

- e. “Database Desktop” , menampilkan program bantu “Database Desktop” yang berguna untuk membuat, memodifikasi file database.

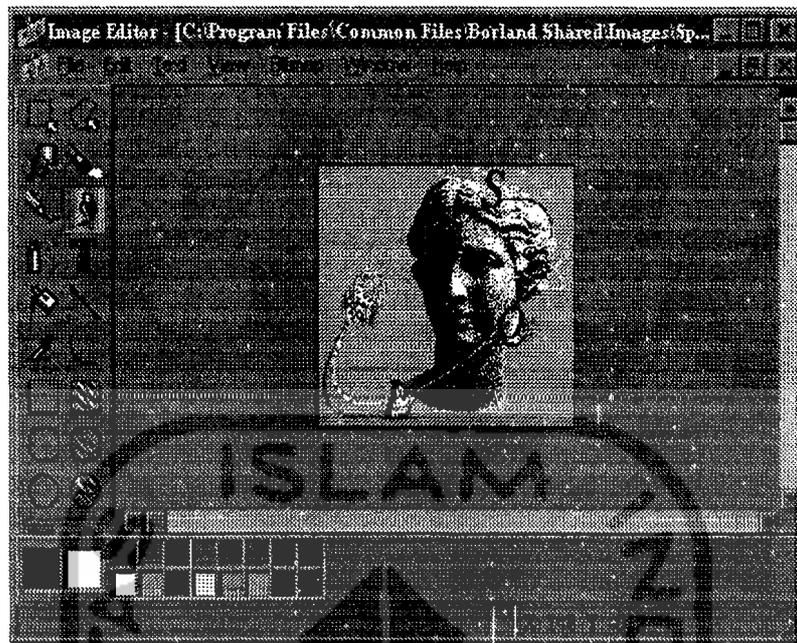


The screenshot shows a window titled "Database Desktop" with a table of biological data. The table has five columns: "Index", "Species No.", "Category", "Common Name", and "Species No.". The data is as follows:

Index	Species No.	Category	Common Name	Species No.
1	00120.00	Triggerfish	Clown Triggerfish	Balloonfish conspicillum
2	00130.00	Snapper	Red Emperor	Eugenia nobis
3	00150.00	Wrasse	Giant-Mouth Wrasse	Chelodactylus unicolor
4	00170.00	Angelfish	Blue Angelfish	Pomacentrus nauruetum
5	00180.00	Cod	Japanese Rockcod	Yaple laai
6	00190.00	Scorpaenid	Flouise	Pteris volans
7	00100.00	Butterflyfish	Orange Butterflyfish	Chaetodon ornatissimus
8	00110.00	Shark	Spotted Shark	Cetorhinus maximus
9	00120.00	Ray	Devil Ray	Megachasma californicum
10	00130.00	Kel	California Moray	Synbranchia morax
11	00140.00	Cod	Langcod	Ophiodon elongatus

Gambar 2.55 “Database Desktop”

- f. “Packages Collection Editor”, untuk menampilkan program bantu “Packages Collection Editor” yang berguna untuk membuat dan mengedit koleksi paket. Koleksi paket adalah cara yang menyenangkan untuk membundel paket dan file-file yang terkait untuk didistribusikan ke pengembang (“Developer”) lainnya.
- g. “Image Editor”, untuk mengaktifkan program “Image Editor” yang berguna untuk membuat atau memodifikasi file grafis misalnya file bitmap, “icon” dan kursor.



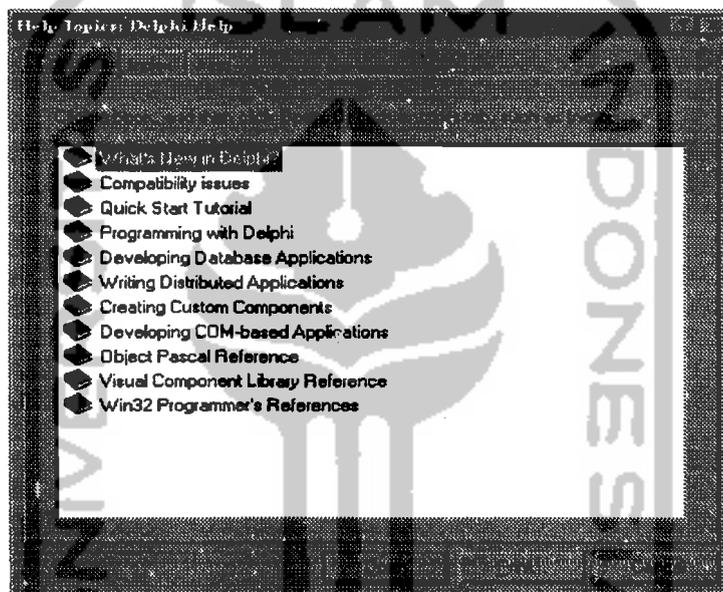
Gambar 2.56 "Image Editor"

10. Menu "WorkGroups"

- a. "Browse PVCS Project", untuk menampilkan jendela "PVCS Project" yang berguna untuk membentuk semua kegiatan tentang kontrol versi dari aplikasi yang sedang dikerjakan.
- b. "Manage Archive Directories", digunakan ketika jendela PVCS sedang terbuka, dan ada proyek yang dipilih, untuk membuat direktori untuk menampung arsip PVCS, direktori arsip tersebut digunakan untuk menyimpan semua perubahan kerja pribadi.
- c. "Add Project To Version Control", untuk membuat file proyek PVCS baru bagi program yang sedang dikerjakan.
- d. "Set Data Directories", untuk menentukan direktori tempat menyimpan file kerja.

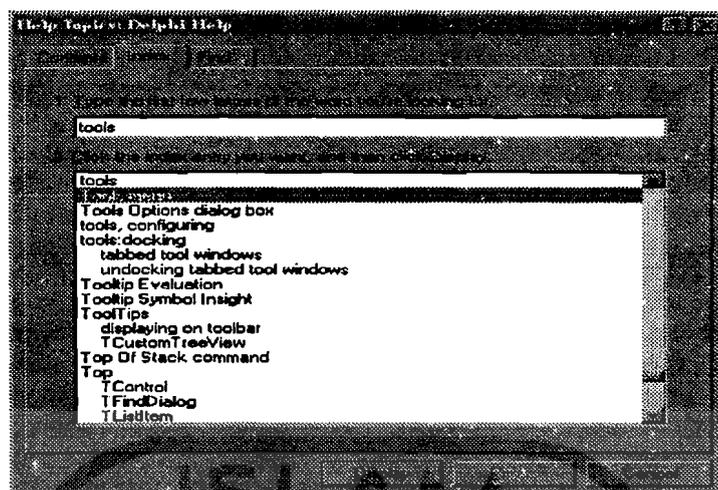
11. Menu “Help”

- a. “Content”, untuk menampilkan topik bantuan Delphi dengan tiga cara, pertama halaman “content” menampilkan urutan yang hirarkis, yang kedua dengan mengklik halaman “index” untuk menampilkan indeks daftar topik secara alfabet dan terakhir dengan mengklik halaman “find” untuk pencarian topik sesuai kebutuhan dengan memasukkan kata kunci.



Gambar 2.57 “Help Topic” pada halaman “Content”

- b. “Index”, digunakan untuk mencari bantuan yang lebih spesifik dengan mengetik kata indeks dari topik yang dicari.



Gambar 2.58 “Help Topic” pada halaman “Index”

- c. “Inprise Home Page”, untuk membuka browser “web” dan langsung menuju situs “Inprise” di WWW (“World Wide Web”).
- d. “Delphi Home Page”, untuk mendapat hubungan dengan “Home Page” Delphi pada situs “Inprise” di WWW.
- e. “Inprise Program And Service”, hubungan langsung dengan halaman “Program and Service” pada situs “Inprise” di WWW, yang menyediakan informasi tentang pendidikan, dukungan teknik, konsultasi dan layanan lain yang ditawarkan oleh “Inprise”.
- f. “Delphi Direct”, hubungan langsung dengan halaman web Delphi untuk, “download” program dan mendapatkan informasi terbaru dari Delphi dan “Inprise”.
- g. “Costumize”, menjalankan program bantu untuk mengatur topik mana yang akan dicantumkan pada “help content” dan “index”.
- h. “About”, menampilkan hak cipta, dan informasi versi dari Delphi.

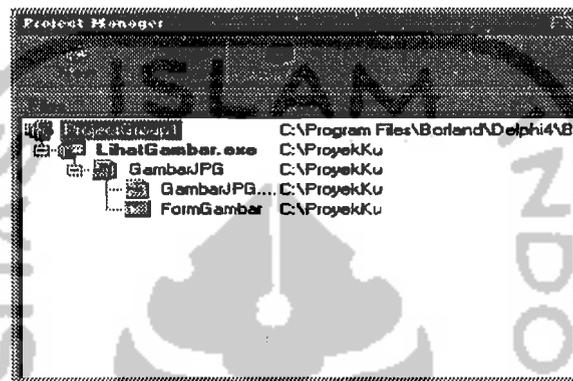
2.3.8 Manajemen Proyek

Sebuah proyek Delphi biasanya terdiri dari beberapa file yang terkait satu dengan lainnya. Ketika proses kompilasi, Delphi juga membuat file-file untuk membangun aplikasi. Dibawah ini jenis file yang terbentuk ketika membangun sebuah aplikasi :

1. File proyek (*.dpr) untuk menyimpan daftar file "unit" dan "form" yang digunakan .
2. File "unit" (*.pas) untuk menyimpan rutin program. Ada file "unit" yang terkait dengan "form", ada juga yang dipakai untuk menyimpan prosedur dan fungsi.
3. File "form" (*.dfm) untuk meyimpan informasi tentang "form".
4. File setting proyek (*.dof) untuk menyimpan informasi tentang setting proyek.
5. File "resource" (*.res) untuk menyimpan informasi tentang file grafis yang digunakan dalam program.
6. File "backup" (*.~dp,*.~df, *.~pa) berisi cadangan dari file proyek,"unit" dan "form".
7. File eksekusi (*.exe) adalah file hasil kompilasi yang merupakan file proyek yang sudah jadi dan bisa berdiri sendiri. Kecuali program yang menggunakan DLL atau komponen ActiveX.
8. File obyek unit (*.dcu) untuk menyimpan file "unit" yang dikompilasi dan akan diubah menjadi file eksekusi.
9. File "dynamic link library" (*.dll). File ini hanya tercipta apabila ada file DLL yang dirancang.

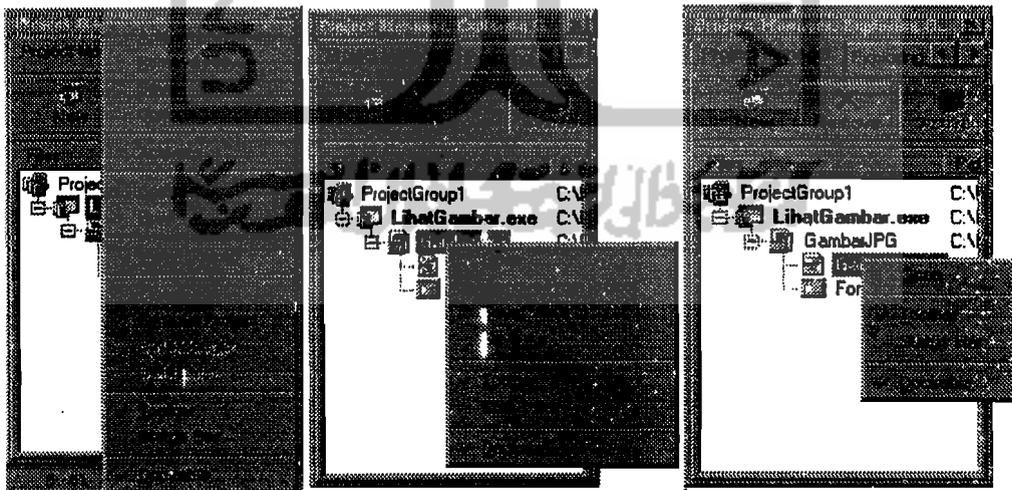
1. Jendela “Project Manager”

“Project Manager” digunakan untuk mengorganisir semua file “unit” dan “form” yang dipakai membangun sebuah aplikasi. File-file tersebut dapat ditambah atau dikurangi sesuai dengan kebutuhan proyek. Jendela ini dapat dibuka melalui menu “View”, “Project Manager”.



Gambar 2.59 “Project Manager”

Pada tiap level pada “Project Manager” apabila diklik kanan akan muncul “Pop-up” menu, untuk memudahkan akses pada tiap file.



Gambar 2.60 Menu “Pop-up” tiap level pada “Project Manager”

2. Modus Pengerjaan Proyek

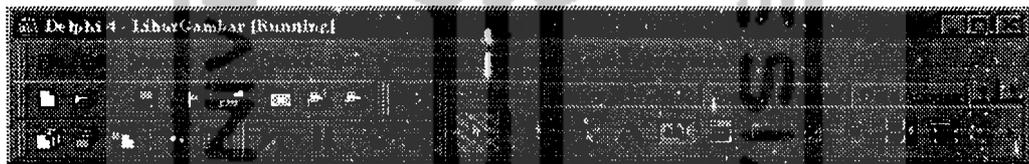
Ada tiga jenis modus pengerjaan ketika membuat sebuah proyek, yaitu : modus disain, “run-time” dan “break”.

- a. Modus disain, adalah modus yang berlaku ketika proyek berada pada tahap rancangan, yaitu merancang “form” dan menuliskan rutin program.



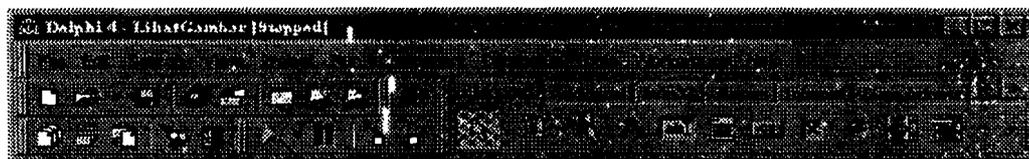
Gambar 2.61 Modus Disain

- b. Modus “run-time”, adalah modus yang berlaku ketika proyek dijalankan (“run”) tapi masih dalam lingkungan Delphi. Pada modus ini rutin program tidak dapat dimodifikasi, hanya bisa dimonitor.



Gambar 2.62 Modus “run-time”

- c. Modus “break”, adalah modus berlaku ketika terjadi penghentian program pada waktu dijalankan (“run-time”). Pada modus ini rutin program bisa dimodifikasi.

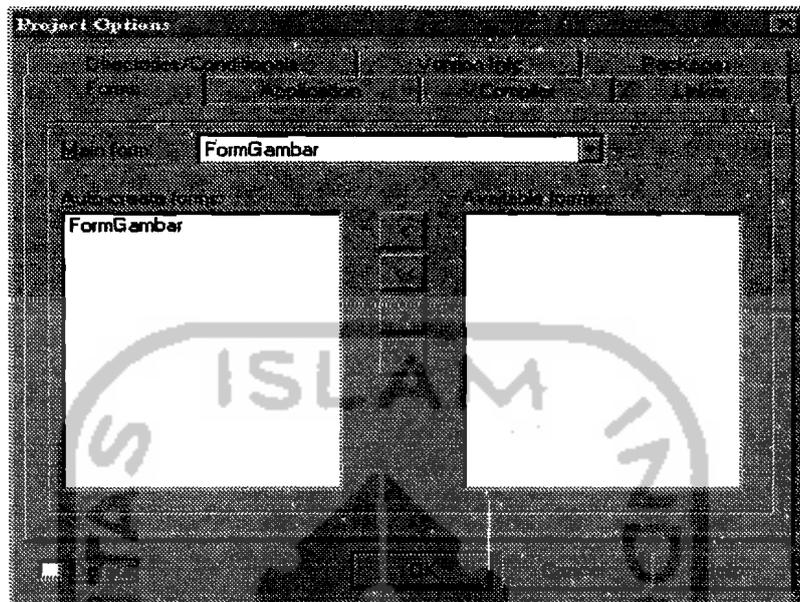


Memulai
“run-time”

Pindah ke
modus “break”

Gambar 2.63 Modus “break”

3. Pilihan “Project Options”



Gambar 2.64 Halaman “Form” pada item menu “Project”| “Options”.

“Project Options” ini digunakan untuk mengatur setting yang berlaku pada proyek yang sedang dikerjakan. “Project Options” ini terbagi menjadi tujuh halaman, yaitu “Form”, “Application”, “Compiler”, “Linker”, “Directories/Conditional”, “Version Info” dan “Packages”.

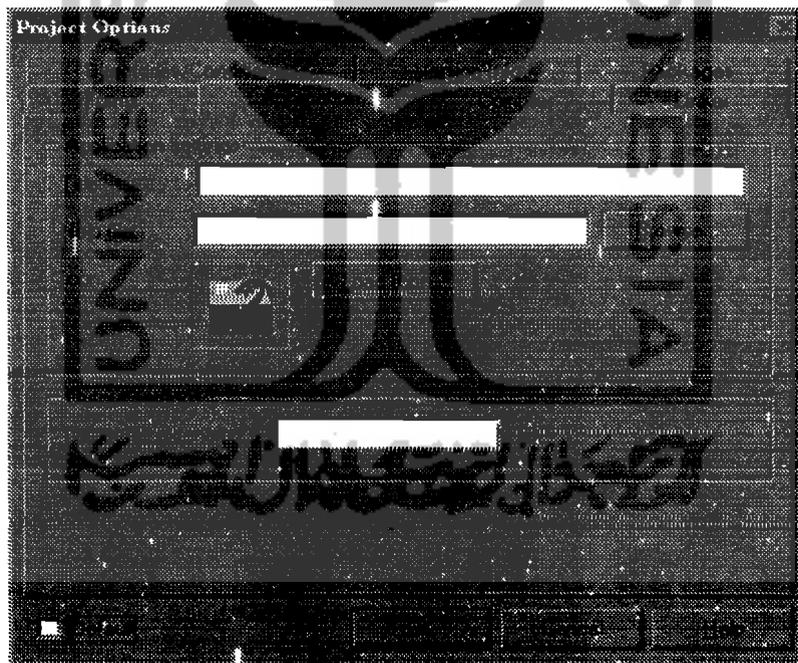
a. “Form”

- 1). Kotak “Main Form” untuk menentukan “form” utama dari program “MDI” (“Multi Document Interface”).
- 2). Kotak “Auto-create forms” untuk menentukan “form” mana yang langsung dibuat ketika program dijalankan.
- 3). Kotak “Avalaible forms” untuk menentukan “form” yang aktif ketika diperlukan saja.
- 4). Kotak “Default” untuk membuat setting tersebut berlaku untuk semua proyek.

b. "Application"

Halaman "Application" mempunyai setting sebagai berikut :

- 1). Kotak "Title" untuk menentukan teks yang muncul pada "Taskbar" ketika program diminimasi.
- 2). Kotak "Help file" untuk menentukan file bantuan yang dipakai aplikasi. Biasanya mempunyai ekstensi *.hlp.
- 3). Kotak "Icon" untuk menentukan "icon" bagi aplikasi. Apabila tidak diubah, akan diberikan "icon" standar Delphi.
- 4). Kotak "Target file extension" untuk menentukan ekstensi file agar bisa bekerja dengan file eksekusi dari program.



Gambar 2.65 Halaman "Application" pada item menu "Project" | "Options"

c. "Compiler"

Pilihan ini untuk mengatur setting “Compiler” dalam mengkompilasi program.

Pilihan ini terbagi dalam beberapa kelompok yaitu “ Code Generation”,

“Runtime Errors”, “Syntax Options”, “Debugging” dan “Messages”.

d. “Linker”

Untuk mengatur cara “Linker” Delphi untuk membuat sebuah program aplikasi.

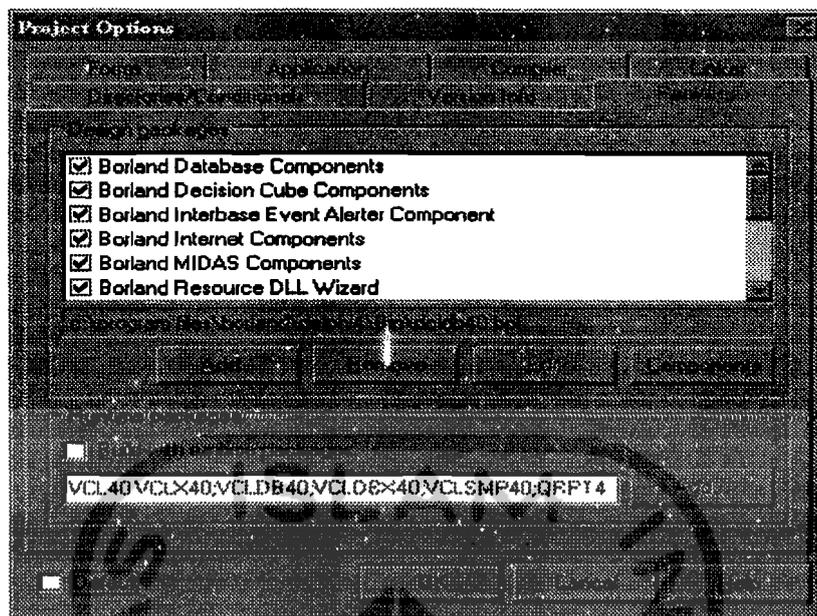
e. “Directories/ Conditionals”

Pilihan ini terbagi menjadi tiga kelompok yaitu : “Directories”, “Conditionals” dan “Aliases”. Kotak “Output directory” digunakan untuk menentukan letak file hasil kompilasi (*.dcu) dan file eksekusi (*.exe) dari program. Apabila tidak diisi, maka kedua file tersebut akan ditempatkan pada direktori tempat file proyek berada.

f. “Version Info”

Untuk menentukan nomor versi dan informasi lain yang ditampilkan pada file eksekusi program. Cara melihat informasi ini dengan mengklik kanan file eksekusi, lalu memilih “Properties”.

g. “Packages”



2.66 Halaman “Packages” dari item menu “Project”/”Options”

Untuk menentukan file paket dan komponen visual yang akan disertakan dalam proyek baik pada modus disain maupun pada modus “runtime”.

2.3.9 Obyek Pascal

1. Konstanta

Konstanta merupakan variabel yang nilai didalamnya selalu tetap. Bisa juga digunakan untuk mempermudah pembacaan program.

Cara mendeklarasikan konstanta sebagai berikut :

Const

```
Nama_konstanta := ekspresi;
```

Dimana ekspresi bisa terdiri dari gabungan bilangan , huruf dan tanda operasi.

Ekpresi yang bertipe data “String” perlu diapit dengan tanda petik (‘ ‘).

2. Variabel

Variabel adalah tempat penampung data yang isinya dapat diubah-ubah.

Cara penulisannya sebagai berikut :

Var

```
Nama_variabel : [jenis tipe data];
```

3. Tipe data

Dalam mendeklarasikan sebuah variabel, biasanya diperlukan penentuan tipe data yang tepat. Hal ini dimaksudkan selain untuk menghemat memori komputer, tetapi juga berkaitan dengan kecepatan eksekusi program.

a. "Simple"

Tipe data ini dibagi menjadi dua yaitu tipe "Ordinal" dan tipe "Integer"

1). "Ordinal"

a). "Integer"

Tipe "Integer" mencakup semua bilangan yang bulat.

Tipe	Jangkauan	Format
Integer	-2147483648..2147483647	Bertanda 32-bit
Cardinal	0..4294967295	Tak bertanda 32-bit
Shortint	-128..127	Bertanda 8-bit
Smallint	-32768..32767	Bertanda 16-bit
Longint	-2147483648..2147483647	Bertanda 32-bit
Int64	$-2^{63}..2^{63}-1$	Bertanda 64-bit
Byte	0..255	Tak bertanda 8-bit

Word	0..65535	Tak bertanda 16-bit
Longword	0..4294967295	

Tabel 2.1 Macam- macam tipe "Integer"

b). "Character"

Tipe dasar "Character" adalah "Char" yang bernilai 8 bit, "ANSI Char" yang bernilai 8 bit dan "Wide Char" yang bernilai 16 bit. Tipe "Character" meliputi semua huruf.

c). "Boolean"

Tipe "Boolean" dipakai untuk menunjukkan nilai "True" (Benar) atau "False" (Salah). Tipe "Boolean" dibagi menjadi 4 macam yaitu : "Boolean" (ukuran 1 byte), "ByteBool" (ukuran 1 byte), "WordBool" (ukuran 2 byte) dan "Long-Bool" (ukuran 4 byte).

d). "Enumerated"

Type "Enumerated" mendefinisikan sekelompok nilai dengan sebuah daftar "identifier"/ pengenal sederhana.

Untuk mendeklarasikan tipe "Enumerated" seperti dibawah ini :

```
type Namatype = (val1, ..., valn);
```

dimana Namatype, val1 sampai valn merupakan pengenal yang sah.

Sebagai contoh :

```
type Kartu = (Sekop, Wajik, Hati, Keriting);
```

e). "Subrange"

Tipe "Subrange" menggambarkan subkelompok nilai yang terdiri dari tipe ordinal lainnya. Biasanya ditandai dengan adanya struktur "Low" (batas bawah).. "High" (batas atas).

Sebagai contoh, deklarasi tipe "enumerated" :

```
type Kartu = (Sekop, Wajik, Hati, Keriting);
```

dapat dibuat tipe "subrange" seperti :

```
type Tkartu_ku = (Sekop..Keriting);
```

2). "Real"

Tipe "Real" mencakup semua angka desimal. Berikut ini jenis-jenis tipe "Real":

Tipe	Jangkauan	Digit yang signifikan	Ukuran byte
Real48	$2.9 \times 10^{-39} .. 1.7 \times 10^{38}$	11-12	6
Single	$1.5 \times 10^{-45} .. 3.4 \times 10^{38}$	7-8	4
Double	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$	15-16	8
Extended	$3.6 \times 10^{-4951} .. 1.1 \times 10^{4932}$	19-20	10
Comp	$-2^{63+1} .. 2^{63} - 1$	19-20	8
Currency	-922337203685477.5808 sampai 922337203685477.5807	19-20	8
Real	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$	15-16	8

Tabel 2.2 Macam- macam type "Real"

b. "String"

Tipe	Panjang Maksimum	Kebutuhan Memori	Dipakai untuk
ShortString	255 karakter	2 ke 256 byte	Kompatibilitas ke belakang
AnsiString	$\sim 2^{31}$ karakter	4 byte ke 2GB	(ANSI) 8-bit
WideString	$\sim 2^{30}$ karakter	4 byte ke 2GB	Karakter Unicode;

Tabel 2.3 Macam-macam tipe "String"

c. Data Terstruktur

Data terstruktur dibagi lagi menjadi :

1). "Set"

Tipe "Set" adalah sekompok nilai yang terdiri dari tipe ordinal yang sama.

Contoh pendeklarasian :

type

```
TSomeInts = 1..250;
```

```
TIntSet = set of TSomeInts;
```

atau :

```
type TIntSet = set of 1..250;
```

atau :

```
var MySet: set of 'a'..'z';
```

...

```
MySet := ['a', 'b', 'c'];
```

2). "Array"

a). "Array" statis

```
array[indexType1, ..., indexTypeN] of baseType
TMatrix = array[1..10, 1..50] of Real;
```

b). "Array" Dinamis

"Array" Dinamis dipakai untuk menentukan besar ruangan yang disiapkan untuk menampung data sesuai dengan kebutuhan program. "Array" Dinamis bisa dideklarasikan sebagai berikut:

```
array of TipeDasar
sebagai contoh,
```

```
var MyFlexibleArray: array of Real;
```

akan mendeklarasikan sebuah array dinamis berdimensi satu dari tipe data "Real". Deklarasi diatas tidak mencantumkan besar ukuran memori yang akan ditempati oleh MyFlexibleArray. Untuk membuat array dalam memori dipakai perintah `SetLength`. Sebagai contoh diberikan deklarasi diatas,

```
SetLength(MyFlexibleArray, 20);
```

3). "Record"

"Record" dalam banyak bahasa bisa disamakan dengan "Structure" yang mewakili kelompok komponen dari tipe yang bermacam-macam. Masing-masing elemen disebut "field". "Record" bisa dideklarasikan dalam bentuk :

```
type NamaTiperecord = record
  fieldList1: tipe1;
```

```
...
```

```
fieldListn: tipen;
```

```
end
```

sebagai contoh :

```
type
```

```
TTanggalRec = record
```

```
  Tahun : Integer;
```

```
  Bulan : (Jan, Feb, Mar, Apr, Mei, Jun,
           Jul, Agu, Sep, Okt, Nop, Des);
```

```
  Hari : 1..31;
```

```
end;
```

4). "File"

Sebuah "File" adalah sekelompok elemen dari tipe data yang sama. Bentuk

umum deklarasi "File" adalah :

```
type fileType_name = file of type
```

Contoh penggunaan :

```
type
```

```
DataTelepon = record
```

```
  NamaDepan, NamaBelakang : string[20];
```

```
  NoTelepon : string[15];
```

```
  Terdaftar : Boolean;
```

```
end;
```

```
DaftarTelepon = file of DataTelepon;
```

5). "Class"

Sebuah "Class" merupakan struktur yang terdiri dari "field", "method"/metoda dan "property"/properti. Contoh sebuah tipe "Class" disebut "Object" atau obyek. Sedangkan "field", "method" dan "property" dari "Class" disebut "Components"/Komponen atau "Members"/Anggota. Deklarasi "Class" memiliki bentuk umum :

```

type Namakelas = class (ancestorClass)
    daftar_anggota
end;

```

sebagai contoh dibawah ini deklarasi kelas TListColumns dari VCL Delphi unit ComCtrls :

```

type
    TListColumns = class(TCollection)
private
        FOwner: TCustomListView;
        function GetItem(Index: Integer): TListColumn;
        procedure SetItem(Index: Integer; Value:
            TListColumn);
protected
        function GetOwner: TPersistent; override;
        procedure Update(Item: TCollectionItem); override;
public
        constructor Create(AOwner: TCustomListView);

```

```

function Add: TListColumn;

property Owner: TCustomListView read FOwner;

property Items[Index: Integer]: TListColumn read
GetItem write SetItem; default;

end;

```

6). "Class Reference"

Tipe "Class Reference" disebut juga "MetaClass" yang ditunjukkan dari bentuk umum deklarasinya :

```
class of tipe; dimana tipe adalah tipe kelas apa saja.
```

Sebagai contoh adalah deklarasi berikut dari "constructor" untuk TCollection (dalam unit kelas VCL) :

```
type TCollectionItemClass = class of TCollectionItem;
```

...

```
constructor Create(ItemClass: TCollectionItemClass);
```

Deklarasi ini menyebutkan bahwa untuk membuat contoh obyek TCollection, harus melewati penurunan nama kelas konstruktor dari TCollectionItem.

Tipe "Class-reference" ini sangat berguna ketika akan memanggil sebuah metode kelas atau konstruktor virtual pada sebuah kelas atau obyek yang memiliki tipe aktual yang dikenal pada saat kompilasi.

7). "Interface"

Tipe "interface" digunakan untuk mendefinisikan metode agar dapat dimanfaatkan oleh sebuah kelas. "Interface" menawarkan kemudahan dalam

pengembangan teknik pewarisan multi (“multiple inheritance”). “Interface” juga sangat penting untuk digunakan pada model obyek terdistribusi, termasuk “COM” (“Component Object Model”) dan “CORBA” (“Common Object Request Broker Architecture”). Semua obyek yang dibentuk lewat Delphi, yang mendukung “Interface” dapat berinteraksi dengan obyek “COM” yang ditulis dengan C++, Java dan bahasa perograman lainnya. Bentuk umum deklarasi “interface” adalah :

```

type Namainterface = interface (ancestorInterface)
    ['{GUID}']
    daftar_anggota
end;

```

Adapun contoh pendeklarasian “interface” :

```

type
    IMalloc = interface(IUnknown)
        ['{00000002-0000-0000-C000-000000000046}']
        function Alloc(Size: Integer): Pointer; stdcall;
        function Realloc(P: Pointer; Size: Integer):
            Pointer; stdcall;
        procedure Free(P: Pointer); stdcall;
        function GetSize(P: Pointer): Integer; stdcall;
        function DidAlloc(P: Pointer): Integer; stdcall;
        procedure HeapMinimize; stdcall;
end;

```

4. "Pointer"

"Pointer" adalah sebuah variabel yang menunjuk ke sebuah alamat memori.

"Pointer" memakai 4 byte memori. Bentuk umum pendeklarasian "Pointer" adalah :

```
type NamaTipepointer = ^tipe
```

Ada dua perintah yang harus dipergunakan pada saat memakai "Pointer", yaitu "New" dan "Dispose". Perintah "New" untuk menyiapkan lokasi memori bagi variabel berpointer, sedangkan "Dispose" untuk mengosongkan kembali memori yang telah dipakai. Contoh:

```

1  var
2  X, Y: Integer; // X dan Y adalah variabel "Integer"
3  P: ^Integer; // P variabel "Integer" ber"pointer"
4  begin
5  X := 17; // memasukkan nilai ke X
6  P := @X; // memasukkan alamat X ke P
7  Y := P^; //mengambil nilai dari P dan memasukkan hasil ke
   Y
8  end;
```

5. Prosedural

Tipe Prosedural berguna untuk memperlakukan prosedur dan fungsi sebagai suatu nilai yang bisa dilewatkan pada prosedur dan fungsi lainnya. Sebagai contoh, misalkan ada sebuah fungsi bernama Calc, memiliki dua parameter "Integer" dan menghasilkan bilangan bulat :

```
function Calc(X,Y: Integer): Integer;
```

Fungsi Calc dapat dimasukkan ke variabel F :

```
var F: function(X,Y: Integer): Integer;
```

```
F := Calc;
```

6. "Variant"

Tipe "Variant" digunakan manakala memanipulasi data dengan banyak tipe atau tipe data tidak dapat ditentukan pada saat kompilasi. Tipe "Variant", memang menawarkan fleksibilitas yang lebih besar, tetapi juga lebih banyak menggunakan memori dibandingkan tipe lainnya. Contoh :

```
var
```

```
V1, V2, V3, V4, V5: Variant;
```

```
I: Integer;
```

```
D: Double;
```

```
S: string;
```

```
begin
```

```
V1 := 1; { nilai "integer" }
```

```
V2 := 1234.5678; { nilai real }
```

```
V3 := 'Hello world!'; { nilai string }
```

```
V4 := '1000'; { nilai value }
```

```
V5 := V1 + V2 + V4; { nilai real 2235.5678 }
```

```
I := V1; { I = 1 (nilai integer) }
```

```
D := V2; { D = 1234.5678 (nilai real) }
```

```
S := V3; { S = 'Hello world!' (nilai string) }
```

```

I := V4; { I = 1000 (nilai integer) }
S := V5; { S = '2235.5678' (nilai string) }
end;

```

7. Struktur Kontrol

Dalam setiap bahasa pemrograman, data dikontrol melalui sebuah struktur kontrol. Ada tiga kelompok Struktur Kontrol pada Delphi yaitu :

a. Struktur Pengambilan Keputusan

1). If...Then...(Else)

Struktur ini digunakan mengeksekusi perintah dari beberapa pilihan yang tersedia. Standar penulisannya adalah :

```

If kondisi/ syarat then
    [blok perintah 1]
Else
    [blok perintah 2];

```

Untuk blok perintah yang lebih dari satu baris, maka harus diawali dengan kata **Begin** dan diakhiri dengan **End**. Untuk struktur kontrol yang bersarang (“nested”) bisa dituliskan sebagai berikut :

```

If kondisi 1 then
    [blok perintah 1];
    Else if kondisi 2 then
        [blok perintah 2];
        Else if kondisi 3 then

```

```

begin
    [blok perintah 3];
...dan seterusnya
end

```

Else

begin

```
[blok perintah n+1];
```

end;

2). Case....of

Sebagai alternatif bisa dipakai struktur Case....of. Standar penulisannya adalah

Case kondisi of

```
Ekspresi 1 : [blok perintah 1];
```

```
Ekspresi 2 : [blok perintah 2];
```

```
..... dan seterusnya
```

Else

```
[blok perintah n+1];
```

end;

b. Struktur Pengulangan

1). Repeat....Until

Struktur Repeat....Until digunakan untuk mengulang sebuah blok perintah yang jumlah pengulangannya tidak diketahui, sampai kondisi dipenuhi. Standar penulisannya :

Repeat

[blok perintah];

Until kondisi;

2). **While...Do**

Hampir sama dengan Repeat....Until, bedanya pengulangan dengan While...Do hanya dilakukan apabila hasil pengujian kondisi bernilai benar. Cara penulisannya adalah :

While kondisi **Do**

Begin

[blok perintah];

end;

3). **ForDo**

Struktur For....Do digunakan apabila jumlah perulangannya diketahui dengan pasti. Struktur penulisannya adalah :

For variabel_lokal = awal **To** akhir **Do**

Begin

[blok perintah];

End;

Dari struktur diatas terlihat struktur For...Do tidak memerlukan kondisi, tetapi memerlukan nilai awal dan nilai akhir. Dengan ketentuan nilai awal lebih besar dari nilai akhir. Apabila dikehendaki variabel penghitung berjalan mundur, maka kata **To** bisa diganti dengan **DownTo**. Dengan ketentuan nilai awal lebih besar dari nilai akhir.

c. Struktur Pencabangan

1). GoTo

Digunakan untuk melompat ke rutin program yang ada dibawah nama label yang dipanggil. Nama label harus sudah dideklarasikan terlebih dahulu. Cara penulisannya adalah :

Label nama_label;

Begin

[blok perintah];

nama_label :

[blok perintah];

Goto nama_label;

End;

2). Break

Perintah ini dilakukan untuk keluar dari "looping"/pengulangan karena ada kondisi yang tidak terpenuhi.

3). Continue

Perintah Continue dilaksanakan untuk menggagalkan pengulangan yang sedang dilaksanakan, untuk memulai langkah pengulangan berikutnya.

4). Exit

Perintah ini digunakan untuk keluar dari rutin program tempatnya berada. Apabila pemanggilannya pada prosedur, maka akan langsung dibawa keluar dari prosedur tersebut.

5). Halt

Perintah Halt digunakan untuk keluar dari program, dimanapun pemanggilannya dilakukan.

6). Run Error

“Run Error” berguna untuk menghentikan jalannya program (melompat ke bagian akhir program) dengan cara membangkitkan suatu kesalahan yang menyebabkan program tersebut terhenti.

d. Tanda Operasi

Tanda operasi/”operator” adalah notasi yang digunakan untuk menghubungkan dua buah nilai atau lebih sehingga membentuk ekspresi. Contoh penggunaannya adalah untuk pengujian kondisi atau perhitungan matematik.

1). Tanda Operasi Aritmatik

Operator	Operasi	Tipe operand	Tipe hasil	Contoh
+	Penjumlahan	Integer, real	Integer, real	$X + Y$
-	Pengurangan	Integer, real	Integer, real	Result - 1
*	Perkalian	Integer, real	Integer, real	$P * L$
/	Pembagian	Integer, real	Real	$X / 2$
Div	Pembagian Integer	Integer	Integer	Angka_hari div 7
Mod	Sisa pem- bagian	Integer	Integer	$Y \text{ mod } 6$
+ (unary)	Tanda positif	Integer, real	Integer, real	+7
- (unary)	Tanda negatif	Integer, real	Integer, real	-X

Tabel 2.4 Operator Aritmatik

2). Tanda Operasi Logika

Operator	Operasi	Tipe operand	Tipe hasil	Contoh
Not	Negasi antar bit	integer	integer	not X
And	And antar	integer	integer	X and Y
Or	Or antar bit	integer	integer	X or Y
Xor	Xor antar bit	integer	integer	X xor Y
Shl	Geser kiri antar bit	integer	integer	X shl 2
Shr	Geser kanan antar bit	integer	integer	Y shr I

Tabel 2.5 Operator Logika

3). Tanda Operasi Relasi

Operator	Operasi	Tipe operand	Tipe Hasil	Contoh
=	Sama dengan	simple, class, class reference, interface, string, packed string	Boolean	I = Max
◇	Tidak sama dengan	simple, class, class reference, interface, string, packed string	Boolean	X ◇ Y
<	Lebih kecil dari	simple, string, packed string, Pchar	Boolean	X < Y
>	Lebih besar dari	simple, string, packed string, Pchar	Boolean	Len > 0
<=	Lebih kecil atau sama dengan	simple, string, packed string, Pchar	Boolean	Cnt <= I
>=	Lebih besar atau sama dengan	simple, string, packed string, Pchar	Boolean	I >= 1

Tabel 2.6 Operator Relasi

4). Tanda Operasi Boolean

Operator	Operasi	Tipe operand	Tipe Hasil	Contoh
Not	Negasi	Boolean	Boolean	not (C in MySet)

And	Persekutuan	Boolean	Boolean	Done and (Total > 0)
Or	Atau	Boolean	Boolean	A or B
Xor	Atau eksklusif	Boolean	Boolean	A xor B

Tabel 2.7 Operator Boolean

5). Tanda Operasi Pointer

Operator	Operasi	Tipe operand	Tipe hasil	Contoh
+	Penambahan pointer	character pointer, integer	character pointer	P + I
-	Pengurangan Pointer	character pointer integer	character pointer, integer	P - Q
^	Pengembalian Pointer	pointer	Tipe dasar pointer	P^
=	Sama dengan	pointer	Boolean	P = Q
◇	Tidak sama dengan	pointer	Boolean	P ◇ Q

Tabel 2.8 Operator Pointer

6). Tanda Operasi Set

Operator	Operasi	Tipe operand	Tipe hasil	Contoh
----------	---------	--------------	------------	--------

+	Gabungan	set	set	Set1 + Set2
-	Perbedaan	set	set	S - T
*	Interseksi	set	set	S * T
\leq	Subset	set	Boolean	Q \leq MySet
\geq	Superset	set	Boolean	S1 \geq S2
=	Sama dengan	set	Boolean	S2 = MySet
\diamond	Tidak sama dengan	set	Boolean	MySet \diamond S1
In	Anggota	Ordinal, set	Boolean	A in Set1

Tabel 2.9 Operator Set

2.3.10 Prosedur dan Fungsi

Dalam pemrograman terstruktur, prosedur mempunyai fungsi yang sangat penting. Pembuatan program menjadi lebih mudah, karena program disusun dari blok- blok kecil, dengan masing- masing bagian mempunyai tugas yang spesifik. Setiap bagian dapat dipelajari dan diubah secara terpisah dari detail program lainnya. Untuk tugas- tugas yang sama atau hampir sama, penggunaan prosedur akan menghindarkan terjadinya duplikasi rutin program (menghemat memori dan mempercepat eksekusi). Dalam Delphi terdapat dua jenis prosedur yaitu "Sub Procedure" atau bisa disingkat dengan "Procedure" , dan "Function Procedure".

"Procedure" adalah prosedur yang tidak mengembalikan nilai, sedangkan "Function Procedure" adalah prosedur yang mengembalikan nilai. Delphi juga menyediakan fungsi dan prosedur "built in" untuk membantu pembuatan program.

1. Prosedur

Prosedur dituliskan diantara judul program (“Program”) dan awal dari program utama (“Begin”). Karena prosedur merupakan program kecil, maka dalam rutinnnya juga boleh ditambah bagian **var**, **type**, **const** atau prosedur lainnya.

Cara penulisan prosedur :

```
Procedure nama_prosedur (parameter : tipe);
```

```
Var
```

```
    Nama_variabel : tipe;
```

```
Const
```

```
    Nama_variabel = ekspresi;
```

```
Begin
```

```
    [blok perintah]
```

```
End;
```

Cara pemanggilan prosedur, dilakukan pada baris terpisah didalam program utama atau prosedur lain .

```
Nama_prosedur (parameter);
```

2. Fungsi

Fungsi biasanya diperlukan pada saat membuat program dengan banyak perhitungan. Tetapi untuk fungsi-fungsi tertentu sudah tersedia dalam pustaka Delphi, misalnya **abs** (untuk mencari harga mutlak), **power** (untuk pemangkatan real), dan sebagainya. Cara penulisan fungsi adalah sebagai berikut :

```
Function nama_fungsi (parameter : tipe) : tipe;
```

```
Var
```

```
Nama_variabel : tipe;  
  
Begin  
  
  [blok perintah];  
  
End;
```

Karena fungsi juga bersifat seperti variabel, untuk pemanggilan fungsi harus digabungkan dengan pernyataan lain. Contoh :

```
Variabel := nama_fungsi(parameter);
```

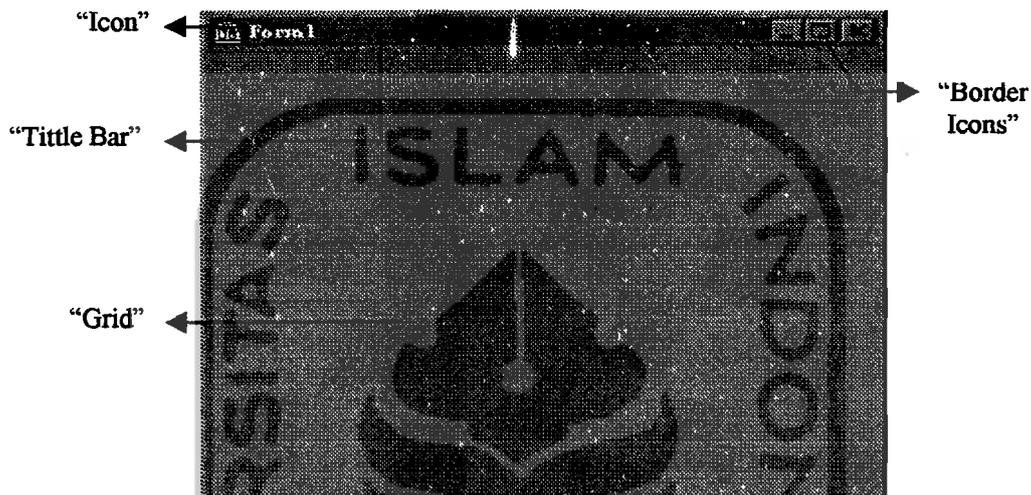
2.3.11 Dasar Pemrograman Visual

Pemrograman Visual memiliki cukup banyak perbedaan dibandingkan dengan pemrograman konvensional, atau non Visual. Dalam pemrograman non Visual, biasanya dibuat dahulu program utamanya, baru dipikirkan tampilannya. Tapi dalam pemrograman Visual yang terjadi adalah sebaliknya. Tampilan bisa langsung direncanakan, baru kemudian perencanaan kaitan antara tampilan dengan program utamanya. Pembuatan prosedur-prosedur yang digunakan untuk memperindah tampilan, digantikan dengan pengisian properti pada komponen. Respon program terhadap pemakai, digantikan dengan prosedur "event", dan banyak lagi yang lainnya. Tetapi karena banyak hal sulit bisa digantikan dengan mudah, membawa konsekuensi pada peningkatan kompleksitas program dan standar pemrograman.

Pada prinsipnya untuk membuat sebuah program secara visual, adalah perencanaan tampilan pada "Form", dengan meletakkan komponen sedemikian, mengatur propertinya, kemudian sentuhan akhir dengan menambahkan prosedur "event".

1. "Form"

"Form" adalah sebuah obyek yang paling mendasar dalam pemrograman Visual. Form bisa diibaratkan sebagai pondasi tempat meletakkan semua komponen visual yang akan membentuk sebuah program.



Gambar 2.57 "Form"

Keterangan gambar :

"Icon" adalah gambar kecil yang mewakili lambang aplikasi yang bisa diubah lewat properti "Icon"

"Title Bar" adalah judul dari lembar "Form" yang bisa diubah lewat properti "Caption".

"Grid" adalah titik-titik pembantu untuk mengatur letak komponen visual pada "form".

"Border Icons" terdiri dari tiga buah tombol, yaitu tombol "Minimize", "Maximize" dan "Close".

a. Properti Penting “Form”

Ada 52 properti pada “Form”, dibawah ini dibahas beberapa properti yang dianggap paling penting.

1). Properti “ BorderIcons”

Properti ini memiliki beberapa subproperti yaitu :

- a). `biSystemMenu`, jika setting subproperti ini “False”, maka pada saat form ini dijalankan tombol-tombol yang biasa ada pada sudut kanan atas jendela tidak akan ditampilkan.
- b). `biMinimize`, jika setting subproperti ini “False”, maka tombol minimize tidak bisa diakses.
- c). `biMaximize`, sama dengan `biMinimize`, hanya yang tidak bisa diakses adalah tombol Maximize.
- d). `biHelp`, jika setting subproperti ini “True”, maka akan ditampilkan tombol Help dipojok kanan atas jendela. Tombol ini hanya bisa digunakan manakala subproperti `biMimize` dan `biMaximize` bernilai “False”.

2). Properti “BorderStyles”

Properti ini juga mempunyai beberapa subproperti yaitu :

- a). `bsDialog`, form yang tidak bisa diubah ukurannya dan dibatasi dengan kotak dialog standar yang hanya memiliki tombol “close” dipojok kanan atas jendela.

- b). `bsNone`, tidak bisa diubah ukurannya, tidak memiliki garis pembatas dan tidak memiliki "Title Bar".
 - c). `bsSingle`, tidak bisa diubah ukurannya, dan dibatasi oleh sebuah garis tunggal, semua tombol "Minimize", "Maximize" dan "Close" muncul pada "Title Bar".
 - d). `bsSizable`, bentuknya mirip dengan `bsSingle`, tetapi ukurannya bisa diubah.
 - e). `bsToolWindow`, mirip dengan `bsSingle` hanya memiliki "Title Bar" lebih kecil, tanpa "Icon" serta hanya tombol "Close" yang muncul di kanan atas jendela.
 - f). `bsSizeToolWin`, seperti `bsSizable` yang bisa diubah ukurannya hanya saja memiliki judul form lebih kecil, tanpa "icon" dan hanya ada tombol "Close".
- 3). Properti "Caption", untuk menampilkan teks pada judul "Form".
- 4). Properti "Enabled", jika setting bernilai "True" maka "Form" bisa diakses dan sebaliknya tidak.
- 5). Properti "Font", secara umum mengatur bentuk huruf pada properti "Caption". Memiliki subproperti lagi antara lain :
- a). Name, untuk memilih nama Font
 - b). Size, untuk menentukan ukuran font.
 - c). Style, untuk menentukan tebal, tipis dan miring sebuah font.
- 6). Properti "HorzScrollBar", memiliki banyak subproperti antara lain :
- "Visible", jika setting "True", maka balok penggulung horisontal akan ditampilkan dibagian bawah "Form".

- 7). Properti "Icon", untuk memilih "icon" yang muncul pada balok judul.
- 8). Properti "Name", untuk menentukan nama "form".
- 9). Properti "VertScrollBar", memiliki banyak subproperti antara lain :
 - "Visible", jika setting "True", maka balok penggulung vertikal akan ditampilkan dibagian samping kanan "Form".
- 10). Properti " Visible", jika setting bernilai "True", maka "Form" akan terlihat dilayar.
- 11). Properti "Window State", memiliki 3 buah subproperti yaitu :
 - a). wsNormal, "form" akan ditampilkan secara normal, dalam bentuk dan ukuran seperti pada waktu didisain.
 - b). wsMinimized, "form" akan ditampilkan dalam bentuk terminimasi.
 - c). wsMaximized, "form" akan ditampilkan memenuhi layar, yaitu dalam ukuran maksimal.

b. Cara Memanggil Form

Adakalanya program yang dibuat membutuhkan "form" lebih dari satu. Digunakan metoda "Show" untuk memanggil "form" tersebut dari "Form" lainnya.

Contoh :

```

Procedure TForm1.Klik (Sender: Tobject);
Begin
    Form2.Show;
End;

```

Nama unit yang terkait dengan “form” yang dipanggil juga harus ditambahkan dibawah kata “implementation”, dengan menggunakan perintah “uses”. Contoh :

```
Implementation
```

```
Uses Unit2;
```

2. “Komponen Visual”

Terdapat 2 jenis visual komponen pada Delphi, yaitu “Visual Component”(“VC”) dan “Non Visual Component”(“NVC”). Perbedaannya “Visual Component” akan terlihat pada saat modus disain dan modus “runtime”. Sedangkan “Non Visual Component” hanya akan terlihat pada modus disain saja. Pada Delphi 4 ini terdapat 14 kelompok, yaitu : “Standard”, “Additional”, “Win32”, “System”, “Internet”, “Data Access”, “Data Controls”, “Midas”, “Decision Cube”, “QReport”, “Dialogs”, “Win31”, “Sample” dan “ActiveX”. Dibawah ini akan dijelaskan beberapa komponen yang dipakai dalam pembuatan program, yaitu :

a. Kelompok “Standard”

- 1).  Main Menu, digunakan untuk membuat menu bar pada “form”.
- 2).  PopupMenu, digunakan untuk membuat menu “popup” pada saat klik kanan mouse.
- 3).  Label, untuk membuat teks pada “form”.
- 4).  Edit, digunakan sebagai input/ output pada “form”.
- 5).  Button, untuk mer.buat tombol dengan fungsi tertentu.

- 6).  ComboBox, digunakan untuk menyederhanakan input, agar pemakai tidak perlu mengetik, cukup memilih dari daftar yang disediakan.

b. Kelompok “Additional”

- 1).  Image, untuk menampilkan data gambar berupa “icon”, “Bitmap” atau “metafile”.
- 2).  Bevel, untuk membuat kotak pembatas, untuk memperindah tampilan program.

c. Kelompok “Win32”

- 1).  ImageList, untuk menampung gambar - gambar yang dipakai sebagai “icon” pada program.
- 2).  RichEdit, untuk membuat kotak editor, bisa untuk input/ output dan hasilnya bisa dicetak ke printer.
- 3).  Animate, untuk menjalankan file-file AVI tanpa suara.
- 4).  StatusBar, untuk membuat area tempat menampilkan informasi pada bagian bawah “form”.
- 5).  ToolBar, untuk membuat tombol- tombol dengan “icon” untuk mempercepat eksekusi sebuah perintah.

d. Kelompok “System”

-  Timer, untuk mengatur interval waktu pada sebuah fungsi atau prosedur.

e. Kelompok “Dialogs”

- 1).  OpenDialog, untuk menampilkan kotak dialog untuk membuka sebuah file dalam bentuk standar.
- 2).  SaveDialog, menampilkan kotak dialog penyimpanan file.
- 3).  FontDialog, untuk menampilkan kotak dialog memilih font.
- 4).  PrintDialog, untuk menampilkan kotak dialog untuk pencetakan.

f. Kelompok “Samples”

-  Gauge, untuk menampilkan indikator kemajuan proses yang sedang dikerjakan program. Biasanya ditampilkan pola perubahan warna dan prosentase.

3. “Properti”

a. Beberapa properti penting

Properti adalah bagian-bagian penyusun sebuah komponen yang diatur sedemikian dalam “Object Inspector”, berguna untuk merubah tampilan atau perilaku dari komponen tersebut. Setiap komponen visual memiliki properti yang berbeda. Dibawah ini disajikan beberapa properti penting dan bersifat umum, yaitu :

- 1). Caption, untuk menentukan teks yang akan menjadi judul sebuah obyek atau komponen.
- 2). Color, untuk menentukan warna komponen.

- 3). Font, untuk memilih jenis huruf beserta ukuran, warna dan “style” bagi teks yang akan ditampilkan pada komponen.
- 4). Enabled, untuk menentukan bisa tidaknya sebuah komponen diakses. Jika setting Enabled bernilai “True”, maka komponen bisa diakses, dan sebaliknya tidak.
- 5). Height, untuk menentukan ukuran komponen dari sisi atas kebawah.
- 6). Left, untuk menentukan letak obyek dari kiri.
- 7). Name, untuk memberi nama komponen, yang akan digunakan sebagai variabel pada sebuah prosedur.
- 8). Picture, untuk memilih gambar yang akan ditampilkan pada obyek.
- 9). Text, mirip dengan Caption tapi digunakan untuk menampilkan pesan, bukan judul.
- 10). Top, untuk mengatur jarak komponen dengan tepi atas layar.
- 11). Visible, untuk menentukan terlihat atau tidaknya komponen, pada saat program dijalankan.
- 12). Width, untuk menentukan lebar obyek.

b. Teknik pengisian properti

1). Cara Langsung

Teknik pengisian properti secara langsung dilakukan dengan memilih properti yang akan diisi atau diubah pada “Object Inspector”, kemudian diisi atau diubah sesuai dengan kebutuhan.

2). Cara Tidak Langsung

Sedangkan pengisian properti dengan tidak langsung, dilakukan dengan menu-
liskannya pada rutin program. Cara penulisannya adalah :

```
Nama_obyek.nama_properti.nama_subproperti :=  
setting_properti;
```

4. "Event"

"Event" adalah aktivitas yang terjadi saat program berjalan. Contoh "event" adalah penekanan tombol mouse, penekanan tombol keyboard, penggeseran mouse dan lain-lain.

a. Macam-macam "event"

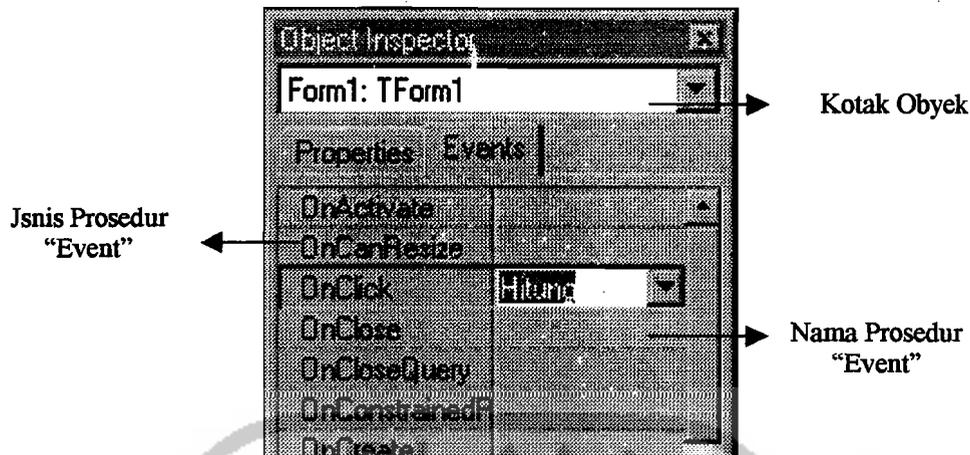
Setiap komponen visual pada Delphi memiliki "event" yang khas dan berbeda. Dibawah inii beberapa "event" yang sering digunakan dalam pembuatan program:

- 1). Event untuk mouse
 - a). OnClick, terjadi pada saat pemakai menekan tombol mouse.
 - b). OnDblClick, terjadi pada saat pemakai menekan/klik ganda tombol mouse.
 - c). OnMouseDown, terjadi bila pemakai menekan dan menahan tombol mouse.
 - d). OnMouseUp, terjadi bila pemakai melepas tombol mouse.
 - e). OnDragDrop, terjadi bila pemakai menekan dan menahan tombol mouse, kemudian menggeser mouse ketempat lain baru kemudian dilepas.
 - f). OnDragOver, terjadi bila pemakai menekan dan menahan tombol mouse pada obyek, kemudian menggeser mouse melewati obyek tersebut.
- 2). Event untuk keyboard

- a). OnKeyPress, terjadi bila pemakai menekan tombol keyboard lalu dilepaskan.
 - b). OnKeyDown, terjadi pada saat pemakai menekan dan menahan tombol keyboard.
 - c). OnKeyUp, terjadi jika pemakai melepaskan tekanan pada sebuah tombol keyboard.
- 3). Event karena sebab lain
- a). OnActivate, terjadi pada saat pemakai membuat sebuah "form" menjadi jendela yang aktif.
 - b). OnCreate, terjadi pada saat pemakai membuat "form" pada saat "runtime"
 - c). OnDeactivate, terjadi saat berpindah dari aplikasi aktif ke aplikasi windows yang lain.
 - d). OnResize, terjadi pada saat pemakai mengubah ukuran "form".
 - e). OnScroll, terjadi pada saat pemakai menggunakan balok penggulung/ "scroll bar" untuk menggulung layar.
 - f). OnTimer, terjadi saat sebuah interval waktu terpenuhi. Prosedur event yang terkait dengan "event" ini akan dieksekusi.

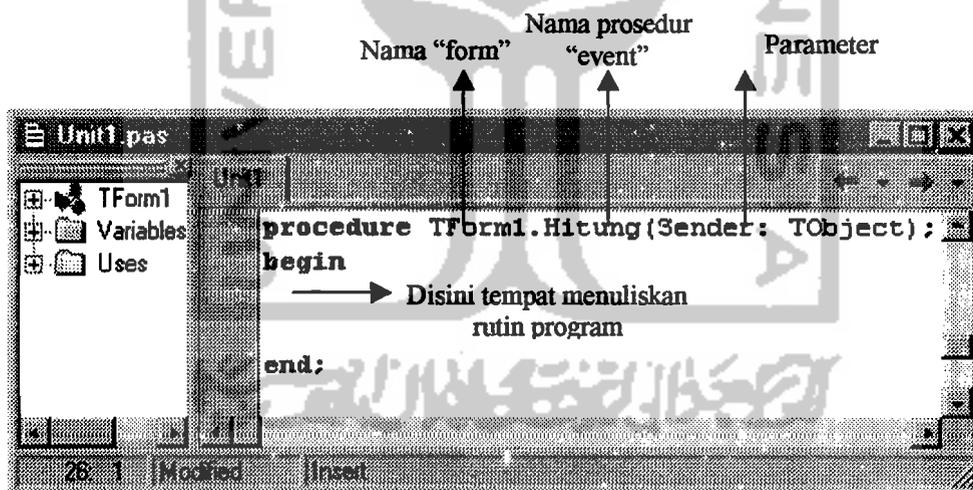
b. Prosedur "Event"

Jika sebuah "event" dibangkitkan oleh pemakai, maka suatu prosedur yang terasosiasi dengan "event" tersebut akan dipanggil. Prosedur ini dibuat dengan mengklik nama prosedur "event" yang terdapat pada "Object Inspector".



Gambar 2.68 Pembuatan Prosedur “Event” lewat “Object Inspector”

Apabila nama prosedur “event” belum ada, maka nama tersebut bisa langsung diketik pada kolom nama pada “Object Inspector”, dengan segera Delphi akan menempatkan kursor pada “Code Editor”, sehingga rutin program bisa dituliskan pada bagian tersebut.



Gambar 2.69 Penulisan rutin prosedur “Event” pada “Code Editor”.

c. Parameter pada prosedur “Event”

Setiap prosedur “event” memiliki parameter yang khas. Karena itu tidak mungkin dibahas semuanya, pada bagian ini akan diberikan contoh parameter, yaitu pada “Event” mouse. Ketika sebuah aplikasi VCL mendeteksi sebuah aksi mouse,

maka akan dipanggil prosedur “event” yang telah didefinisikan untuk menangani “event” tersebut, dan melewati lima buah parameter. Kelima buah parameter tersebut adalah :

Parameter	Arti
Sender	Obyek yang mendeteksi aksi mouse
Button	Menunjukkan tombol mana yang ditekan yaitu : mbLeft, mbMiddle, atau mbRight
Shift	Menunjukkan status tombol Alt, Ctrl, dan Shift pada saat aksi mouse terjadi
X, Y	Kordinat dimana “event” tersebut terjadi

Tabel 2.10 Parameter “event” Mouse.

Seringkali pengembalian kordinat diperlukan dalam penanganan “event” mouse, tetapi kadang-kadang perlu juga untuk mengecek tombol mana yang menyebabkan terjadinya “event”.

5. Pembuatan program secara Visual

Secara prinsip pembuatan program secara visual terdiri dari beberapa langkah, dapat dilihat pada diagram berikut :



Gambar 2.70 Diagram pembuatan program secara Visual.