

PERTEMUAN KE-12  
File dan Teks Pemrosesan String

Tujuan:

1. Menampilkan file teks menggunakan objek kotak teks
2. Menyimpan catatan dalam file teks
3. Menggunakan teknik pemrosesan teks untuk mensortir dan mengenkripsi file teks

A. Menampilkan File Teks Menggunakan Objek Kotak Teks

Cara paling mudah untuk menampilkan file teks pada program adalah menggunakan objek kotak teks. Anda bisa membuat objek teks dalam berbagai ukuran. Jika isi file teks terlalu besar, Anda juga bisa menambahkan scroll bar pada kotak teks agar pemakai bisa membaca seluruh file. Untuk me-load isi dari file teks ke dalam kotak teks, Anda perlu menggunakan tiga pernyataan dan satu fungsi. Perintah yang sesuai dijelaskan pada tabel di bawah ini:

Perintah	Keterangan
Open	Membuka file teks untuk input atau output
Line Input	Membaca baris input file teks
EOF	Memeriksa akhir file teks
Close	Menutup file teks

**Membuka File Teks untuk Input**

Sebuah file teks mengandung satu atau lebih baris-baris yang berisi angka, kata, atau karakter. File teks berbeda dari file dokumen yang mengandung kode-kode pemformatan, dan dari file executable yang mengandung instruksi untuk sistem operasi. File teks umum pada komputer akan dikenali oleh Microsoft Windows Explorer sebagai "Text Documents" atau memiliki ekstensi nama .txt, .ini, .log, .inf, .dat, atau .bat. karena file teks hanya mengandung karakter biasa yang bisa dikenali, Anda bisa menampilkannya dengan mudah menggunakan objek kotak teks (text box).

Anda bisa memberikan pemakai pilihan apakah ingin membuka file teks pada program menggunakan objek common dialog untuk meminta pemakai mengetikkan nama path dari file tersebut. Objek common dialog mendukung metode ShowOpen yang menampilkan kotak dialog Open pada layar. Setelah pemakai memilih file pada kotak dialog, nama path nya akan dikembalikan kepada program pada properti FileName, dan Anda bisa menggunakan nama ini untuk membuka file. Objek common dialog tidak membuka file, tetapi hanya mengambil nama path dari file tersebut.

**Pernyataan Open**

Setelah mendapatkan nama path dari pemakai, Anda membuka file dalam program menggunakan pernyataan Open. Sintaks untuk pernyataan Open adalah sebagai berikut:

```
Open pathname For mode As #filenumber
```

**Keterangan:**

- Pathname adalah nama path Windows yang valid
- Mode adalah perintah yang menunjukkan bagaimana file akan digunakan. (Anda akan menggunakan mode Input dan Output)
- Filenumber adalah ilai integr dari 1 sampai 255

Nomor file akan dihubungkan dengan file apabila dibuka. Kemudian Anda menggunakan

nomor file ini dalam kode program setiap kali Anda ingin merujuk kepada file yang terbuka tersebut. Selain hubungan ini, nomor file tidak memiliki arti apa-apa. Visual Basic hanya menggunakan nomor file ini untuk mencatat file-file yang dibuka dalam program.

Pernyataan Open yang menggunakan objek common dialog adalah sebagai berikut:

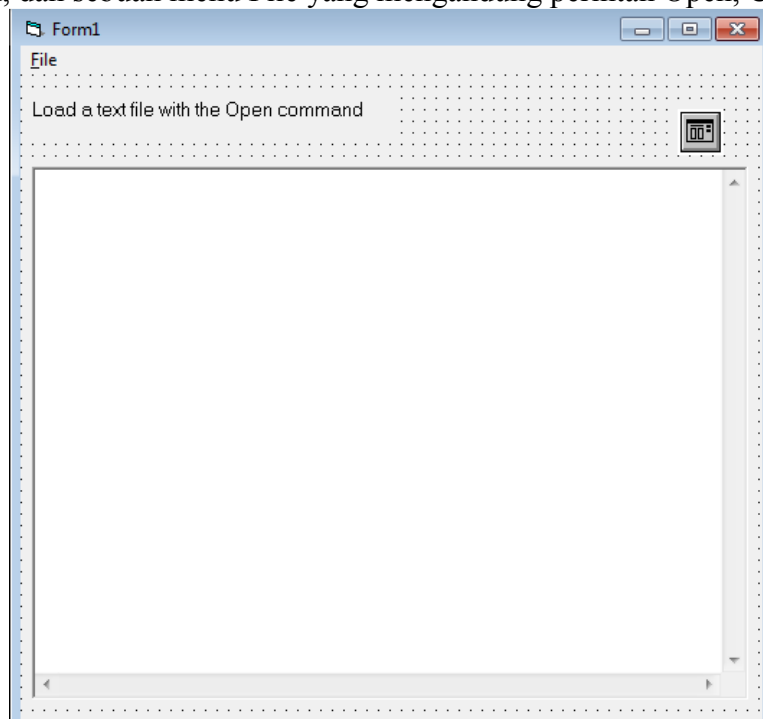
```
Open CommonDialog1.FileName For Input As #1
```

Disini properti CommonDialog1.FileName merepresentasikan nama path, Input adalah modusnya, dan 1 adalah nomor file.

Praktek berikut ini menunjukkan bagaimana Anda bisa menggunakan objek common dialog dan perintah Open untuk membuka sebuah file teks. Praktek ini juga menunjukkan bagaimana Anda bisa menggunakan perintah Line Input dan EOF untuk menampilkan isi file teks dalam kotak teks, dan bagaimana Anda bisa menggunakan perintah Close untuk menutup file.

### Menjalankan program Text Browser

1. Jalankan Visual Basic
2. Buatlah form yang terdiri atas, sebuah objek kotak besar yang memiliki scrollbar. Sebuah objek common dialog, sebuah label yang memberikan petunjuk penggunaan program, dan sebuah menu File yang mengandung perintah Open, Close, dan Exit.



3. Berikut ini adalah setting properti:

Objek	Properti	Setting
TxtFile	Enabled	False
	Multiline	True
	Name	txtFile
	ScrollBars	3-Both
	Text	(Empty)
mnuItemClose	Enabled	False

	Name	mnuItemClose
<b>lblFile</b>	Caption	“Load a text file with the Open command”
	Name	LblFile
<b>Form1</b>	Caption	“Text Browser”

4. Ketikkan pernyataan berikut:

```

Private Sub mnuItemOpen_Click()
    Wrap$ = Chr$(13) + Chr$(10) 'create wrap character
    CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
    CommonDialog1.ShowOpen 'display Open dialog box
    If CommonDialog1.FileName <> "" Then
        Form1.MousePointer = 11 'display hourglass
        Open CommonDialog1.FileName For Input As #1
        On Error GoTo TooBig: 'set error handler
        Do Until EOF(1) 'then read lines from file
            Line Input #1, LineOfText$
            AllText$ = AllText$ & LineOfText$ & Wrap$
        Loop
        lblFile.Caption = CommonDialog1.FileName
        txtFile.Text = AllText$ 'display file
        txtFile.Enabled = True
        mnuItemClose.Enabled = True
        mnuItemOpen.Enabled = False
    Cleanup:
        Form1.MousePointer = 0 'reset mouse
        Close #1 'close file
    End If
    Exit Sub
TooBig: 'error handler displays message
    MsgBox ("The specified file is too large.")
    Resume Cleanup: 'then jumps to Cleanup routine
End Sub

Private Sub mnuItemClose_Click()
    txtFile.Text = "" 'clear text box
    lblFile.Caption = "Load a text file with the Open command."
    mnuItemClose.Enabled = False 'dim Close command
    mnuItemOpen.Enabled = True 'enable Open command
    txtFile.Enabled = False 'disable text box
End Sub

Private Sub mnuItemExit_Click()
    End 'quit program
End Sub

```

5. Buatlah beberapa tulisan dan simpan dalam format .txt dan simpan

6. Klik tombol start, dan jalankan. Bukalah file .txt tersebut.

### Memeriksa kode program ShoText

Perintah-perintah Visual Basic tertulis di dalam tanda kurung:

1. Meminta pemakai mengetikkan nama path menggunakan objek common dialog
2. Membuka file tertentu untuk input (Open...For Input)
3. Menyalin file per baris ke dalam string bernama AllText\$ (line Input)
4. Menyalin baris-baris sampai akhir file dicapai (EOF) atau sampai tidak ada lagi ruang pada string. String AllText\$ memiliki ruang 64 KB karakter
5. Menampilkan string AllText\$ pada kotak tek dan menyalakan scrollbar
6. Menangani setiap error yang terjadi (On Error GoTo)
7. Memperbarui perintah menu File dan pointer mouse serta menutup file (Close)

Unuk melihat keterangan mengenai perintah tersebut, maka terangilah kata yang Anda

inginkan lalu tekan F1.

Error handler TooBig: pada prosedur menampilkan sebuah pesan dan membatalkan proses loading apabila pemakai memilih file yang berukuran lebih besar dari 64 KB. Error handler ini penting karena batasan string 64 KB dari objek kotak teks. (untuk file yang lebih besar dari 64 KB, Anda perlu menggunakan kontrol Rich TextBox)

Jika Anda memilih file yang panjangnya beberapa halaman, Visual Basic akan butuh waktu me-loadnya. Untuk alasan ini, digunakan properti MousePointer untuk mengubah bentuk pointer menjadi jam pasir sampai file ditayangkan pada layar.

Program sederhana tersebut selanjutnya akan digunakan template untuk utilitas tingkat lanjut untuk memproses file teks.

### **Membuat File Teks Baru pada Disk**

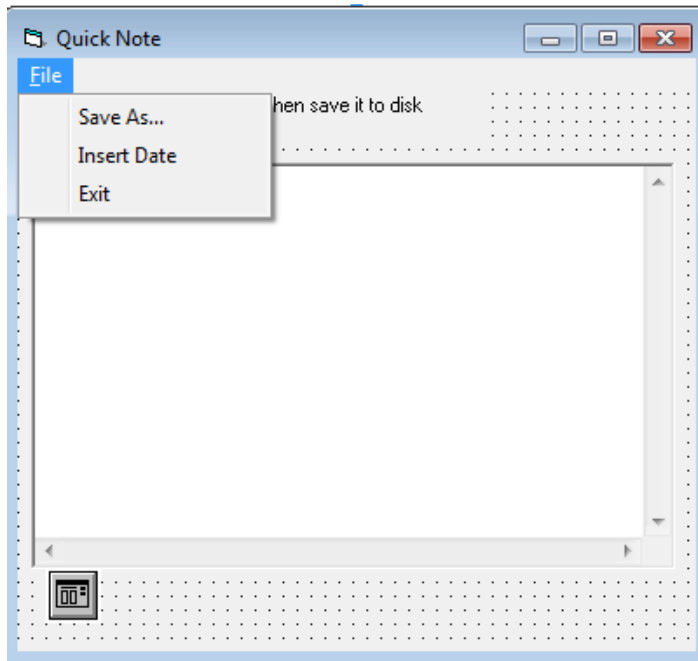
Berikut adalah langkah-langkah yang Anda butuhkan untuk membuat program tersebut:

1. Ambil masukan dari pemakai atau melakukan perhitungan matematik, atau lakukan keduanya
2. Kirimkan hasil pemrosesan tersebut ke satu atau lebih variabel. Sebagai contoh, Anda bisa mengirimkan isi kotak teks ke variabel bernama InputForFile\$
3. Minta pemakai menentukan nama path menggunakan kotak dialog Save As. Untuk menampilkan kotak dialog ini, gunakan metode ShowSave dari objek common dialog.
4. Gunakan nama path yang diterima pada kotak dialog untuk membuka file untuk output (Open...For Output)
5. Gunakan perintah Print # untuk menyimpan satu atau lebih nilai epada file yang dibuka (Print #)
6. Tutuplah file apabila sudah selesai (Close)

Praktek berikut ini menunjukkan bagaimana Anda bisa menggunakan objek kotak teks dan common dialog serta perintah Open, Print #, dan Close untuk membuat sebuah program pencatat sederhana. Anda bisa menggunakan program ini untuk mencatat di rumah atau di tempat kerja lalu

### **Menjalankan program Qnote**

1. Buatlah form berikut:



2. Aturlah properti seperti berikut ini:

Objek	Properti	Setting
TxtNote	Multiline	True
	Name	TxtNote
	ScrollBar	3-Both
	Text	(Empty)
Label1	Caption	"Type your note and then save it to disk"
Form1	Caption	"Quick Note"

3. Ketikkan pernyataan berikut:

```

Private Sub mnuItemDate_Click()
Wrap$ = Chr$(13) & Chr$(10) 'add date to string
    TxtNote.Text = Date$ & Wrap$ & TxtNote.Text
End Sub

Private Sub mnuItemExit_Click()
End
End Sub

Private Sub mnuItemSave_Click()
'note: the entire file is stored in a string
CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
CommonDialog1.ShowSave 'display Save dialog
If CommonDialog1.FileName <> "" Then
    Open CommonDialog1.FileName For Output As #1
    Print #1, TxtNote.Text 'save string to file
    Close #1 'close file
End If
End Sub

```

4. Jalankan program, dan ketikkan teks Anda
5. Gunakan perintah pada menu File
6. Klik perintah Save As
7. Ketikkan dengan format.txt

### Memeriksa kode progrm Qnote

1. Periksalah event\_procedure mnulItemSave\_Click.

Blok pernyataan ini menggunakan objek common dialog untuk menampilkan kotak dialog Save As, membuka file untuk output sebagai file no.1, menulis nilai pada properti txtNote ke dalam disk menggunakan perintah Print #, lalu menutup file teks. Perhatikan pernyataan ini:

```
Print #1, txtNote.Txt
```

Pernyataan tersebut mengirimkan seluruh isi kotak teks kepada file yang terbuka. Print # serupa dengan metode Print, kecuali ia mengarahkan output ke file tertentu, bukan ke layar atau printer.

2. Periksa perintah Open, Print #, dan Close.

### Memproses String Teks dengan Kode Program

Pada bagian ini akan dipelajari cara mengambil informasi dari string teks, menyalin sekelompok string ke dalam array string, serta melindungi informasi penting dengan mengenkripsi string.

### Mengurutkan Teks

Konsep dasar dalam pengurutan adalah sederhana, dengan membuat daftar item yang ingin diurutkan, lalu membandingkan setiap item satu per satu sampai daftar diurutkan dalam urutan naik, turun, alfabetis maupun numerik. Pada Visual Basic, setiap item dibandingkan menggunakan operator relasional yang sama untuk membandingkan nilai numerik.

### Memproses String dengan Pernyataan dan Fungsi

Tugas paling umum yang dilakukan terhadap string adalah menggabungkannya menggunakan operator & (penggabungan). Sebagai contoh program berikut ini menggabungkan tiga ekspresi string dan mengirimkan hasilnya (Bring on the circus!) ke dalam variabel string bernama slogan\$.

```
slogan$ = "Bring" & " on the " & "circus!"
```

anda juga bisa memodifikasi ekspresi string menggunakan beberapa pernyataan khusus, fungsi, serta perator dalam kode program. Tabel berikut melampirkan daftar perintah yang paling berguna.

Perintah	Keterangan	Keterangan
<b>Ucase</b>	Mengubah huruf string menjadi huruf besar	Ucase("Kim") menghasilkan KIM
<b>Lcase</b>	Mengubah huruf string menjadi huruf kecil	Lcase("Kim") menghasilkan kim
<b>Len</b>	Menentukan panjang sebuah string (dalam karakter)	Len("Mississippi") menghasilkan 11
<b>Right</b>	Menghasilkan karakter yang dihitung dari sisi kanan string	Right("Budapest",4) menghasilkan pest
<b>Left</b>	Menghasilkan karakter yang dihitung dari sisi kiri string	Left("Budapest",4) menghasilkan Buda
<b>Mid</b>	Menghasilkan karakter yang dihitung dari tengah-tengah string, dimulai dari titik awal tertentu	Mid("Sommers",4,3) menghasilkan mer
<b>InStr</b>	Mencari titik awal dari sebuah string dalam string besar	start% = InStr("bob", "bobby") menghasilkan 1 ke dalam variabel start%
<b>String</b>	Mengulangi karakter string	String(8,"*") menghasilkan *****
<b>Asc</b>	Menghasilkan huruf ASCII dari huruf tertentu	Asc("A") menghasilkan 65
<b>Chr</b>	Menghasilkan karakter untuk kode ASCII tertentu	Chr\$(65) menghasilkan A
<b>Xor</b>	Menghasilkan operasi "exclusive or" dari dua angka, menghasilkan nilai yang bisa digunakan untuk mengenkripsi dan dekripsi teks.	115 Xor 50 menghasilkan 65

**Tabel operator relasional**

<b>Operator</b>	<b>Arti</b>
< >	Tidak sama dengan
=	Sama dengan
<	Lebih kecil dari
>	Lebih besar dari
< =	Lebih kecil dari atau sama dengan
> =	Lebih besar dari atau sama dengan

Suatu karakter “lebih besar dari” karakter lain jika kode ASCII nya lebih tinggi. Sebagai contoh, nilai ASCII huruf “B” lebih besar dari nilai ASCII huruf “A” sehingga ekspresi di bawah ini adalah benar:

“A” < “B” dan ekspresi di samping ini adalah salah “A” > “B”

“Mike” > “Michael”

“AAAAA” > “AAA”

### **Mengurutkan Teks dalam Kotak Teks**

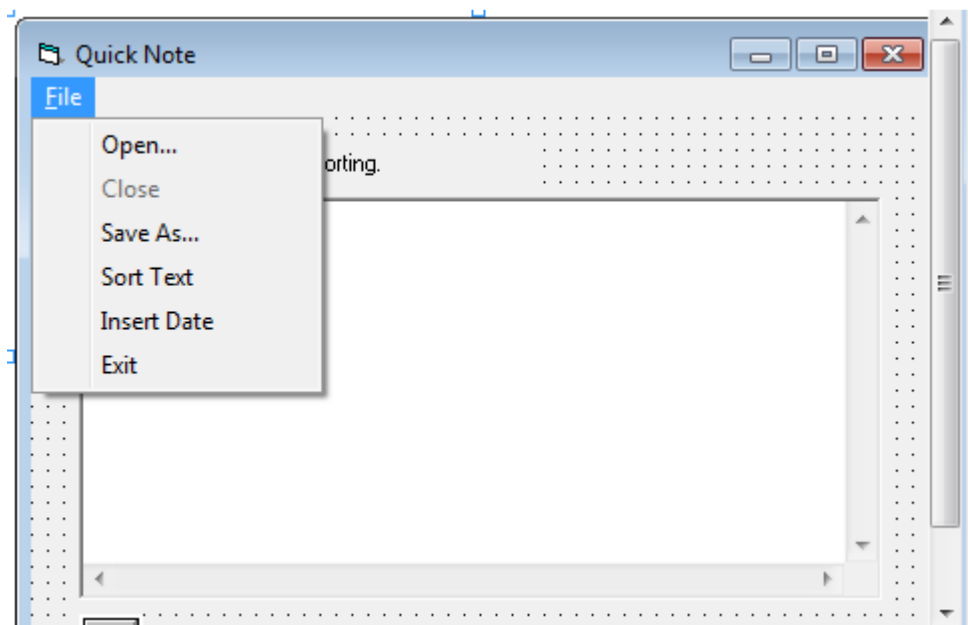
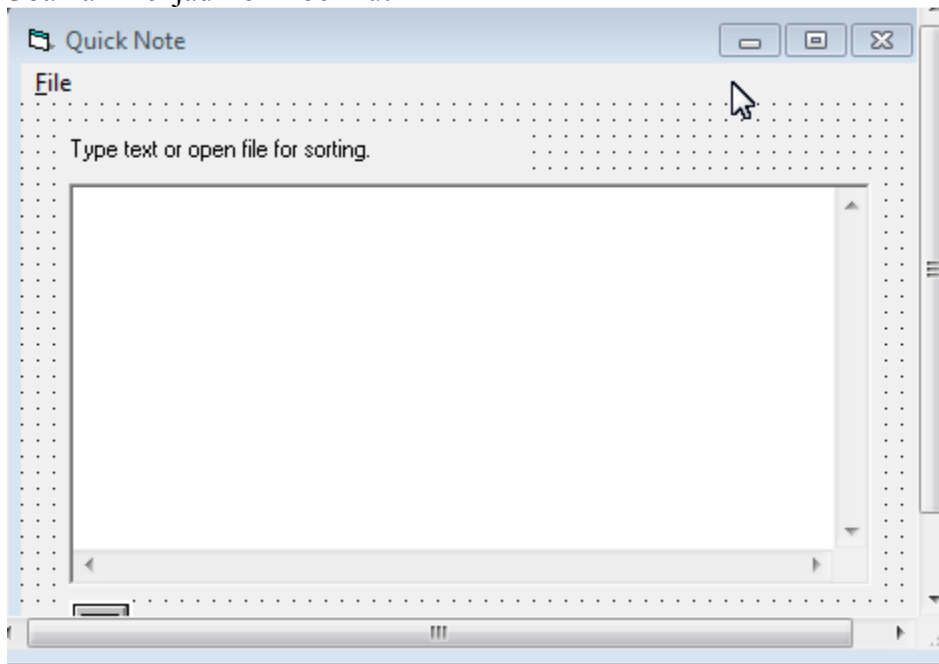
Praktek berikut ini menunjukkan bagaimana Anda bisa menggunakan operator relasional dan beberapa fungsi string untuk mengurutkan baris-baris teks pada kotak teks Visual Basic. Program ini adalah peningkatan dari program Qnote dan mengandung sarana Open yang mengizinkan Anda membuka file yang sudah ada. Pada program ini juga terdapat perintah Sort Text pada menu File yang berfungsi mengurutkan teks yang sedang ditampilkan pada kotak teks. Karena seluruh isi kotak teks Visual Basic disimpan dalam satu string, program pertama-tama harus memecah string yang panjang tersebut menjadi string-string individual yang lebih kecil. String-string ini kemudian bisa diurutkan menggunakan subprogram ShellSort, sebuah rutin pengurutan berdasarkan algoritma yang diciptakan oleh Donald Shell pada tahun 1950. Modul standar yang mendefinisikan array string dinamis menyimpan setiap baris dalam kotak teks digunakan untuk menyederhanakan program ini. Kode di bawah ini menggunakan fungsi Chr untuk mendeteksi karakter carriage return di akhir setiap baris.

```
'determine number of lines in text box object (txtNote)
lineCount% = 0 'this variable holds the total number of lines
charsInFile% = Len(txtNote.Text) 'get total character in box
For i% = 1 To charsInFile% 'move one char at a time through box
    letter$ = Mid(txtNote.Text, i%,1) 'put next char in letter$
    If letter$ = Chr(13) Then 'if carriage ret found (end of line!)
        lineCount% = lineCount% + 1 'go to next line (add to count)
        i% = i% + 1 'skip linefeed char (which always follows cr)
    End If
Next i%
```

Rutin ini mengirimkan jumlah baris pada kotak teks ke dalam variabel lineCount%. Nilai ini dapat digunakan untuk mengubah ukuran array dinamis pada program untuk menyimpan setiap string teks individual. Array string hasilnya kemudian dikirimkan ke subprogram ShellSort untuk diurutkan, dan ShellSort mengembalikan array string tersebut dalam bentuk alfabetis. Setelah array string diurutkan, kemudian tinggal menyalinnya kembali ke kotak teks menggunakan loop For.

## Menjalankan program SortDemo

1. Gunakan kembali program Qnote
2. Ubahlah menjadi form berikut



3. Setting Properti

Objek	Properti	Setting
TextNote	Multiline	True
	Name	txtNote
	ScrollBars	3-Both
	Text	(Empty)
Label1	Name	lblFile
	Caption	"Type text or open file sorting"

4. Ketikkan pernyataan berikut



```
Project1 - Form1 (Code)
mnuItemClose Click

Private Sub mnuItemClose_Click()
    TxtNote.Text = "" 'clear text box
    lblFile.Caption = "Type text or open file for sorting."
    mnuItemClose.Enabled = False 'dim Close command
    mnuItemOpen.Enabled = True 'enable Open command
End Sub

Private Sub mnuItemDate_Click()
    Wrap$ = Chr$(13) & Chr$(10) 'add date to string
    TxtNote.Text = Date$ & Wrap$ & TxtNote.Text
End Sub

Private Sub mnuItemExit_Click()
End
End Sub
```

```
Project1 - Form1 (Code)
mnuItemOpen Click

Private Sub mnuItemOpen_Click()
    Wrap$ = Chr$(13) + Chr$(10) 'create wrap character
    CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
    CommonDialog1.ShowOpen 'display Open dialog box
    If CommonDialog1.FileName <> "" Then
        Form1.MousePointer = 11 'display hourglass
        Open CommonDialog1.FileName For Input As #1
        On Error GoTo TooBig: 'set error handler
        Do Until EOF(1) 'then read lines from file
            Line Input #1, LineOfText$
            AllText$ = AllText$ & LineOfText$ & Wrap$
        Loop
        lblFile.Caption = CommonDialog1.FileName
        TxtNote.Text = AllText$ 'display file
        TxtNote.Enabled = True
        mnuItemClose.Enabled = True
        mnuItemOpen.Enabled = False 'enable scroll
    Cleanup:
        Form1.MousePointer = 0 'reset mouse
        Close #1 'close file
    End If
End Sub
```

```
Project1 - Form1 (Code)
mnuItemSave Click
Private Sub mnuItemSave_Click()
    'note: the entire file is stored in a string
    CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
    CommonDialog1.ShowSave 'display Save dialog
    If CommonDialog1.FileName <> "" Then
        Open CommonDialog1.FileName For Output As #1
        Print #1, TxtNote.Text 'save string to file
        Close #1 'close file
    End If
End Sub

Private Sub mnuItemSortDate_Click()

End Sub

mnuItemSortText Click
Private Sub mnuItemSortText_Click()
    'determine number of lines in text box object (txtNote)
    lineCount% = 0 'this variable holds the total number of lines
    charsInFile% = Len(TxtNote.Text) 'get total characters in box
    For i% = 1 To charsInFile% 'move one char at a time through box
        letter$ = Mid(TxtNote.Text, i%, 1) 'put next char in letter$
        If letter$ = Chr$(13) Then 'if carriage ret found (end of line!)
            lineCount% = lineCount% + 1 'go to next line (add to count)
            i% = i% + 1 'skip linefeed char (which always follows cr)
        End If
    Next i%
    'build an array to hold the text in the text box
    ReDim strArray$(lineCount%) 'create array of proper size
    curline% = 1
    ln$ = "" 'use ln$ to build lines one character at a time
    For i% = 1 To charsInFile% 'loop through text again
        letter$ = Mid(TxtNote.Text, i%, 1)
        If letter$ = Chr$(13) Then 'if carriage return found
            curline% = curline% + 1 'increment line count
            i% = i% + 1 'skip linefeed char
            ln$ = "" 'clear line and go to next
        Else
            ln$ = ln$ & letter$ 'add letter to line
            strArray$(curline%) = ln$ 'and put in array
        End If
    Next i%
    'sort array
    ShellSort strArray$(), lineCount%
    'then display sorted array in text box
    TxtNote.Text = ""
    Wrap$ = Chr$(13) & Chr$(10) 'add date to string
    curline% = 1
    For i% = 1 To lineCount%
        TxtNote.Text = TxtNote.Text & strArray$(curline%) & Wrap$
        curline% = curline% + 1
    Next i%
End Sub
```

5. Simpan program dengan nama SortDemo
6. Jalankan program, dan ketikkan teks berikut:

**Zebra            Banana**  
**Gorilla         Apple**  
**Moon            Turtle**

Pastikan menekan Enter setelah kata "Turtle" sewaktu mengetikkannya, agar Visual Basic bisa menghitung jumlah baris dengan tepat.

7. Buatlah sebuah file .txt dan simpan dalam disk dengan nama abc.txt

**K is for kite**  
**Y is for yellow**  
**3 is for me**  
**Q is for quick**  
**N is for nerd**  
**X is for Xenix**  
**V is for van**  
**I is for fun**  
**T is for Trevor**  
**S is for Sommers**  
**P is for phone**  
**8 like to roller skate**  
**O is for Oscar**

8. Panggilah file abc.txt dengan menu Open
9. Array strArray\$ dideklarasikan pada modul standar (SortDemo.bas) yang juga merupakan bagian program ini. Menggunakan pernyataan ReDim untuk mengubah ukuran strArray\$ sebagai string dinamis dengan variabel lineCount%. Pernyataan ini membuat array yang memiliki jumlah elemen yang sama dengan baris teks pada otak teks (kebutuhan untuk subprogram ShellShort). Menggunakan loop For dan variabel ln\$, digunakan untuk memeriksa kotak teks kembali, mencari karakter carriage return, kemudian menyalin setiap baris yang ditemukan ke dalam strArray\$. Setelah array penuh dengan teks, subprogram dipanggil ShellSort yang telah dibuat sebelumnya pada modul SortDemo.bas.

Subprogram ShellShort menggunakan menggunakan operator relasional <= untuk membandingkan elemen-elemen array dan menukar elemen yang urutannya salah. Subprogram tersebut adalah sebagai berikut:

```

Option Base 1      'start array at 1
Public strArray$( ) 'declare dynamic array for sort

Sub ShellSort(sort$( ), numOfElements%)
'The ShellSort subprogram sorts the elements of sort$( )
'array in descending order and returns it to the calling
'procedure.

span% = numOfElements% \ 2
Do While span% > 0
    For i% = span% To numOfElements% - 1
        j% = i% - span% + 1
        For j% = (i% - span% + 1) To 1 Step -span%
            If sort$(j%) <= sort$(j% + span%) Then Exit For
            'swap array elements that are out of order
            temp$ = sort$(j%)
            sort$(j%) = sort$(j% + span%)
            sort$(j% + span%) = temp$
        Next j%
    Next i%
    span% = span% \ 2
Loop
End Sub

```

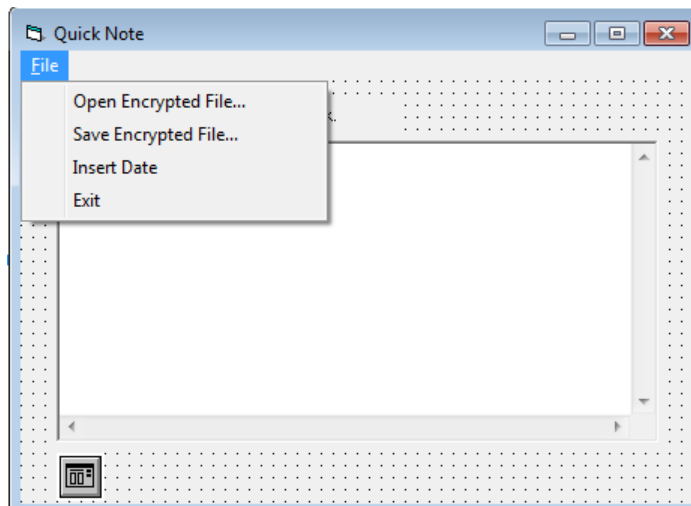
Metode pengurutan ini secara terus-menerus membagi daftar elemen ke dalam subdaftar yang ukurannya setengah lebih kecil. Pengurutan kemudian membandingkan bagian atas dan bawah subdaftar untuk melihat apakah elemen-elemennya tidak terurut. Jika bagian atas dan bawah tidak terurut, maka akan ditukar. Hasil akhirnya adalah sebuah array (sort\$) yang diurutkan secara alfabetis pada aturan menurun. Untuk mengubah aturan pengurutan, ubahlah operator relasional (ubah <= menjadi >=).

### **Melindungi Teks dengan Enkripsi**

Anda bisa muli menulis rutin enkripsi sederhana yang mengubah kode-kode ASCII dokumen Anda dan “mengacak” teks untuk menyembunyikannya dari orang jahil. Proses ini (disebut enkripsi atau encryption) secara matemats mengubah karakter dlam file, membuat enkripsi dengan sukses, Anda juga perlu mengembalikan prosesnya.

### **Menekripsi teks dengan mengubah kode ASCII**

1. Bukalah kembali program Qnote
2. Ubahlah form menjadi seperti berikut:



3. Ketikkan pernyataan berikut:

```

Private Sub mnuOpenItem_Click()
    Wrap$ = Chr$(13) + Chr$(10) 'create wrap character
    CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
    CommonDialog1.ShowOpen      'display Open dialog box
    If CommonDialog1.FileName <> "" Then
        Form1.MousePointer = 11 'display hourglass
        Open CommonDialog1.FileName For Input As #1 'open file
        On Error GoTo Problem: 'set error handler
        Do Until EOF(1)         'copy each line of text to
            Line Input #1, LineOfText$ 'AllText$ string
            AllText$ = AllText$ & LineOfText$ & Wrap$
        Loop
        'now, decrypt string by subtracting one from ASCII code
        decrypt$ = "" 'initialize string for decryption
        charsInFile = Len(AllText$) 'get length of string
        For i% = 1 To charsInFile 'loop once for each char
            letter$ = Mid(AllText$, i%, 1) 'get char with Mid
            decrypt$ = decrypt$ & Chr$(Asc(letter) - 1) 'subtract 1
        Next i% 'and build new string
        txtNote.Text = decrypt$ 'then display converted string
        txtNote.Enabled = True 'and enable scroll bars
        lblFile.Caption = CommonDialog1.FileName 'set caption
    Cleanup: 'when finished...
        Form1.MousePointer = 0 'reset mouse
        Close #1 'close file
        CommonDialog1.FileName = "" 'clear filename
    End If
    Exit Sub
Problem: 'if there is a problem, display appropriate message
MsgBox "Error Opening File", , Err.Description
lblFile.Caption = "" 'remove caption
txtNote.Text = "" 'clear text box
Resume Cleanup: 'finally, finish with Cleanup routine

```

```

Private Sub mnuItemSave_Click()
    CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
    CommonDialog1.ShowSave 'display Save dialog
    If CommonDialog1.FileName <> "" Then
        Form1.MousePointer = 11 'display hourglass
        lblFile.Caption = CommonDialog1.FileName
        'save text with encryption scheme (ASCII code + 1)
        encrypt$ = "" 'initialize encryption string
        charsInFile% = Len(txtNote.Text) 'find string length
        For i% = 1 To charsInFile% 'for each character in file
            letter$ = Mid(txtNote.Text, i%, 1) 'read next char
            'determine ASCII code of char and add one to it
            encrypt$ = encrypt$ & Chr$(Asc(letter$) + 1)
        Next i%
        Open CommonDialog1.FileName For Output As #1 'open file
        Print #1, encrypt$ 'save encrypted text to file
        txtNote.Text = encrypt$
        Close #1 'close file
        CommonDialog1.FileName = "" 'clear filename
        Form1.MousePointer = 0 'reset mouse
    End If

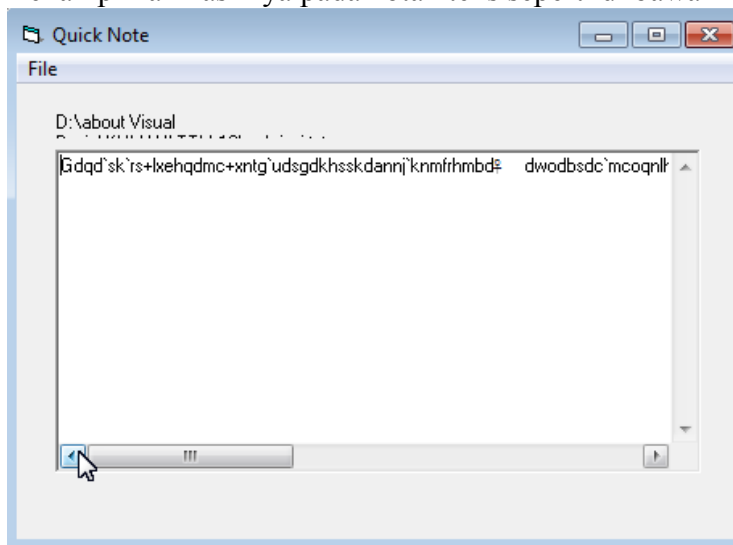
```

4. Ketikkan teks berikut

Here at last, my firend, you have the little book along since expected and promised, a little book on vast matter, namely, "on my own ignorance and that of many others."

Francesco Petraca, c.1368

5. Simpan dalam disk dengan nama padua.txt
6. Pada saat Anda menyimpan file teks, program akan mengacak kode ASCII dan menampilkan hasilnya pada kotak teks seperti di bawah ini:



7. Untuk mengemblikan fie ke bentuk aslinya, pilih perintah Open Encrypt File, lalu buka file padua.txt

### Memeriksa kode program encrypt

1. Buka event procedure mnuItemSave pada jendela Code untuk melihat kode program yang menghasilkan enkripsi yang telah Anda lihat di atas. Fungsi Asc dan Chr serta loop For, ditambahkan satu angka ke dalam ASCII untuk setiap karakter pada kotak teks, lalu menyimpan string yang telah dienkrpsi tersebut ke dalam file teks tertentu.

```

'save text with encryption scheme (ASCII code + 1)
encrypt$ = "" 'initialize encryption string
charsInFile% = Len(txtNote.Text) 'find string length
For i% = 1 To charsInFile% 'for each character in file
    letter$ = Mid(txtNote.Text, i%, 1) 'read next char
    'determine ASCII code of char and add one to it
    encrypt$ = encrypt$ & Chr$(Asc(letter$) + 1)
Next i%
Open CommonDialog1.FileName For Output As #1 'open file
Print #1, encrypt$ 'save encrypted text to file
txtNote.Text = encrypt$

```

2. Bukalah event procedure mnuOpenItem. Kode program ini hampir mirip dengan yang terdapat pada perintah Save, tetapi ia mengurangi 1 dari kode ASCII untuk setiap huruf, bukan menambahkannya.

```

'now, decrypt string by subtracting one from ASCII code
decrypt$ = "" 'initialize string for decryption
charsInFile = Len(AllText$) 'get length of string
For i% = 1 To charsInFile 'loop once for each char
    letter$ = Mid(AllText$, i%, 1) 'get char with Mid
    decrypt$ = decrypt$ & Chr$(Asc(letter) - 1) 'subtract 1
Next i% 'and build new string
txtNote.Text = decrypt$ 'then display converted string

```

### Mengekripsi dengan operator XOR

1. Masih dengan form yang sama dengan program sebelumnya.
2. Buatlah pernyataan berikut pada event procedure mnuOpenItem

```

Wrap$ = Chr$(13) + Chr$(10) 'create wrap character
CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
CommonDialog1.ShowOpen 'display Open dialog box
If CommonDialog1.FileName <> "" Then
    'get encryption code to convert coded file to text
    code = InputBox("Enter encryption code", , 1)
    If code = "" Then Exit Sub 'if Cancel chosen, exit sub
    Form1.MousePointer = 11 'display hourglass
    Open CommonDialog1.FileName For Input As #1 'open file
    On Error GoTo Problem: 'set error handler
    decrypt$ = "" 'initialize string for decryption
    Do Until EOF(1) 'until end of file reached
        Input #1, Number& 'read encrypted numbers
        e$ = Chr$(Number& Xor code) 'convert with Xor
        decrypt$ = decrypt$ & e$ 'and build string
    Loop
    lblFile.Caption = CommonDialog1.FileName 'set caption
    txtNote.Text = decrypt$ 'display converted string
    txtNote.Enabled = True 'and enable scroll bars
CleanUp: 'when finished...
    Form1.MousePointer = 0 'reset mouse
    Close #1 'close file
    CommonDialog1.FileName = "" 'clear filename
End If
Exit Sub
Problem: 'if there is a problem, display appropriate message
If Err.Number = 5 Then 'Chr$ problem means bad key
    MsgBox ("Incorrect Encryption Key")
Else 'for other problems (like file too big) show error
    MsgBox "Error Opening File", , Err.Description
End If
Resume CleanUp: 'finally, finish with CleanUp routine
End Sub

```

3. Buatlah pernyataan berikut pada event procedure mnulItemSave

```

Private Sub mnuItemSave_Click()
    CommonDialog1.Filter = "Text files (*.TXT)|*.TXT"
    CommonDialog1.ShowSave 'display Save dialog
    If CommonDialog1.FileName <> "" Then
        'get encryption code and use it to encrypt file
        code = InputBox("Enter Encryption Code", , 1)
        If code = "" Then Exit Sub 'if Cancel chosen, exit sub
        Form1.MousePointer = 11 'display hourglass
        charsInFile% = Len(txtNote.Text) 'find string length
        Open CommonDialog1.FileName For Output As #1 'open file
        For i% = 1 To charsInFile% 'for each character in file
            letter$ = Mid(txtNote.Text, i%, 1) 'read next char
            'convert to number w/ Asc, then use Xor to encrypt
            Print #1, Asc(letter$) Xor code; 'and save in file
        Next i%
        Close #1 'close file when finished
        CommonDialog1.FileName = "" 'clear filename
        Form1.MousePointer = 0 'reset mouse
    End If
End Sub

```

Pada perintah Print #1, operator Xor digunakan untuk mengubah setiap huruf pada kotak teks menjadi kode numerik, yang kemudian disimpan dalam disk.

4. Ketikkan teks berikut untuk file yang akan dienkripsi:  
 Rothair's Edict (Lombard Italy, c.643)  
 296. on Stealing Grapes. He who takes more than three grapes  
 from another man's vine shall pay six soldi as composition. He  
 who takes less than three shall bear no guilt.
5. Pada menu File, klik perintah Save Encrypted file, dan simpanlah file tersebut dengan nama oldlaws.txt
6. Jalankan program
7. Ketikkan 500 atau angka lalu tekan Enter
8. Hapus teks, restore file yang telah dienkripsi.
9. Klik perintah Open Encrypted File
10. Ketikkan kode enkripsi.
11. Bukalah file terenkripsi menggunakan Notepad.
12. Perhatikan apa yang terjadi.