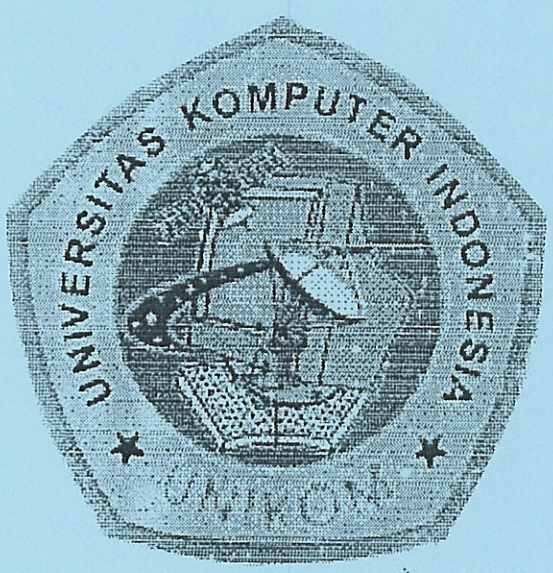
UNIT PELAKSANA TEKNIS PERPUSTAKAAN * UNIKOM *

DAFTAR PUSTAKA

KADIVRE III, *Produk Jasa Telekomunikasi*, TELKOM Bandung, 2000

TKD-Voice Department, *Smart Voice Processing System Operation Manual*, TKD-Voice Department, 1997


UNIVERSITAS KOMPUTER INDONESIA

UNIT PELAKSANA TEKNIS PERPUSTAKAAN
UNIKOM

PANDUAN PEMROGRAMAN SCRIPT

1. Komentar

Komentar atau yang biasa kita kenal dengan *comment* sangat berguna untuk mengingatkan kita akan fungsi-fungsi bagian-bagian script yang kita tulis. Namun ada keterbatasan bahwa yang berlaku hanya berupa komentar *single-line*, dimulai dengan karakte ';' hingga akhir baris.

Contoh : Substr(S$,N$,1,3) ; **Copy 3 karakter S$ dari posisi 1**

2. Tipe data

Tipe data yang tersedia adalah :

a. String

Tipe data String dimulai dan diakhiri dengan aphosthrophe (")

Misalnya : File$="USER.VOC"

Identifier yang dipakai harus diakhiri dengan karakter '$' pada waktu deklarasinya.

Missal : Declare **Name$, Account$, S$**

Catatan :

panjang maksimum konstanta string adalah 65359 karakter.

b. Integer

Tipe data Integer adalah semua nilai numeric

Missal : RetryCount=**25**

Berbeda dengan tipe data String, identifier yang dipakai tidak boleh diakhiri dengan karakter '$' pada waktu deklarasinya.

Misalnya : Declare **Name, Account, S**

Catatan :

Rangenya adalah dari –32768 sampai 32767.

3. Identifier

Adalah suatu token yang mewakili nama suatu variable ataupun subroutine/function. Identifier harus dimulai dengan alphabet dan selanjutnya dapat dikombinasikan dengan alphanumeric. Panjang maksimum dari suatu identifier adalah 32 karakter. Identifier tidak mengindahkan perbedaan huruf kecil dan kapital.

4. Operator

Operator berguna untuk mewakili operasi-operasi yang akan anda lakukan dengan dua operand atau lebih.

Secara umum syntaxnya adalah :

Operand-kiri **OPERATOR** *operand-kanan*

atau

operand-kiri **OPERATOR** [{*operand-kanan*}]

batasan : operand-kiri dan operand-kanan harus se-tipe.

Menurut jenisnya, operator dibedakan menjadi 2, yaitu :

a. Arithmatic operator

Syntax : *operand-kiri* **OPERATOR** *operand-kanan*

Pengisian, direpresentasikan dengan '='

Berlaku untuk tipe data String maupun Integer.

Contoh :

1. Kode = 3
2. Account$ = "0109281"

Pengisian sesudah penjumlahan, direpresentasikan dengan '+='

Berlaku untuk tipe data String maupun Integer.

Contoh :

1. I + = 3
2. Account$ + = KodeCabang$

Pengisian sesudah pengurangan, direpresentasikan dengan '-='

Berlaku hanya untuk tipe data Integer.

Contoh :

I - = 3

Penjumlahan, direpresentasikan dengan '+'

Berlaku untuk tipe data String maupun Integer.

Contoh :

1. I + = 3 + 5 + 7
2. Notelp$ + = "567" + "2342"

Pengurangan, direpresentasikan dengan '-'

Berlaku hanya untuk tipe data Integer.

Contoh :

I + = 7 – 2 – 4

b. Comparative (perbandingan) operator

Lebih besar, direpresentasikan oleh '>'

Contoh :

```
if I > 5
...
endif
```

Kecil dari, direpresentasikan oleh '<'

Contoh :

```
if I < 5
...
endif
```

Lebih besar atau sama dengan, direpresentasikan oleh '>='

Contoh :

```
if A >= 10
...
endif
```

Kecil dari atau sama dengan, direpresentasikan oleh '<='

Contoh :

```
if A <= 10
...
endif
```

Sama dengan, direpresentasikan oleh '= ='

Contoh :

```
if S = = 5
...
endif
```

Tidak sama dengan, direpresentasikan oleh '!='

Contoh :

```
if S != 5
...
endif
```

5. **Ekspresi**

   Syntax : *operand-kiri* **COMPARATIVE-OPERATOR** *operand-kanan*

   Operand-kiri harus merupakan suatu identifier dari variable.

   Operand-kanan dapat berupa nilai konstanta, variable, ataupun function.

   Kaidah yang berlaku :

   Operand-kiri dan operand-kanan harus mempunyai tipe yang sama, artinya apabila operand-kiri adalah variable String maka operand-kanan harus merupakan konstanta String, atau variable String, demikian juga untuk operand Integer.

   Ekspresi dapat lebih dari satu, dengan memakai *CONJUNCTION* atau konjungsi sebagai berikut :

   1. **AND** untuk konjungsi **Dan**
   2. **OR** untuk konjungsi **Atau**

6. **Reserve words**

   Reserve words adalah syntax yang dipergunakan didalam pembuatan script. Reserver word ini hanya dipakai sebagai pengenal bagi kompiler, tidak dapat digunakan sebagai variable atau identifier yang dipakai dan diolah oleh user.

a. **Sub**

Digunakan untuk menandai awal dari sebuah subroutine/procedure. Sebuah procedure bias tanpa parameter, namun dapat juga memiliki satu atau beberapa parameter. Apabila ada parameter yang dipakai, maka parameter-parameter tersebut dituliskan di belakang nama identifier procedure dengan dipisahkan tanda koma satu dengan lainnya.

Syntax : **Sub <Identifier> [{parameter}]**

Catatan : sebagai procedure utama dalam sebuah VPS script adalah **Sub Main** yang keberadaannya adalah mutlak. Selain itu, sebuah VPS script juga memiliki sebuah procedure yang dipanggil ketika terjadi *on-hook* yang dinamai **Sub Exitproc**. Berbeda dengan Main, Exitproc hanya bersifat optional. Untuk diketahui, Main dan Exitproc tidak memiliki parameter sama sekali.

b. **EndSub**

Untuk menandai akhir dari subroutine/procedure.

Syntax : **EndSub**

Contoh : Sub HitungUang Rupiah$, Sen$

...

EndSub

c. **CallSub**

Dalam metode pemrograman terstruktur, sebuah program yang baik hanya memiliki sebuah pintu masuk dan hanya sebuah pintu

keluar. Untuk itu, tidak dikenal instruksi yang melompat dari satu bagian program ke program lainnya. Untuk memanggil suatu subroutine/procedure dari program tersebut digunakanlah **CallSub**. Setelah procedure yang dipanggil selesai, program akan kembali ke instruksi yang berada dibawah **CallSub**. Dengan demikian kaidah satu pintu masuk dan satu pintu keluar tetap terjaga.

Syntax : **CallSub <NamaSubRoutine> [{parameter}]**

Contoh : **CallSub** Time ; melompat ke sub time

d. Declare

Untuk proses-proses perhitungan dan manipulasi data, dibutuhkan variable-variable sebagai *temporary storage*. Untuk mendeklarasikan dan memesan storage tersebut, kita gunakan **Declare**. Dengan sebuah perintah **Declare** kita dapat memesan satu atau lebih sekaligus.

Syntax : **Declare <Identifier[(size)] [,Identifier...]>**

Contoh : **Declare** Account$, Buffer$(2000), Number, I

e. If

Pada suatu saat, program perlu melakukan seleksi untuk menentukan percabangan. Pada sebuah percabangan sederhana, **If** adalah pilihan yang tepat untuk memulai sebuah ekspresi percabangan. Apabila nilai ekspresi adalah **True**, maka instruksi-instruksi yang berada dibawahnya, sebelum adanya instruksi **Elseif** atau **EndIf**, akan dieksekusi.

Syntax : **If <Ekspression [Conjunction <Expression...>]>**

Contoh : **If** A$ = = "1"

...

EndIf

f. **ElseIf**

Instruksi ini terutama berguna bila pada saat ekspresi **If** bernilai **False** perlu dilakukan pengujian lagi dan instruksi-instruksi yang dieksekusi pun berbeda. Adapun lingkup instruksi yang dieksekusi adalah yang berada dibawah **ElseIf** hingga sebuah **ElseIf** lainnya atau **EndIf**. Dengan demikian jelaslah bahwa **ElseIf** bersifat optional namun dapat lebih dari satu dalam sebuah percabangan **If**.

Syntax : **ElseIf <Expression [Conjunction {Expression}]>**

Contoh : If A$ = = "1"

Print("A bernilai 1")

ElseIf A$ = = "2"

Print("A bernilai 2")

EndIf

g. **Else**

Sebagaimana halnya **ElseIf**, **Else** juga bersifat optional. Namun, dalam sebuah percabangan **If**, hanya sebuah **Else** yang boleh muncul. Semua kondisi yang gagal dipenuhi oleh **If** maupun **ElseIf** akan terjaring disini. Lingkup instruksi-instruksi yang akan dieksekusi adalah yang berada dibawah **Else** hingga **EndIf**.

Syntax : **Else**

Contoh :

```
If A$ = = "1"
        Print("A bernilai 1")
ElseIf A$ = = "2"
        Print("A bernilai 2")
Else
        Print("A tidak diketahui")
EndIf
```

h. EndIf

Menandai akhir dari sebuah **If**.

Syntax : **EndIf**

Contoh :

```
If A$ = = "1"
        Print("A bernilai 1")
ElseIf A$ = = "2"
        Print("A bernilai 2")
Else
        Print("A tidak diketahui")
EndIf
```

Catatan : sebuah percabangan **If** dapat berisi percabangan-percabangan **If** lainnya.

i. **SelectCase**

Adakalanya, tidaklah efektif menggunakan **ElseIf** untuk percabangan majemuk tertentu. Untuk itu tersedia sebuah instruksi yang khusus diperuntukkan percabangan majemuk.

Syntax : **SelectCase <IntegerVariable>**

Contoh : **SelectCase** A

j. **Case**

Untuk menentukan kondisi yang diinginkan dengan seleksi majemuk **SelectCase**. Dalam sebuah percabangan **SelectCase**, minimal terdapat sebuah **Case** dengan maksimal tak terbatas. Lingkup instruksi yang akan dieksekusi, bila **Case** memenuhi syarat adalah instruksi yang berada dibawah **Case** hingga instruksi **Case** berikutnya atau instruksi **ElseCase** atau akhir **SelectCase** yang ditandai dengan instruksi **EndCase**.

Syntax : **Case <Value [-Value]>**

Contoh : SelectCase A

Case 1

... ; job no1

Case 2

... ; job no 2

EndCase

k. **ElseCase**

Instruksi ini menjaring semua kondisi yang tidak memenuhi Case yang tersedia. Lingkup instruksi yang akan dieksekusi adalah yang berada dibawah instruksi ini hingga **EndCase**.

Syntax : **ElseCase**

Contoh : SelectCase A

Case 1

... ; job no1

Case 2

... ; job no 2

ElseCase

... ; another job

EndCase

l. **EndCase**

Instruksi ini menandai berakhirnya blok percabangan majemuk yang dimulai dengan **SelectCase**. Sebuah percabangan **SelectCase** juga dapat berisi percabangan majemuk lainnya.

Syntax : **EndCase**

Contoh : SelectCase A

Case 1

... ; job no1

Case 2

... ; job no 2

ElseCase

... ; do anything else

EndCase

m. **For** dan **TO**

Memulai suatu proses iterasi yang sudah pasti dan sudah ditentukan jumlah iterasinya.

Syntax : **For** < x = ***Start*** **To** ***Max***>

Keterangan : x : Variable Integer sebagai counter

Start : Nilai Integer untuk startup counter

Max : Nilai akhir iterasi

Contoh : **For** c = 1 **To** 10

n. **EndFor**

Menandakan akhir dari suatu proses iterasi yang dimulai dengan **For**.

Syntax : **EndFor**

Contoh : For c = 1 To 10

... ; do something

EndFor

o. **Repeat**

Memulai blok dari suatu iterasi, daimana iterasi diharapkan terjadi minimal satu kali, hingga kondisi tidak memungkinkan lagi.

Syntax : **Repeat**

p. **Until**

Mengakhiri proses iterasi yang diawali dengan **Repeat**. Iterasi akan berakhir bila ekspresi yang berada pada **Until** telah bernilai **True**.

Syntax : **Until <Expression [Conjunction {Expression}]>**

Contoh :
```
Repeat
...      ; something to be done at least once
Until X = = 5 or PIN$ = = "111"
```

q. **While**

Memulai blok dari suatu iterasi dengan terlebih dahulu memeriksa apakah kondisi memungkinkan terjadinya iterasi tersebut, sehingga blok tersebut bias jadi tidak pernah dieksekusi.

Syntax : **While <Expression [Conjunction {Expression}]>**

r. **EndWhile**

Mengakhiri blok yang mengalami iterasi yang diawali dengan **While**.

Syntax : **EndWhile**

Contoh :
```
While X = = 5 or PIN$ = ="111"
...
EndWhile
```

7. **Internal function**

Internal function adalah fungsi yang sudah ada pada VpsComp sehingga bisa langsung digunakan.

a. **GetTime**

Mengambil jam dari sistem.

Syntax : **GetTime(StringVariable)**

Contoh : **GetTime**(Time$)

Catatan : Hasil dari **GetTime** mempunyai format "HHMMDD"

Dimana HH = Jam, MM = Menit, DD = Detik

b. **GetDate**

Mengambil tanggal dari sistem.

Syntax : **GetDate(StringVariable)**

Contoh : **GetDate**(Date$)

Catatan : Hasil dari **GetDate** mempunyai format "YYYYMMHH"

Dimana YYYY = Tahun, MM = Bulan, HH = Hari

8. **Function**

Beberapa dari function-function di bawah ini mempunyai return value sebagai berikut :

0 = Operation Completed

1 = Line is busy (Dial Only)

2 = Target number is Invalid (Dial Only)

3 = No signal tone generated (Dial Only)

4 = No answer (Dial Only)

5 = No dial tone (Dial Only)

6 = Operation Time Out

7 = Remote Hang Up

8 = DTMF detected

9 = No ring back (Dial Only)

10 = Fax tone detected (Dial Only)

225 = Unknown error

a. **GetDTMF**

menunggu pemasukan DTMF dari line tersambung.

Syntax : **GetDTMF (TrgtStrVar, IntCount, DTMFAccept, DTMFTerminator)**

Dimana :

1. TrgtStrVar adalah String Buffer untuk menampung DTMF yang didapat.
2. IntCount adalah Panjang DTMF yang akan ditampung.
3. DTMFAccept adalah DTMF-List yang diterima.
4. DTMFTerminator adalah DTMF-List untuk mengakhiri proses.

Return : Operation Completed, DTMF Detected, Operation Time Out.

Catatan : DTMF-List adalah daftar DTMF keypad yang akan digunakan, valid DTMF adalah : 0,1,…,9,#,*.

Cara penulisan DTMF adalah sebagai berikut :

a. 012345 berarti 0,1,2,3,4,5
b. 0-5#* berarti 0,1,2,3,4,5,#,*
c. @ berarti 0,1,…,9,#,*

b. **Play**

Instruksi untuk melafalkan satu atau lebih file message. Pelafalan message dapat dihentikan oleh pemakai dengan menekan satu atau lebih tombol DTMF. Bila DTMFTerminators dikosongkan, berarti pelafalan tidak dapat dihentikan.

Syntax : **{VarInt} = Play (FileName, DTMFTerminators)**

Dimana :

1. VarInt adalah variable penampung hasil dari play
2. FileName adalah variable string atau Direct String yang berisi nama file yang akan di-play. Jika lebih dari satu file, pisahkan antara tiap file dengan spasi.
3. DTMFTerminators adalah DTMF-List dimana play akan dihentikan apabila salah satu dari DTMF-List ditekan oleh pemakai.

Return : Operation Completed, DTMF Detected

Contoh : **Play** ("Morning Greet",@)

9. **Miscelanous function**

Di bawah ini adalah fungsi-fungsi tambahan yang digunakan untuk mengoptimalkan dan membantu keseluruhan script.

a. **SetMsgPath**

Merubah sistem default Message File Path.

Syntax : **SetMsgPath(StrVal)**

Dimana, StrVal adalah Direct String Value.

Contoh : **SetMsgPath** ("C:\ENG") Play("Hello.Wav",@)

; sama saja dengan

Play ("C:\ENG\Hello.Wav",@)

TELKOM

KANTOR DAERAH PELAYANAN TELEKOMUNIKASI CIANJUR

JL. IR . H . JUANDA NOMOR . 9 TELEPON. (0263) 261301 Fax : 262401

SURAT KETERANGAN

Nomor : Tel. 001 / PD520 / RE3 - DO3C / 2002

Diberikan kepada

FAUZARRAMADHAN SYAM

NIM : 10700110

Yang bersangkutan telah mengikuti Praktek Kerja Lapangan (PKL) pada PT. TELKOM KANDATEL CIANJUR

Lama Waktu : 21 Hari Kerja mulai tanggal, 01 September s/d 30 September 2002

Sebagai laporan Akhir yang bersangkutan telah menyusun buku laporannya dengan bantuan Teknik SISIFO

Demikian surat keterangan ini diberikan untuk dapat dipergunakan sebagaimana mestinya

Cianjur, 22 Nopember 2002

TELKOM

DEDI EFFENDI

KABAG SDM

UNIVERSITAS KOMPUTER INDONESIA
JURUSAN TEKNIK INFORMATIKA
KAMPUS I : JL. DIPATI UKUR 112 TELP. (022) 2504119, 2503430 BANDUNG 40132
KAMPUS II : JL. DIPATI UKUR 116 TELP. (022) 2506634, 2503371 BANDUNG 40132
KAMPUS III : JL. DIPATI UKUR 102 TELP. (022) 2506637, FAX. 2533754 BANDUNG 40132
KAMPUS IV : JL. DIPATI UKUR 114 TELP. (022) 2503054, 2508805 BANDUNG 40132

PENILAIAN KERJA PRAKTEK
UNIVERSITAS KOMPUTER INDONESIA

Nama Mahasiswa/i : Fauzar Ramadhan. S
NIM : 10700110
Jurusan : Teknik Informatika
Nama Perusahaan : PT. TELKOM KANDATEL Cianjur
Alamat Perusahaan & Telepon : Jl. Jr. H. Juanda no. 9 Cianjur
Waktu Kerja Praktek : 09.00 - 15.00 WIB
Nama Pejabat Penilai : WIDODO YULIONO
Jabatan : SPE APLIKASI TSFO

NO	ASPEK YANG DINILAI	NILAI	INDEK PRESTASI
1	Disiplin Dalam Kerja	80	A
2	Kemampuan Menyelesaikan Tugas	87	A
3	Prestasi Kerja	83	A
4	Inisiatif	90	A
5	Kepribadian	80	A

JUDUL LAPORAN KERJA PRAKTEK

Program Layanan Terpadu PT. TELKOM

Komentar Pejabat Penilai Atas Prestasi Kerja Mahasiswa/i UNIKOM
Baik. Perlu ditingkatkan kemampuan praktis mhs.

NILAI	INDEKS	KETERANGAN
80 ≤ NA ≤ 100	A	Sangat Baik
70 ≤ NA ≤ 79	B	Baik
55 ≤ NA ≤ 69	C	Cukup
41 ≤ NA ≤ 54	D	Kurang
0 ≤ NA ≤ 40	E	Buruk

CIANJUR, 15 - JANUARI 2002

Mengetahui,
Ketua Jurusan Teknik Informatika

Budhi Irawan S.Si
NIP. 41.27 70 06 003

Koordinator Kerja Praktek

TELKOM

(Nama, Tanda Tangan & Cap)

UNIVERSITAS KOMPUTER INDONESIA
JURUSAN TEKNIK INFORMATIKA
KAMPUS I : JL. DIPATI UKUR 112 TELP. (022) 2504119, 2503430 BANDUNG 40132
KAMPUS II : JL. DIPATI UKUR 116 TELP. (022) 2506634, 2503371 BANDUNG 40132
KAMPUS III : JL. DIPATI UKUR 102 TELP. (022) 2506637, FAX. 2533754 BANDUNG 40132
KAMPUS IV : JL. DIPATI UKUR 114 TELP. (022) 2503054, 2508805 BANDUNG 40132

ABSENSI KEHADIRAN MAHASISWA/I

Nama Perusahaan : PT. TELKOM KANDATEL Cianjur
Alamat Perusahaan : Jl. Ir. H. Juanda no. 3 Cianjur
Pembimbing di Perusahaan : Widodo Siswoyo
Nama Mahasiswa : Fauzar Ramadhan
NIM : 10700110

No	Hari/Tanggal	Jam Masuk	Paraf	Jam Pulang	Paraf	Paraf Pembimbing
1	Senin / 2 Sept 2002	09.00 WIB		15.00 WIB		
2	Selasa / 3 Sept 2002	09.00 WIB		15.00 WIB		
3	~~Kamis~~ Rabu / 4 Sept 2002	09.00 WIB		15.00 WIB		
4	Kamis / 5 Sept 2002	09.00 WIB		15.00 WIB		
5	Jum'at / 6 Sept 2002	09.00 WIB		15.00 WIB		
6	Senin / 9 Sept 2002	09.00 WIB		15.00 WIB		
7	Selasa / 10 Sept 2002	09.00 WIB		15.00 WIB		
8	Rabu / 11 Sept 2002	09.00 WIB		15.00 WIB		
9	Kamis / 12 Sept 2002	09.00 WIB		15.00 WIB		
10	Jum'at / 13 Sept 2002	09.00 WIB		15.00 WIB		
11	Senin / 16 Sept 2002	09.00 WIB		15.00 WIB		
12	Selasa / 17 Sept 2002	09.00 WIB		15.00 WIB		
13	Rabu / 18 Sept 2002	09.00 WIB		15.00 WIB		
14	Kamis / 19 Sept 2002	09.00 WIB		15.00 WIB		
15	Jum'at / 20 Sept 2002	09.00 WIB		15.00 WIB		
16	Senin / 23 Sept 2002	09.00 WIB		15.00 WIB		
17	Selasa / 24 Sept 2002	09.00 WIB		15.00 WIB		
18	Rabu / 25 Sept 2002	09.00 WIB		15.00 WIB		
19	Kamis / 26 Sept 2002	09.00 WIB		15.00 WIB		
20	Jum'at / 27 Sept 2002	09.00 WIB		15.00 WIB		
21	Senin / 30 Sept 2002	09.00 WIB		15.00 WIB		
22		~~09.00 WIB~~		~~15.00 WIB~~		
23						
24						
25						
26						
27						
28						
29						
30						

Bandung, ... 2002

Mengetahui,
Ketua Jurusan Teknik Informatika

Budhi Irawan, S.Si
NIP. 4127 70 06 003

Koordinator Kerja Praktek

(Y. S. T.)
NIK. 740269