


MAKALAH

UNSUR-UNSUR BAHASA BASIC

M L K UPT PERPUSTAKAAN IKIP PADANG	
DATE MA TEL OKTOBER 1991	
SUMBER H R A HADIAH	
KODIK I KKI	
NOI VE TARIS 1419/HD/91-40(2)	
CALL NO 001.642 NYK 40	

M L K UPT PERPUSTAKAAN
IKIP PADANG

oleh :

Drs. Kasman Rukun, M.Pd

*Disampaikan pada Penataran Penggunaan Komputer
pada Perguruan Dinyyah Putri
Padang Panjang
Tanggal 23 Februari 1991 s/d. 23 Maret 1991*

UNSUR-UNSUR BAHASA BASIC *)

oleh : Drs. Kasman Rukun, MPd.

Seperti bahasa pemrograman lainnya, BASIC mempunyai tata bahasa dan perbendaharaan kata. Tata bahasa merupakan aturan yang berhubungan dengan bentuk-bentuk pernyataan/statement Basic, sedangkan perbendaharaan kata berisi sekumpulan simbol atau tanda-tanda yang digunakan dalam pemrograman BASIC.

Makalah ini akan mempelajari unsur-unsur bahasa BASIC, seperti :Konstanta, variabel, rumus-rumus (ekspresi), fungsi serta penggunaan array dan statement DIM.

1. Konstanta

Dalam BASIC, ada tiga jenis konstanta yaitu :

(1) Konstanta Numerik

Konstanta numerik terdiri dari satu atau lebih bilangan (riil), baik positif maupun negatif yang menggunakan titik desimal (tanda koma) maupun yang tidak menggunakan titik desimal. Bila tanda desimalnya tidak dicantumkan, biasanya tanda desimal itu otomatis ditempatkan sebelah kanan bilangan, misalnya bilangan 20 sama dengan 20.00.

Bilangan riil ini dapat berpresisi tunggal (single precision) atau berpresisi ganda(double precision). Bilangan berpresisi tunggal adalah bilangan yang memiliki salah satu ketentuan dari ketentuan berikut :

- bentuk eksponensial ditandai dengan huruf E
- maksimal terdiri dari tujuh angka

*)Disampaikan pada Penataran Penggunaan Komputer Pada Perguruan Dinyyah Putri Padang Panjang tanggal 23 Februari 1991 s/d 23 Maret 1991

- diakhiri tanda !

Contoh : 2, 73, .87, -6.84, 2.3E5, -1.3E-2

Konstanta numerik yang digunakan dalam BASIC mulai dari 10^{-38} s/d 10^{38} atau ditulis $10^{(-38)} < n < 10^{(38)}$, dimana n adalah konstanta numerik. Besar kecilnya bilangan, ditandai dengan menggunakan pangkat positif atau pangkat negatif. Bentuk umumnya adalah $(\pm)X.XXXE(\pm)nn$, dimana + adalah tanda untuk bilangan positif dan - adalah tanda untuk bilangan negatif, X adalah bilangan dari 0 sampai 9, E menunjukkan "perkalian 10 pangkat" dan nn adalah pangkatnya (10 pangkat nn), misalnya : 2E-5 artinya 2×10^{-5} atau 0,00002.

Bilangan yang berpresisi ganda adalah bilangan yang memiliki salah satu ketentuan berikut :

- minimal terdiri dari delapan angka
- diakhiri tanda #
- bentuk eksponensial ditandai dengan huruf D.

Contoh : 12.3#, 1.23D-2, 234123567

(2) Konstanta Integer

Konstanta "integer" adalah semua bilangan yang ditulis tanpa menggunakan tanda desimal. Tanda persen (%) digunakan untuk membedakan konstanta integer dari konstanta numerik. Range untuk konstanta integer adalah -32768 sampai 32767%.

Contoh konstanta integer : 24%, 12345%, -7%, 906%.

(3) Konstanta String

Konstanta string (disebut juga literal) terdiri dari susunan karakter alphanumerik dan/atau beberapa karakter khusus yang dibatasi oleh tanda kutip (tunggal maupun ganda).

Selama pelaksanaan program (saat dieksekusi), karakter-karakter yang berada diantara tanda kutip akan dicetak seperti apa adanya.

Contoh : 'TEMPAT TINGGAL', "KERAMAIAN", "BANDUNG 40113"

akan dicetak berupa :

TEMPAT TINGGAL , KERAMAIAN dan BANDUNG 40113

2 Variabel

Variabel adalah kuantitas yang dapat diubah-ubah selama pelaksanaan program, yang disimpan dalam lokasi memory. Variabel dapat berupa bilangan atau huruf. Ada 4 jenis variabel, yaitu :

(1) Variabel Numerik

Variabel numerik hanya dapat berisi konstanta numerik yang dituliskan berupa huruf tunggal atau huruf tunggal yang diikuti sebuah bilangan bulat.

Contoh : A1, C, R7, Z2 dst.

(2) Variabel Integer

Variabel integer dituliskan berupa huruf tunggal atau huruf tunggal yang diikuti sebuah bilangan dan diakhiri dengan tanda %. Nilai-nilai yang digunakan untuk variabel integer haruslah dalam bentuk konstanta integer, dalam hal ini variabel integer tidak menggunakan bilangan pecahan. Jika ditulis A1% = 19.5 nilai yang ditunjukkan untuk A1 adalah 19, ini adalah contoh penulisan variabel integer yang salah.

Contoh penulisan variabel integer yang benar : A6%, E3%.

(3) Variabel String

Penulisan variabel string sama dengan penulisan variabel

numerik yang diikuti dengan tanda dollar (\$). Variabel string terdiri dari karakter alphanumerik, tetapi bilangan yang digunakan hanya bilangan bulat, misalnya : B\$, C2\$, S3\$. Dapat juga dituliskan seperti berikut : B\$="SAYA", C2\$="MAKAN" dan S3\$="NASI". Panjang variabel string berkisar dari 0 sampai 255 karakter dan karakter-karakter yang boleh digunakan dalam nama variabel adalah huruf, bilangan, titik desimal dan karakter-karakter khusus yang digunakan untuk memberi tanda/ciri variabel tersebut.

(4) Variabel Subscript

Variabel subscript adalah variabel dengan satu atau dua buah subscript yang berupa variabel numerik, variabel integer atau variabel string. Subscript-nya berupa suatu ekspresi positif dengan range 0 s/d 32767.

Kumpulan data yang tersusun secara matrik dapat kita simpan dalam suatu variabel yang disebut array. Ada dua jenis array, yaitu array satu dimensi dan array dua dimensi. Array satu dimensi merupakan jajaran variabel, sedangkan array dua dimensi merupakan suatu matriks yang terdiri dari baris dan kolom.

Array mempunyai nama dan subscript. Pada array dua dimensi, subscript digunakan untuk membedakan elemen-elemen pada array dan untuk menunjukkan lokasi dalam baris dan kolom. Penulisan subscript selalu dipisahkan dengan tanda koma dan berada dalam tanda kurung.

Contoh : A(1), A\$(8), B(12,16), C%(2,9) dst.

A, A\$, B dan C% adalah nama array dan bilangan-bilangan yang berada dalam tanda kurung adalah subscript-nya. Jika dalam

tanda kurung tersebut terdapat dua buah subscript, subscript pertama menunjukkan nomor baris dan subscript kedua menunjukkan nomor kolom.

Untuk mendefinisikan array dalam suatu program, digunakan pernyataan DIM. Bentuk penulisannya :

nomor program DIM variabel (integer 1, integer 2)

Misalkan kita akan mendefinisikan suatu array dengan perintah: DIM A(2,3), hasilnya akan berupa array 3x4 seperti digambarkan dalam Tabel 4.1, dari contoh ini dapat kita lihat, variabel array selalu dihitung dari nol (0). A adalah nama array, di dalam array A terdapat 9 elemen, yaitu A(0,0) sampai A(2,3) dengan angka terbesar adalah 3.

Table 1 : Array 3x4

Kolom		0	1	2	3
Baris	0	A(0,0)	A(0,1)	A(0,2)	A(0,3)
	1	A(1,0)	A(1,1)	A(1,2)	A(1,3)
	2	A(2,0)	A(2,1)	A(2,2)	A(2,3)

Catatan :

- Integer 2, hanya dipakai untuk array dua dimensi.
- DIM digunakan untuk menyediakan tempat untuk variabel-variabel yang merupakan elemen atau anggota jajaran variabel serta memberikan nilai awal 0 bagi seluruh variabel.
- Jika pernyataan DIM tidak digunakan, maka nilai subscript maksimum yang dapat digunakan adalah 10 untuk array satu dimensi dan (10,10) untuk array dua dimensi.
- Pernyataan DIM harus diberikan sebelum jajaran variabel

yang bersangkutan digunakan.

- Pernyataan DIM dalam satu jajaran variabel, hanya boleh diberikan satu kali.

Selain menggunakan karakter-karakter khusus seperti \$, %, .!, dan # untuk mendefinisikan jenis-jenis variabel, dapat juga digunakan pernyataan DEF (=DEFine).

Contoh :

```
DEFINT <huruf1-huruf2,huruf3,...>
```

- INT kependekan dari INTeger
- Fungsinya untuk mendefinisikan/mendeklarasikan variabel bilangan bulat

```
DEFSNG <huruf1-huruf2,huruf3...>
```

- SNG kependekan dari SiNGle precision
- Fungsinya untuk mendefinisikan variabel bilangan riil berpresisi tunggal

```
DEFDBL <huruf1-huruf2,huruf3...>
```

- DBL kependekan dari DouBLE precision
- Digunakan untuk mendefinisikan variabel bilangan riil berpresisi ganda

```
DEFSTR <huruf1-huruf2,huruf3...>
```

- STR kependekan dari STRing
- Untuk mendefinisikan variabel string

Catatan :

- Pendefinisian jenis variabel menggunakan pernyataan di atas, tidak akan mengubah jenis variabel yang didefinisikan dengan menggunakan karakter-karakter \$, %, ! dan #.
- <huruf1-huruf2,huruf3...> merupakan huruf awal nama-nama

variabel yang hendak didefinisikan jenis-jenisnya.

Contoh :

Cara I:

```
10 DEFINT A-D
20 DEFSNG E-H
30 DEFDBL I-L
40 DEFSTR M-P
50 B = 4/7
60 G = 4/7
70 K = 4/7
80 M = "Alamat:"
90 Print B
100 Print G
110 Print K
120 Print M
RUN
```

```
1
.5714286
.5714285714285714
OK
```

Cara II:

```
10 B% = 4/7
20 G! = 4/7
30 K# = 4/7
40 M$ = "Alamat:"
50 Print B
60 Print G
70 Print K
80 Print M
RUN
```

```
1
.5714286
.5714285714285714
OK
```

Pada cara I, semua variabel yang dimulai dari huruf A sampai D akan berupa bilangan bulat, variabel yang dimulai dari huruf E sampai H berupa bilangan presisi tunggal (single precision), sedangkan variabel dengan huruf I sampai L adalah bilangan presisi ganda (double precision) dan huruf M sampai P berupa variabel string. Untuk mendefinisikan variabel string sebaiknya digunakan tanda \$, seperti pada cara II, agar perbedaannya terlihat dengan jelas, terutama jika programnya cukup panjang.

3. Ekspresi-ekspresi BASIC

Dalam bahasa BASIC, digunakan simbol-simbol khusus yang disebut operator, untuk melaksanakan operasi aritmatik. Operator ini digunakan untuk menghubungkan bilangan-bilangan atau variabel bilangan, dan membentuknya menjadi suatu persamaan (ekspresi). Berdasarkan jenis operasinya, kita dapat

menggunakan operator-operator tersebut untuk ekspresi aritmatik, ekspresi relasional dan ekspresi string.

(1) Ekspresi Aritmatik

Operator aritmatik digunakan untuk ekspresi aritmatik. BASIC mempunyai 5 operasi aritmatik dasar yang dijelaskan dalam Tabel 2:

Tabel 2 : Operator Aritmatik

Operator		Contoh	Pengertian	Persamaan Aljabar
Simbol	A r t i			
+	Penjumlahan	$A + B$	Tambahkan A pada B	$a + b$
-	Pengurangan	$A - B$	Kurangkan B dari A	$a - b$
*	Perkalian	$A * B$	Kalikan A dengan B	$a . b$
/	Pembagian	A/B	Bagi A dengan B	a/b
** atau ^	Pemangkatan	A^B	A dipangkatkan dengan B	a^b

Berbeda dengan pembagian biasa yang menggunakan operator pembagi (/), maka pembagian bilangan bulat, yaitu pembagian yang memberikan hasil berupa bilangan bulat menggunakan operator *back slash* (\).

Contoh :

```
10 A = 5/2
20 B = 5
30 C = 12.5/2.31
40 D = 12.5.31
50 PRINT A, B
60 PRINT C, D
RUN.
```

```
2.5          2
5.411256    6
OK
```

Dari contoh di atas, dapat dilihat, jika kita menggunakan operator pembagi bilangan bulat (tanda \), maka hasil-

nya akan berupa bilangan bulat. Pada baris program nomor 40, pembagian bilangan dilakukan dengan membulatkan lebih dahulu penyebut dan pembilangnya kemudian hasil akhirnya juga dibulatkan.

Untuk memperoleh sisa suatu pembagian bilangan bulat, digunakan operator MOD (MODulus). Penggunaannya dapat dilihat pada program berikut :

```
10 A = 5/2
15 B = 5
20 C = 5 MOD 2
25 D = 12.5/2.31
30 E = 12.5.31
40 F = 12.5 MOD 2.31
50 Print A, B, C
60 Print D, E, F
RUN
```

```
2.5          2          1
5.411256    6          1
OK
```

Nilai C yang besarnya sama dengan 1 diperoleh dari sisa pembagian 5 dan nilai F pada contoh di atas adalah sisa pembagian 12.5.31.

Dalam suatu operasi aritmatik digunakan perhitungan-perhitungan aritmatika yang pelaksanaannya disusun menurut urutan hirarki sbb :

(i) Pemangkatan

Seluruh operasi pemangkatan dilaksanakan lebih dahulu.

(ii) Perkalian dan pembagian

Operasi perkalian dan pemangkatan dilaksanakan setelah pelaksanaan operasi pemangkatan.

(iii) Penjumlahan dan pengurangan

Operasi ini dilaksanakan setelah pelaksanaan operasi perka-

lian dan pembagian.

Catatan :

- Dua buah operator yang letaknya berurutan, harus dipisahkan oleh tanda kurung.
- Operasi dalam tanda kurung, dilakukan terlebih dahulu.

Contoh :

Ekspresi : $A/B * C$

Ekspresi di atas sama dengan ekspresi matematik $(a/b)c$, operasi dilaksanakan dari kiri ke kanan.

Agar lebih jelas, urutan-urutan operasi aritmatik ini dapat dilihat pada tabel berikut.

Tabel 3 : Urutan Operasi Aritmatik

Operator	Urutan
()	Pertama
** atau ^	Kedua
* , /	Ketiga
+ , -	Terakhir

Contoh :

$$X = 12^2 + 5 * 4 - 12 / 3 - (75 + 25) / (50 - 25)$$

Persamaan di atas diselesaikan dengan urutan berikut :

- $(75 + 25) = 100$; berada dalam tanda kurung letaknya lebih di sebelah kiri
- $(50 - 25) = 25$; berada dalam tanda kurung
- $12^2 = 144$; bilangan berpangkat
- $5 * 4 = 20$; perkalian (letaknya di sebelah kiri pembagian)
- $12 / 3 = 4$; pembagian (letaknya lebih di sebelah kiri)

(vi) $100/25 = 4$; pembagian

(vii) $144 + 20 = 164$; penjumlahan (letaknya lebih di sebelah kiri)

(viii) $164 - 4 = 160$; pengurangan (letaknya lebih di sebelah kiri)

(ix) $160 - 4 = 154$; pengurangan.

Dari contoh di atas, dapat kita simpulkan, jika dalam suatu persamaan terdapat 2 buah operasi aritmatik dengan operator yang sama, maka operasi lebih dulu dilakukan pada persamaan yang letaknya di sebelah kiri.

(2) Ekspresi String

Ekspresi string dapat dibentuk dengan operasi penggabungan, misalnya penambahan dua string. Operator untuk operasi ini adalah tanda + (plus) atau gabungan (&). Contoh ekspresi string : C\$ = A\$ + B\$, Z\$ = "Selamat" + "Siang".

BASIC menggunakan kode ASCII untuk membandingkan karakter masing-masing string.

(3) Ekspresi Relasional

Ekspresi relasional dibentuk dengan menggabungkan variabel dan konstanta dengan menggunakan operator relasional. Jenis variabel string dapat digabungkan hanya dengan jenis variabel string. BASIC mempunyai 6 operator relasional yang dapat dilihat dalam Tabel 4

Operator relasional dapat juga digunakan untuk membandingkan dua buah nilai karakter alphanumerik. Hasil perbandingan ini adalah nilai 1 jika benar atau nilai 0 jika salah. Dalam hal ini operator relasional disebut operator string relasional dan ekspresinya disebut ekspresi string relasional.

Penggunaan operator relasional ini adalah untuk keperluan suatu kontrol yang akan kita pelajari lebih lanjut

Tabel 4.1 : Operator Relasional

Operator Relasional	Pemakaian	Persamaan dgn. pers. Aljabar	A r t i
=	$A = B$	$a = b$	A sama dengan B
<	$A < B$	$a < b$	A lebih kecil dari B
>	$A > B$	$a > b$	A lebih besar dari B
<= atau =<	$A <= B$	$a < b$	A lebih kecil atau sama dengan B
>= atau =>	$A >= B$	$a > b$	A lebih besar atau sama dengan B
<> atau ><	$A >< B$	$a = b$	A tidak sama dengan B

4 Fungsi

BASIC dilengkapi dengan fungsi-fungsi lain yang juga merupakan fungsi numerik, yaitu :

- INT(X) : - Merupakan singkatan dari INTeger
 - Digunakan untuk mengubah bilangan riil berpecahan (X) menjadi bilangan bulat (integer)
 - Fungsi INT selalu membulatkan nilai X ke bawah, sehingga $INT(X) \leq X$.
- FIX(X) : Fungsinya untuk menghilangkan bilangan desimal dari suatu bilangan riil.
- CINT(X) : - Adalah singkatan dari Convert to INTeger
 - Fungsinya untuk mengubah bilangan riil berpecahan (X) menjadi bilangan bulat, jika lebih besar atau sama dengan 0.5, dibulatkan ke atas

tetapi jika lebih kecil dari 0.5, dibulatkan ke bawah.

Contoh :

```
10 A = 1.34 : B = -1.34
20 PRINT A, INT(A), FIX(A) , CINT(A)
30 PRINT B, INT(B), FIX(B) , CINT(B)
40 END
RUN
```

```
1.34      1      1      1
-1.34     -2     -1     -1
OK
```

CSNG(X) : - Merupakan singkatan dari Convert to Single precision.

- Fungsinya untuk mengubah X menjadi bilangan riil berpresisi tunggal.

CDBL(X): - Singkatan dari Convert to Double precision

Contoh :

```
10 A# = 2.5436217 : B = 12.5 : C! = 5.34 : D = 32
20 PRINT A#, CSNG(A#)
30 PRINT B , CDBL(B)
40 PRINT C!, CSNG(C!), CDBL(C!)
50 PRINT D , CSNG(D) , CDBL(D)
RUN
```

```
2.5436217      2.543622
12.5           12.5
5.34          5.34          5.340000152587891
32            32           32
OK
```

SQR(X) : - Singkatan dari Square Root

- Fungsinya untuk menghasilkan nilai akar kwadrat X dan nilai X harus ≥ 0 .

Contoh :

```
10 A = 5
20 PRINT A, SQR(A)
30 END
```

```
RUN
5      2.236068
OK
```

ABS(X) : - Singkatan dari ABSolute
- Gunanya untuk memberikan harga mutlak dari X

Contoh :

```
10 X = -2
20 PRINT ABS(X)
30 END
```

```
RUN
4
OK
```

SIN(X) : Memberi nilai fungsi trigonometri SINUS sudut X,
X dalam satuan radian.

COS(X) : Memberikan nilai fungsi trigonometri COSINUS sudut
X, X dalam satuan radian.

TAN(X) : Memberikan nilai fungsi trigonometri TANGENT sudut
X, X dalam satuan radian.

ATN(X) : Memberikan nilai fungsi trigonometri ARC TANGENT
sudut X (dalam radian). nilai X berada dalam
interval $-\pi/2$ sampai $\pi/2$; $\pi = 3.141593$

Catatan :

Untuk mengubah derajat menjadi radian, kalikan dengan $180/\pi$

Contoh 1:

```
10 X = .8
20 PI = 3.141593
30 Y = X * 180/PI
40 PRINT Y, SIN(Y), COS(Y), TAN(Y), ATN(Y)
```

```
RUN
45.83662      .9600759      -.2797398      -3.432032      1.548983
OK
```

Contoh 2:

```
10 X = .8
20 PRINT SIN(X), COS(X), TAN(X), ATN(X)
```

```
RUN
.7173561      .6967067      1.029639      0
```

Keterangan :

Pada contoh 1, besaran sudut menggunakan derajat, sedangkan pada contoh 2, besaran sudutnya dalam radian.

LOG(X) : - Singkatan dari LOGaritma

- Untuk memberikan nilai logaritma dengan bilangan pokok e dari X ($e \log X = \ln X$), $e = 2.71828$.
- Untuk mendapatkan nilai logaritma dengan bilangan pokok 10 dari X, bagilah LOG(X) dengan LOG(10), X harus lebih besar dari 0.

Contoh 1:

```
10 X = 10
20 Y = LOG(X)
30 PRINT Y
RUN
```

```
2.302585
OK
```

Contoh 2:

```
10 X = 100
20 Y = LOG(X)/LOG(10)
30 PRINT Y
RUN
```

```
2
OK
```

Keterangan :

Contoh 1, adalah program untuk menghitung nilai logaritma dengan bilangan pokok e dari 10, sedangkan contoh 2 adalah program untuk menghitung nilai logaritma dengan bilangan pokok 10 dari 100.

EXP(X) : - Adalah singkatan dari EXPonent

- Fungsinya untuk memberikan nilai e dipangkatkan X.

Contoh :

```
10 X = 2
20 Y = EXP(X)
30 PRINT Y
```



```
RUN
7.389056
OK
```

SGN(X) : - Singkatan dari SiGN .
- Fungsinya untuk memberikan nilai: 1 jika $X > 0$
0 jika $X = 0$
-1 jika $X < 0$

Contoh :

```
10 A = 5 : B = -2 : C = -3 : D = 12 : E = 0
20 PRINT SGN(A), SGN(B), SGN(C), SGN(D), SGN(E)
30 END
RUN
1      -1      -1      1      0
OK
```

RND[(X)] : - Adalah singkatan dari RaNDom
- Untuk menghasilkan bilangan acak yang bernilai antara 0 dan 1. Setiap kali angka random dimunculkan, maka nilainya akan berbeda dengan angka random sebelumnya, tetapi jika program diulang dari awal, angka random yang dihasilkan akan sama
- X merupakan ekspresi numerik, tapi pada umumnya tidak digunakan.

RANDOHIZE : - Pernyataan yang digunakan bila kita menginginkan angka random yang selalu berbeda jika kita menjalankan program dari awal (mengulangi program dari awal).

Contoh 1:

```
10 A = 2 : B = 3 : C = 4
20 PRINT RND(A)
30 PRINT RND(B)
40 PRINT RND(C)
```

001.642
KUR
Li

```
RUN
.7151002
.683111
.4821425
OK
```

```
RUN
.7151002
.683111
.4821425
OK
```

Contoh 2 :

```
10 RANDOMIZE
20 A = 2 : B = 6
30 PRINT RND(A)
40 PRINT RND(B)
50 END
```

```
RUN
Random number seed (-32768 to 32767)? 345
.3359736
.5235548
OK
```

```
RUN
Random number seed (-32768 to 32767)?34
.1565455
3.840159E-02
OK
```

Dari kedua contoh di atas, dapat kita lihat penggunaan pernyataan RANDOMIZE, akan memberikan hasil keluaran yang berbeda dalam pengulangan pelaksanaan program, jika benih yang diberikan (seed) itu berbeda. Untuk memperoleh kumpulan bilangan acak yang berbeda, harus dimasukkan benih (seed) yang berbeda, sedangkan pada contoh 1, pelaksanaan program yang dilakukan berulang-ulang, tetap menghasilkan keluaran yang sama.

MILIK UPT PERPUSTAKAAN
IKIP. PADANG

DAFTAR PUSTAKA

Byron S. Gotfried, Theory and Problem of Programming with BASIC, Second Edition, Schaums Out Line Series MC Graw Hill, Singapore, 1982.

Yogiyanto, Teori dan Aplikasi Program Komputer Bahasa BASIC, Andi Offset, Yogyakarta, 1986.

MILIK UPT PERPUSTAKAAN
IKIP. PADANG