



**IKO31204**  
**Pemrograman Sistem**  
**Jilid 2: Scripting**

Fakultas Ilmu Komputer - Universitas Indonesia  
Sep 2011

# topik

standard stream (in, out, err)

pipe & redirection

operation

control

awk & sed

# standard stream

Text terminal

Keyboard

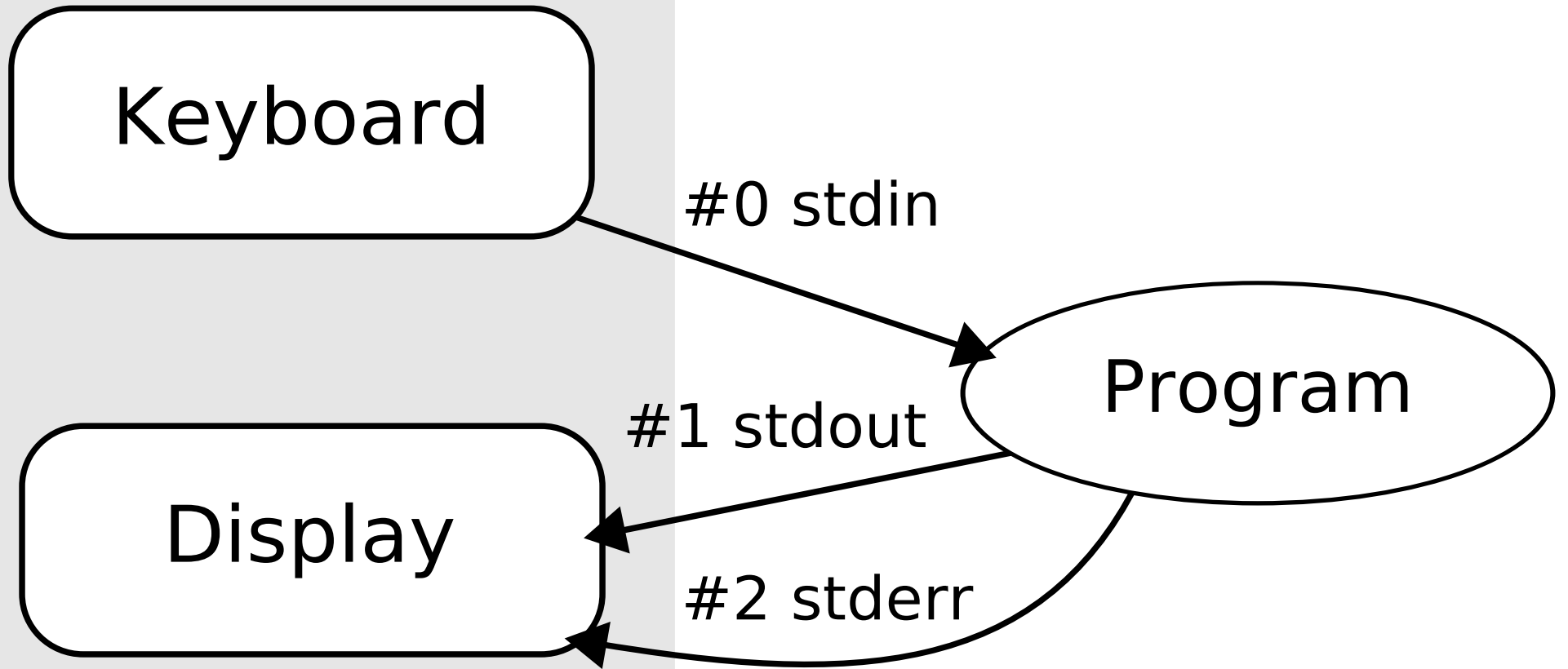
#0 stdin

Display

#1 stdout

#2 stderr

Program



# stdin

standard input is **data (often text)** going into a program. The program requests data transfers by use of the **read** operation.

# stdin

tidak semua program butuh input  
cth: ls, dir, mv, dll

dapat berasal dari keyboard ATAU  
output dari program lain

file descriptor 0 (nul)

# stdout

standard output is the **stream** where a program **writes its output data**. The program requests data transfer with the **write** operation.

# stdout

tidak semua program ada outputnya  
cth: mv, mkdir, mount, dll

menampilkan ke monitor ATAU  
menjadi input bagi program lain

file descriptor 1 (satu)

# stderr

standard error is **another output stream** typically used by programs to output **error messages** or **diagnostics**. It is a stream independent of standard output and **can be redirected separately**.



# stderr

the usual destination is the **text terminal which started the program** to provide the best chance of being seen even if standard output is redirected (so **not readily observed**).

# stderr

tidak semua program ada stderr nya  
cth yg ada: curl, grep, dll

menampilkan ke monitor ATAU  
menjadi input bagi program lain

file descriptor 2 (dua)

# redirection

new file

```
cth: ls > isi-dari-ls
```



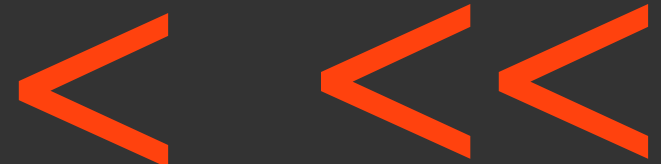
append file

```
cth: ls -al >> isi-dari-ls
```



stdin dari file

```
cth: sort < isi-dar-ls
```



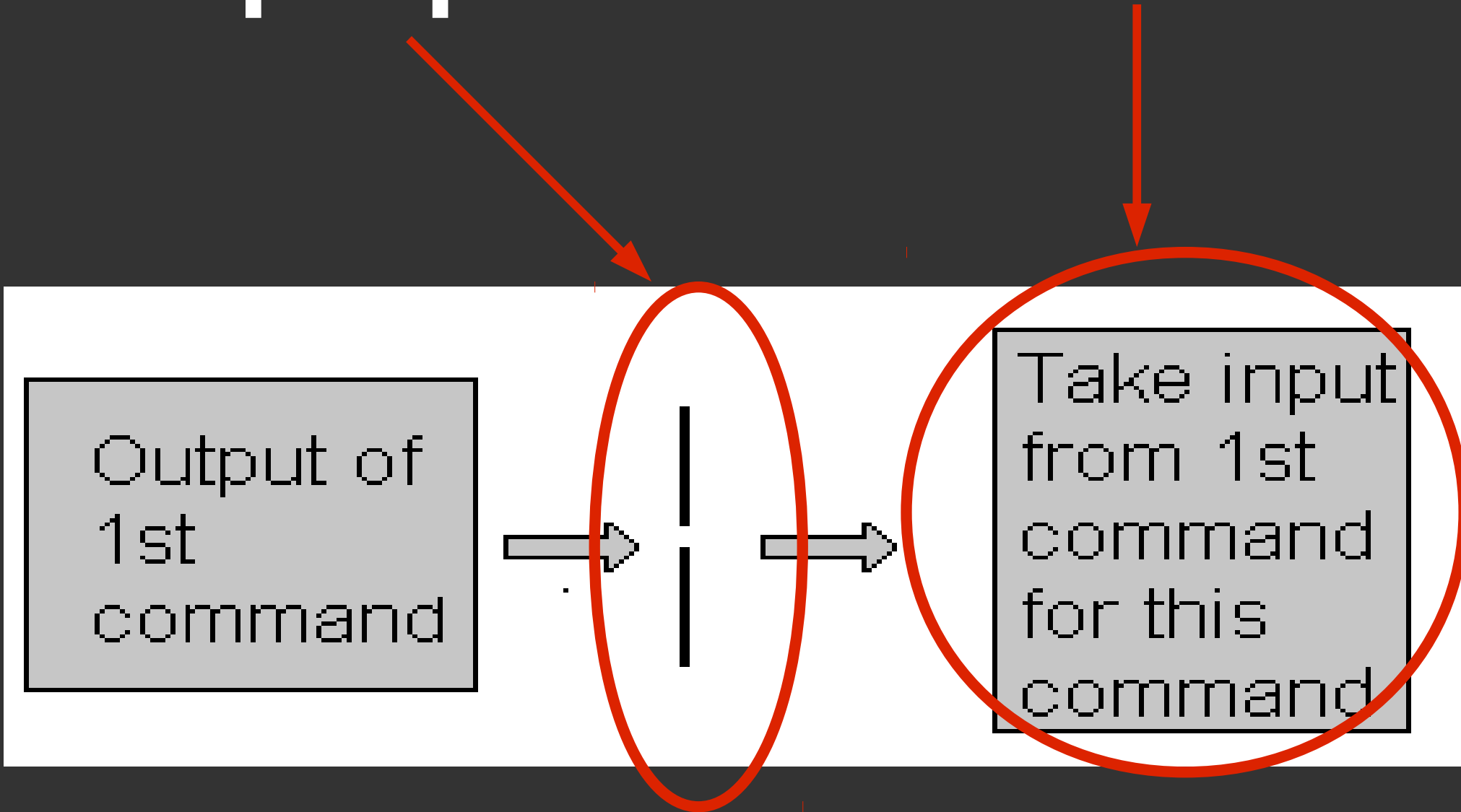
# pipe

a way to connect the **output** of one program to the **input** of another program **without any temporary file**

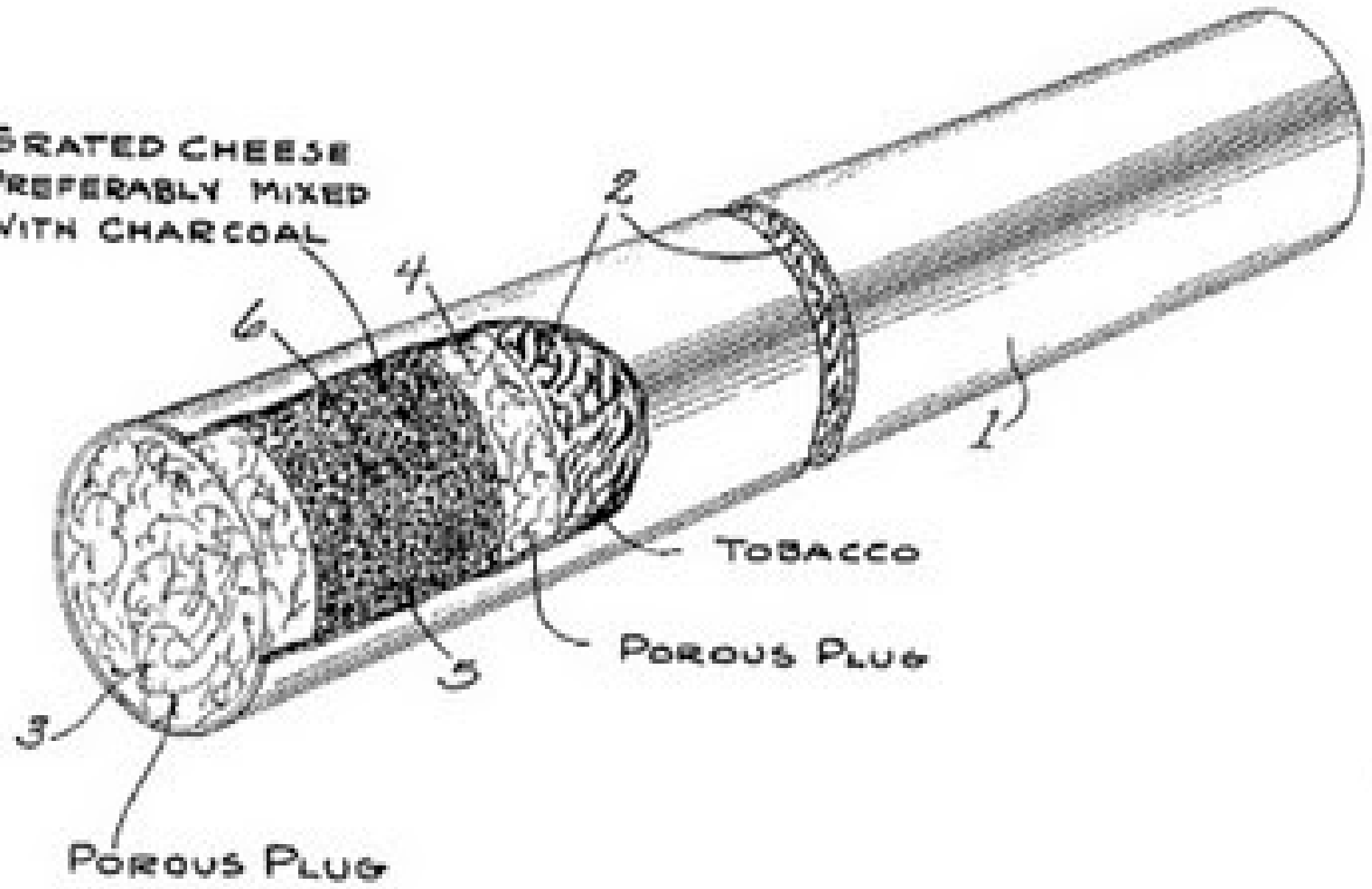
# filter

a filter performs some kind of process on the input and gives output

# pipe & filter



GRATED CHEESE  
PREFERABLY MIXED  
WITH CHARCOAL



TOBACCO

POROUS PLUG

POROUS PLUG

# simple\_read.c

```
#include <unistd.h>
#include <stdlib.h>

int main(){
    char buffer[128];
    int nread;

    nread = read(0, buffer, 128);
    if (nread == -1)
        write(2, "Read error\n", 11);
    if ((write(1,buffer,nread)) != nread)
        write(2, "Write error\n",12);
    exit(0);
}
```



# simple\_writeX.c

```
#include <unistd.h>
#include <stdlib.h>

int main()
{
    write(1, "This is Standard Output\n", 24);
    write(2, " This is Standard Error\n", 23);
    exit(0);
}
```

# pipe & filter

cth:

berkas logs.txt (10000 baris)

ambil baris ke 423 s/d 3221

```
# head -n 3221 < logs.txt | tail -n +423
```

# operation

```
for { variable name } in { list }  
do
```

```
    execute one for each item in  
    the list until the list is not  
    finished (and repeat all  
    statement between do and done)
```

```
done
```

sintaks #1 :: for

# operation

```
for ( ( expr1; expr2; expr3 ) )  
do  
    .....  
    ...  
    repeat all statements between do  
    and done until expr2 is TRUE  
done
```

sintaks #2 :: for

# operation

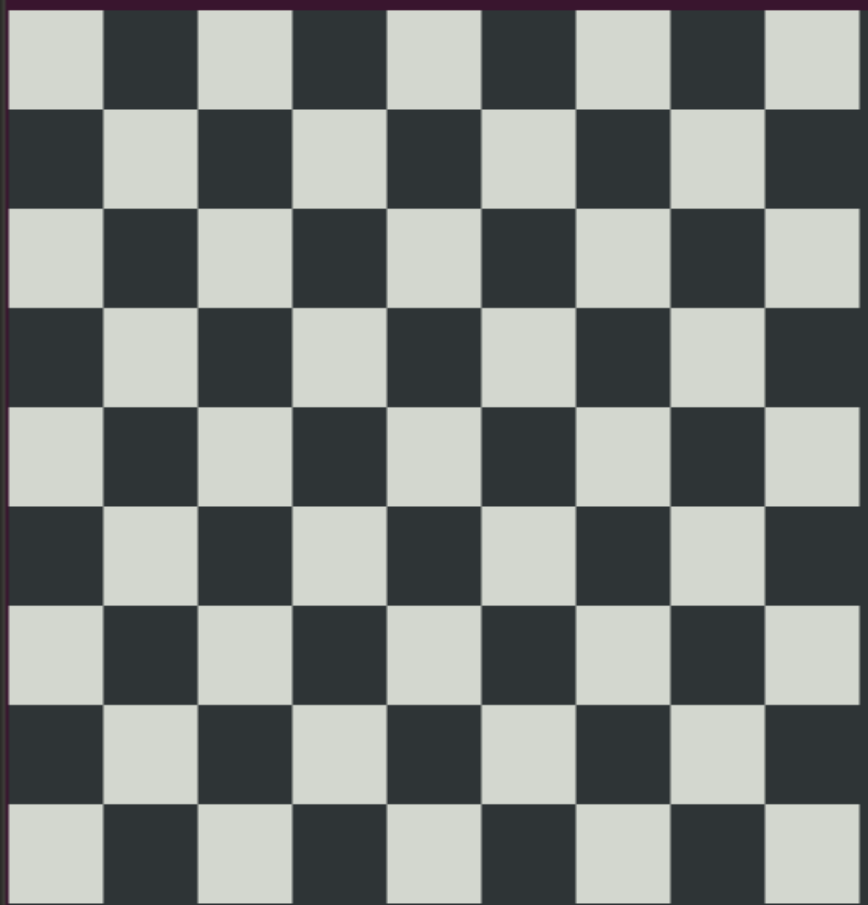
```
while [ condition ]  
do  
    command1  
    command2  
    command3  
    . . .  
done
```

sintaks :: while

# nested loop

```
for (( i = 1; i <= 9; i++ )) ### Outer for loop ###
do
    for (( j = 1 ; j <= 9; j++ )) ### Inner for loop ###
    do
        tot=`expr $i + $j`
        tmp=`expr $tot % 2`
        if [ $tmp -eq 0 ]; then
            echo -e -n "\033[47m "
        else
            echo -e -n "\033[40m "
        fi
    done
    echo -e -n "\033[40m" ##### set back background colour to black
    echo "" ##### print the new line ###
done
```

```
echo -e -n "\033[40m" ### set back background  
echo " done" the new line ###
```



adin@adin: ~\$

perkakas  
tambahan



# BASH SCRIPT

```
#!/bin/sh
for file in *.xxx; do
    # exit if there are no files
    if [ ! -f $file ]; then
        exit
    fi
    b=`basename $file .xxx`
    echo Converting $b.xxx to $b.c...
    mv $b.xxx $b.c
done
```

```
#!/bin/sh
```

```
CHALLENGE="challenge.txt"
```

```
RESPFILE="response.txt"
```

```
echo -n "Challenge Number? -- "
```

```
read CHALL
```

```
echo $CHALL > $CHALLENGE
```

```
hitung `cat $CHALLENGE` > $RESPFILE
```

```
echo -n "challenge.txt: "
```

```
cat $CHALLENGE
```

```
echo -n "response.txt: "
```

```
cat $RESPFILE
```

```
exit 0
```

```
#!/bin/bash
```

```
DEL="d"
```

```
for JUMLAH in {0..16} ; do
```

```
    DIR=`printf "M%2.2d" $JUMLAH`
```

```
    [ -d $DIR ] && [ "$1" = "$DEL" ] && rmdir $DIR
```

```
    [ -d $DIR ] || [ "$1" = "$DEL" ] || mkdir $DIR
```

```
done
```

```
for DIR in INFO TEST ; do
```

```
    [ -d $DIR ] && [ "$1" = "$DEL" ] && rmdir $DIR
```

```
    [ -d $DIR ] || [ "$1" = "$DEL" ] || mkdir $DIR
```

```
done
```

```
exit 0
```

# sed: stream editor

- editor NON interaktif

- `sed 's/funtion/function/g' < mainx.c > main.c`
- `sed -e '4d' -e '2d' hapus-b4-b3.txt`
- `sed -e '1d' -e '$d' -e '/^$/d' hapus-b1-kosong.txt`
- `sed 's/\([^:]*\).*\/\1/' /etc/passwd`
- `sed 's/\(^|[0-9.]\)\([0-9]\+\)\([0-9]\{3\}\)/\1\2,\3/g' numbers`

# diff dan patch

- `diff -Naur modul-b2 modul-b2-modified`  
    `- > modul-b2-patch.diff`
- `patch -p1 < modul-b2-patch.diff`

# Makefile

```
all: hello
```

```
hello: main.o hello.o
```

```
    gcc main.o hello.o -o hello
```

```
main.o: main.c
```

```
    gcc -c main.c
```

```
hello.o: hello.c
```

```
    gcc -c hello.c
```

```
clean:
```

```
    rm -rf *.o hello
```

# REGEX (REGular EXpression)

- Bagaimana caranya mencari kata “kambing” dalam sebuah berkas?
- Bagaimana caranya mencari sebuah HTML TAG dalam sebuah berkas?
- Bagaimana caranya mencari sebuah alamat email dalam sebuah berkas?
- Gunakan REGEX!
- Masihkah ingat TBA?

# PERMASALAHAN

- Si Didi makan Pisang
  - Bagaimana cara menemukan “i” pertama?
  - Bagaimana cara mencari “Pisang” namun bukan “pisang”?
- Dst...



# AWAS!

- Lain Padang, Lain Belawan, Lain Pula Lubuk Linggau! Terdapat beberapa dialek REGEX.
  - PERL, PHP, JDK, .NET
- Asumsi Pengolahan Per BARIS (historis)
- Silakan Google SINI dan SANA untuk mencari tutorial REGEX.
- Perbanyak JAM TERBANG. REGEX seharusnya membantu kita, jangan menghafal REGEX!
- Juga digunakan di GRE (?)

# BEBERAPA CONTOH

- **Sumber** <http://www.regular-expressions.info/tutorial.html>
- **Alamat Email**
  - `\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b`
- **<TAG HTML> ZCZC BLAH BLAH</TAG>**
  - `<TAG\b[^>]*>(.*?)</TAG>`
- **PASANGAN HTML**
  - `<([A-Z][A-Z0-9]*)\b[^>]*>(.*?)</\1>`
- **CASELESS**
  - `\1`

# Karakter Khusus 1

- `[ \ ^ $ . | ? * + ( )`
- Gunakan 'escape' “\”
  - contoh: “ 2 + 2 = 4 “
  - menjadi: “ 2 \+ 2 = 4”
- Digit tunggal: `\d`
  - 0 1 2 3 4 5 6 7 8 9
- `\w` = A-Za-z0-9\_
- `\s` = white karakter

# Karakter Khusus 2

- `\t` = tab
- `\r` = `<CR>`
- `\n` = Line Feed
- `\xFF` = HEX
- `\uFFFF` = Unicode
- `\b` = word boundary
- `\B` = does not match `\b`

# Karakter Umum

- Pal[au] = Pala atau Palu
- [0123456789] = 0 atau 1 atau 2 atau ...
- [0-9] = [0123456789]
- [A-Z] = A atau B atau ...
- [A-Za-z] = satu huruf besar atau kecil
- Ak[^a] = asal bukan Ak dan Aka.

# DOT

- “.” untuk segala karakter
- `\d\d.\d\d.\d\d\d\d`
  - 10/10/2010 namun juga 10a10a2010
- `\d\d[/- .]\d\d[/- .]\d\d\d\d`
  - 10/10/2010 namun juga 10/10.2010
  - juga 99/99/9999
- `[0-3]\d[/- .][0-1]\d[/- .]\d\d\d\d`
  - 31/12/2010 namun juga 33/13/9999

# STAR & PLUS

- “\*”

- $\langle [A-Za-z][A-Za-z0-9]^* \rangle$

- “+”

- $\langle [A-Za-z0-9]^+ \rangle$

-

?

- colou?r = color dan colour
- Nov(ember)? = Nov dan November
- Feb(ruary)? 23(rd)?
  - Feb 23
  - Feb 23rd
  - February 23
  - February 23rd



^ & \$

- ^a aku
- g\$ belakang
- \d+ sdsdjh345kjkjk
- ^\d+\$ 345

|

- `cat|dog|fish|cow`
- `\b(cat|dog)\b`
-

{ }

- **\b[1-9][0-9]{3}\b**
  - 1000 - 9999
- **\b[1-9][0-9]{2,4}\b**
  - 100 - 99999

# umask

- Digit 1: 4-2-1-0
  - user r/w/x/nouser
- Digit 2: 4-2-1-0
  - group r/w/x/nogroup
- Digit 3: 4-2-1-0
  - other r/w/x/no other
- Example:
  - `umask 077`
  - `umask 032`

# AWK

- alat bantu pembuatan laporan
- filter berkas
- struktur proses per BARIS:

```
pattern { action }
```

```
BEGIN { print "MULAI" }
```

```
{ print "PROSES" }
```

```
END { print "SELESAI" }
```

```
$ last
```

```
rms46      pts/0          jembatan.cs.ui.a Sun Sep 25 22:16   still logged in
julia.ed   pts/0          kawung.cs.ui.ac. Sun Sep 25 18:35 - 18:35   (00:00)
rizki.ma   pts/0          kawung.cs.ui.ac. Sun Sep 25 15:54 - 16:25   (00:30)
rizki.ma   pts/0          kawung.cs.ui.ac. Sun Sep 25 15:52 - 15:54   (00:01)
rizki.ma   pts/0          kawung.cs.ui.ac. Sun Sep 25 15:29 - 15:51   (00:22)
rizki.ma   pts/0          kawung.cs.ui.ac. Sun Sep 25 15:28 - 15:28   (00:00)
rizki.ma   pts/0          kawung.cs.ui.ac. Sun Sep 25 14:50 - 15:27   (00:37)
adrianto   pts/0          kawung.cs.ui.ac. Sat Sep 24 23:19 - 02:34   (03:15)
adrianto   pts/0          kawung.cs.ui.ac. Sat Sep 24 23:14 - 23:16   (00:01)
adrianto   pts/0          kawung.cs.ui.ac. Sat Sep 24 22:50 - 23:01   (00:11)
adrianto   pts/0          kawung.cs.ui.ac. Sat Sep 24 22:46 - 22:47   (00:00)
andrea.b   pts/1          kawung.cs.ui.ac. Sat Sep 24 20:12 - 22:25   (02:12)
```

```
$ last | awk '{print $1}'
```

```
rms46
julia.ed
rizki.ma
rizki.ma
rizki.ma
rizki.ma
rizki.ma
adrianto
adrianto
adrianto
adrianto
andrea.b
```

```
$ last | awk '{print $1}' | sort -u
```

```
abdel.ja  
ade.rahm  
adilla.w  
adrian.a  
adrianto  
aji.prad  
anandra.  
andika.w  
andrea.b  
ardhi.pu  
ardhiwib  
arif.fai
```

tanya jawab