

Referensi Cepat Bash

Perintah, scripting, pipe, redirection, job control

Dasar

echo & Navigasi

```
echo "Hello, World!" # print text
pwd                 # print working directory
cd /path/to/dir    # change directory
cd ..              # go up one level
cd ~               # go to home directory
cd -               # go to previous directory
```

Listing & Membuat

```
ls                 # list files
ls -la            # long format, show hidden
ls -lh           # human-readable sizes
mkdir mydir      # create directory
mkdir -p a/b/c   # create nested directories
```

Salin, Pindah & Hapus

```
cp file.txt copy.txt # copy file
cp -r dir/ backup/   # copy directory recursively
mv old.txt new.txt   # rename / move
rm file.txt          # delete file
rm -r dir/           # delete directory recursively
rm -rf dir/          # force delete (no prompt)
```

Variabel & Ekspansi

Variabel

```
name="Alice"       # assign (no spaces!)
echo "$name"       # variable expansion
echo "${name}_file" # braces for clarity
readonly PI=3.14   # constant
unset name         # delete variable
```

Variabel Khusus

```
$_              Nama script
$1 $2 ...      Argumen posisional
 $#             Jumlah argumen
 $@            Semua argumen (kata terpisah)
 $*           Semua argumen (satu string)
 $?           Status exit perintah terakhir
 $$          ID proses saat ini
 $!         PID proses background terakhir
```

Command Substitution & Aritmatika

```
files=$(ls)      # capture output
today=$(date +%Y-%m-%d) # command substitution
count=$((5 + 3)) # arithmetic: 8
echo $(10 / 3)   # integer division: 3
echo $(10 % 3)   # modulo: 1
```

Operasi String

```
${#str}         Panjang string
${str:0:5}      Substring (offset:panjang)
${str/old/new}  Ganti kemunculan pertama
${str//old/new} Ganti semua kemunculan
${str^^}       Huruf besar
${str,,}       Huruf kecil
```

Kondisional

if / elif / else

```
if [[ "$name" == "Alice" ]]; then
  echo "Hi Alice"
elif [[ "$name" == "Bob" ]]; then
  echo "Hi Bob"
else
  echo "Who are you?"
fi
```

Operator Test

```
-eq -ne  Integer sama / tidak sama
-lt -gt  Integer lebih kecil / lebih besar
-le -ge  Integer kurang/lebih atau sama
== !=   String sama / tidak sama
-z "$str" String kosong
-n "$str" String tidak kosong
-f file File ada dan merupakan file biasa
-d dir  Direktori ada
-e path Path ada (tipe apa pun)
-r -w -x Bisa dibaca / ditulis / dieksekusi
&& ||   AND / OR logika
```

Loop

for Loop

```
for fruit in apple banana cherry; do
  echo "$fruit"
done

for f in *.txt; do
  echo "File: $f"
done
```

for Loop Gaya C

```
for ((i=0; i<5; i++)); do
  echo "$i"
done
```

while Loop

```
count=0
while [[ $count -lt 5 ]]; do
  echo "$count"
  ((count++))
done
```

Kontrol Loop

```
break      Keluar dari loop
continue   Lanjut ke iterasi berikutnya
```

Fungsi

Mendefinisikan & Memanggil

```
greet() {
  echo "Hello, $1!" # $1 = first arg
  return 0          # exit status
}
greet "Alice"      # Hello, Alice!
```

Variabel Lokal & Return Value

```
add() {
  local sum=$(( $1 + $2 ))
  echo "$sum" # "return" via stdout
}
result=$(add 3 5) # capture: 8
```

Pipe & Redirection

Pipe

```
ls -l | grep ".txt" # pipe output
cat log | sort | uniq # chain commands
cmd1 | tee out.txt   # pipe + save to file
```

Redirection

```
cmd > file      Redirect stdout (timpa)
cmd >> file     Redirect stdout (tambah)
cmd < file      Redirect stdin dari file
cmd 2> file     Redirect stderr
cmd 2>&1        Redirect stderr ke stdout
cmd &> file     Redirect stdout + stderr
cmd << EOF     Here document (input inline)
/dev/null      Buang output: cmd > /dev/null
```

Operasi File

Melihat File

```
cat file.txt      # print entire file
head -n 10 file.txt # first 10 lines
tail -n 10 file.txt # last 10 lines
tail -f log.txt   # follow (live updates)
less file.txt     # paginated viewer
```

Menghitung & Mencari

```
wc -l file.txt    # count lines
wc -w file.txt    # count words
wc -c file.txt    # count bytes
find . -name "*.txt" # find by name
find . -type d    # find directories
find . -mtime -7  # modified in last 7 days
```

Perintah File Lainnya

```
touch file       Buat file / perbarui timestamp
stat file        Metadata file (ukuran, tanggal)
file img.png     Deteksi tipe file
diff a.txt b.txt Bandingkan dua file
sort file.txt    Urutkan baris
uniq             Hapus duplikat yang bersebelahan
cut -d: -f1      Ekstrak field berdasarkan delimiter
tr 'a-z' 'A-Z'  Terjemahkan / ganti karakter
```

Pemrosesan Teks

grep

```
grep "error" log.txt # search for pattern
grep -i "error" log.txt # case-insensitive
grep -r "TODO" src/ # recursive search
grep -n "func" file.go # show line numbers
grep -c "error" log.txt # count matches
grep -v "debug" log.txt # invert match
```

sed

```
sed 's/old/new/' file # replace first per line
sed 's/old/new/g' file # replace all
sed -i 's/old/new/g' file # edit in place
sed -n '5,10p' file # print lines 5-10
sed '/pattern/d' file # delete matching lines
```

awk

```
awk '{print $1}' file # print first field
awk -F: '{print $1}' file # custom delimiter
awk '$3 > 100' file # filter by field value
awk '{sum+=$1} END{print sum}' file # sum column
```

Referensi Cepat Bash

Izin

chmod

```
chmod 755 script.sh # rwxr-xr-x
chmod +x script.sh # add execute
chmod -w file.txt # remove write
chmod u+x,g-w file # user +exec, group -write
```

Referensi Izin

r (4)	Baca
w (2)	Tulis
x (1)	Eksekusi
u / g / o	User / Group / Others
755	Owner: rwx, Group/Other: r-x
644	Owner: rw-, Group/Other: r--

Kepemilikan

```
chown user file.txt # change owner
chown user:group file.txt # change owner + group
chown -R user:group dir/ # recursive
```