PEMBAHASAN UAS

1. Membuat script untuk melakukan pengecekan service ssh secara berkala dengan interval waktu tiap 10 detik dan memberikan notifikasi ke layar jika service ssh mati.
    - Membuat file untuk diisi perintah

      ```
      dellarfh@della:~$ nano 46_monssh.sh
      ```

    - Menambahkan isi seperti di bawah dan disimpan

      ```
        GNU nano 7.2
      #!/bin/bash

      while true; do
              STATUS=$(systemctl is-active ssh)

              if [ "$STATUS" != "active" ]; then
                   echo " [$(date)] SERVICE SSH MATI! "
              else
                   echo " [$(date)] SSH aktif. "
              fi

              sleep 10
      done_
      ```

    - Membuka akses agar bisa diedit atau dijalankan

      ```
      dellarfh@della:~$ chmod +x 46_monssh.sh
      ```

    - Output

      ```
      dellarfh@della:~$ ./46_monssh.sh
       [Wed May 28 12:19:20 PM UTC 2025] SSH aktif.
       [Wed May 28 12:19:30 PM UTC 2025] SSH aktif.
      ```

2. Membuat script backup direktori tertentu dan jalankan backup secara berkala setiap 15 detik dengan backup ke file 1 sd 10 dan kembali lagi menimpa file 1 jika sudah file 10 dijalankan selama 1 jam.
    - Membuat direktori data dan file contoh

      ```
      dellarfh@della:~$ mkdir -p ~/data
      dellarfh@della:~$ echo "Example File" > ~/data/
      ```

    - Membuat direktori backup

      ```
      dellarfh@della:~$ mkdir -p ~/backup
      ```

    - Membuat dan mengedit script backup

      ```
      dellarfh@della:~$ nano 46_backup.sh
      ```

      ```
        GNU nano 7.2                                                    46_
      #!/bin/bash

      src_dir = "$HOME/data"
      backup_dir = "$HOME/backup"
      max_files = 10
      interval = 15
      runtime = 3600 #1jam = 3600 detik

      start_time = $(date + %s)

      while [ $(($(date + %s) - start_time)) -lt $runtime ]; do
              index=$(( (($(date + %s) -start_time) / interval) % max_files + 1 ))
              filename="file${index}.tar.gz"
              tar -czf "$backup_dir/$filename" -C "$src_dir" .
              echo "$(date): Backup disimpan ke $backup_dir/$filename"
              sleep $interval
      done
      ```

- Memberikan izin eksekusi pada script

```
dellarfh@della:~$ chmod +x 46_backup.sh
```

- Menjalankan script

```
dellarfh@della:~$ ./46_backup.sh
./46_backup.sh: line 3: src_dir: command not found
./46_backup.sh: line 4: backup_dir: command not found
./46_backup.sh: line 5: max_files: command not found
./46_backup.sh: line 6: interval: command not found
./46_backup.sh: line 7: runtime: command not found
date: extra operand ‘%s’
Try 'date --help' for more information.
./46_backup.sh: line 9: start_time: command not found
./46_backup.sh: line 12: syntax error near unexpected token `)'
./46_backup.sh: line 12: `       index=$(( ($date + 5s) -start_time) / interval
dellarfh@della:~$
```

- Mengedit crontab untuk menjalankan script secara berkala (setelah script diperbaiki)

```
> crontab -e
```

```
  GNU nano 7.2                                                    /tmp/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 1 * * 0 /tmp/crontab.GMWauV/crontab/46_backup.sh
```

- Melihat entri crontab yang sudah diinstal

```
crontab: installing new crontab
> crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 1 * * 0 /tmp/crontab.GMWauV/crontab/46_backup.sh
>
```

3. Membuat script awk dari suatu file, lalu menghitung rata-rata dan menampilkan nilai maksimum dari file tersebut.
    - Menuliskan sebuah data yang nantinya akan diinputkan ke file data.txt

```
dellarfh@della:~$ cat > data.txt <<EOF
> timestamp suhu kelembapan cahaya
> 2025-05-27T08:00 24.5 60 800
> 2025-05-27T08:15 25.0 62 850
> 2025-05-27T08:30 26.2 65 900
> 2025-05-27T08:45 27.1 67 950
> EOF
dellarfh@della:~$
```

    - Membuat sebuah file untuk diisi perintah

```
dellarfh@della:~$ nano 46_suhukelembapan.sh
```

    - Menambahkan isi seperti di bawah dan disimpan

```
  GNU nano 7.2                                                    46_
#!/bin/bash

echo -n "Suhu maksimum: "
awk 'NR>1 { if ($2 > max) max=$2 } END { print max }' data.txt

echo -n "Rata-rata kelembapan: "
awk 'NR>1 { total += $3; count++ } END { print total/count }' data.txt
```

    - Output

```
ellarfh@della:~$ ./46_suhukelembapan.sh
uhu maksimum: 27.1
ata-rata kelembapan: 63.5
```

4. Membuat perintah di linux untuk melihat isi file /etc/passwd, lalu melakukan filter yang memiliki directory home dan mengambil nama user nya, dan melakukan filtering lagi.
    - Melihat isi file /etc/passwd

```
dellarfh@della:~$ cat /etc/passwd
```

    - Memfilter baris yang mengandung /home

```
dellarfh@della:~$ grep "/home" /etc/passwd
dellarfh@della:~$ grep "/home" /etc/passwd
dellarfh:x:1000:1000:dellaa,22,0878663509449,0343125678:/home/dellarfh:/bin/bash
paul:x:1001:1001::/home/paul:/bin/sh
jane:x:1002:1001::/home/jane:/bin/sh
alice:x:1003:1002::/home/alice:/bin/sh
derek:x:1004:1002::/home/derek:/bin/sh
```

    - Memfilter baris dan mengekstrak nama user dengan cut

```
dellarfh@della:~$ grep "/home" /etc/passwd | cut -d: -f1
dellarfh
paul
jane
alice
derek
dellarfh@della:~$
```

    - Membuat dan mengedit script shell, lalu disimpan

```
dellarfh@della:~$ nano 46_filter.sh
```

```
  GNU nano 7.2
#!/bin/bash

echo "Daftar user dengan direktori /home: "
awk -F: '/\/home/ { print $1 }' /etc/passwd
```

- Memberikan izin eksekusi pada script

```
dellarfh@della:~$ chmod +x 46_filter.sh
```

- Output

```
dellarfh@della:~$ ./46_filter.sh
Daftar user dengan direktori /home:
dellarfh
paul
jane
alice
derek
dellarfh@della:~$
```