Project Sesi 7 Bootcamp DevOps Engineer Digital Skola

Deskripsi:

Anda adalah seorang DevOps Engineer di sebuah perusahaan yang memiliki banyak file log di direktori /var/logs. Anda diminta untuk membuat bash script untuk:

1.Membuat direktori baru dengan nama backup_ddmmyyyy berdasarkan tanggal saat ini.

2.Memindahkan semua file .log dari /var/logs ke direktori backup.

3.Membuat arsip zip dari direktori backup.

4.Menampilkan pesan sukses setelah proses selesai.

Tugas Peserta:

Buat Bash Script berdasarkan langkah di atas. Pastikan script:

-Dapat dijalankan berulang kali tanpa menyebabkan konflik nama direktori.

-Menangani error, misalnya jika direktori atau file tidak ditemukan.

Srcipt lengkap untuk melakukan backup file log dengan nama file **backup_logs.sh** adalah sebagai berikut:

```
#!/bin/bash

# Direktori sumber file log
LOG_DIR="/var/log"
# Dapatkan tanggal saat ini dalam format DDMMYYYY
DATE=$(date +%d%m%Y)
# Direktori backup berdasarkan tanggal
BACKUP_DIR="/home/kaisenberg/tugas_sesi7/backup_${DATE}"
# File arsip zip
ZIP_FILE="/home/kaisenberg/tugas_sesi7/backup_${DATE}.zip"

# Buat direktori backup, jika belum ada
if [ ! -d "$BACKUP_DIR" ]; then
  mkdir -p "$BACKUP_DIR"
  if [ $? -ne 0 ]; then
    echo "Error: Gagal membuat direktori backup: $BACKUP_DIR" >&2
    exit 1
  fi
fi

# Periksa apakah direktori log ada
if [ ! -d "$LOG_DIR" ]; then
  echo "Error: Direktori log tidak ditemukan: $LOG_DIR" >&2
  exit 1
fi
```

```
# Cari dan pindahkan file .log ke direktori backup
LOG_FILES=$(find "$LOG_DIR" -type f -name "*.log")

if [ -z "$LOG_FILES" ]; then
  echo "Error: Tidak ada file .log yang ditemukan di $LOG_DIR untuk di backup." >&2
  exit 1
else
  find "$LOG_DIR" -type f -name "*.log" -exec mv {} "$BACKUP_DIR" \;
  if [ $? -ne 0 ]; then
    echo "Error: Gagal memindahkan file log dari $LOG_DIR ke $BACKUP_DIR" >&2
    exit 1
  fi
  echo "Sukses: File log dari $LOG_DIR telah dipindahkan ke $BACKUP_DIR"
fi


# Buat arsip zip dari direktori backup
zip -r "$ZIP_FILE" "$BACKUP_DIR"
if [ $? -ne 0 ]; then
  echo "Error: Gagal membuat arsip zip: $ZIP_FILE" >&2
  exit 1
fi

# Tampilkan pesan sukses
echo "========================================"
echo "Backup file log berhasil diselesaikan!"
echo "Direktori Backup: $BACKUP_DIR"
echo "File Arsip Zip: $ZIP_FILE"
echo "========================================"

exit 0
```

Penjelasan langkah-langkah Script adalah sebegai berikut:

1. Shebang dan Persiapan variabel:

```
#!/bin/bash

# Direktori sumber file log
LOG_DIR="/var/log"
# Dapatkan tanggal saat ini dalam format DDMMYYYY
DATE=$(date +%d%m%Y)
# Direktori backup berdasarkan tanggal
BACKUP_DIR="/home/kaisenberg/tugas_sesi7/backup_${DATE}"
# File arsip zip
ZIP_FILE="/home/kaisenberg/tugas_sesi7/backup_${DATE}.zip"
```

- **#!/bin/bash**: Baris ini adalah *shebang*. Ini memberitahu sistem operasi untuk menjalankan script ini menggunakan interpreter bash.
- Komentar-komentar (# ...): Baris-baris yang diawali dengan # adalah komentar. Ini digunakan untuk memberikan penjelasan tentang script dan variabel yang didefinisikan.
    - **LOG_DIR="/var/log":** Mendefinisikan variabel **LOG_DIR** yang menyimpan path ke direktori sumber file log. Dalam kasus ini, direktori sumber adalah **/var/log**.
    - **DATE=$(date +%d%m%Y):** Mendefinisikan variabel **DATE** yang menyimpan tanggal saat ini. Perintah **date +%d%m%Y** digunakan untuk mendapatkan tanggal dengan format DDMMYYYY (contoh: 26022025).
    - **BACKUP_DIR="/home/kaisenberg/tugas_sesi7/backup_${DATE}":** Mendefinisikan variabel **BACKUP_DIR** yang menentukan path lengkap direktori backup. Direktori backup akan dibuat di dalam **/home/kaisenberg/tugas_sesi7/** dengan nama backup_ diikuti tanggal saat ini (contoh: /home/kaisenberg/tugas_sesi7/backup_25082024).
    - **ZIP_FILE="/home/kaisenberg/tugas_sesi7/backup_${DATE}.zip":** Mendefinisikan variabel **ZIP_FIL**E yang menentukan path lengkap dan nama file arsip zip. File zip akan disimpan di **/home/kaisenberg/tugas_sesi7/** dengan nama backup_ diikuti tanggal saat ini dan ekstensi .zip (contoh: /home/kaisenberg/tugas_sesi7/backup_26022025.zip).

2. Pembuatan Direktori Backup:

```
# Buat direktori backup, jika belum ada
if [ ! -d "$BACKUP_DIR" ]; then
  mkdir -p "$BACKUP_DIR"
  if [ $? -ne 0 ]; then
    echo "Error: Gagal membuat direktori backup: $BACKUP_DIR" >&2
    exit 1
  fi
fi
```

- Bagian ini bertujuan untuk membuat direktori backup harian jika direktori tersebut belum ada.
- **if [! -d "$BACKUP_DIR"]; then**: Memeriksa apakah direktori yang path-nya disimpan dalam variabel $BACKUP_DIR **tidak** (!) ada (-d).
- **mkdir -p "$BACKUP_DIR"**: Jika direktori backup belum ada, perintah mkdir -p akan membuat direktori tersebut. Opsi -p memastikan direktori induk juga dibuat jika belum ada, dan tidak akan menghasilkan error jika direktori sudah ada.
- **if [$? -ne 0]; then ... fi**: Memeriksa nilai *exit status* dari perintah mkdir. $? adalah variabel khusus yang menyimpan *exit status* perintah terakhir yang dijalankan. Jika nilai $? **tidak sama dengan** (-ne) 0, berarti perintah mkdir gagal.

- **echo "Error: Gagal membuat direktori backup: $BACKUP_DIR" >&2**: Jika pembuatan direktori gagal, pesan error akan ditampilkan ke *standard error* (>&2).
- **exit 1:** Script akan keluar dengan *exit code* 1. *Exit code* bukan nol menandakan bahwa script selesai dengan error.

3. Pemeriksaan Direktori Log Sumber:

```
# Periksa apakah direktori log ada
if [ ! -d "$LOG_DIR" ]; then
  echo "Error: Direktori log tidak ditemukan: $LOG_DIR" >&2
  exit 1
fi
```

- Bagian ini memastikan bahwa direktori sumber log (/var/log) benar-benar ada sebelum melanjutkan proses backup.
- **if [! -d "$LOG_DIR"];** then: Memeriksa apakah direktori yang path-nya disimpan dalam variabel $LOG_DIR **tidak** (!) ada (-d).
- **echo "Error: Direktori log tidak ditemukan: $LOG_DIR" >&2:** Jika direktori log tidak ditemukan, pesan error akan ditampilkan ke *standard error*.
- **exit 1**: Script akan keluar dengan *exit code* 1.

4. Pencarian dan Penyalinan File Log:

```
# Cari dan pindahkan file .log ke direktori backup
LOG_FILES=$(find "$LOG_DIR" -type f -name "*.log")

if [ -z "$LOG_FILES" ]; then
  echo "Error: Tidak ada file .log yang ditemukan di $LOG_DIR untuk di backup." >&2
  exit 1
else
  find "$LOG_DIR" -type f -name "*.log" -exec mv {} "$BACKUP_DIR" \;
  if [ $? -ne 0 ]; then
    echo "Error: Gagal memindahkan file log dari $LOG_DIR ke $BACKUP_DIR" >&2
    exit 1
  fi
  echo "Sukses: File log dari $LOG_DIR telah dipindahkan ke $BACKUP_DIR"
fi
```

- Bagian ini bertugas mencari file dengan ekstensi .log di direktori sumber dan memindahkannya ke direktori backup.
- **LOG_FILES=$(find "$LOG_DIR" -type f -name "*.log"):** Perintah find digunakan untuk mencari file.
  - **"$LOG_DIR":** Direktori tempat pencarian dimulai (direktori sumber log).

- **-type f**: Hanya mencari file (bukan direktori atau jenis file lain).
- **-name "*.log":** Hanya mencari file yang namanya berakhiran .log.
- **LOG_FILES=$(...):** *Command substitution*. Output dari perintah find (yaitu daftar file .log yang ditemukan) disimpan ke dalam variabel LOG_FILES.

- **if [-z "$LOG_FILES"]; then ... else ... fi**: Memeriksa apakah variabel LOG_FILES kosong (-z). Jika kosong, berarti tidak ada file .log yang ditemukan.
  - **echo "Error: Tidak ada file .log yang ditemukan di $LOG_DIR untuk di backup." >&2**: Jika tidak ada file .log ditemukan, pesan error akan ditampilkan ke *standard error*.
  - **exit 1**: Script akan keluar dengan *exit code* 1.
  - **else**: Jika variabel LOG_FILES tidak kosong (ada file .log yang ditemukan), blok else akan dieksekusi.
  - **find "$LOG_DIR" -type f -name "*.log" -exec mv {} "$BACKUP_DIR" \;**: Perintah find digunakan lagi untuk mencari file .log dan kemudian mengeksekusi perintah cp untuk setiap file yang ditemukan.
    - **mv {} "$BACKUP_DIR":** Perintah mv (move) digunakan untuk menimdahkan setiap file yang ditemukan ({}) ke direktori backup ("$BACKUP_DIR").
    - **-exec ... \;**: Opsi -exec pada find digunakan untuk menjalankan perintah cp pada setiap file yang ditemukan. \; menandakan akhir dari perintah yang akan dieksekusi oleh -exec.
  - **if [$? -ne 0]; then ... fi**: Memeriksa *exit status* dari perintah find -exec cp .... Jika gagal, pesan error ditampilkan dan script keluar.
  - **echo "Sukses: File log dari $LOG_DIR telah dipindahkan ke $BACKUP_DIR"**: Jika penyalinan file berhasil, pesan sukses ditampilkan ke *standard output*. Perhatikan bahwa pada baris sukses ini, tidak dialihkan ke stderr karena ini adalah pesan informasi normal, bukan error.

5. Pembuatan Arsip Zip:

```
# Buat arsip zip dari direktori backup
zip -r "$ZIP_FILE" "$BACKUP_DIR"
if [ $? -ne 0 ]; then
  echo "Error: Gagal membuat arsip zip: $ZIP_FILE" >&2
  exit 1
fi
```

- Bagian ini membuat file arsip zip dari direktori backup yang telah berisi file log.
- **zip -r "$ZIP_FILE" "$BACKUP_DIR"**: Perintah zip digunakan untuk membuat arsip zip.

- **-r**: Opsi rekursif, yang berarti akan mengarsipkan direktori dan semua isinya secara rekursif.
- **"$ZIP_FILE"**: Nama file arsip zip yang akan dibuat (path dan nama file telah didefinisikan dalam variabel ZIP_FILE).
- **"$BACKUP_DIR"**: Direktori yang akan diarsipkan (direktori backup).

- **if [$? -ne 0]; then ... fi**: Memeriksa *exit status* dari perintah zip. Jika gagal, pesan error ditampilkan dan script keluar.

6. Menampilkan Pesan Sukses Akhir:

```
# Tampilkan pesan sukses
echo "========================================"
echo "Backup file log berhasil diselesaikan!"
echo "Direktori Backup: $BACKUP_DIR"
echo "File Arsip Zip: $ZIP_FILE"
echo "========================================"

exit 0
```

- Bagian ini menampilkan pesan sukses ke *standard output* setelah semua langkah backup berhasil diselesaikan.
- **echo "=="**: Menampilkan baris pemisah untuk memperjelas output.
- **echo "Backup file log berhasil diselesaikan!"**: Menampilkan pesan utama bahwa backup berhasil.
- **echo "Direktori Backup: $BACKUP_DIR"**: Menampilkan path direktori backup tempat file log disalin.
- **echo "File Arsip Zip: $ZIP_FILE"**: Menampilkan path file arsip zip yang telah dibuat.
- **echo "=="**: Menampilkan baris pemisah penutup.
- **exit 0**: Script keluar dengan *exit code* 0. *Exit code* 0 menandakan bahwa script berhasil dijalankan tanpa error.

Penanganan Error:

Script ini memiliki beberapa mekanisme penanganan error:

- Pemeriksaan keberadaan direktori backup sebelum pembuatan.
- Pemeriksaan keberadaan direktori log sumber.
- Pemeriksaan apakah ada file .log yang ditemukan sebelum proses penyalinan.
- Pemeriksaan keberhasilan setiap perintah penting (mkdir, find -exec cp, zip) melalui *exit status*.

- Jika terjadi error pada langkah-langkah kritikal (pembuatan direktori, penyalinan file, pembuatan zip, tidak ditemukannya log), script akan menampilkan pesan error ke *standard error* (>&2) dan keluar dengan *exit code* bukan nol (exit 1).

Sebelum menggunakan cronjob script tersebut dicoba dijalankan manual dengan cara berikut:

1. Jadikan script dapat dieksekusi dengan perintah chmod +x backup_log.sh.

```
root@MENOMEN:/home/kaisenberg/tugas_sesi7# chmod +x backup_logs.sh
```

2. Jalankan script dengan perintah ./backup_log.sh.

```
root@MENOMEN:/home/kaisenberg/tugas_sesi7# ./backup_logs.sh > ./backup_$(date +%d%m%Y).log 2>&1
```

3. Validasi dengan command **ls -l** untuk melihat apakah direkori backup berhasil dibuat

```
root@MENOMEN:/home/kaisenberg/tugas_sesi7# ls -l
total 16
drwxr-xr-x 2 root root 4096 Feb 26 23:57 backup_26022025
-rw-r--r-- 1 root root  691 Feb 26 23:57 backup_26022025.log
-rw-r--r-- 1 root root 2405 Feb 26 23:57 backup_26022025.zip
-rwxr-xr-x 1 root root 1570 Feb 26 23:37 backup_logs.sh
```

Cronjob untuk otomatisasi backup log:

Buka file cronjob dengan perintah **crontab -e** lalu edit file tersebut dengan command cronjob berikut: **57 12 * * * /home/kaisenberg/tugas_sesi7/backup_logs.sh >> /home/kaisenberg/tugas_sesi7/backup.log 2>&1.** Perintah cronjob tersebut berarti: "Setiap hari, pada pukul 12:57 siang, jalankan script bash **/home/kaisenberg/tugas_sesi7/backup_logs.sh** dan simpan semua output (baik keluaran normal maupun pesan kesalahan) dari script tersebut ke dalam file log **/home/kaisenberg/tugas_sesi7/backup.log** (dengan cara menambahkan output baru ke akhir file log yang sudah ada)."

```
57 12 * * * /home/kaisenberg/tugas_sesi7/backup_logs.sh >> /home/kaisenberg/tugas_sesi7/backup.log 2>&1
```

Dengan command **ls -l** dapat diverifikasi bahwa direktori **backup_ddmmyyy**, arsip **backup_ddmmyyy.zip**, dan **backup.log** log dari eksekusi script telah berhasil dibuat.

```
root@MENOMEN:/home/kaisenberg/tugas_sesi7# ls -l
total 40
-rw-r--r-- 1 root root  1028 Feb 27 12:57 backup.log
drwxr-xr-x 2 root root  4096 Feb 26 23:57 backup_26022025
-rw-r--r-- 1 root root   691 Feb 26 23:57 backup_26022025.log
-rw-r--r-- 1 root root  2405 Feb 26 23:57 backup_26022025.zip
drwxr-xr-x 2 root root  4096 Feb 27 12:57 backup_27022025
-rw-r--r-- 1 root root 15838 Feb 27 12:57 backup_27022025.zip
-rwxr-xr-x 1 root root  1570 Feb 26 23:37 backup_logs.sh
```

Isi dari **backup.log** dapat dilihat dengan perintah **cat backup.log**. Telah terverifikasi bahwa proses backup dengan script secara otomatis sukses dilakukan.

```
root@MENOMEN:/home/kaisenberg/tugas_sesi7# cat backup.log
Sukses: File log dari /var/log telah dipindahkan ke /home/kaisenberg/tugas_sesi7/backup_27022025
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/ (stored 0%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/unattended-upgrades-dpkg.log (stored 0%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/discover.log (deflated 80%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/ubuntu-advantage.log (deflated 91%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/auth.log (deflated 91%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/unattended-upgrades-shutdown.log (stored 0%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/kern.log (deflated 77%)
  adding: home/kaisenberg/tugas_sesi7/backup_27022025/unattended-upgrades.log (deflated 47%)
=======================================
Backup file log berhasil diselesaikan!
Direktori Backup: /home/kaisenberg/tugas_sesi7/backup_27022025
File Arsip Zip: /home/kaisenberg/tugas_sesi7/backup_27022025.zip
=======================================
```