

MATA KULIAH
PRAKTIKUM MANAJEMEN DATA

Writing Interactive Scripts & Funtion
On Bash Beginner Guide Modul

Dosen Pengampu :

Isbat Uzzin Nadhori, S.Kom, M.T., & Prasetyo Wibowo, S.ST., M.Kom



Disusun Oleh:

Intan Dwi Anggreini (3324600006)

1 D4 SAINS DATA TERAPAN

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

Jl. Raya ITS, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111

- 1) Write a script that asks for the user's age. If it is equal to or higher than 16, print a message saying that this user is allowed to drink alcohol. If the user's age is below 16, print a message telling the user how many years he or she has to wait before legally being allowed to drink.

```
GNU nano 7.2 exercisel.sh
#!/bin/bash

# Meminta inputan usia pengguna
echo -n "Masukkan usia Anda: "
read usia

# Validasi input: harus angka bulat positif
if ! [[ "$usia" =~ ^[0-9]+$ ]]; then
    echo "Input tidak valid. Harap masukkan angka bulat positif."
    exit 1
fi

# Logika utama
if [ "$usia" -ge 16 ]; then
    echo "Anda diperbolehkan untuk minum alkohol."

    # Hitung konsumsi bir statistik
    tahun_minum=$((usia - 16))
    konsumsi=$((tahun_minum * 100))
    echo "Anda telah mengonsumsi sekitar $konsumsi liter alkohol sejak usia $usia tahun dalam kurun waktu konsumsi $tahun_minum tahun."
else
    sisa=$((16 - usia))
    echo "Anda harus menunggu $sisa tahun lagi sebelum boleh minum alkohol secara legal."
fi

Help Write Out Where Is Cut Execute Location Undo Set Mark
Exit Read File Replace Paste Justify Go To Line Redo Redo Copy
```

Membuat program bash script dalam file exercisel.sh dengan analisa program dibawah ini:

```
# Meminta inputan usia pengguna
echo -n "Masukkan usia Anda: "
read usia
```

- Dengan command *echo* akan menampilkan prompt input “Masukkan usia Anda: “ yang nantinya nilai (usia) yang diinputkan akan disimpan ke variabel *usia*.
- Option -n pada command *echo* mencegah echo pindah baris agar input pengguna berada di baris yang sama.

```
GNU nano 7.2 exercisel.sh
# Validasi input: harus angka bulat positif
if ! [[ "$usia" =~ ^[0-9]+$ ]]; then
    echo "Input tidak valid. Harap masukkan angka bulat positif."
    exit 1
fi
```

- Mengecek bahwa input hanya berisi digit angka ($^[0-9]+$$) dengan conditional statement IF. Jika tidak valid, program akan menampilkan pesan “Input tidak valid. Harap masukkan angka bulat positif” dan kemudian keluar (*exit 1*) dan tidak melanjutkan proses.
- Membuat logika dengan kondisional statement IF yang akan menentukan diperbolehkan atau tidak diperbolehkannya konsumsi alkohol berdasarkan usia pengguna.

```
if [ "$usia" -ge 16 ]; then
    echo "Anda diperbolehkan untuk minum alkohol."

    # Hitung konsumsi bir statistik
    tahun_minum=$((usia - 16))
    konsumsi=$((tahun_minum * 100))
    echo "Anda telah mengonsumsi sekitar $konsumsi liter alkohol sejak usia $usia tahun dalam kurun waktu konsumsi $tahun_minum tahun."
```

Chapter 8. Writing Interactive Scripts

- 1) Jika nilai dalam variabel usia (nilai yang diinputkan user) lebih dari sama dengan (-ge : greater than or equal to) 16 maka akan menampilkan pesan “Anda diperbolehkan untuk minum alkohol”. Lalu juga akan menampilkan pesan yang berisi banyaknya alkohol dalam satuan liter yang telah dikonsumsi dalam rentang waktu satu (1) tahun.

```
else
    sisa=$((16 - usia))
    echo "Anda harus menunggu $sisa tahun lagi sebelum boleh minum alkohol secara legal."
fi
```

Namun jika nilai dalam variabel usia (nilai yang diinputkan user) tidak memenuhi kondisional statement pertama maka akan menjalankan program dalam scope else, yang menampilkan pesan jarak berapa tahun lagi yang dibutuhkan untuk diperbolehkan mengonsumsi alkohol.

```
intan@ds2024:~/BashGuide/chapter8$ ls -l
total 4
-rw-r--r-- 1 root root 756 May  2 04:06 exercisel.sh
intan@ds2024:~/BashGuide/chapter8$ chmod +x exercisel.sh
chmod: changing permissions of 'exercisel.sh': Operation not permitted
intan@ds2024:~/BashGuide/chapter8$ sudo chmod +x exercisel.sh
intan@ds2024:~/BashGuide/chapter8$ ls -l
total 4
-rwxr-xr-x 1 root root 756 May  2 04:06 exercisel.sh
```

- Merubah permission dari file exercisel.sh agar program bash script dalam file dapat dijalankan.

As an extra, calculate how much beer an 18+ user has drunk statistically (100 liters/year) and print this information for the user.

```
intan@ds2024:~/BashGuide/chapter8$ ./exercisel.sh
Masukkan usia Anda: 15
Anda harus menunggu 1 tahun lagi sebelum boleh minum alkohol secara legal.
intan@ds2024:~/BashGuide/chapter8$ ./exercisel.sh
Masukkan usia Anda: 19
Anda diperbolehkan untuk minum alkohol.
Anda telah mengonsumsi sekitar 300 liter alkohol sejak usia 19 tahun dalam kurun waktu konsumsi 3 tahun.
intan@ds2024:~/BashGuide/chapter8$
```

- 2) Write a script that takes one file as an argument. Use a *her* document that presents the user with a couple of choices for compressing the file. Possible choices could be **gzip**, **bzip2**, **compress** and **zip**.

```

GNU nano 7.2      exercise2.sh
#!/bin/bash

# Mengecek apakah argumen file diberikan
if [ $# -ne 1 ]; then
    echo "Program Bash Script $0 hanya memproses 1 file"
    echo "Program menolak memproses file lebih dari 1: $1, $2"
    exit 1
fi

FILE="$1"

# Mengecek apakah file benar-benar ada
if [ ! -f "$FILE" ]; then
    echo "File '$FILE' tidak ditemukan."
    exit 1
fi

# Menampilkan menu dengan here document
cat << EOF
Pilih metode kompresi:
1) gzip
2) bzip2
3) compress
4) zip
EOF

```

```

# Baca pilihan dari pengguna
read -p "Masukkan pilihan [1-4]: " pilihan

case "$pilihan" in
    1)
        gzip "$FILE"
        echo "File telah dikompres dengan gzip."
        ;;
    2)
        bzip2 "$FILE"
        echo "File telah dikompres dengan bzip2."
        ;;
    3)
        compress "$FILE"
        echo "File telah dikompres dengan compress."
        ;;
    4)
        zip "${FILE}.zip" "$FILE"
        echo "File telah dikompres dengan zip."
        ;;
    *)
        echo "Pilihan tidak valid."
        exit 1
        ;;
esac

```

Berikut analisa dari program bash script diatas:

```

if [ $# -ne 1 ]; then
    echo "Program Bash Script $0 hanya memproses 1 file"
    echo "Program menolak memproses file lebih dari 1: $1, $2"
    exit 1
fi

```

```

intan@ds2024:~/BashGuide/chapter8$ ./exercise2.sh yes1.txt yes2.txt
Program Bash Script ./exercise2.sh hanya memproses 1 file
Program menolak memproses file lebih dari 1: yes1.txt, yes2.txt

```

Mengecek apakah argumen yang dimasukkan hanya 1 argumen (file), karena program bash script hanya menerima satu file sebagai argumen menggunakan kondisional statement IF dan logika tidak sama dengan (!=) atau -ne. Jika argumen yang diberikan memenuhi kondisional statement IF (file yang akan diproses lebih dari 1) maka akan menampilkan 2 pesan yang terletak setelah command *echo*.

```

FILE="$1"

# Mengecek apakah file benar-benar ada
if [ ! -f "$FILE" ]; then
    echo "File '$FILE' tidak ditemukan."
    exit 1
fi

```

```

intan@ds2024:~/BashGuide/chapter8$ ./exercise2.sh yes3.txt
File 'yes3.txt' tidak ditemukan.

```

Variabel \$FILE berisi nama file dari argumen yang dimasukkan user yang sebelumnya tertampung di variabel \$1. Untuk mengecek apakah file yang akan diproses oleh bash script exercise2.sh tersedia dengan conditional statement IF > *if [! -f \$FILE]*, jika memenuhi kondisional statement ini maka menampilkan pesan bahwa file yang akan diproses tidak ditemukan. Option -f ini untuk memeriksa apakah file yang diberikan ada dan merupakan file reguler.

```
cat << EOF
Pilih metode kompresi:
1) gzip
2) bzip2
3) compress
4) zip
EOF
```

Kemudian karena untuk menu kompresi antara gzip, bzip2, compress, dan zip menggunakan here document sehingga menggunakan command “*cat << EOF* *EOF*” pada program bash script.

```
read -p "Masukkan pilihan [1-4]: " pilihan
```

Untuk meminta inputan dari user, dimana user akan memilih file akan dikompresi antara gzip, bzip2, compress, atau zip maka kita menggunakan command *-p* “*Masukkan pilihan [1-4]:*”. Kemudian untuk menyimpan inputan dari user tersebut ke dalam variabel pilihan kita menggunakan command *read pilihan*.

```
case "$pilihan" in
  1)
    gzip "$FILE"
    echo "File telah dikompres dengan gzip."
    ;;
  2)
    bzip2 "$FILE"
    echo "File telah dikompres dengan bzip2."
    ;;
  3)
    compress "$FILE"
    echo "File telah dikompres dengan compress."
    ;;
  4)
    zip "${FILE}.zip" "$FILE"
    echo "File telah dikompres dengan zip."
    ;;
  *)
    echo "Pilihan tidak valid."
    exit 1
    ;;
esac
```

Dengan command *case* akan mencocokkan nilai dalam variabel pilihan (pilihan kompresi yang diinginkan user) dengan pola yang sesuai dan menjalankan blok kode yang sesuai dengan pilihan tersebut. Misalnya, jika variabel pilihan menampung nilai 1 maka akan menjalankan blok kode 1 yang akan mengompresi file menjadi gzip dan kemudian menampilkan pesan “File telah dikompresi dengan gzip.”.

Berikut ini percobaan saya menjalankan file bash script exercise2.sh dengan argument file cobaExercise2.txt:

```

GNU nano 7.2      cobaExercise2.txt *
Assalamu'alaikum Wr. Wb.

Saya Intan Dwi Anggreini dari SDT A 2024.
Disini saya membuat file TXT dalam rangka praktikum mata kuliah manajemen data.

اَرْسَىٰ رَسْوَ لَا عَمَّ نَبَا , اَرْسَىٰ رَسْوَ لَا عَمَّ نَبَا

"Maka sesungguhnya bersama kesulitan ada kemudahan.
Sesungguhnya bersama kesulitan ada kemudahan"
(QS. Al-Insyirah: 5-6).|

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

```

intan@ds2024:~/BashGuide/chapter8$ ./exercise2.sh cobaExercise2.txt
Pilih metode kompresi:
1) gzip
2) bzip2
3) compress
4) zip
Masukkan pilihan [1-4]: 1
File telah dikompres dengan gzip.
intan@ds2024:~/BashGuide/chapter8$ ls -l
total 12
-rw-r--r-- 1 intan intan 289 May  2 06:38 cobaExercise2.txt.gz
-rwxr-xr-x 1 root  root  706 May  2 06:26 exercisel.sh
-rwxrwxr-x 1 intan intan 914 May  2 06:27 exercise2.sh

```

- 3) Write a script called homebackup that automates **tar** so the person executing the script always uses the desired options (cvp) and backup destination directory (/var/backups) to make a backup of his or her home directory. Implement the following features:

Mendefinisikan beberapa variabel konstanta :

```

BACKUP_DIR="/var/backups"
HOME_DIR="$HOME"
FULL_BACKUP="${BACKUP_DIR}/full_backup.tar"
INCR_BACKUP="${BACKUP_DIR}/incremental backup.tar"

```

- Test for the number of arguments. The script should run without arguments. If any arguments are present, exit after printing a usage message.

```

if [ $# -ne 0 ]; then
    echo "Program bash script homebackup.sh hanya bisa dijalankan tanpa argumen"
    exit 1
fi

```

Mengecek agar skrip dijalankan tanpa argumen dengan kondisional statemen IF. Jika jika bash script dijalankan dengan argumen maka kondisional statement IF terpenuhi, maka akan menampilkan pesan “Program bash scripttanpa argumen” dan keluar dari program bash script.

- Determine whether the backups directory has enough free space to hold the backup.

```

if [ ! -d "$BACKUP_DIR" ]; then
    echo "Membuat direktori backup di $BACKUP_DIR"
    sudo mkdir -p "$BACKUP_DIR"
    sudo chown "$USER" "$BACKUP_DIR"
fi

```

Mengecek apakah directory tujuan (/var/backups) telah tersedia atau belum. Jika belum maka kondisional statement `if [! -d "$BACKUP_DIR"]` terpenuhi dan program akan membuat /var/backups jika belum serta memberi akses write (*chown*) user pada direktori tersebut.

```
FREE_SPACE_KB=$(df "$BACKUP_DIR" | awk 'NR==2 {print $4}')
intan@ds2024:~$ df /var/backups
Filesystem              1K-blocks    Used Available Use% Mounted on
/dev/mapper/ubuntu--vg-ubuntu--lv 11758760 9550988   1588664   86% /
intan@ds2024:~$ df /var/backups | awk 'NR==2 {print $4}'
1588664
```

Mengecek size ruang kosong di /var/backups dengan command `BACKUP_DIR | awk 'NR == 2 {print $4}'` lalu menyimpan output command tersebut ke variabel `FREE_SPACE_KB` (variabel menyimpan size sisa ruang kosong dari /var/backups).

```
REQUIRED_SPACE_KB=500000

if [ "$FREE_SPACE_KB" -lt "$REQUIRED_SPACE_KB" ]; then
    echo "Tidak cukup ruang di $BACKUP_DIR. Dibutuhkan minimal 5000 MB."
    exit 1
fi
```

Membuat variabel `REQUIRED_SPACE_KB` yang menentukan size minimum ruang kosong yang diperlukan yakni sebesar 500000 KB (500 MB). Kemudian membandingkannya nilai antara dengan `FREE_SPACE_KB` dan `REQUIRED_SPACE_KB` dengan logika `-lt` atau less than (<), jika memenuhi kondisi menandakan ruang kosong pada direktori /var/backups kurang dari 500 MB sehingga backup tidak dapat dijalankan.

- Ask the user whether a full or an incremental backup is wanted. If the user does not have a full backup file yet, print a message that a full backup will be taken. In case of an incremental backup, only do this if the full backup is not older than a week.

```
echo "Backup penuh atau incremental? (ketik: full / incr)"
read -r BACKUP_TYPE
```

Menawarkan jenis backup kepada user antara full dan incremental. Inputan user terkait tawaran ini akan disimpan dalam variabel `BACKUP_TYPE`.

```
GNU nano 7.2                               homebackup.sh *
if [ "$BACKUP_TYPE" = "full" ]; then
    echo "Melakukan backup secara keseluruhan"
    tar --create --verbose --preserve-permissions \
        --file="$FULL_BACKUP" \
        --listed-incremental="$SSNAPSHOT_FILE" "$HOME_DIR"

    BACKUP_FILE="$FULL_BACKUP"
```

Jika user memilih opsi “full”, maka akan menjalankan backup data directory home secara keseluruhan yang kemudian akan ditampung dalam variabel `FULL_BACKUP`. File dalam variabel `FULL_BACKUP` akan disimpan berdasarkan path dan nama file yang telah ditetapkan dalam variabel `BACKUP_FILE`.

```

GNU nano 7.2                                     homebackup.sh
if [ "$BACKUP_TYPE" = "full" ]; then
    echo "Melakukan backup full atau backup secara keseluruhan"
    tar --create --verbose --preserve-permissions \
        --file="$FULL_BACKUP" \
        --listed-incremental="$SNAPSHOT_FILE" "$HOME_DIR"

    BACKUP_FILE="$FULL_BACKUP"

elif [ "$BACKUP_TYPE" = "incr" ]; then
    # Cek apakah full backup ada
    if [ ! -f "$FULL_BACKUP" ]; then
        echo "Belum ada backup full. Backup full akan dilakukan."
        tar --create --verbose --preserve-permissions \
            --file="$FULL_BACKUP" \
            --listed-incremental="$SNAPSHOT_FILE" "$HOME_DIR"
        BACKUP_FILE="$FULL_BACKUP"
    else
        # Cek umur backup penuh
        ONE_WEEK_AGO=$(date -d "7 days ago" +%s)
        BACKUP_TIME=$(stat -c %Y "$FULL_BACKUP")

        if [ "$BACKUP_TIME" -lt "$ONE_WEEK_AGO" ]; then
            echo "Backup full lebih dari seminggu yang lalu. Backup full baru akan dibuat."
            tar --create --verbose --preserve-permissions \
                --file="$FULL_BACKUP" \
                --listed-incremental="$SNAPSHOT_FILE" "$HOME_DIR"
            BACKUP_FILE="$FULL_BACKUP"
        else
            echo "Melakukan backup incremental."
            tar --create --verbose --preserve-permissions \
                --file="$INCR_BACKUP" \
                --listed-incremental="$SNAPSHOT_FILE" "$HOME_DIR"
            BACKUP_FILE="$INCR_BACKUP"
        fi
    fi
else
    echo "Input tidak valid. Hanya menyediakan opsi 'full' atau 'incr'."
    exit 1
fi

```

Jika user memilih opsi “incr” atau incremental, maka program akan menjalankan backup data directory home secara sebagian atau yang mengalami perubahan saja. Namun backup incremental ini hanya akan langsung dijalankan jika sebelumnya user telah melakukan full backup, jika belum pernah sama sekali maka akan dilakukan backup full terlebih dahulu sebelum melakukan backup incremental. Selain itu, jika backup full berusia lebih dari 1 minggu (tidak update) maka akan dibuatkan backup full baru sebelum melakukan backup incremental.

- Compress the backup using any compression tool. Inform the user that the script is doing this, because it might take some time, during which the user might start worrying if no output appears on the screen.

```

echo "Mengompresi file backup (harap tunggu)."
```

```

gzip -f "$BACKUP_FILE"

```

Setelah file backup dibuat, maka file backup akan di kompresi dengan gzip.

- Print a message informing the user about the size of the compressed backup.

```

SIZE=$(du -h "$BACKUP_FILE" | cut -f1)
echo "Backup berhasil. Ukuran file backup: $SIZE"

```

Setelah semua program backup dieksekusi maka program terakhir akan menampilkan pesan yang menunjukkan size file backup yang telah dibuat.

Merubah permission dari bash script homebackup.sh sebelum menjalankan file bash.

```
intan@ds2024:~/BashGuide/chapter8$ ls -l homebackup.sh
-rw-r--r-- 1 root root 2677 May  2 09:31 homebackup.sh
intan@ds2024:~/BashGuide/chapter8$ sudo chmod +x homebackup.sh
[sudo] password for intan:
intan@ds2024:~/BashGuide/chapter8$ ls -l homebackup.sh
-rwxr-xr-x 1 root root 2677 May  2 09:31 homebackup.sh
```

Melakukan alias file bash script homebackup.sh menjadi command homebackup:

```
intan@ds2024:~/BashGuide/chapter8$ alias homebackup="bash /home/intan/BashGuide/chapter8/homebackup.sh"
```

Percobaan menjalankan bash script :

```
intan@ds2024:/$ homebackup
Backup penuh atau incremental? (ketik: full / incr)
full
Melakukan backup full atau backup secara keseluruhan
tar: : Cannot open: No such file or directory
tar: /home/intan: Directory is new
tar: /home/intan/.cache: Directory is new
tar: /home/intan/.config: Directory is new
tar: /home/intan/.local: Directory is new
tar: /home/intan/.ssh: Directory is new
tar: /home/intan/BashGuide: Directory is new
tar: /home/intan/backups: Directory is new
tar: /home/intan/marketing: Directory is new
tar: /home/intan/mydir: Directory is new
tar: /home/intan/oke: Directory is new
tar: /home/intan/play: Directory is new
tar: /home/intan/sales: Directory is new
tar: /home/intan/support: Directory is new
tar: /home/intan/.config/procps: Directory is new
tar: /home/intan/.local/share: Directory is new
tar: /home/intan/.local/share/nano: Directory is new
tar: /home/intan/BashGuide/chapter8: Directory is new
tar: Removing leading `/' from member names
/home/intan/
/home/intan/.cache/
/home/intan/.config/
/home/intan/.config/procps/
/home/intan/.local/
/home/intan/.local/share/
/home/intan/.local/share/nano/
/home/intan/.ssh/
/home/intan/BashGuide/
/home/intan/BashGuide/chapter8/
/home/intan/backups/
/home/intan/marketing/
/home/intan/mydir/
```

```
/home/intan/.pl.sh.swo
/home/intan/.profile
/home/intan/.sudo_as_admin_successful
/home/intan/.viminfo
/home/intan/bye.txt
/home/intan/directories.txt
/home/intan/errors.txt
/home/intan/files
/home/intan/halo.txt
tar: Removing leading `/' from hard link targets
/home/intan/hello.txt
/home/intan/list
/home/intan/myerror.txt
/home/intan/pl.sh
/home/intan/p2.sh
/home/intan/p3.sh
/home/intan/ploop.sh
/home/intan/pwaktu.sh
/home/intan/sorter
/home/intan/terminal
/home/intan/yes.txt
/home/intan/z
/home/intan/.cache/motd.legal-displayed
/home/intan/.ssh/authorized_keys
/home/intan/BashGuide/chapter8/cobaExercise2.txt.gz
/home/intan/BashGuide/chapter8/exercise1.sh
/home/intan/BashGuide/chapter8/exercise2.sh
/home/intan/BashGuide/chapter8/homebackup.sh
/home/intan/BashGuide/chapter8/yes1.txt
/home/intan/BashGuide/chapter8/yes2.txt
/home/intan/backups/full_backup.tar
/home/intan/mydir/halo.juga
/home/intan/oke/iya
/home/intan/play/terminal
Mengompresi backup (harap tunggu)...
Backup berhasil. Ukuran file backup: 5.4M
intan@ds2024:/$
```

- 4) Write a script called simple-useradd.sh that adds a local user to the system. This script should:
- Take only one argument, or else exit after printing a usage message.

```
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 username"
    exit 1
fi

USERNAME="$1"
```

Memastikan hanya satu argumen (username) diberikan dan jika argumen yang diberikan lebih dari satu maka akan menampilkan pesan penggunaan dan keluar.

- Check /etc/passwd and decide on the first free user ID. Print a message containing this ID.

```
NEXT_UID=$(awk -F: '$3>=1000 {uids[$3]=1} END {for(i=1000;i<60000;i++) if(!uids[i]) {print i; break}}' /etc/passwd)
echo "User ID yang akan digunakan: $NEXT_UID"
```

Command `“awk -F: '$3>=1000 {uids[$3]=1} END {for(i=1000;i<60000;i++) if(!uids[i]) {print i; break}}' /etc/passwd”`

Mencari UID ≥ 1000 yang belum digunakan di /etc/passwd, dimana UID 1000+ biasanya digunakan untuk user biasa (bukan sistem). Hal ini bertujuan untuk menghindari tabrakan dengan UID yang sudah ada.

Command `echo "User ID yang akan digunakan: $NEXT_UID"`

Menampilkan user ID yang akan digunakan oleh user baru yang akan dibuat.

- Create a private group for this user, checking the /etc/group file. Print a message containing the group ID.

Melakukan pengecekan apakah grup sudah tersedia dengan kondisional statement IF ELSE:

```
if grep -q "^$USERNAME:" /etc/group; then
    echo "Grup $USERNAME sudah ada."
```

Mengecek apakah group dari user sudah tersedia di file /etc/group jika ada maka memenuhi kondisional statement IF ini dan akan menampilkan suatu pesan.

```
else
    echo "$USERNAME:x:$NEXT_UID:" >> /etc/group
    echo "Grup privat '$USERNAME' dibuat dengan GID $NEXT_UID"
fi
```

Apabila tidak memenuhi kondisional statement IF maka group user belum tersedia sehingga akan menjalankan program ELSE. Diantaranya membuat grup dengan nama sama seperti username dan GID akan disamakan dengan UID user (model privat group di Linux).

- Gather information from the operator user: a comment describing this user, choice from a list of shells (test for acceptability, else exit printing a message), expiration date for this account, extra groups of which the new user should be a member.

```
read -p "Masukkan komentar/deskripsi user: " COMMENT
echo "Pilih shell (misal: /bin/bash, /bin/sh, /bin/zsh):"
read -p "Shell: " SHELL

if ! grep -qx "$SHELL" /etc/shells; then
    echo "Shell tidak valid. Pilihan harus dari /etc/shells"
    exit 1
fi

read -p "Masukkan tanggal kedaluwarsa akun (format YYYY-MM-DD): " EXPIRY
read -p "Masukkan grup tambahan (dipisah koma, atau kosong): " EXTRA_GROUPS
```

Melakukan input beberapa data yang diperlukan dalam pembuatan user dan menyimpannya dalam beberapa variabel berikut:

- COMMENT: Berisi inputan deskripsi tentang user.
- SHELL: Berisi inputan path bash dimana user akan menjalankan berbagai command atau program bash script (/etc/shells).

- EXPIRY: Berisi batas waktu Tahun-Bulan-Tanggal aktivasi dari user (/etc/shadow)
 - EXTRA_GROUPS : Berisi nama group dari user (user mau dimasukkan ke group mana) namun bisa dikosongkan.
- With the obtained information, add a line to /etc/passwd, /etc/group and /etc/shadow; create the user's home directory (with correct permissions!); add the user to the desired secondary groups.

```
HOME_DIR="/home/$USERNAME"
```

Membuat path direktori home untuk user yang tersimpan dalam variabel HOME_DIR.

```
echo "$USERNAME:x:$NEXT_UID:$NEXT_UID:$COMMENT:$HOME_DIR:$SHELL" >> /etc/passwd
```

Menambahkan informasi yang telah diinputkan pada program sebelumnya ke file /etc/passwd.

- Set the password for this user to a default known string.

```
ENCRYPTED_PASS=$(openssl passwd -6 "Password123!")
EXPIRE_DAYS=$(date -d "$EXPIRY" +%s)
EXPIRE_DAYS=$((EXPIRE_DAYS / 86400))
echo "$USERNAME:$ENCRYPTED_PASS:$(date +%s):0:99999:7:::$EXPIRE_DAYS" >> /etc/shadow
```

Membuat password default yang dienkripsi (algoritma SHA-512). Kemudian tanggal kedaluwarsa password dikonversi ke format epoch days. Lalu menambah entri nama username baru, password username yang telah dienkripsi dan expiry password pada file /etc/shadow.

```
mkdir -p "$HOME_DIR"
chown "$USERNAME:$USERNAME" "$HOME_DIR"
chmod 700 "$HOME_DIR"
```

Membuat direktori home untuk user baru berdasarkan path directory dari variabel HOME_DIR. Direktori home dari user baru dibuat dengan izin 700 (hanya owner). Dan hak milik home directory diberikan ke user yang baru dibuat.

```
if [ -n "$EXTRA_GROUPS" ]; then
    IFS=',' read -ra GROUPS <<< "$EXTRA_GROUPS"
    for group in "${GROUPS[@]"; do
        if ! grep -q "^$group:" /etc/group; then
            echo "$group:x:$((++NEXT_UID)):" >> /etc/group
            echo "Grup tambahan '$group' dibuat."
        fi
        sed -i "/^$group:/ s/$/, $USERNAME/" /etc/group
    done
fi
```

Program diatas menangani penambahan group dari user baru. Jika grup yang akan ditetapkan pada user baru tidak ada maka akan dibuat. Lalu user ditambahkan sebagai anggota grup tersebut.

```
echo "User '$USERNAME' berhasil dibuat dengan password default 'Password123!','"
```

Program diatas menampilkan pesan bahwa user berhasil ditambahkan dan diberi password default.

- Merubah permission dari file bash script simple-useradd.sh agar dapat di eksekusi.

```
intan@ds2024:~/BashGuide/chapter8$ sudo chmod +x simple-useradd.sh
intan@ds2024:~/BashGuide/chapter8$ ls -l simple-useradd.sh
-rwxr-xr-x 1 root root 1974 May  3 03:17 simple-useradd.sh
```

- Membuat alias dari eksekusi bash script simple-useradd.sh.

```
root@ds2024:/# alias tambahuser="bash /home/intan/BashGuide/chapter8/simple-useradd.sh"
```

- Pecobaan menjalankan bash script simple-useradd.sh

```
root@ds2024:/# tambahuser cobaUser
User ID yang akan digunakan: 1008
Grup cobaUser sudah ada.
Masukkan komentar/deskripsi user: userCobaCoba
Pilih shell (misal: /bin/bash, /bin/sh, /bin/zsh):
Shell: /bin/bash
Masukkan tanggal kedaluwarsa akun (format YYYY-MM-DD): 2033-03-03
Masukkan grup tambahan (dipisah koma, atau kosong): kosong
Grup tambahan '0' dibuat.
User 'cobaUser' berhasil dibuat dengan password default 'Password123!'.
```

- Mengecek user yang telah dibuat.

```
root@ds2024:/home/intan/BashGuide/chapter8# awk -F: '$3 >= 1000 { print $1 }' /etc/passwd
nobody
intan
cobaUser
```

- 5) Rewrite the script from [Section 7.2.1.4](#) so that it reads input from the user instead of taking it from the first argument.

```
GNU nano 7.2 exercise5.sh
#!/bin/bash

read -p "Enter the filename: " FILENAME

echo "Properties for $FILENAME:"

if [ -f "$FILENAME" ]; then
    echo "Size is $(ls -lh "$FILENAME" | awk '{ print $5 }')"
    echo "Type is $(file "$FILENAME" | cut -d":" -f2-)"
    echo "Inode number is $(ls -li "$FILENAME" | awk '{ print $1 }')"
    echo "$(df -h "$FILENAME" | awk 'NR==2 { print "On", $1, "which is mounted as the", $6, "partition." }')"
else
    echo "File does not exist."
fi
```

Perubahan sintax hanya terdapat pada command yang mulanya *FILENAME="\$1"* menjadi *read -p "Enter the filename: " FILENAME* dimana user dapat menjalankan bash script ini untuk mencari informasi suatu file secara bebas dengan input user namun input user ini dijalankan dalam program bash script yang tengah berjalan bukan sebagai argument.

- Input Nama File

```
read -p "Enter the filename: " FILENAME
```

Pengguna diminta untuk memasukkan nama file yang ingin diperiksa.

- Menampilkan Properti File

```
echo "Properties for $FILENAME:"
```

Menampilkan judul informasi untuk file yang dimaksud.

- Pengecekan Apakah File Ada

```

if [ -f "$FILENAME" ]; then
    echo "Size is $(ls -lh "$FILENAME" | awk '{ print $5 }')"
    echo "Type is $(file "$FILENAME" | cut -d":" -f2-)"
    echo "Inode number is $(ls -li "$FILENAME" | awk '{ print $1 }')"
    echo "$(df -h "$FILENAME" | awk 'NR==2 { print "On", $1 ", which is mounted as the", $6, "partition." }')"
else
    echo "File does not exist."
fi

```

Mengecek apakah file tersebut benar-benar ada dan merupakan file biasa.

- Jika file ada maka akan menampilkan Informasi dari file seperti ukuran file, tipe file, nomor inode, dan informasi partisi tempat file berada.

```

if [ -f "$FILENAME" ]; then
    echo "Size is $(ls -lh "$FILENAME" | awk '{ print $5 }')"
    echo "Type is $(file "$FILENAME" | cut -d":" -f2-)"
    echo "Inode number is $(ls -li "$FILENAME" | awk '{ print $1 }')"
    echo "$(df -h "$FILENAME" | awk 'NR==2 { print "On", $1 ", which is mounted as the", $6, "partition." }')"

```

- Jika File Tidak Ada

```

else
    echo "File does not exist."
fi

```

Dibawah ini kami merubah permission dari exercise5.sh agar program bash didalam file tersebut dapat dieksekusi.

```

intan@ds2024:~/BashGuide/chapter8$ ls -l exercise5.sh
-rw-r--r-- 1 root root 444 May  3 05:00 exercise5.sh
intan@ds2024:~/BashGuide/chapter8$ chmod +x exercise5.sh
chmod: changing permissions of 'exercise5.sh': Operation not permitted
intan@ds2024:~/BashGuide/chapter8$ sudo chmod +x exercise5.sh
intan@ds2024:~/BashGuide/chapter8$ ls -l exercise5.sh
-rwxr-xr-x 1 root root 444 May  3 05:00 exercise5.sh

```

Dibawah ini merupakan percobaan menjalankan program bash script exercise5.sh :

```

intan@ds2024:~/BashGuide/chapter8$ ./exercise5.sh
Enter the filename: /etc/groups
Properties for /etc/groups:
File does not exist.
intan@ds2024:~/BashGuide/chapter8$ ./exercise5.sh
Enter the filename: /etc/group
Properties for /etc/group:
Size is 1.1K
Type is ASCII text
Inode number is 133024
On /dev/mapper/ubuntu--vg-ubuntu--lv, which is mounted as the / partition.

```

Chapter 11. Functions

Here are some useful things you can do using functions:

1. Add a function to your ~/.bashrc config file that automates the printing of man pages. The result should be that you type something like **printman <command>**, upon which the first appropriate man page rolls out of your printer. Check using a pseudo printer device for testing purposes.

```

GNU nano 7.2 /home/intan/.bashrc *
# Chapter 11. Functions
# Exercise 11.4.1

# Create unction to print a man page to a pseudo printer
printman() {
    if [ -z "$1" ]; then
        echo "Usage: printman <command> [section]"
        return 1
    fi

```

```

CMD="$1"
SECTION="$2"

# Jika section diberikan, gunakan; jika tidak, abaikan
if [ -n "$SECTION" ]; then
    MANPAGE=$(man "$SECTION" "$CMD" | col -b)
else
    MANPAGE=$(man "$CMD" | col -b)
fi

# Jika tidak berhasil, menampilkan pesan warning pada pengguna
if [ -z "$MANPAGE" ]; then
    echo "Man page not found for '$CMD' in section '$SECTION'."
    return 1
fi

# Simulasi pencetakan ke file di /tmp
OUTPUT="/tmp/${CMD}_manpage.txt"
echo "$MANPAGE" > "$OUTPUT"

echo "Man page for '$CMD' has been 'printed' to $OUTPUT."
)|

```

[^]G Help [^]O Write Out [^]W Where Is [^]K Cut [^]T Execute
[^]X Exit [^]R Read File [^]\ Replace [^]U Paste [^]J Justify

```

GNU nano 7.2 /home/intan/.bashrc *
# Chapter 11. Functions
# Exercise 11.4.1

# Create unction to print a man page to a pseudo printer
printman() {

```

Mendefinisikan function printman

```

    if [ -z "$1" ]; then
        echo "Usage: printman <command> [section]"
        return 1
    fi

```

Memastikan bahwa hanya satu argumen (command) diberikan untuk menjalankan fungsi. Jika argumen yang diberikan lebih dari satu maka akan menampilkan petunjuk pemakaian dan keluar dari function dengan syntax error 1 (return 1).

```

CMD="$1"
SECTION="$2"

```

Menyimpan nama perintah (\$1) dan opsi section (\$2) ke dalam variabel.

```

    if [ -n "$SECTION" ]; then
        MANPAGE=$(man "$SECTION" "$CMD" | col -b)
    else
        MANPAGE=$(man "$CMD" | col -b)
    fi

```

Jika section diberikan, maka akan dijalankan man <section> <command>.

Namun jika tidak, maka akan menjalankan man <command>. Kemudian pipeline ke col - b digunakan untuk membersihkan karakter backspace dari output man, agar hasil bersih dan bisa dicetak dengan benar.

```
if [ -z "$MANPAGE" ]; then
    echo "Man page not found for '$CMD' in section '$SECTION'."
    return 1
fi
```

Memastikan apakah output dari man kosong. Jika ya, maka akan menampilkan pesan bahwa halaman manual tidak ditemukan.

```
OUTPUT="/tmp/${CMD}_manpage.txt"
echo "$MANPAGE" > "$OUTPUT"
```

Menyimpan hasil printman ke file \$command_manpage.txt di direktori /tmp. File ini mensimulasikan perangkat printer.

```
    echo "Man page for '$CMD' has been 'printed' to $OUTPUT."
}|
```

Program berakhir dengan menampilkan pesan yang memberi tahu pengguna bahwa halaman manual telah "dicetak".

Berikut dibawah ini percobaan yang saya lakukan dari program bash script yang baru saya buat:

```
intan@ds2024:/$ printman ls
Man page for 'ls' has been 'printed' to /tmp/ls_manpage.txt.
intan@ds2024:/$ cat /tmp/ls_manpage.txt
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by
    default). Sort entries alphabetically if none of -cftuvSUX
    nor --sort is specified.

    Mandatory arguments to long options are mandatory for short
    options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file
```

As an extra, build in a possibility for the user to supply the section number of the man page he or she wants to print.

```
intan@ds2024:/$ printman printf 3
Man page for 'printf' has been 'printed' to /tmp/printf_manpage.txt.
intan@ds2024:/$ cat /tmp/printf_manpage.txt
printf(3)                Library Functions Manual                printf(3)

NAME
    printf, fprintf, dprintf, sprintf, snprintf, vprintf, vf-
    printf, vdprintf, vsprintf, vsnprintf - formatted output
    conversion

LIBRARY
    Standard C library (libc, -lc)

SYNOPSIS
    #include <stdio.h>

    int printf(const char *restrict format, ...);
    int fprintf(FILE *restrict stream,
                const char *restrict format, ...);
    int dprintf(int fd,
                const char *restrict format, ...);
    int sprintf(char *restrict str,
                const char *restrict format, ...);
```

2. Create a subdirectory in your home directory in which you can store function definitions. Put a couple of functions in that directory. Useful functions might be, amongs others, that you have the same commands as on DOS or a commercial UNIX when working with Linux, or vice versa. These functions should then be imported in your shell environment when `~/bashrc` is read.

- Membuat subdirektori `chapter11` pada direktori home kita yang akan menampung beberapa file bash script yang akan kita buat `mkdir -p ~/BashGuide/chapter11`.

```
intan@ds2024:/$ mkdir -p ~/BashGuide/chapter11
```

- Membuat file bash script yang berisi beberapa alias command OS Debian linux yang dibuat meniru OS DOS. Command dari OS yang akan ditiru diantaranya `dir`, `copy`, `del`, `md`, `rd`.


```

GNU nano 7.2          chapter11/dos_aliases.sh
# ~/.bash_funcs/dos_aliases.sh
# Membuat alias ala DOS jika Anda terbiasa di Windows
# Contoh: dir → ls, copy → cp, del → rm

dos_aliases() {
    alias dir='ls -l'
    alias copy='cp'
    alias del='rm'
    alias md='mkdir'
    alias rd='rmdir'
    echo "DOS-style aliases loaded."
}

```

- Membuat file bash script yang berisi beberapa alias command OS Debian linux yang dibuat meniru OS UNIX. Command dari OS yang akan ditiru diantaranya ll, psg, h dan “..”.

```

GNU nano 7.2          chapter11/unix_aliases.sh
# ~/.bash_funcs/unix_to_linux.sh
# Membuat alias ala commercial UNIX (HP-UX/AIX) ke Linux
# Contoh: ll → ls -la, dfh → df -h

unix_aliases() {
    alias ll='ls -la'
    alias psg='ps -ef | grep'
    alias h='history'
    alias ..='cd ..'
    echo "UNIX-style aliases loaded."
}

```

- Menambahkan loader.
 Disini kita akan mendaftarkan fungsi-fungsi yang terdapat pada file bash script (.sh) dalam direktori “~/BashGuide/chapter11” ke lingkungan Bash (~/.bashrc).
 Ketersediaan direktori akan dicek terlebih dahulu dengan kondisional statement `if [-d "$HOME//BashGuide/chapter11"]`. Jika ada, semua .sh di dalamnya di-source, jadi fungsi seperti dos_aliases dan unix_aliases tersedia untuk dipanggil.

```

GNU nano 7.2          /home/intan/.bashrc
# Load custom function definitions from ~/BashGuide/chapter11

# Periksa direktori
if [ -d "$HOME/BashGuide/chapter11/" ]; then
    for file in "$HOME/BashGuide/chapter11"/*.sh; do
        # Sumber (source) setiap file jika ada
        [ -r "$file" ] && source "$file"
    done
fi

```

- Melakukan reload .bashrc.

```

intan@ds2024:/$ source ~/.bashrc
Profile dari .bashrc

```

- Melakukan pemanggilan fungsi

```
intan@ds2024:/$ dos_aliases
DOS-style aliases loaded.
intan@ds2024:/$ unix_aliases
UNIX-style aliases loaded.
```

Berikut percobaan penggunaan command alias dari dos_aliases function:

```
intan@ds2024:~$ dir
total 562888
drwxrwxrwx 2 root root      4096 May  2 12:03 backups
drwxrwxr-x 4 intan intan    4096 May  3 07:08 BashGuide
lrwxrwxrwx 1 intan intan      1 Apr  3 11:34 bye.txt -> z
-rw-rw-r-- 1 intan intan 10754412 Apr  8 02:07 directories.txt
-rw-rw-r-- 1 intan intan   53150 Apr  8 02:07 errors.txt
-rw-rw-r-- 1 intan intan 10835989 Apr 18 11:48 files
-rw-rw-r-- 3 intan intan     16 Apr  3 11:31 halo.txt
-rw-rw-r-- 1 intan intan     12 Apr 11 02:24 hello.txt
-rw-rw-r-- 1 intan intan     97 Apr 20 00:24 list
drwxrwxr-x 2 intan intan    4096 Apr 20 04:59 marketing
drwxrwxr-x 2 intan intan    4096 Apr  3 11:32 mydir
-rw-rw-r-- 1 intan intan    186 Apr  3 12:01 myerror.txt
drwxrwxr-x 2 intan intan    4096 Apr 20 09:08 oke
-rwxrwxr-x 1 intan intan     40 Apr 20 03:02 pl.sh
-rwxrwxr-x 1 intan intan     41 Apr 20 03:11 p2.sh
-rw-rw-r-- 1 intan intan     35 Apr 15 15:25 p3.sh
drwxrwxr-x 2 intan intan    4096 Apr 11 02:23 play
-rwxrwxr-x 1 intan intan     55 Apr 20 03:58 ploop.sh
-rw-r--r-- 1 root root       58 Apr 18 10:45 pwaktu.sh
drwxrwxr-x 2 intan intan    4096 Apr 20 04:59 sales
-rwxrwxr-x 1 intan intan     37 Apr 19 00:33 sorter
drwxrwxr-x 2 intan intan    4096 Apr 20 04:59 support
-rw-rw-r-- 1 intan intan     12 Apr 11 02:25 terminal
-rw-rw-r-- 1 intan intan 554663936 Apr  5 05:23 yes.txt
-rw-rw-r-- 3 intan intan     16 Apr  3 11:31 z
```

Berikut percobaan penggunaan command alias dari unix_aliases function:

```
intan@ds2024:~$ psg intan
root      956      955  0 01:48 ?      00:00:00 sshd: intan [priv]
intan     961        1  0 01:48 ?      00:00:00 /usr/lib/systemd/systemd --
user
intan     962      961  0 01:48 ?      00:00:00 (sd-pam)
intan    1068      956  0 01:48 ?      00:02:49 sshd: intan@pts/0
intan    1069    1068  0 01:48 pts/0    00:00:00 -bash
root     1073      955  0 01:48 ?      00:00:00 sshd: intan [priv]
intan    1126    1073  0 01:48 ?      00:00:02 sshd: intan@pts/1
intan    1127    1126  0 01:48 pts/1    00:00:00 -bash
root     1907    1733  0 04:59 pts/0    00:00:00 su - intan
intan    1908    1907  0 04:59 pts/0    00:00:00 -bash
intan    2843    1908 50 08:33 pts/0    00:00:00 ps -ef
intan    2844    1908  0 08:33 pts/0    00:00:00 grep --color=auto intan
```

MATA KULIAH PRAKTIKUM MANAJEMEN DATA

How to Used Docker ?

Dosen Pengampu :

Isbat Uzzin Nadhori, S.Kom, M.T., & Prasetyo Wibowo, S.ST., M.Kom



Disusun Oleh:

Intan Dwi Anggreini (3324600006)

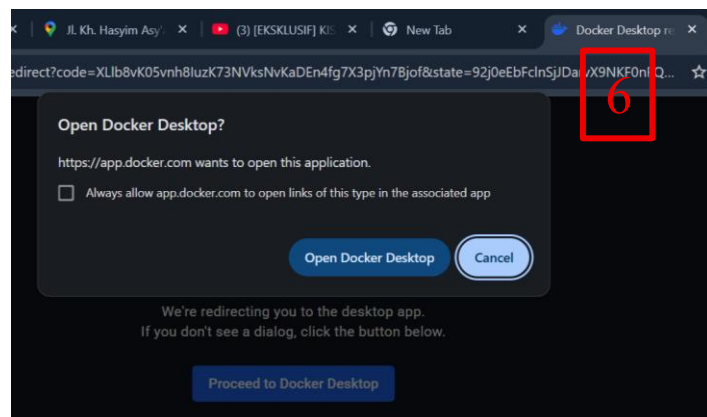
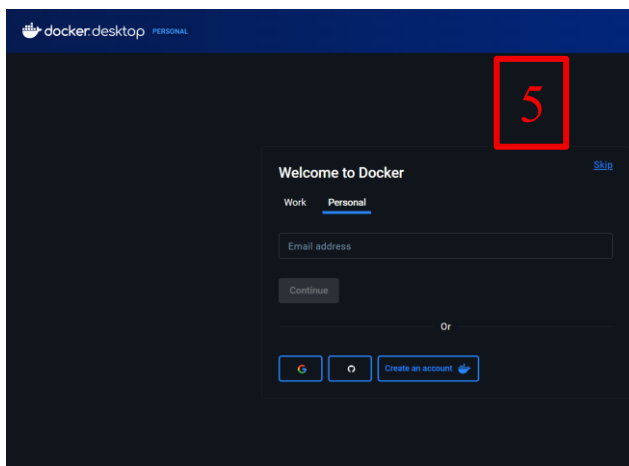
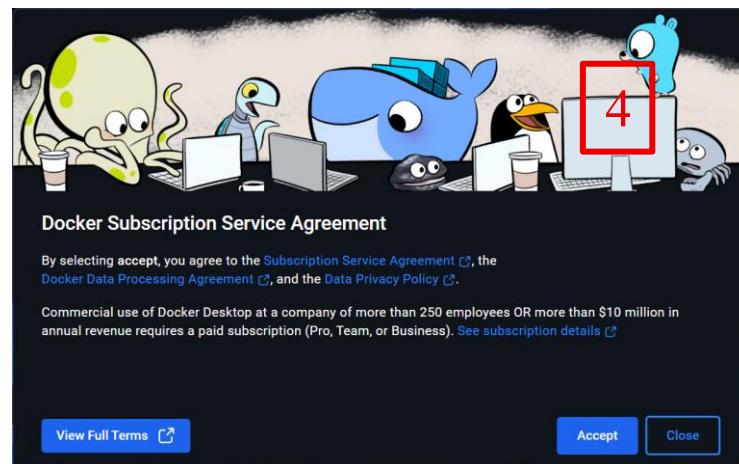
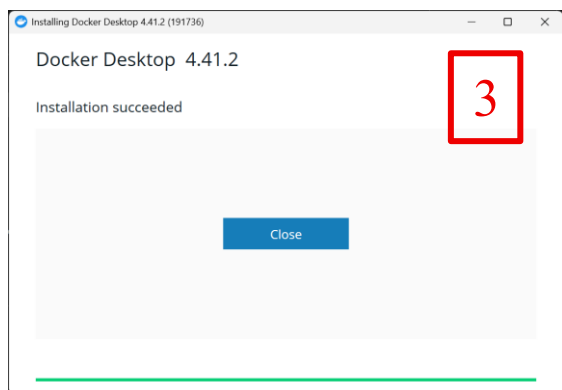
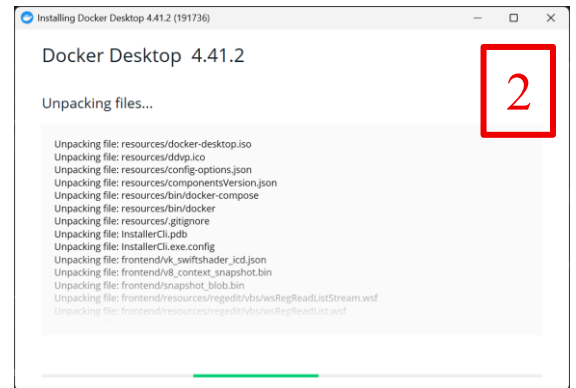
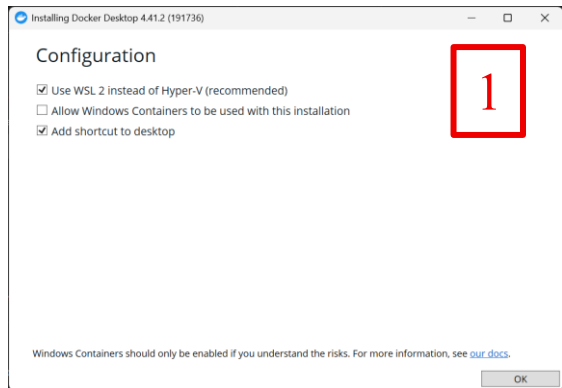
1 D4 SAINS DATA TERAPAN

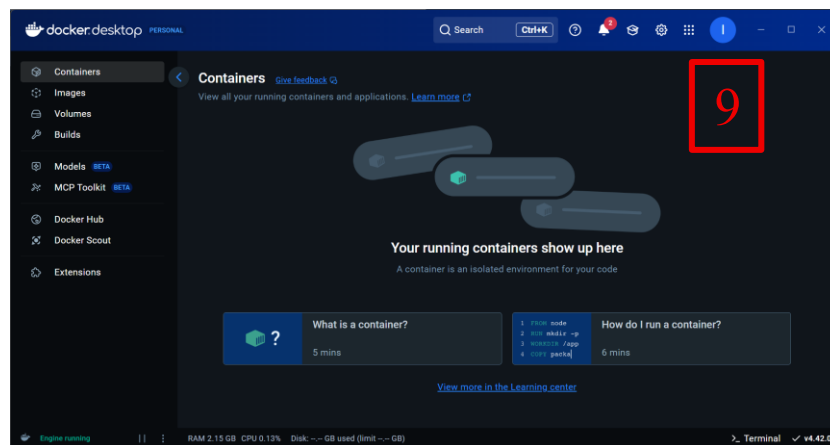
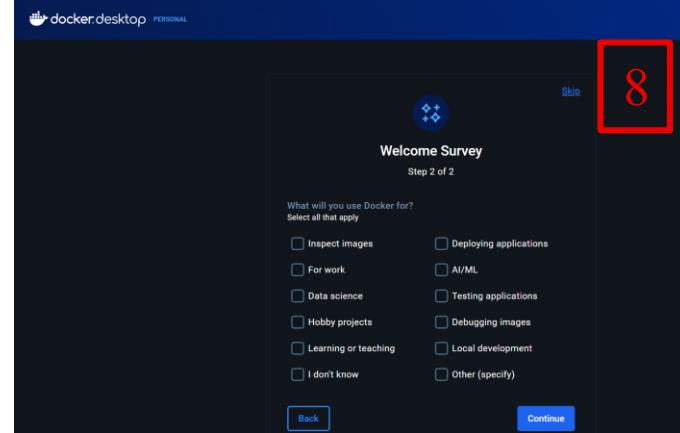
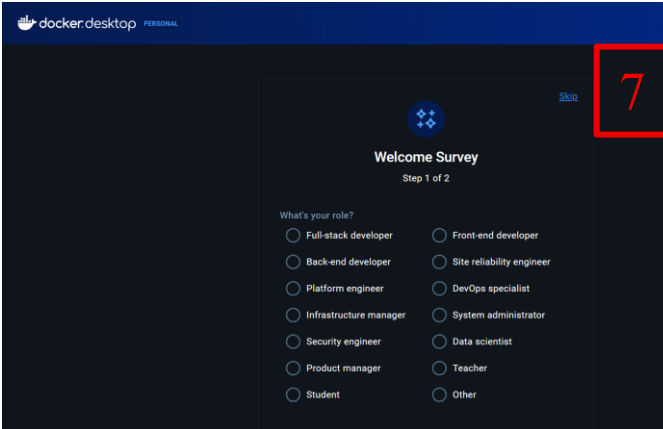
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

Jl. Raya ITS, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111

A. Melakukan Instalasi Docker Desktop

1) Proses intallasi :





- 2) Pengecekan untuk memastikan docker berhasil diinstall dan mengecek versi docker

```
PS C:\WINDOWS\system32> docker --version
Docker version 28.2.2, build e6534b4
```

- 3) Melakukan login ke docker di CMD sebelum melanjutkan berbagai konfigurasi docker.

```
PS C:\WINDOWS\system32> docker login
Authenticating with existing credentials... [Username: intanangreini99]

Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
```

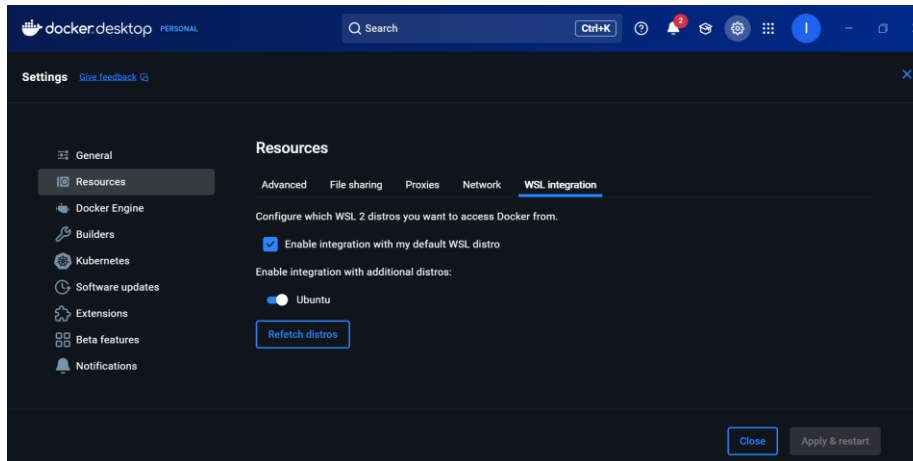
- 4) Melakukan instalasi wsl ubuntu

```
PS C:\WINDOWS\system32> wsl --install
Downloading: Ubuntu
Installing: Ubuntu
Distribution successfully installed. It can be launched via 'wsl.exe -d Ubuntu'
Launching Ubuntu...
Provisioning the new WSL instance Ubuntu
This might take a while...
Create a default Unix user account: dintanangreini
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Mengecek keberhasilan installasi

```
PS C:\WINDOWS\system32> wsl --version
WSL version: 2.5.7.0
Kernel version: 6.6.87.1-1
WSLg version: 1.0.66
MSRDC version: 1.2.6074
Direct3D version: 1.611.1-81528511
DXCore version: 10.0.26100.1-240331-1435.ge-release
Windows version: 10.0.26100.4349
```

5) Mengintegrasikan WSL dengan docker



B. Melakukan Instalasi Image

1) Mengecek daftar image.

```
PS C:\WINDOWS\system32> docker images ls
REPOSITORY TAG IMAGE ID CREATED SIZE
No images found matching "ls": did you mean "docker image ls"?
```

2) Melakukan beberapa instalasi image yang dibutuhkan yaitu postgres, mysql, dan ubuntu.

- Instalasi image postgres

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
f4ce9941f6e3: Pull complete
dad67da3f26b: Pull complete
b47a445a47f0: Pull complete
928d00623a6e: Pull complete
3664068a9b37: Pull complete
db3ab53631e4: Pull complete
eb3a531023c8: Pull complete
c6def2c6e21d: Pull complete
603ef9fcdd8e: Pull complete
05b641b3bdab: Pull complete
c95f49cc11b3: Pull complete
64e8f1b2b243: Pull complete
8a1f652e0c97: Pull complete
abfd68ef219e: Pull complete
Digest: sha256:6cf6142afacfa89fb28b894d6391c7dcbf6523c33178bdc33e782b3b533a9342
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

- Instalasi image mysql

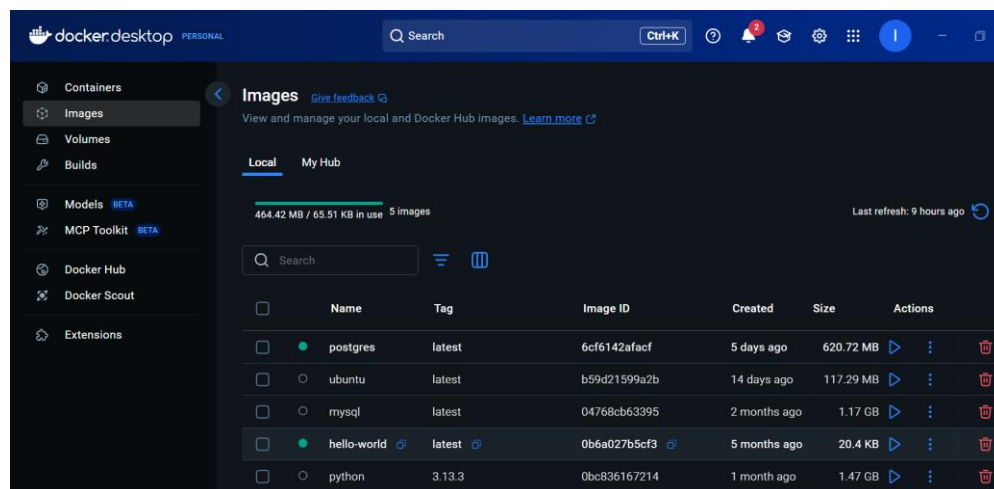
```
PS C:\WINDOWS\system32> docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
5056ce4ab875: Pull complete
72ac912b8a2e: Pull complete
b288ccce2510: Pull complete
7488ffd7127f: Pull complete
4bd1fb59dd90: Pull complete
8a50ff4ab30c: Pull complete
d23320eed97a: Pull complete
b097427f1ebe: Pull complete
7074f55c9a02: Pull complete
9845df06f911: Pull complete
Digest: sha256:04768cb63395f56140b4e92cad7c8d9f48dfa181075316e955da75aadca8a7cd
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

- Instalasi image ubuntu

```
PS C:\WINDOWS\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d9d352c11bbd: Pull complete
Digest: sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Mengecek daftar image yang telah berhasil diinstallasi :

```
PS C:\WINDOWS\system32> docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
postgres      latest    6cf6142afacf   4 days ago     621MB
ubuntu        latest    b59d21599a2b   13 days ago    117MB
mysql         latest    04768cb63395   8 weeks ago    1.17GB
hello-world   latest    0b6a027b5cf3   4 months ago   20.4kB
```



C. Menjalankan Container Postgres dalam Docker

- 1) Mengecek daftar container

```
PS C:\WINDOWS\system32> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\WINDOWS\system32>
```

2) Menjalankan container postgres.

Container dibuat dengan format command :

docker run

--name {nama_container}

-e POSTGRES_PASSWORD={password_postgresdb}

-p {port_host}:{port_container}

-d (mengaktifkan detached mode, container dijalankan di background)

{nama_image}

docker run --name postgres-main -e POSTGRES_PASSWORD=ds99999 -p 5432:5432 -d postgres

```
PS C:\WINDOWS\system32> docker run --name postgres-main -e POSTGRES_PASSWORD=ds99999 -p 5432:5432 -d postgres
5103aa09b59a47cd8dff7b9a95cc04704ff84ffae5ed12780c01e417ff6a9217
```

Mengecek daftar container yang telah dijalankan.

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|----------|--------------------------|----------------|---------------|---|---------------|
| 5103aa09b59a | postgres | "docker-entrypoint.s..." | 59 seconds ago | Up 59 seconds | 0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp | postgres-main |

3) Menginstall image dan menjalankan container pgadmin agar dapat memonitoring database postgres secara GUI.

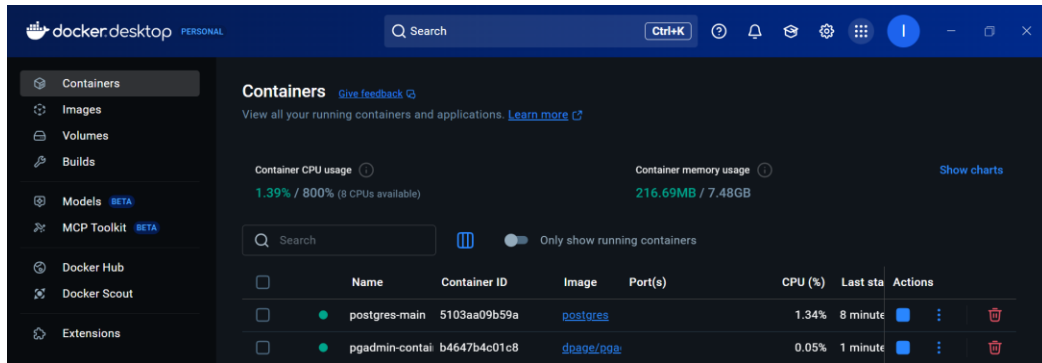
- Melakukan instalasi image pgadmin.

```
Select Administrator: Windows PowerShell
PS C:\WINDOWS\system32> docker pull dpape/pgadmin4
Using default tag: latest
latest: Pulling from dpape/pgadmin4
7c48c5894605: Pull complete
e32d37330f9b: Pull complete
8f4e00ca85d9: Pull complete
f18232174bc9: Pull complete
3c2f6c5c12fe: Pull complete
f3888f615432: Pull complete
19c31ea47c7b: Pull complete
abdb8f7da7a0: Pull complete
20690bbb262d: Pull complete
494696b06e5e: Pull complete
35d02455d858: Pull complete
02f79b30c1ff: Pull complete
9b4f03ce919b: Pull complete
77da47c991ce: Pull complete
218ba7b7b321: Pull complete
5c5e4eb233fb: Pull complete
Digest: sha256:6b1c0db09695758c72abcd5160b7c4cc6d0847ab0c90a2403bdbf951f0defb5a
Status: Downloaded newer image for dpape/pgadmin4:latest
docker.io/dpape/pgadmin4:latest
```

Mengecek daftar image yang telah diinstall

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|----------------|--------|--------------|-------------|-------|
| postgres | latest | 6cf6142afacf | 5 days ago | 621MB |
| ubuntu | latest | b59d21599a2b | 2 weeks ago | 117MB |
| dpape/pgadmin4 | latest | 6b1c0db09695 | 2 weeks ago | 787MB |

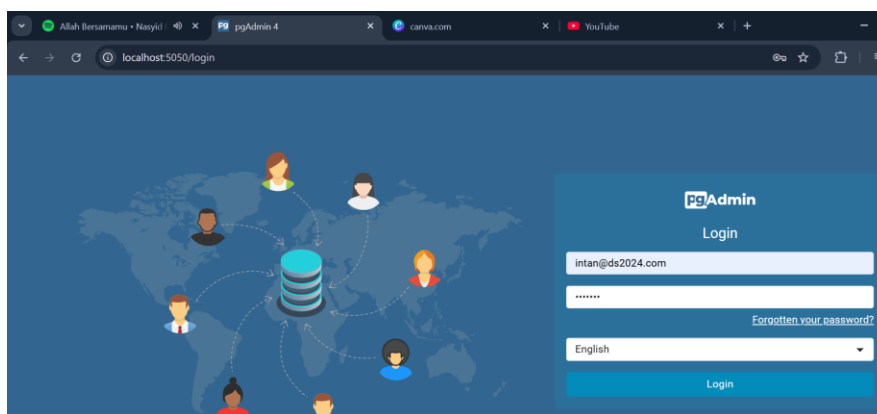
- Menjalankan container pgadmin dengan command :
docker run --name pgadmin-container -p 5050:80
-e PGADMIN_DEFAULT_EMAIL=intan@ds2024.com
-e PGADMIN_DEFAULT_PASSWORD=ds99999 -d dpape/pgadmin4



- Mengecek daftar container yang telah dijalankan.

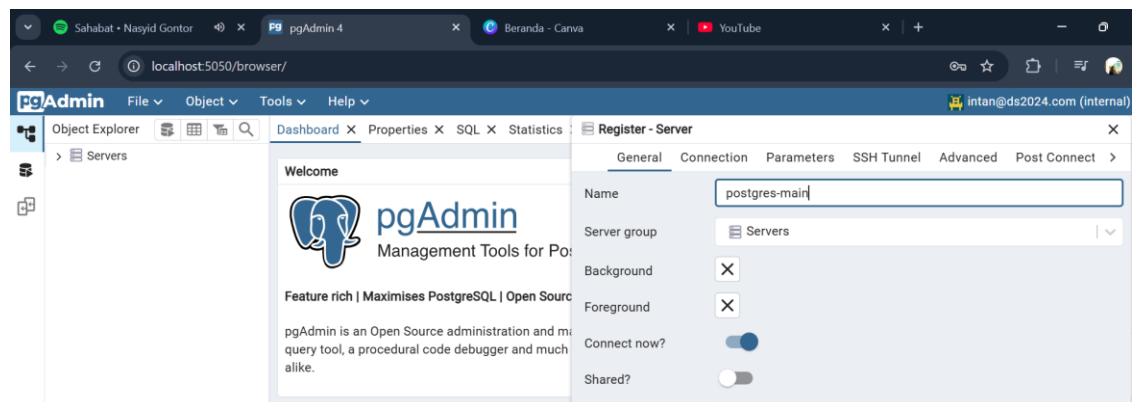
```
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
b4647b4c01c8   dpage/pgadmin4 "/entrypoint.sh"        6 seconds ago Up 5 seconds   0.0.0.0:5050->80/tcp, [::]:5050->80/tcp   pgadm
in-container   postgres      "docker-entrypoint.s..." 6 minutes ago Up 6 minutes   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp   postg
res-main
```

- Login ke halaman web pgAdmin.



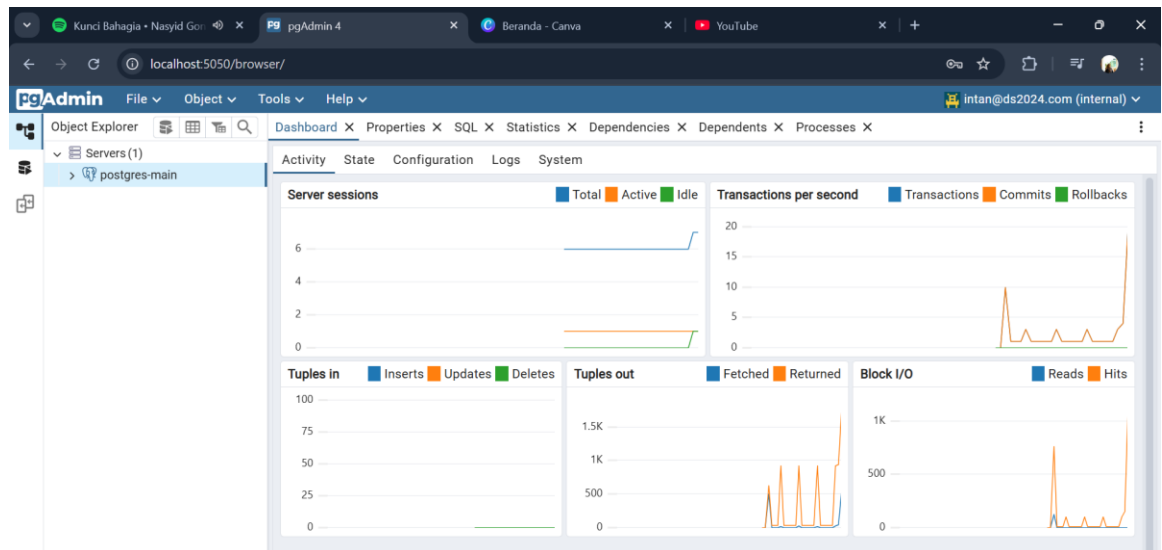
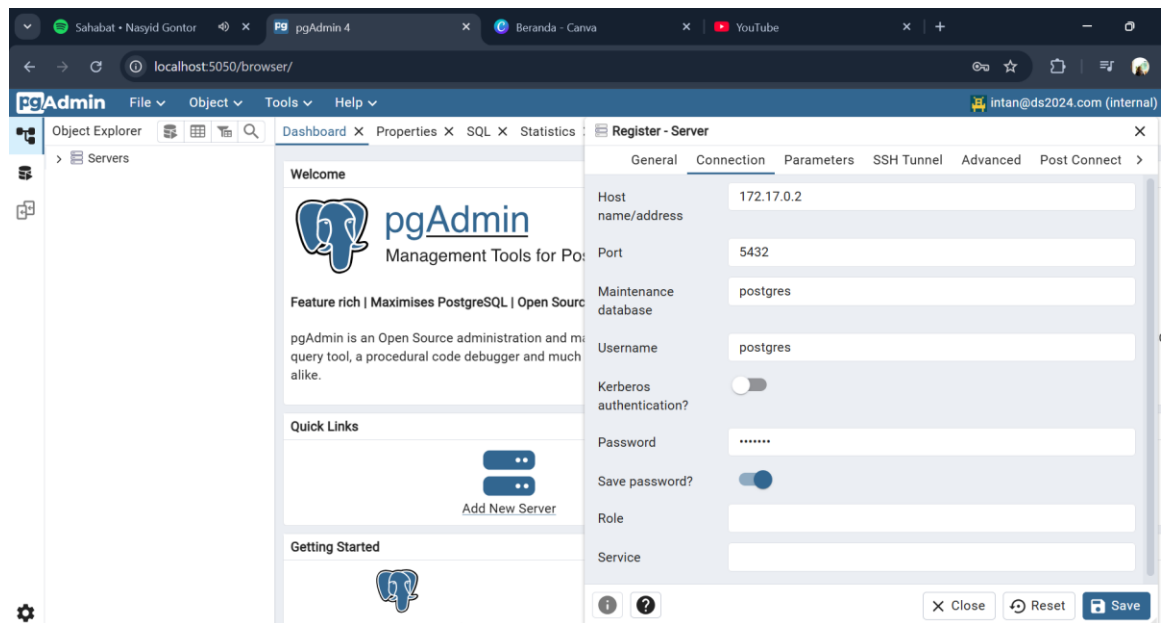
Membuat server baru untuk koneksi ke container postgres yang telah dibuat sebelumnya dengan memasukkan beberapa hal:

- **Name:** Nama Server yang dimasukkan harus sama dengan nama container postgres.
- Mengaktifkan fitur **Connect now**



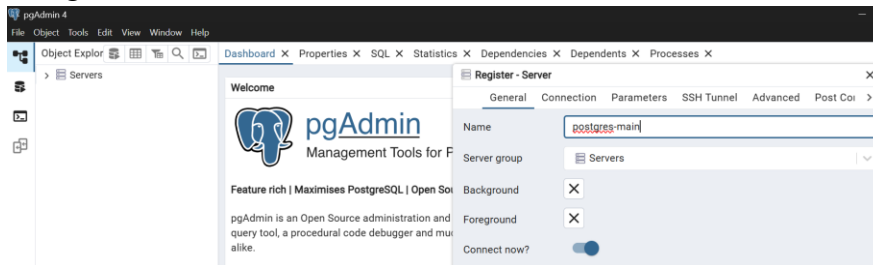
- **Host name/address:** Memasukkan alamat ip dari container dengan command:
`docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' postgres-main`

```
PS C:\WINDOWS\system32> docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' postgres-main
172.17.0.2
```
- **Port:** Memasukkan nomor port dari container postgres
- **Maintenance Database, dan Username** diisi dengan nama database, dan nama user database bawaan postgres.
- **Password:** Memasukkan password postgres yang telah kita tetapkan ketika menjalankan docker postgres-main sebelumnya.
- **Klik Save.**

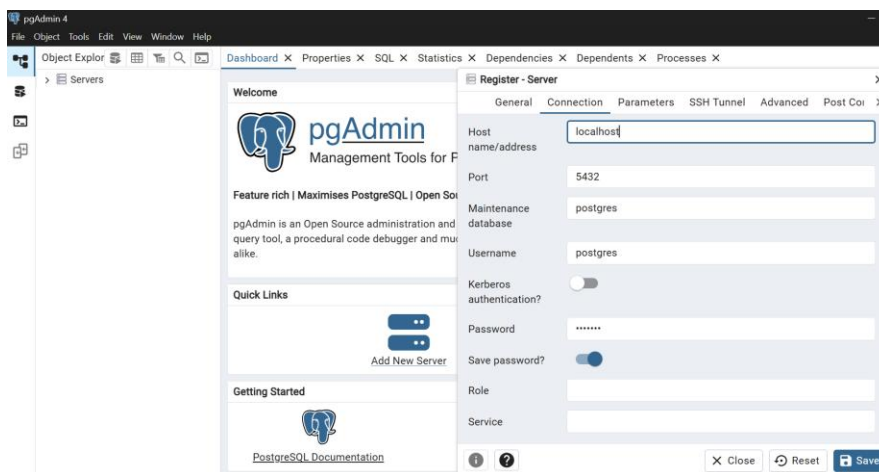


D. Menghubungkan Container Postgres dalam Docker ke pgAdmin Windows.

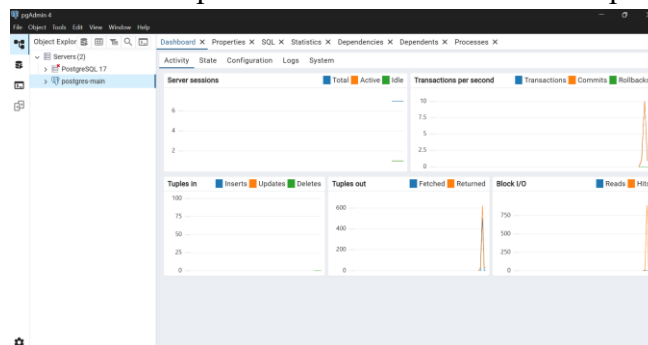
- 1) Klik **Add New Server**: Membuat server baru untuk koneksi ke container postgres yang telah dibuat sebelumnya dengan menginputkan beberapa hal:
- 2) **Name**: Nama Server yang dimasukkan harus sama dengan nama container postgres.
- 3) Mengaktifkan fitur **Connect now**



- 4) **Host name/address**: Memasukkan alamat 127.0.0.1
- 5) **Port**: Memasukkan nomor port dari container postgres
- 6) **Maintenance Database, dan Username** diisi dengan nama database, dan nama user database bawaan postgres.
- 7) **Password**: Memasukkan password postgres yang telah kita tetapkan ketika menjalankan docker postgres-main sebelumnya.
- 8) **Klik Save**.



- 9) Server berhasil dibuat dan menampilkan dashboard dari container postgres dalam docker.



E. Menjalankan Container *httpd* dalam Docker

1) Menginstall image *httpd*

```
PS C:\WINDOWS\system32> docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
c06cec1379c2: Pull complete
be5c5a616c3a: Pull complete
d0a755bf09a1: Pull complete
d1042d58e186: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:f6557a77ee2f16c50a5ccbb2564a3fd56087da311bf69a160d43f73b23d3af2d
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

Mengecek daftar image yang telah diinstall

```
PS C:\WINDOWS\system32> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
postgres latest 6cf6142afacf 5 days ago 621MB
ubuntu latest b59d21599a2b 2 weeks ago 117MB
dpage/pgadmin4 latest 6b1c0db09695 2 weeks ago 787MB
httpd latest f6557a77ee2f 4 months ago 221MB
```

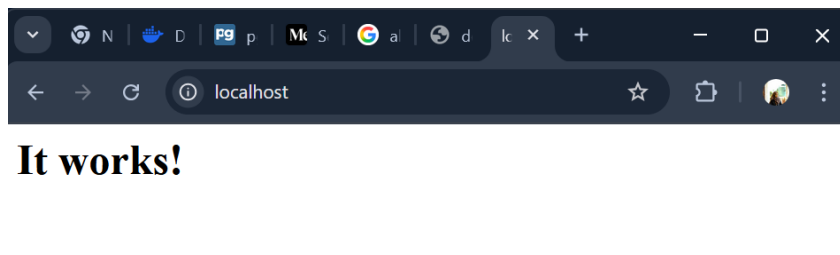
2) Menjalankan container *httpd*

`docker run --name apache-main -p 80:8080 -d httpd`

```
PS C:\WINDOWS\system32> docker run --name apache-main -p 80:8080 -d httpd
78f87b9335d6212c427a4fedfd2d18dbf78652940c745518a416359c0ecc2e68
```

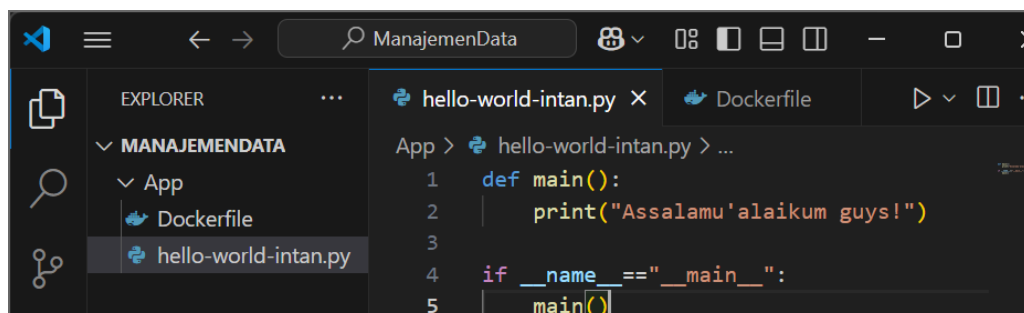
```
PS C:\WINDOWS\system32> docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
78f87b9335d6 httpd "httpd-foreground" 35 seconds ago Up 34 seconds 0.0.0.0:80->8080/tcp, [::]:80->8080/tcp apache-main
```

Mencoba membuka halaman web dari container *httpd*

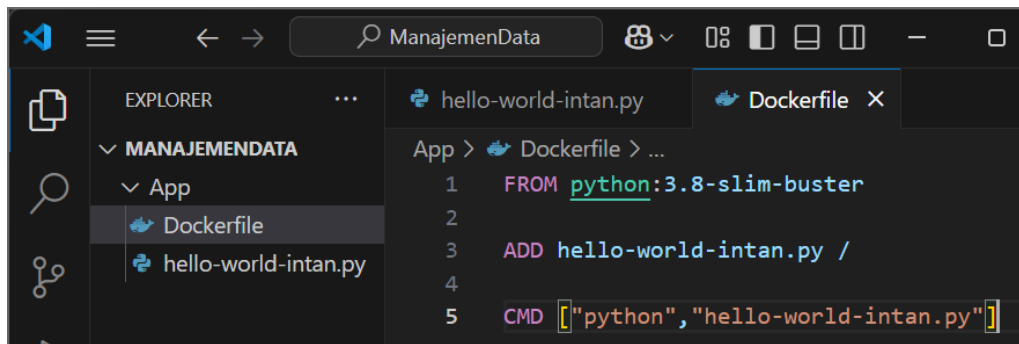


F. Membangun aplikasi python

- 1) Pastikan vscode ada atau download dan install jika belum ada
- 2) Buat file `hello-world.py`



3) Membuat Dockerfile

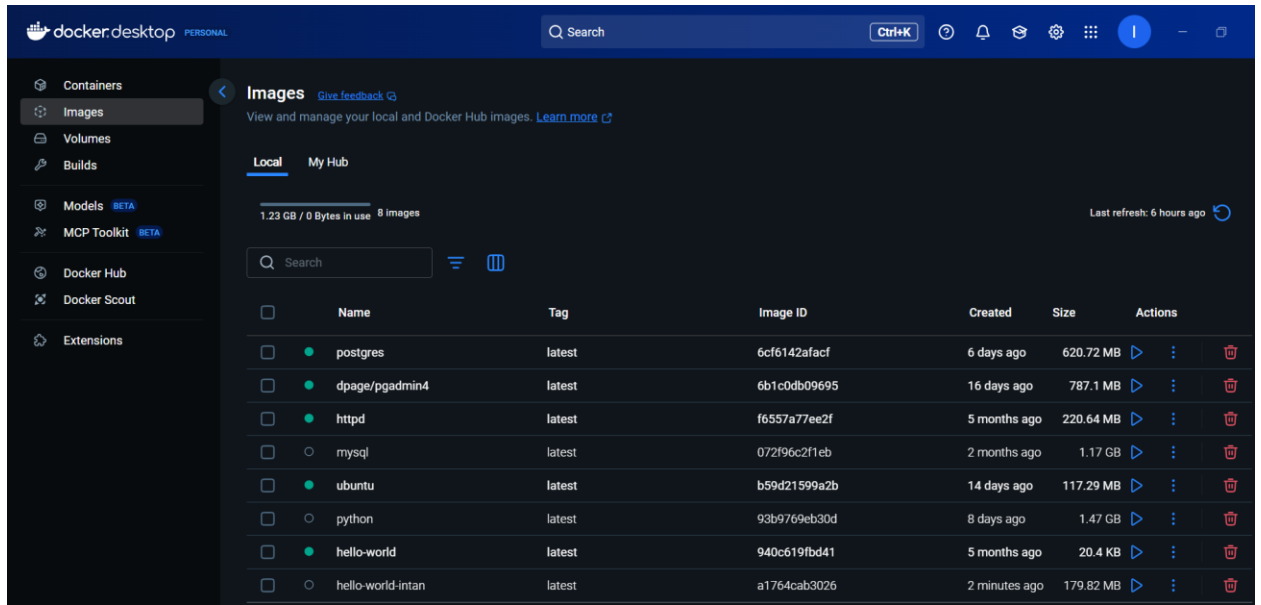


4) Membuat image hello-world

```
PS D:\ManajemenData\App> docker build -t hello-world-intan .
[+] Building 34.6s (7/7) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 135B                                0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim-buster 1.2s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 135B                                      0.0s
=> [1/2] FROM docker.io/library/python:3.8-slim-buster@sha256:8799b0564103a9f36cfb8a 32.5s
=> => resolve docker.io/library/python:3.8-slim-buster@sha256:8799b0564103a9f36cfb8a 0.0s
=> => sha256:276dfcf5deffff3c5d540a8e0d9a18656a4c03637a8b4f4eec1f4a1 3.14MB / 3.14MB 11.5s
=> => sha256:cbfea27109a8b1136059a7973ccb8243889faf162ebc173a05909dcb0bec 244B / 244B 1.1s
=> => sha256:8f777578c172d018077d3dc22d6654911fff60066097943fe8c46 12.89MB / 12.89MB 31.8s
=> => sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d 2.78MB / 2.78MB 17.6s
=> => sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e48571 27.14MB / 27.14MB 26.1s
=> => extracting sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e5fc96 1.1s
=> => extracting sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e257e 0.1s
=> => extracting sha256:8f777578c172d018077d3dc22d6654911fff60066097943fe8c4697ecf8aa 0.4s
=> => extracting sha256:cbfea27109a8b1136059a7973ccb8243889faf162ebc173a05909dcb0bec0 0.0s
=> => extracting sha256:276dfcf5deffff3c5d540a8e0d9a18656a4c03637a8b4f4eec1f4a147799c 0.2s
=> [2/2] ADD hello-world-intan.py /                                0.1s
=> exporting to image                                              0.2s
=> => exporting layers                                              0.1s
=> => exporting manifest sha256:5538a58d606695efe6b13e8ea9e36f43a5ab7271048502da38790 0.0s
=> => exporting config sha256:1f257cf3ed8313c72232275898995ec85099810226f83a0b3a09ebd 0.0s
=> => exporting attestation manifest sha256:8f13e7457cd0778f1902f990e7ebfedb74ab2c8ee 0.0s
=> => exporting manifest list sha256:a1764cab30260ac1a3f35d2ca83a365227f1194021876f8f 0.0s
=> => naming to docker.io/library/hello-world-intan:latest        0.0s
=> => unpacking to docker.io/library/hello-world-intan:latest     0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/qq6r66ni7eofgvc73gfzketad
```

```
PS D:\ManajemenData\App> docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
hello-world-intan   latest      a1764cab3026  2 minutes ago  180MB
postgres            latest      6cf6142afacf  5 days ago    621MB
python              latest      93b9769eb30d  8 days ago    1.47GB
ubuntu              latest      b59d21599a2b  2 weeks ago   117MB
dpage/pgadmin4      latest      6b1c0db09695  2 weeks ago   787MB
mysql               latest      072f96c2f1eb  8 weeks ago   1.17GB
httpd               latest      f6557a77ee2f  4 months ago  221MB
hello-world         latest      940c619fbd41  4 months ago  20.4kB
```



5) Menjalankan container hello-world

```
PS D:\ManajemenData\App> docker run hello-world-intan  
Assalamu'alaikum guys!
```

ftpuser1@192.168.6.6 - FileZilla Pro

File Edit View Transfer Server Bookmarks Help New version available!

Host: 192.168.6.6

Username: ftpuser1

Password: ●●●●●●

Port:

Quickconnect

Status: Starting download of /home/ftpuser1/BashGuide/chapter8/exercise1.sh

Status: File transfer successful, transferred 999 bytes in 1 second

Status: File transfer successful, transferred 707 bytes in 1 second

Local site: D:\ManajemenData\

file_from_git

GDGoC_ITS

Include

Lib

ManajemenData

mongodb-win8.0.9

| Filename | Filesize | Filetype | Last modified |
|-----------|----------|-------------|---------------|
| .. | | | |
| App | | File folder | 6/12/2025 ... |
| chapter11 | | File folder | 6/13/2025 ... |
| chapter8 | | File folder | 6/13/2025 ... |

3 directories

| Server/Local file | Dire... | Remote file | Size | Prio... | Status |
|-------------------|---------|-------------|------|---------|--------|
|-------------------|---------|-------------|------|---------|--------|

Queued files

Failed transfers

Successful transfers (20)

Remote site: /home/ftpuser1/BashGuide

ftpuser1

BashGuide

chapter11

chapter8

ftp

intan

samba

sambauser

| File na... | Filesize | Filetype | Last mod... | Permis... | Owner... |
|------------|----------|-------------|-------------|-----------|----------|
| cha... | | File fol... | 6/13/202... | drwxr... | 0 0 |
| cha... | | File fol... | 6/13/202... | drwxr... | 0 0 |

Selected 2 directories.

FileZilla Pro

i

Transfers finished
All files have been successfully transferred

Queue: empty