

How to install Arch Linux without Archinstall



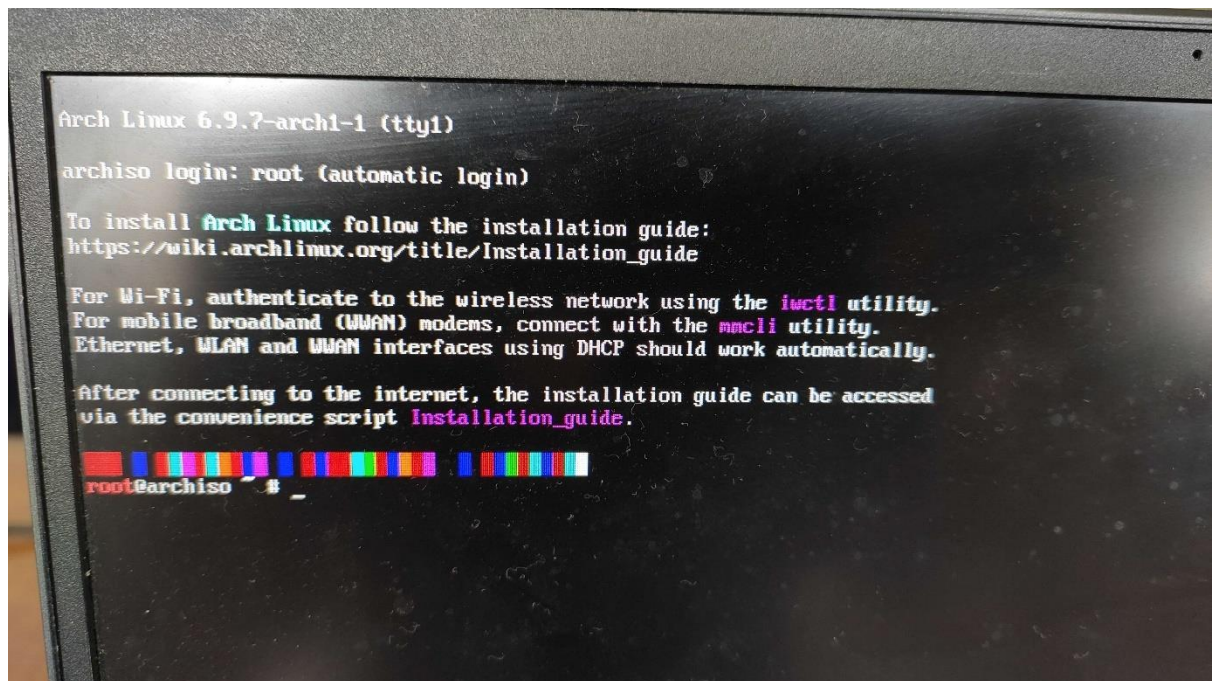
Art by: u/RealShovelKun

Disclaimer: Aku buat tutorial ini dengan asumsi kalian sudah punya bootable flashdisk dengan image arch linux didalamnya.

1. Masuk ke dalam archiso

Langkah pertama adalah untuk masuk kedalam archiso, archiso adalah tools untuk membuat arch linux melalui live CD/USB ISO image.

Cara untuk masuk ke dalam archiso adalah dengan memasukkan flashdisk atau bootable device yang berisi image ISO arch linux ke komputer yang ingin di install arch dan lalu di reboot. Tergantung komputernya tekan tombol esc atau F10 untuk masuk ke boot menu dan pilih USB boot.



Jika gambar tersebut sudah muncul, artinya sudah masuk ke dalam archiso dan kita bisa memulai instalasi.

2. Connect ke internet

Pastikan untuk memiliki kabel ethernet agar dapat terhubung ke internet secara otomatis, jika tidak kalian dapat menggunakan wifi. Untuk menghubungkan ke wifi kalian dapat membuka utility **iwctl**.

Gunakkan command **device list** untuk melihat device apa saja yang ada. Lalu gunakan command **station devicename scan** dan **station devicename get-networks** untuk melihat network apa saja yang tersedia di sekitar.

```

[fwl]# device list

```

Devices				
Name	Address	Powered	Adapter	Mode
wlan1	00:91:9e:70:24:5c	on	phy1	station

```

[fwl]# station wlan1 scan
[fwl]# station wlan1 get-networks

```

Available networks		
Network name	Security	Signal
> Wiswis	psk	00:00:00
ITB Guest	open	00:00:00
ITB Hotspot	8021x	00:00:00
eduroam	8021x	00:00:00
ITB IoT	open	00:00:00
Infinix SMART 8	psk	00:00:00
Ideahub_Informatika	open	00:00:00
OPPO A5 2020	psk	00:00:00
EB93E234-fEBDSya08q1ADuT	psk	00:00:00
zara	psk	00:00:00
TP-SPOT HS	psk	00:00:00
EB939878-fEBDSyauXu1A0Da	psk	00:00:00
EB93E73C-fEBDSya0Sy1A0XX	psk	00:00:00
Dosen-SPOT	psk	00:00:00
dpl	psk	00:00:00
BCNLounge	open	00:00:00

Setelah itu kalian dapat **station devicename connect ssid** dan ketik password dari ssid tersebut untuk terhubung ke internet. Untuk mengecek nya kalian dapat **exit** untuk keluar dari iwctl dan menggunakan command **ping google.com** untuk melihat koneksi kalian.

```

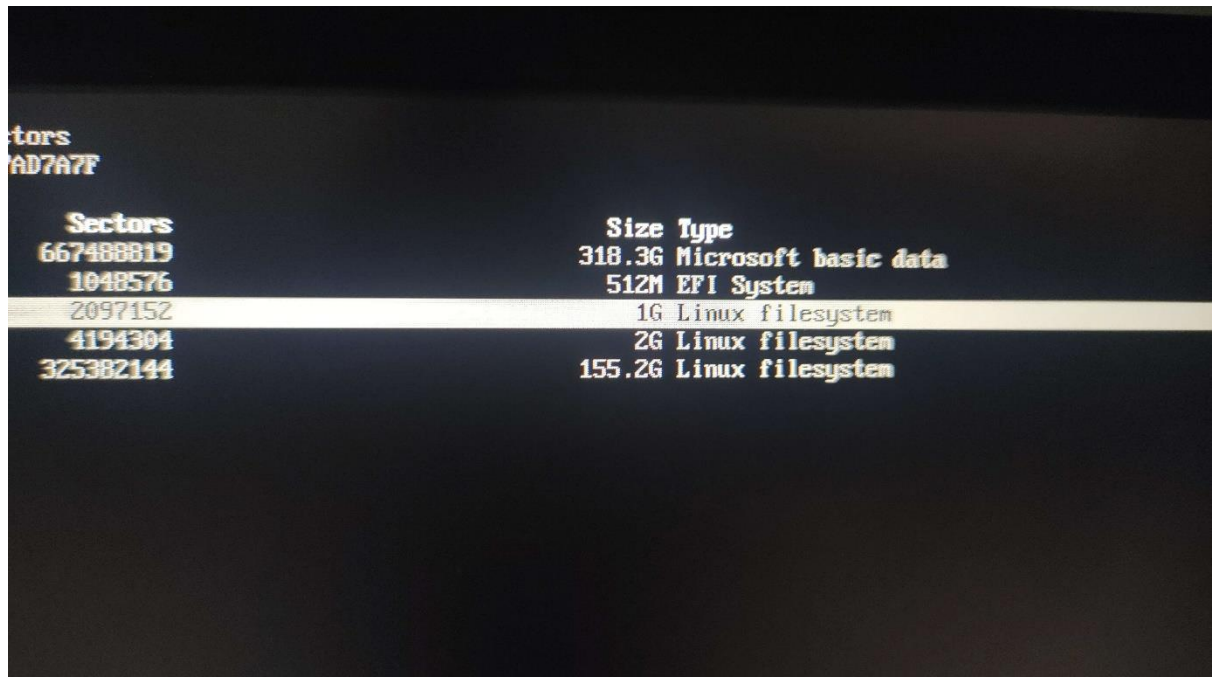
[fwl]# station wlan1 connect Wiswis
Type the network passphrase for Wiswis psk.
Passphrase: 
Terminate
[fwl]#
1 root@archiso ~ # ping google
ping: google: Name or service not known
2 root@archiso ~ # ping google.com
PING google.com (142.251.175.138) 56(84) bytes of data:
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=1 ttl=107 time=64.2 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=2 ttl=107 time=58.4 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=3 ttl=107 time=65.5 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=4 ttl=107 time=56.6 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=5 ttl=107 time=70.4 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=6 ttl=107 time=47.0 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=7 ttl=107 time=62.4 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=8 ttl=107 time=53.8 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=9 ttl=107 time=51.6 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=10 ttl=107 time=41.6 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=11 ttl=107 time=49.2 ms
64 bytes from sh-in-f138.1e100.net (142.251.175.138): icmp_seq=12 ttl=107 time=36.4 ms
^C
--- google.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11013ms
rtt min/avg/max/ndev = 36.409/56.443/82.446/12.276 ms
root@archiso ~ #

```

Seharusnya akan muncul balasan dari google.com seperti gambar diatas, untuk memberhentikan ping cukup tekan ctrl + C.

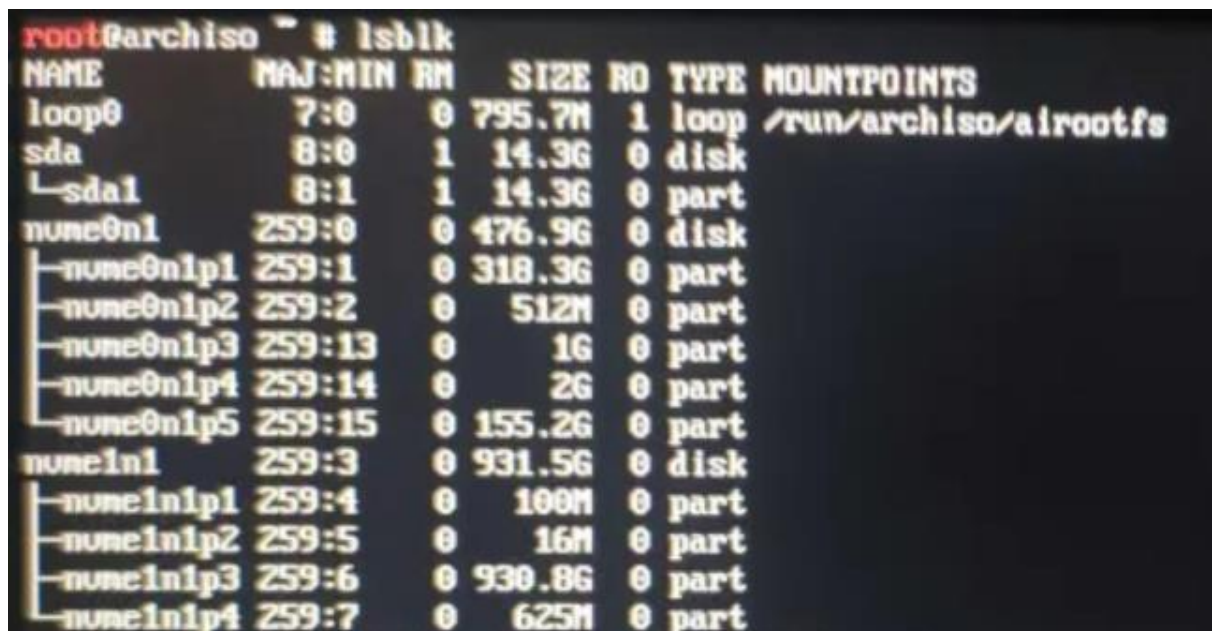
3. Membuat partisi

Langkah berikutnya adalah membuat partisi untuk menyiapkan tempat arch di install. Untuk membuat partisi kalian dapat menggunakan **cfdisk /dev/diskname**. Disitu saya menggunakan 1gb untuk efi, 2gb untuk swap (**untuk swap sebaiknya gunakan $\frac{1}{2}$ dari RAM kalian jika kalian ingin daily drive arch sebagai OS kalian**), dan sisanya untuk root.



Sectors	Size	Type
667488819	318.3G	Microsoft basic data
1048576	512M	EFI System
2097152	1G	Linux filesystem
4194304	2G	Linux filesystem
325382144	155.2G	Linux filesystem

Setelah mempartisi, pastikan tiap partisi telah di write. Untuk mengeceknya kalian dapat menggunakan **lsblk**.



```
root@archiso ~ # lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINTS
loop0	7:0	0	795.7M	1	loop	/run/archiso/alrootfs
sda	8:0	1	14.3G	0	disk	
└─sda1	8:1	1	14.3G	0	part	
nunc0n1	259:0	0	476.9G	0	disk	
├─nunc0n1p1	259:1	0	318.3G	0	part	
├─nunc0n1p2	259:2	0	512M	0	part	
├─nunc0n1p3	259:13	0	1G	0	part	
├─nunc0n1p4	259:14	0	2G	0	part	
└─nunc0n1p5	259:15	0	155.2G	0	part	
nunc1n1	259:3	0	931.5G	0	disk	
├─nunc1n1p1	259:4	0	100M	0	part	
├─nunc1n1p2	259:5	0	16M	0	part	
├─nunc1n1p3	259:6	0	930.8G	0	part	
└─nunc1n1p4	259:7	0	625M	0	part	

Setelah itu kalian harus format tiap partisi. Untuk root gunakan **mkfs.ext4 /dev/root_partition**, untuk swap gunakan **mkswap /dev/swap_partition**, dan untuk efi gunakan **mkfs.fat -F 32 /dev/efi_system_partition**.

```

root@archiso ~ # mkfs.ext4 /dev/nvme0n1p5
mke2fs 1.47.1 (20-May-2024)
Discarding device blocks: done
Creating filesystem with 40672768 4k blocks and 10174464 inodes
Filesystem UUID: 610d5c58-595f-4e76-9cb1-2522ec4ed142
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mkfs.fat -F 32 /dev/nvme0n1p3
mkfs.fat 4.2 (2021-01-31)
root@archiso ~ # mkswap /dev/nvme0n1p4
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=498919ae-276b-488d-b5c9-ea94e5dd6aa5
root@archiso ~ #

```

4. Mount filesystem

Setelah membuat partisi, kalian harus mount tiap filesystem ke /mnt. /mnt adalah lokasi tempat kita melakukan instalasi nanti. Cara untuk mount adalah dengan menggunakan **mount /dev/root_partition /mnt** untuk root, **mkdir -p /mnt/boot/efi** dan **mount /dev/efi_partition /mnt/boot/efi** untuk efi, dan **swapon /dev/swap_partition** untuk swap. Setelah itu kalian dapat menggunakan **lsblk** untuk mengecek tiap filesystem.

```

root@archiso ~ # mount /dev/nvme0n1p5 /mnt
root@archiso ~ # mkdir -p /mnt/boot/efi
root@archiso ~ # mount /dev/nvme0n1p3 /mnt/boot/efi
root@archiso ~ # swapon /dev/nvme0n1p4
root@archiso ~ # lsblk

```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
loop0	7:0	0	795.7M	1	loop	/run/archiso/airootfs
sda	8:0	1	14.3G	0	disk	
└─sda1	8:1	1	14.3G	0	part	
nvme0n1	259:0	0	476.9G	0	disk	
├─nvme0n1p1	259:1	0	318.3G	0	part	
├─nvme0n1p2	259:2	0	512M	0	part	
├─nvme0n1p3	259:13	0	1G	0	part	/mnt/boot/efi
├─nvme0n1p4	259:14	0	2G	0	part	[SWAP]
└─nvme0n1p5	259:15	0	155.2G	0	part	/mnt
nvme1n1	259:3	0	931.5G	0	disk	
├─nvme1n1p1	259:4	0	100M	0	part	
├─nvme1n1p2	259:5	0	16M	0	part	
├─nvme1n1p3	259:6	0	930.8G	0	part	
└─nvme1n1p4	259:7	0	625M	0	part	

```

root@archiso ~ #

```


5. Instalasi

Sekarang kita harus menginstall semua package yang esensial. Mulai dari linux hingga network manager. Untuk melakukannya gunakan **pacstrap /mnt base linux linux-firmware sof-firmware base-devel grub efibootmgr nano networkmanager**. Pacstrap adalah script agar arch dapat menginstall package, /mnt adalah lokasi mount point root, base adalah package agar arch dapat berjalan, linux adalah package berisi linux kernel, linux-firmware adalah firmware file agar kernal dapat bekerja dengan wifi adapter, GPU, etc, sof-firmware untuk onboard audio, base-devel untuk development tools dan library seperti gcc, make, binutils, etc, grub adalah package bootloader untuk dualboot, efibootmgr untuk manage tiap UEFI boot entries, nano untuk sebagai text editor, dan networkmanager untuk mengatur koneksi network.

```
no label, UUID=198919ac-276b-488d-b5c9-ea94e5dd6aa5
root@archiso ~ # mount /dev/nvme0n1p5 /mnt
root@archiso ~ # mkdir -p /mnt/boot/efi
root@archiso ~ # mount /dev/nvme0n1p3 /mnt/boot/efi
root@archiso ~ # swapoff /dev/nvme0n1p1
root@archiso ~ # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0  795.7M  1 loop /run/archiso/airootfs
sda          8:0      1   14.3G  0 disk
├─sda1       8:1      1   14.3G  0 part
└─nvme0n1    259:0    0  476.9G  0 disk
   ├─nvme0n1p1 259:1    0  318.3G  0 part
   ├─nvme0n1p2 259:2    0    512M  0 part
   ├─nvme0n1p3 259:13   0    16G  0 part /mnt/boot/efi
   ├─nvme0n1p4 259:14   0    2G  0 part [SWAP]
   └─nvme0n1p5 259:15   0  155.2G  0 part /mnt
nvme1n1     259:3    0  931.5G  0 disk
   ├─nvme1n1p1 259:4    0   100M  0 part
   ├─nvme1n1p2 259:5    0    16M  0 part
   ├─nvme1n1p3 259:6    0   930.8G  0 part
   └─nvme1n1p4 259:7    0    625M  0 part
root@archiso ~ # pacstrap /mnt base linux linux-firmware sof-firmware base-devel grub efibootmgr nano networkmanager
```

Setelah itu tunggu hingga semuanya telah diinstall.

```
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [fsck]
=> Generating module dependencies
=> Creating zstd-compressed initcpio image: '/boot/initramfs-linux.img'
-> Early uncompressed CPPIO image generation successful
=> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
=> Using default configuration file: '/etc/mkinitcpio.conf'
-> -k /boot/vmlinuz-linux -g /boot/initramfs-linux-fallback.img -S autodetect
=> Starting build: '6.9.10-arch1-1'
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [microcode]
-> Running build hook: [modconf]
-> Running build hook: [kms]
=> WARNING: Possibly missing firmware for module: 'ast'
-> Running build hook: [keyboard]
=> WARNING: Possibly missing firmware for module: 'xhci_pci'
-> Running build hook: [keymap]
-> Running build hook: [consolefont]
=> WARNING: consolefont: no font found in configuration
-> Running build hook: [block]
=> WARNING: Possibly missing firmware for module: 'bfa'
=> WARNING: Possibly missing firmware for module: 'qla2xxx'
=> WARNING: Possibly missing firmware for module: 'qla2xxx'
=> WARNING: Possibly missing firmware for module: 'qla2xxx'
=> WARNING: Possibly missing firmware for module: 'qla2xxx'
=> WARNING: Possibly missing firmware for module: 'qla2xxx'
=> Running build hook: [filesystems]
-> Running build hook: [fsck]
=> Generating module dependencies
=> Creating zstd-compressed initcpio image: '/boot/initramfs-linux-fallback.img'
-> Early uncompressed CPPIO image generation successful
=> initcpio image generation successful
( 0/10) Reloading system bus configuration...
  Skipped: Running in chroot.
( 9/10) Warn about old perl modules
(10/10) Updating the info directory file...
pacstrap -S /mnt base linux linux-firmware sof-firmware base-devel grub nano 20.46s user 14.23s system 51x cpu 1:07.40 total
root@archiso ~ #
```

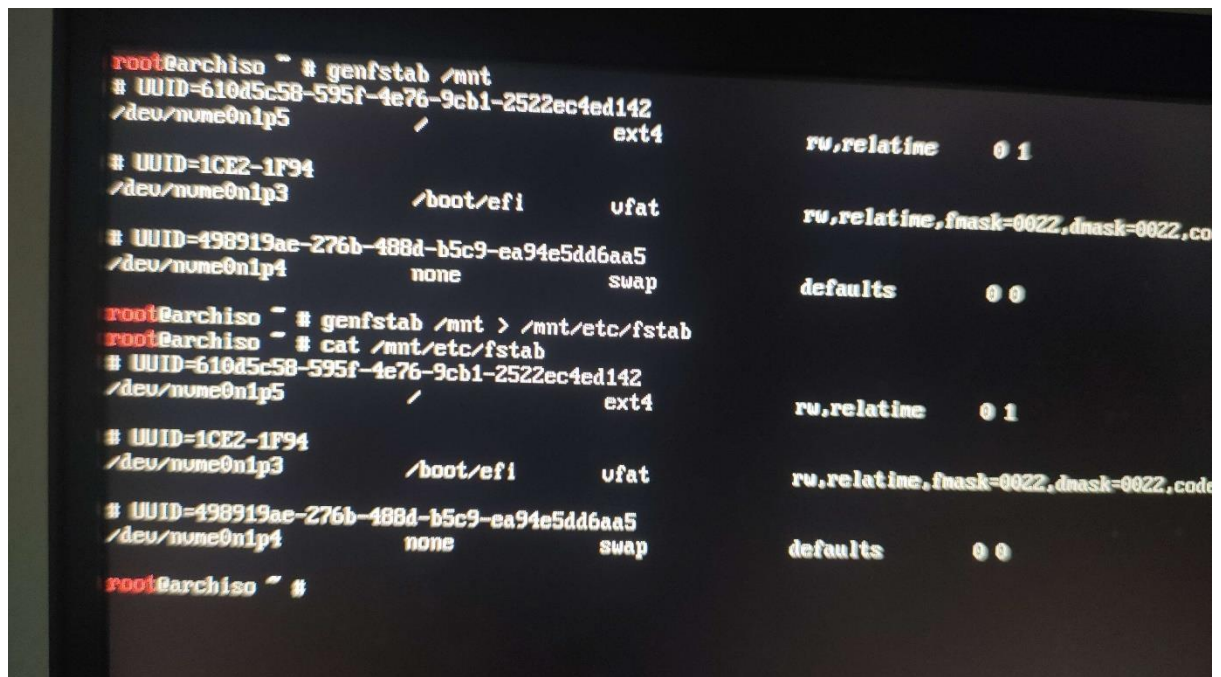
6. Konfigurasi sistem

Setelah menginstall semua package yang diperlukan, langkah berikutnya adalah untuk mengkonfigurasi sistem.

6.1. Fstab

Fstab adalah file system table yang berisi informasi tentang semua disk dan partition yang harus di mount ketika sistem dinyalakan.

Untuk mengkonfigurasi fstab, gunakan **genfstab /mnt** > **/mnt/etc/fstab** untuk mengcopy ke **/mnt/etc/fstab**, untuk mengeceknya kalian dapat menggunakan **cat /mnt/etc/fstab**.



```
root@archiso ~ # genfstab /mnt
# UUID=610d5c58-595f-4e76-9cb1-2522ec4ed142
/dev/nvme0n1p5          /          ext4          rw,relatime          0 1

# UUID=1CE2-1F94
/dev/nvme0n1p3          /boot/efi    ufat          rw,relatime,fnask=002Z,dmask=002Z,code=002Z 0 0

# UUID=498919ae-276b-488d-b5c9-ea94e5dd6aa5
/dev/nvme0n1p4          none         swap          defaults            0 0

root@archiso ~ # genfstab /mnt > /mnt/etc/fstab
root@archiso ~ # cat /mnt/etc/fstab
# UUID=610d5c58-595f-4e76-9cb1-2522ec4ed142
/dev/nvme0n1p5          /          ext4          rw,relatime          0 1

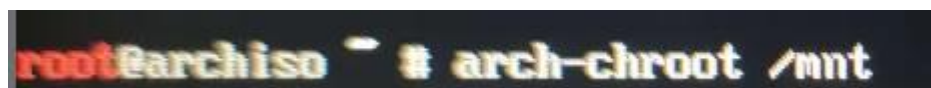
# UUID=1CE2-1F94
/dev/nvme0n1p3          /boot/efi    ufat          rw,relatime,fnask=002Z,dmask=002Z,code=002Z 0 0

# UUID=498919ae-276b-488d-b5c9-ea94e5dd6aa5
/dev/nvme0n1p4          none         swap          defaults            0 0

root@archiso ~ #
```

6.2. Chroot

Sisa konfigurasi harus dilakukan di root arch, gunakan **arch-chroot /mnt** untuk pindah ke root di /mnt.



```
root@archiso ~ # arch-chroot /mnt
```

6.3. Time

Berikutnya adalah konfigurasi waktu. Untuk mengkonfigurasi waktu gunakan **ln -sf /usr/share/zoneinfo/Region/City /etc/localtime** untuk mengubah menjadi local timezone dan **hwclock --systohc** untuk sinkronisasi clock dengan waktu sekarang.

```
[root@archiso /]# ln -sf /usr/share/zoneinfo/Asia/Jakarta /etc/localtime
[root@archiso /]# date
Mon Jul 22 18:53:51 WIB 2024
```

6.4. Localization (kayanya ga wajib tapi lakuin aja :moyai:)

Konfigurasi localization agar dapat handle berbagai bahasa, character encoding, dan regional settings.

Mulai dengan gunakan **nano /etc/locale.gen** dan uncomment dengan menghapus # di en_US.UTF-8 UTF-8. Lalu ctrl + O dan enter untuk write file dan ctrl + x untuk exit dari nano. Berikutnya **locale-gen** untuk generate locales. Setelah itu edit locale.conf dengan **nano /etc/locale.conf** dan tambahkan "LANG=en_US.UTF-8" dan edit vconsole.conf untuk mengatur keyboard layout dengan **nano /etc/vconsole.conf** dan tambahkan "KEYMAP=us" (ga ke ss semua :sob:).

6.5. Konfigurasi network

Tambahkan hostname dengan **sudo /etc/hostname**, file tersebut bisa diisi terserah kalian, punyaku hanya diisi dengan "Arch", temanku ada yang isi "Archie", ada juga yang isi "Arcchh"

Nyalakan network manager dengan **systemctl enable NetworkManager**.

```
[root@archiso /]# systemctl enable NetworkManager
Created symlink '/etc/systemd/system/multi-user.target.wants/NetworkManager.service' → '/usr/lib/systemd/system/NetworkManager.service'
Created symlink '/etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service' → '/usr/lib/systemd/system/org.freedesktop.nm-dispatcher.service'
Created symlink '/etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service' → '/usr/lib/systemd/system/NetworkManager-wait-online.service'
[root@archiso /]# _
```

6.6. Root password

Konfigurasi root password dengan **passwd** lalu masukkan password dan di retype. (GA KEFOTO JUGAAA :sob::sob::moyai:)

6.7. Konfigurasi user

Tambahkan user dengan **useradd -m -G wheel -s /bin/bash username**. -m digunakan untuk menambah user directory di /home/username, -G wheel digunakan untuk menambahkan user ke group wheel agar dapat menggunakan sudo access nantinya, -s /bin/bash digunakan sebagai default shell untuk user, dan username adalah username kalian nantinya.

Lalu tambahkan password ke user dengan **passwd username**. Sama seperti root password, harus di retype.

Untuk menggunakan sudo access, gunakan **EDITOR=nano visudo** untuk mengedit file sudoers. Dibagian paling bawah uncomment %wheel ALL=(ALL:ALL) ALL agar user di group wheel dapat menggunakan access sudo.


```

# Defaults!PRGMAN !intercept, !log_subcmds
##
## Runas alias specification
##
##
## User privilege specification
##
root ALL=(ALL:ALL) ALL

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL:ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# %sudo ALL=(ALL:ALL) ALL

## Uncomment to allow any user to run sudo if they know the password
## of the user they are running the command as (root by default).
# Defaults targetpw # Ask for the password of the target user
# ALL ALL=(ALL:ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d

```

Untuk tes kalian dapat **su username** untuk mengubah user dan **sudo pacman -Syu** untuk mengetes sudo user.

6.8. Konfigurasi bootloader

Bootloader akan diperlukan agar sistem dapat dinyalakan, gunakan **grub-install /dev/diskname** dan **grub-mkconfig -o /boot/grub/grub.cfg** untuk memasukkan grub ke boot.

```

[root@archiso /]# grub-install /dev/nvme0n1
Installing for x86_64-efi platform.
Installation finished. No error reported.
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@archiso /]# _

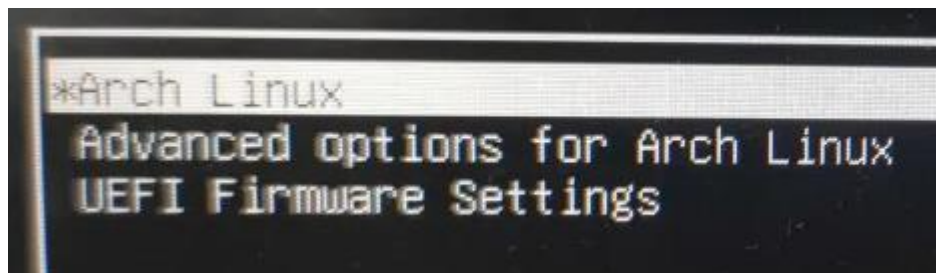
```

7. Reboot

Untuk reboot dan mengetes arch, gunakan **exit** untuk kembali ke archiso, **umount -a** untuk meng-unmount semua filesystem yang telah di mount, dan **reboot** untuk mereboot sistem.

```
root@archiso ~ # umount -a
umount: /run/user/0: target is busy.
umount: /etc/pacman.d/gnupg: target is busy.
umount: /sys/fs/cgroup: target is busy.
umount: /run/archiso/copytoran: target is busy.
umount: /run: target is busy.
umount: /dev: target is busy.
32 root@archiso ~ # reboot_
```

Jika window berikut muncul setelah kalian reboot, tandanya arch telah diinstall.



Yay udah selesai.



Art by RavioliMavioli

Extended guide

Dual Boot

Untuk melakukan dual boot, pertama install os-prober dengan **sudo pacman -Syu os-prober**, untuk mengecek apabila windows terdeteksi gunakan **os-prober**, jika muncul sesuatu edit file grub dengan **sudo nano /etc/default/grub** dan uncomment **"GRUB_DISABLE_OS_PROBER=true"** dengan cara menghapus # di depannya. Setelah itu tinggal reinstall grub dengan **sudo grub-mkconfig -o /boot/grub/grub.cfg** dan saat kalian restart seharusnya windows muncul. Jika windows EFI tidak terdeteksi, kemungkinan karena kalian install windows dan arch di drive yang sama, solusinya dengan mount partisi windows yang memiliki EFI tersebut dan baru di reinstall dengan grub-mkconfig tersebut. Cara mountnya dengan pertama install ntfs-3g dengan **sudo pacman -Sy ntfs-3g** dan **sudo mkdir /mnt/windows** dan **sudo mount -t ntfs-3g /dev/windows_efi_partition /mnt/windows**.

Bonus

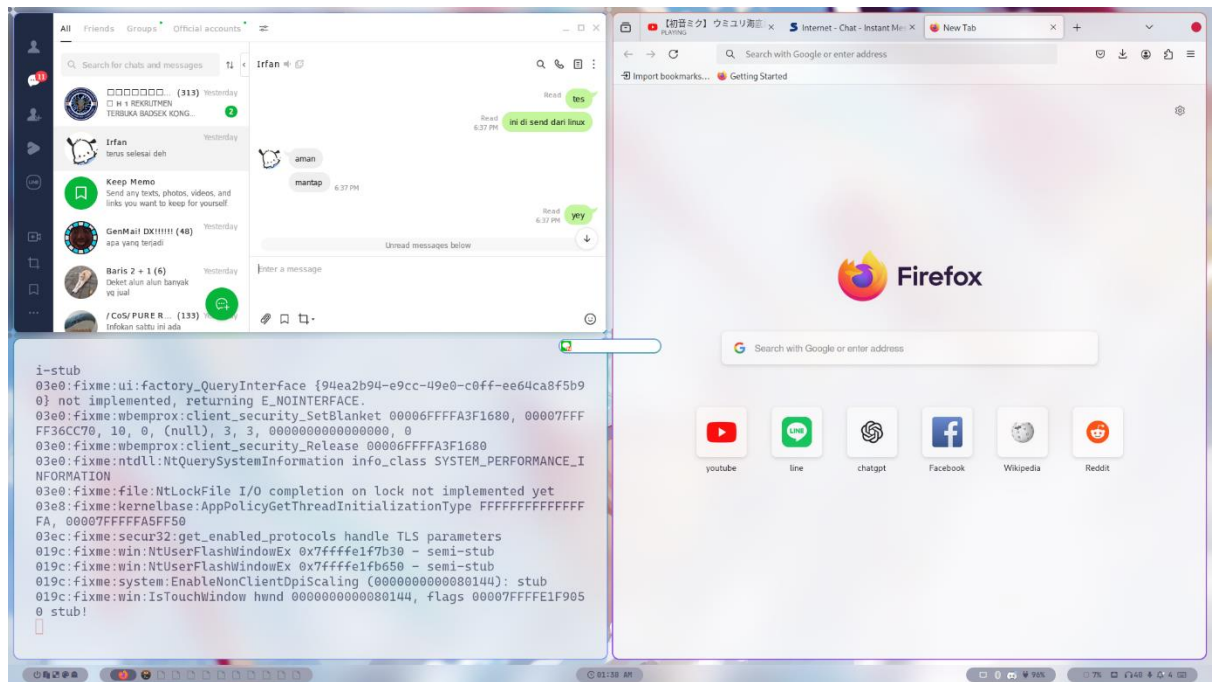
Berikut adalah langkah tambahan yang saya lakukan untuk nilai bonus.

Untuk bonus DHCP sudah masuk saat menginstall network manager di awal.

Saya langsung terjun menginstal wayland sebagai GUI saya. Wayland yang saya gunakan adalah <https://github.com/prasanthangan/hyprdots>. Disitu tinggal ikuti saja instalasi mulai dari install git, clone reponya, dan menjalankan ./install.sh. Jika bingung pilih saja 1 sebagai default, jika ada opsi "installed [i]" pilih saja opsi itu. Di repo tersebut, sudah ada bonus:

1. Mengganti shell default menjadi zsh/fish, saya menggunakan fish
2. Memasang graphical text editor dan web browser, diberikan VS Code dan Mozilla Firefox
3. Menggunakan tiling window manager, wayland sendiri adalah tiling window manager
4. Menggunakan wayland, ya udah pake wayland





Bonus menginstall custom grub theme juga cukup mudah, saya menggunakan tema firefly dari honkai star rail (firefly my beloved <3) <https://github.com/voidlhf/StarRailGrubThemes>. Ikuti saja langkah yang diberikan di github tersebut mulai dari download dan unzip theme nya, lalu meng-edit grub dengan **sudo nano /etc/default/grub** dan menambahkan theme dibawah dengan “**GRUB_THEME=/usr/share/grub/themes/Guinaifen/theme.txt**”, dan melakukan **sudo grub-mkconfig -o /boot/grub/grub.cfg**. Setelah kalian restart seharusnya grub telah berubah.

