**OPPDRAG 1 – Dokumentasjon**

**OPPGAVE 1**

**1. Koble til nettverket**

Jeg koblet meg først til 2IMD-ruteren.

 **Hva:** Jeg koblet Raspberry Pi-en til et trådløst nettverk.

 **Hvordan:** Jeg valgte 2IMD-ruteren i nettverksinnstillingene.

 **Hvorfor:** Pi-en trenger internettforbindelse for å kunne kommunisere med PC-er og laste ned oppdateringer.

**2. Statisk IP-adresse** Jeg satte en statisk IP-adresse på Pi-en ved å kjøre:

sudo ip addr add 192.168.13.16/16 dev wlan0

**Hva:** Jeg satte en fast IP-adresse på Pi-en.

**Hvordan:** Jeg kjørte kommandoen:

sudo ip addr add 192.168.13.16/16 dev wlan0

**Hvorfor:** En statisk IP gjør det enklere å alltid vite hvilken adresse Pi-en har, noe som er viktig når andre enheter skal kommunisere med den.

**3. Se adresser**

Ved å kjøre ip a kunne jeg se de forskjellige adressene, men senere lærte at « hostname -I» Bare viser ip addressen.

**Hva:** Jeg ville kontrollere hvilke adresser Pi-en hadde.

**Hvordan:** Først med ip a, senere med hostname -I.

**Hvorfor:** hostname -I gir en rask oversikt over Pi-ens aktive IP-adresse uten ekstra informasjon.

**4. Feilmeldinger**

Jeg fikk en feilmelding om at jeg ikke fikk kontakt. Dette var fordi PC-ene mine ikke var på samme nettverk eller ikke hadde riktig DNS/GATEWAY.

**Hva:** Jeg fikk feilmelding om manglende kontakt med nettverket.

**Hvordan:** Meldingen dukket opp da jeg prøvde å pinge mellom PC-er.

**Hvorfor:** Problemet oppsto fordi PC-ene ikke var på samme subnett, eller hadde feil DNS/Gateway.

**5. Flere IP-adresser**

Under konfigurasjonen klarte jeg å sette 6 IP-adresser på Pi-en. Jeg måtte derfor søke hvordan man sletter en adresse. Jeg fant i en cheatsheet at kommandoen sudo ip addr del

(ip-adresse) dev eth0/wlan0 sletter den.

**Hva:** Jeg la merke til at Pi-en hadde flere IP-adresser under konfigurasjonen.

**Hvordan:** For å fjerne unødvendige adresser brukte jeg:

sudo ip addr del <ip-adresse> dev eth0/wlan0

**Hvorfor:** Dette rydder opp i nettverkskonfigurasjonen og unngår konflikter.

**6. Til slutt** Til slutt klarte jeg å pinge mellom PC-ene.

**Hva:** Jeg testet om PC-ene kunne kommunisere med Pi-en.

**Hvordan:** Ved å kjøre ping mellom enhetene.

**Hvorfor:** Pinging bekrefter at nettverket fungerer som det skal.

**Bilder**

Her kan du sette inn skjermbilder av:

* **Kommandoen for å sette IP-adresse**
* **Utdata fra ip a**

A computer screen with white text

AI-generated content may be incorrect.

* 
* **Ping mellom PC-ene**

A computer screen shot of a number

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

**Oppsummering/det jeg lærte**

Wlan0/Eth0 står får wireless og ethernet nettverk.

Sette statisk ip-addresse på pi-en gjennom terminal og manuelt.

**OPPGAVE 2**

**OPPGAVE 2**

**1. Oppdater systemet**  
Jeg oppdaterte programvaren og sikkerheten på Pi-en ved å kjøre:

sudo apt update

**Kommando for å oppdatere pakkelisten med updates**

sudo apt upgrade -y

**Kommando for å laste ned updates som blir funnet av «sudo apt update»**

Dette sørget for at systemet hadde de nyeste pakkene.

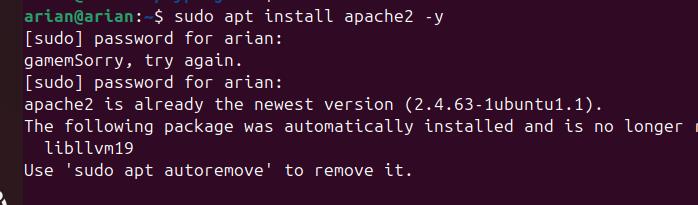
**Hva:** Jeg sørget for at Pi-en hadde siste versjon av alle pakker og sikkerhetsoppdateringer.

**Hvordan:**

**Hvorfor:** Oppdateringer gir sikkerhet og stabilitet.

**2. Installer en webserver**  
Jeg valgte å installere en enkel webserver med navn «Apache» ved å bruke komandoen.

sudo apt install apache2 -y



**Hva:** Jeg installerte en enkel webserver, Apache.

**Hvordan:** sudo apt install apache2 -y

**Hvorfor:** Apache gjør det mulig å vise nettsider fra Pi-en til andre enheter på nettverket.

**3. Finn IP-adressen til Pi-en**  
Jeg fant IP-adressen ved å kjøre:

hostname -I



Deretter skrev jeg denne adressen inn i en nettleser på en enhet på samme nettverk og fikk opp standard “It works”-siden fra Apache. Jeg fant ut at Apache starter som en tjeneste med en gang du laster den ned. Som betyr at webserveren starter med en gang.

**Hva:** Jeg trengte å vite Pi-ens adresse for å åpne nettsiden.

**Hvordan:** hostname -I

**Hvorfor:** Nettleseren trenger IP-en for å koble til webserveren.

**4. Dokumentrot (hvor nettsiden ligger)**

Her begynner jeg å ekperimentere fordi oppgaven ba om en webserver, så jeg tenkte «kan jeg legge inn html?»

Ja, jeg gjorde litt søk og fant ut hvordan veldig raskt.

Apache blir lagret i et standard dokument som er:

/var/www/html

Ved å bruke «nano» komandoen kunne jeg legge mine egne HTML-filer inn.

**Hva:** Jeg fant hvor nettsidene faktisk lagres på Pi-en.

**Hvordan:** Apache bruker som standard mappen /var/www/html.

**Hvorfor:** Alle filer her blir servert til nettleseren når man åpner Pi-ens IP.

**5. Gi brukeren min tilgang til webmappen**  
Som standard eies denne mappen av “root”. For å kunne redigere uten sudo kjørte jeg:

sudo chown -R pi:pi /var/www/html

Men I mitt tilfelde måtte jeg bruke brukernavnet mitt på pi-en.



Sudo er som en admin fant jeg ut senere.

og å kjøre «sudo» foran en komando så sier du til systemet «Kjør denne kommandoen som om jeg var admin eller superbruker.

(samme prinsipp som i Oppgave 1: sudo = kjør som administrator, chown = endre eier, pi:pi = bruker og gruppe “pi”, -R = rekursivt).

**Hva:** Jeg endret eier av webmappen slik at jeg kunne redigere uten sudo.

**Hvordan:** sudo chown -R pi:pi /var/www/html

**6. Lag din egen nettside**  
Jeg redigerte/lage index.html ved hjelp av chatgpt fordi jeg ikke har tid til å redigere html nå.

nano /var/www/html/index.html

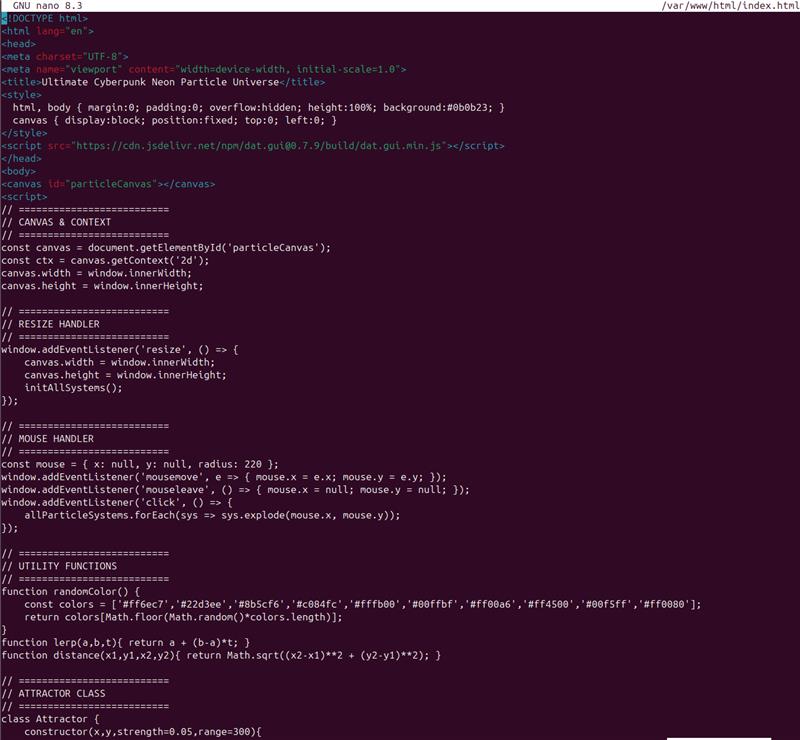


**Hva:** Jeg laget en enkel HTML-side.

Hvordan: nano /var/www/html/index.html

Limte inn HTML, lagret (Ctrl+O), lukket (Ctrl+X).

**Hvorfor:** Dette viser hvordan man kan publisere innhold på sin egen webserver.



Og med det var nettsiden mit oppdatert og ser nå sånn ut:



**9. Oppsummering av viktige begreper**

* Webserver = program som sender filer til nettleseren
* Dokumentrot = mappen der nettsiden ligger
* sudo = kjør som administrator
* chown = endre hvem som eier filer/mapper
* IP-adresse = adressen du bruker i nettleseren for å nå Pi-en

**Oppgave 3**

1. Publisering av Python-mappen til GitHub

* Først måtte Python-mappen publiseres til GitHub.
* Kommandoene som ble brukt:

**A computer screen with white text

AI-generated content may be incorrect.**

* Dette ble gjort med hjelp fra Ludvik. **SELV om jeg kunne klare det helt selv.**

**Hva:** Jeg lastet opp Python-prosjektet til GitHub.

**Hvordan:** Brukte Git-kommandoer (init, add, commit, push) med hjelp fra Ludvik.

**Hvorfor:** Dette gjør prosjektet tilgjengelig på internett og mulig å hente ned på Pi-en

1.1 Problemer underveis

* Feilmeldinger oppsto fordi man forsøkte å pushe gjennom main-branchen.
* Løsning: Pushet gjennom master-branchen i stedet, og da fungerte det.

**Hva: Feil oppsto ved push til main-branchen.**

**Hvordan: Pushet i stedet til master-branchen.**

**Hvorfor: GitHub repo hadde en annen default branch enn forventet.**

**2. Fjerne en fil fra repo**

* En fil måtte slettes fra repository fordi den ikke skulle være der.
* Kommandoene som ble brukt:

**A black screen with white text

AI-generated content may be incorrect.**

**Hva: En fil måtte fjernes fra repo.**

**Hvordan: Brukte git rm og pushet endringen.**

**Hvorfor: Filen skulle ikke være synlig i repo.**

**3. Kloning på Raspberry Pi**

* Opprettet en ny mappe for prosjektene:

**A black background with white text

AI-generated content may be incorrect.**

**A black background with white text

AI-generated content may be incorrect.**

Problemet: Kun en README-fil var synlig, Python-filene manglet.

**Hva:** Jeg ville hente Python-prosjektet til Pi-en.

**Hvordan:** Opprettet ny mappe, kjørte git clone <repo-url>.

**Hvorfor:** For å kunne kjøre og redigere Python-filene lokalt på Pi-en.

**3.1 Løsning på manglende filer**

* Problemet var at default branch på GitHub var main, mens Python-filene var pushet til master.
* Løsning: Byttet branch på Raspberry Pi:
* A black background with a white object in the middle

  AI-generated content may be incorrect.

Etter det var Python-mappene tilgjengelige.

**Hva:** Bare README-filen var synlig.

**Hvordan:** Byttet branch til master på Pi-en.

**Hvorfor:** Python-filene lå på master, ikke main.

**4. Kjøre og endre Python-filene**

* Navigerte inn i mappene og kjørte Python-filene.
* Noen koder måtte endres. Redigering ble gjort med nano «kommandoen».
* For å sikre at alle oppdateringer fra repo var med:
* A black rectangular object with white text

  AI-generated content may be incorrect.

**Hva:** Jeg testet og redigerte Python-skriptene.

**Hvordan:** Navigerte til mappen og brukte nano <filnavn>.py for å redigere.

**Hvorfor:** For å sikre at alle skripter fungerte og kunne tilpasses.

Scripter som blir kjørt på raspberry 