



UiT Norges arktiske universitet

Fakultet for humaniora, samfunnsvitenskap og lærerutdanning

## **Å bruke variabler i programmering i matematikk**

En observasjons og intervjustudie om variabler i tekstbasert og visuell programmering

Dag-Martin Berg

LER-3913, Vår 2023





## Forord

Fem år med lærerutdanning har gått fort, og masteren markerer avslutningen på en hektisk, men lærerik periode i livet mitt. I løpet av disse fem årene har jeg møtt på mange teknologiske løsninger i matematikkfaget og valgte derfor å skrive om programmering.

Mens jeg har jobbet med masteren har jeg vært i fulltidsjobb som lærer i grunnskolen, samt at masterskrivingen er en fulltidsjobb i seg selv, så jeg ser fram til en litt roligere hverdag der jeg har litt mer tid til familie og venner. En takk til familie og venner som har støttet meg i min utdanning og utvist stor tålmodighet mens jeg har vært opptatt i både arbeidstid og fritid.

En grunn til at jeg har hatt mulighet til å studere ved siden av jobb er at UIT Alta har tilbudt en fleksibel studiemodell som har lagt opp til dette. Jeg vil takke UIT Alta for å ha gitt meg denne muligheten. Jeg vil også rette en takk til min arbeidsgiver som har lagt til rette for at jeg skulle ha mulighet til å gjennomføre utdanningen. Min arbeidsgiver har gitt meg permisjon når det er nødvendig og mine kolleger har stilt opp som vikarer mens jeg har studert.

Ikke bare har disse årene gitt meg mye læring, men de har og gitt meg mulighet til å møte mange mennesker. Jeg har hatt mange fine stunder med gode diskusjoner med medstudenter og forelesere. Så selv om det er fint å være ferdig med studiene, så vil jeg savne stundene med alle som har vært en del av min studiehverdag. Takk til mine medstudenter og lærere ved lærerstudiet ved UIT Alta. Det har vært lærerikt å samarbeide med dere.

Jeg vil takke min veileder Steinar Thorvaldsen for samtaler, hjelp og tilbakemeldinger i min forskningsperiode og masterskrivingen. Takk for hjelpen

*Mai, 2023*

*Dag-Martin Berg*



## Sammendrag

### Tittel

«Å bruke variabler i programmering i matematikk»

### Bakgrunn og formål

Bakgrunnen for masteren er innføring av programmering i matematikkfaget ved innføringen av LK20. Det som er mest brukt i programmering er variabler (Karlsen, A. i Sund, 2023).

Derfor ville jeg se nærmere på bruk av variabler i programmering i matematikkfaget.

Formålet er å se om det er forskjell på å jobbe med variabler i visuellprogrammering i form av blokkprogrammering og tekstbasert programmering i for av programmeringsspråket Python. Jeg ønsker å se om bruk av variabler i programmering gir en større forståelse for variabler i algebra. I tillegg ønsker jeg å se litt nærmere på fordeler og ulemper ved bruk av teknologi i undervisningsøyemed.

### Forskningsspørsmål

- 1) Er det forskjell på å jobbe med variabler i visuell programmering og tekstbasert programmering?
- 2) Trenger elevene mer støtte fra lærer når de arbeider med den ene metoden for programmering, tekstbasert eller visuell programmering?
- 3) Er det forskjell på å lære variabler i visuell programmering og tekstbasert programmering?

### Utvalg

Seks elever fra niende trinn og seks elever fra tiende trinn ved en ungdomsskole i Troms og Finnmark, tolv elever totalt.

## Metode

Kvalitativ metode i form av observasjon og intervju i form av semistrukturert intervju med en intervjuguide.

## Funn

Elevene som programmerte med tekstbasert programmering opplevde bruk av variabler som enklere enn elever som programmerte med variabler. En funksjoner som skilte seg, var hvordan man legger inn kommentarer i programmet for å forklare stegene. I Python kan man enkelt legge inn en kommentar ved å bruke tegnet # (firkant/hashtag/sharp), etterfulgt av kommentaren de ønsket å sette inn. I visuell programmering er det ikke en innebygd kommentarfunksjon, da satte elevene inn en tekstblokk som ikke var festet til programmet for å skrive kommentarer.

Elever som programmerte med tekstbasert programmering opplevde feil i programmet de lagde og måtte drive feilsøking, noe elevene som programmerte med visuell programmering ikke møtte på. Elevene opplevde behov for en viss form for lærerstøtte uavhengig av programmeringsmetode, men kun de som programmerte med tekst trengte støtte til feilsøking i starten av prosjektet. Både de som programmerte visuelt og tekstbasert opplevde at de fikk en større forståelse for variabler i algebra etter å ha brukt variabler i programmering.

## Konklusjon

Det er noen forskjeller i hvordan man lager og bruker variabler fra visuell programmering og tekstbasert programmering. Elever som programmerte med tekst opplevde det å lage og bruke variabler som tungvint, mens elevene som programmerte med tekst opplevde det å lage og bruke variabler som effektivt. Kommentarfunksjonen som er i Python finnes ikke i den visuelle programmeringen vi brukte, men elevene løste dette med en løs tekstblokk. Både visuell programmering og tekstbasert programmering ga en opplevelse av større forståelse for variabler i algebra. Programmering med variabler kan hjelpe elevene å få en større forståelse for variabler i algebra, men kan skape og forsterke misoppfatninger om dette ikke implementeres på en god måte.

## Innhold

Forord .....	2
Sammendrag.....	4
1. Innledning.....	10
1.1 Bakgrunn for masteren .....	11
1.2 Formål og forskningsspørsmål .....	12
1.3 Tittel og oppbygging av oppgaven.....	13
2. Teori .....	14
2.1 Digital didaktikk.....	14
2.1.1 Digital etikk.....	14
2.2 Programmering i skolen .....	15
2.2.1 Variabler og algebra .....	16
2.2.2 Variabler i programmering og Python .....	17
2.2.3 Tidligere studier om programmering og algebra.....	18
2.3 Oppgaver, motivasjon og samarbeid.....	19
2.3.1 Oppgaver .....	19
2.3.2 Kort om motivasjon og samarbeid .....	19
2.3.2 Skolens og lærerens plass og rolle .....	20
3 Metode.....	22
3.1 Vitenskapsteoretisk paradigme .....	22
3.2 Kvalitativ tilnærming .....	22
3.3 Datainnsamling.....	23
3.3.1 Observasjon.....	23
3.3.2 Intervju .....	24
3.3.3 Teknologiske hjelpemidler i studien .....	26
3.4 Analyse.....	27
3.4.1 Fenomenologisk analyse .....	27

3.4.2 Tematisk analyse .....	28
3.4.3 Bearbeiding i analysen .....	29
3.5 Utvalg .....	30
3.6 Gjennomføringen .....	30
3.6.1 Elevenes forkunnskap .....	31
3.6.2 Prosjektets oppbygging .....	32
3.6.3 Gjennomføring i timen .....	32
3.7 Validitet, relabilitet og overførbarhet.....	33
3.7.1 Validitet.....	33
3.7.2 Relabilitet .....	34
3.7.1 Overførbarhet .....	34
3.8 Et kritisk blikk på studien.....	35
3.9 Etske betraktninger.....	37
4. Funn.....	39
4.1 Observasjoner.....	39
4.1.1 Fra kalkulator til variabler.....	39
4.1.2 Bruk av kommentarer i programmet .....	42
4.1.3 Feilsøking .....	43
4.1.4 Tidsbruk, en forskjell på Python og blokk .....	45
4.1.5 Samarbeid i og på tvers av læringspar .....	45
4.1.6 Motivasjon.....	45
4.2 Intervjuer .....	46
4.2.1 Generelle spørsmål om programmering .....	46
4.2.2 Programmering og variabler.....	47
4.2.3 Motivasjon og samarbeid i programmering .....	47
4.2.4 Knytte programmering til annen matematikk .....	48
4.2.5 Tabell.....	48



4.3 Andre funn.....	49
4.3.1 Egen kompetanse.....	50
4.3.2 Digital etiske utfordringer kan påvirke funn .....	50
5 Drøfting .....	52
5.1.1 Forskjell på å bruke variabler i visuell og tekstbasert programmering .....	52
5.1.2 Lærerstøtte i visuell og tekstbasert programmering .....	54
5.1.3 .....	60
5.2 Andre betraktninger.....	62
5.2.1 Utfordringer som underviser .....	62
5.2.2 Etske utfordringer med digitale plattformer.....	65
5.3 Tidligere forskning og veien videre .....	66
6 Konklusjon .....	70
7 Referanseliste .....	73
8 Vedlegg .....	76
Vedlegg 1 Intervjuguide.....	76
Vedlegg 2 Samtykkeskjema .....	77
Vedlegg 3 Godkjenning fra NSD .....	81

## Tabbeloversikt

Tabell 1: Forenklet tabell om nordisk tilnærming til algoritmisk tankegang.....	14
Tabell 2: Intervjufunn.....	48

## Figuroversikt

Bilde 1: Bruk av Python som kalkulator.....	38
Bilde 2: Bruk av variabler i Python.....	39
Bilde 3: Del 1 av blokk som kalkulator.....	39
Bilde 4: Del 2 av blokk som kalkulator.....	39
Bilde 5: Steg 1 av å lage variabel i blokk.....	40
Bilde 6: Steg 2 av å lage variabler i blokk..	40
Bilde 7: Bruk av variabler i blokk.....	41
Bilde 8: Kommentar i Python.....	41
Bilde 9: Kommentar i blokk.....	42
Bilde 10: Syntaksfeil i Python.....	43
Bilde 11: Desimaltallfeil i Python.....	43

## 1. Innledning

Samfunnet utvikler seg raskt og ny teknologi kommer på banen. Dette utfordrer oss på alle områder, skolen inkludert, som må henge med i utviklingen. Skolens mål er å gi elevene den nødvendige kompetansen som framtiden krever (Ødegård & Nøvik, 2019). Et av de nye områdene som satses på i skoleverket er programmering.

I løpet av min studieperiode så kom programmering inn i matematikkfaget ved innføringen av LK20 og jeg har måttet lære meg denne ferdigheten. Derfor har jeg valgt å forske på programmering for å gjøre meg bedre i stand til å undervise i dette temaet. Det er allerede skrevet flere masteroppgaver om programmering i matematikk etter at dette ble en del av matematikkundervisningen, noe som kan tyde på at det er en interesse blant lærerstudenter for programmering i matematikkfaget. Det er positivt at nyutdannede lærere har interesse for temaet og bringer kompetansen de har fra studien med ut i skoleverket. Jeg har sett på flere tidligere mastere for å se hvilket tema som allerede er dekt, og jeg har etter denne undersøkelsen havnet på å skrive om programmering og variabler og om det er noe forskjell på å jobbe med det i Python eller i blokkprogrammering. Samtykkeskjema ble sendt ut til foreldre før jeg endret tittel på oppgaven, men de andre faktorene var lik.

Ved innføringen av LK20 kom programmering inn i matematikkfaget. I kjerneelementer i læreplanen for matematikk 1-10 under utforskning og problemløsning nevnes algoritmisk tankegang: «Algoritmisk tenking er viktig i prosessen med å utvikle strategier og framgangsmåtar for å løse problem og inneber å bryte ned eit problem i delproblem som kan løysast systematisk. Vidare inneber det å vurdere om delproblema best kan løysast med eller utan digitale verktøy.» (Kunnskapsdepartementet, 2019a). Algoritmisk tankegang er et sentralt tema innenfor programmering. Fokuset i oppgaven er bruk av variabler i visuell og tekstbasert programmering. Studien ble gjennomført ved hjelp av observasjon og semi-strukturert intervju med en intervjuguide.

## 1.1 Bakgrunn for masteren

Valget av tema for masteren baserer seg på min interesse for bruk av teknologi i undervisning av matematikk. Programmering er en ny ferdighet for meg og jeg ønsker å øke min kompetanse innenfor programmering og hvordan undervise programmering i matematikkfaget. I min utdanning har det vært lite programmering så dette er et område som jeg ønsker å utvikle videre. For å utvikle den nødvendige kompetansen så må jeg først øke min egen kompetanse om programmering i seg selv, men også øke min undervisningskompetanse innenfor programmering. Gjennom min forskning får jeg utfordret og utviklet begge former for kompetanse.

Programmering er innført i matematikkfaget ved innføringen av LK20. Det er i hovedsak to metoder for programmering elevene møter på når de skal løse matematiske oppgaver, visuell programmering og tekstbasert programmering. Jeg ønsket i utgangspunktet å se på hvilke av disse typer programmering som ga lavest inngangsterkel til programmering, men valgte å spisse det inn på hvordan bruk av variabler oppleves innenfor de to metodene for programmering. En av grunnene til at jeg ikke valgte å se på hvilke typer programmering som gir lavest inngangsterkel er at mange elever begynner med visuell programmering når de først skal lære programmering, men går over til tekstbasert programmering etter hvert. Derfor er det første møtet med programmering visuell og deretter tekstbasert. En annen grunn til at jeg spisset det inn var at mange aspekter ved programmering er det allerede skrevet masteroppgaver om.

Variabler er det mest brukte funksjonen i programmering (Karlsen, A. i Sund, 2023). Derfor ønsker jeg å se nærmere hvordan bruk av variabler innenfor visuell programmering og tekstbasert programmering fungerer i undervisningen, samt se om dette styrker elevenes forståelse for variabler innenfor annen matematikk, da særlig algebra.

Programmering legger opp til algoritmisk tankegang og bruk av variabler, og i min oppgave vil jeg ha fokus på bruken av variabler. Det å kunne bruke variabler er svært nyttig algoritmisk tankegang innenfor programmering. Videre skal jeg se om eleven får en større forståelse av variabler i algebra ved at de jobber med variabler i programmering.

## 1.2 Formål og forskningsspørsmål

Programmering er blitt en del av matematikkundervisningen ved innføringen av LK20

Formålet med studien er å finne ut hvordan elever opplever bruke av variabler i programmering, og da se om det er noen forskjell på bruk av variabler i visuell programmering og tekstbasert programmering. Et annet aspekt ved studien er å se om elevene klarer å knytte bruk av variabler i programmering til annen matematikk og opplever en læring for variabler i algebra.

Studien ser også på om det er forskjell i behov for lærerstøtte i visuell programmering eller tekstbasert programmering, og hva disse eventuelle forskjellene er. Jeg vil også se på hva som kan gjøre elevene mer selvstendige slik at de blir mer autonome og dermed for mindre behov for lærerstøtte i selve oppgaveløsningen. Behovet for lærerstøtte vil alltid være til stede i form av oppmuntring, oppgaveutforming og hint om elevene står fast, men målet er at elevene i størst mulig grad kan lære av hverandre og mestre innenfor sitt nærmeste utviklingsområde uten at læreren instruerer eller gir dem svaret (Skaalvik & Skaalvik, 2018).

I studien vil jeg også se på fordeler og ulemper ved bruk av ny teknologi i undervisningssammenheng samt utfordringer som lærere møter på i forbindelse med nye teknologiske løsninger. Disse ulempene er knyttet til personvern i møte med ny teknologi og utfordringene er læreres mulighet til å utvikle egen kompetanse i møte med nye teknologiske løsninger.

Forskningsspørsmål jeg stiller er:

- 1) Er det forskjell på å jobbe med variabler i visuell programmering og tekstbasert programmering?
- 2) Trenger elevene mer støtte fra lærer når de arbeider med den ene metoden for programmering, tekstbasert eller visuell programmering?
- 3) Er det forskjell på å lære variabler i visuell programmering og tekstbasert programmering?

## 1.3 Tittel og oppbygging av oppgaven

Å bruke variabler i programmering i matematikk ser nærmere på om det er forskjell på å jobbe med variabler og lære variabler i Python og blokkprogrammering. Den ser også på om det er forskjell i behovet for lærerstøtte i de to måtene å programmere på. Studien ser også på noen fordeler og ulemper ved bruk av ny teknologi i undervisningen.

I kapittel 1 er det en introduksjon til oppgaven med bakgrunn for masteren, formålet med masteren og forskningsspørsmål som stilles.

I kapittel 2 er teori som er brukt til drøftingen.

Kapittel 3 er metodedelen som viser hvordan studien er gjennomført, samt en etisk refleksjon og et kritisk blikk på studien, samt validitet, reliabilitet og overførbarhet.

Kapittel 4 viser funnene av studien og er delt opp i tre underkategorier, funn fra observasjon, funn fra intervju, samt andre funn som ikke handler om elever, men funn om meg som underviser, samt personvernsutfordringer. Dette funnet baserer seg på observasjon og refleksjon.

I kapittel 5 er det drøfting av funn. Drøftingen er delt opp i delkapitler som drøfter de tre forskningsspørsmålene, samt drøfting av andre funn.

Kapittel 6 er en konklusjon basert på drøftingen av funnene.

## 2. Teori

Når vi jobber med programmering i matematikk så jobber vi på digitale plattformer. I møtet med det digitale møter vi på nye muligheter, men og nye utfordringer.

### 2.1 Digital didaktikk

Digital didaktikk bygger på Vygotskijs sosiokulturelle læringsteori om den proksimale utviklingssonen. Det vil gi dypere læring og større læringsutbytte når lærer og elever samarbeider og hjelper hverandre og at man lærer med teknologien og ikke av teknologien. Digital didaktikk fører til at lærerrollen i større grad blir å tilrettelegge for elevstyrt læring (Jahnke & Nordberg, 2013, i Johanson & Karlsen, 2018). Målet er selvregulert læring og større eierskap til læringen fra elevenes side (Johanson & Karlsen, 2018). I utformingen av prosjektet mitt så var elevstyrt læring sentralt.<sup>1</sup>

Som sagt er digital didaktikk bygd på Vygotskijs sosiokulturelle læringsteori. Ifølge Vygotskij så bør læringen skje i elevens nærmeste utviklingszone, elevene trenger veiledning og støtte (Scaffolding) i sin aktivitet. Veiledningen og støtten bør gis på en slik måte at elevene selv kan finne løsningen ved hjelp av hint, forklaring og forslag til hvordan de kan løse oppgaven (Skaalvik & Skaalvik, 2018). Videre så fremheve Vygotskij viktigheten av språket for å lære. For det første brukes språket til å kommunisere med andre som for eksempel medelever eller lærer. For det andre klargjøres tanken når språket brukes. Ifølge Vygotskij så igangsettes læringsprosessen i sosiale settinger, for så å internaliseres hos individet (Postholm i Steen-Olsen & Postholm, 2015).

#### 2.1.1 Digital etikk

I møte med digitale verktøy så utfordres etikken. I digital etikken er målet å identifisere fordelene med ny teknologi, samt identifisere og begrense risiko (Bergsjø, Eilifsen, Tønnesen & Vik, 2020). Dette blir stadig mer aktuelt i skolen da ny teknologi som programmering tas i bruk. I mitt prosjekt blir særlig hensynet til personvern aktuelt. En utfordring er å beskytte brukere av programvare mot unødvendig datainnsamling (Bergsjø et. al., 2020). I mitt prosjekt brukte vi tredjeparts nettsteder til å programmere og datainnsamling er da et utfordringsområde. For å unngå noe av denne utfordringen så programmerte vi på nettsidene

---

<sup>1</sup> Hentet fra min prosjektskisse

uten å opprette en brukerkonto, da opprettelsen av en slik konto krever personopplysninger som navn, e-postadresse og alder.<sup>2</sup>

Det var ikke mulig å lagre arbeidet uten å opprette konto, men dette ble løst med å ta skjermbilder for visuell programmering og kopiere tekst fra nettside og lime inn i et tekstdokument for tekstbasert programmering.

## 2.2 Programmering i skolen

Nordiske land har en bred forståelse hva programmering i skolen innebærer, inkludert nøkkelkonseptet algoritmisk tenkemåte som består av modellering, abstrahering og feilsøking. I læreplanene i nordiske land, Norge inkludert, så ses algoritmisk tenking som en tankeprosess og er dermed uavhengig av teknologi. Algoritmisk tenkemåte er en kompetanse som kan overføres på tvers av fag og disipliner. Videre så er algoritmisk tenking ikke bare en problemløsningsmetode, men kan også brukes til å uttrykke seg i digitale medier.

Nordiske land har prioritert å innføre programmering og algoritmisk tankegang (Bocconi, S., Chiocciariello, A. og Earp, J., 2018)

Studien ser også nærmere på hvordan de forskjellige nordiske landene jobber med programmering og nevner at flere universiteter tilbyr programmering i lærerutdanning. De presenterer en tabell som tar for seg hvordan de forskjellige landene løser fire kategorier for programmering og hjelpemidlene som brukes. Her er en forenklet tabell som viser hvordan Norge løser dette:

Algoritmisk tankegang og programmering av datamaskiner.	Algoritmer, løkker, betingelser, variabler og funksjoner, feilsøking, utvikle enkle simuleringer.
Algoritmisk tankegang og programmering av fysiske objekter.	Lego Mindstorms Arduino, Droner, Micro:bit
Tekstbasert programmering	Python, Javascript
Visuell programmering.	Scratch, Code Studio, Micro:bit

Tabell 1: Forenklet tabell fra artikkelen «*The Nordic approach to introducing Computational Thinking and programming in compulsory education*» av Bocconi, S., Chiocciariello, A. og Earp, J. fra 2018<sup>3</sup>

---

<sup>2</sup> Hentet fra min prosjektskisse

<sup>3</sup> Tabellen er oversatt og forenklet av meg



Vinnervik og Bungum (2022) ser på hvordan den norske og svenske skolen inkluderer algoritmisk tankegang i artikkelen «Computational thinking as part of compulsory education: How is it represented in Swedish and Norwegian curricula?»

I funnene sine ser de at begge land har hatt stort fokus på reformer som inkluderer fremtidsrettede ferdigheter som problemløsning, kreativitet og digitale ferdigheter.

Programmering er et viktig verktøy for å lære algoritmisk tankegang i både Sveriges og Norges læreplaner og begge land henviser til programmering som en digital kompetanse, og mer presist et verktøy, ikke bare for problemløsning, men også for å lære tradisjonelle faglig innhold. Dette kan bety at programmering eller algoritmisk tankegang ikke er fullt ut akseptert som en ferdighet i seg selv.

I den norske læreplanen er algoritmer og noen grunnleggende programmeringskonsepter underlagt matematikkfaget, mens i andre fag er programmering sett på som et verktøy for å lære faginnhold. Det er et større potensiale i skolen for å lære algoritmisk tankegang og programmering enn det som kommer fram i læreplaner, men det er avhengig av lærernes tolking av kompetansemål, prioriteringer og lærernes egne ferdigheter.

Et av utfordringene for å inkludere programmering på en bred måte i skolen er økonomi, tid og opplæring av lærere innenfor programmering (Vinnervik & Bungum, 2022).<sup>4</sup>

I den Norske skole har det vært en stor debatt om bruk av IKT i skolen. Diskusjonen har gått ut på om digitale hjelpemidler fremmer eller hindrer læring. Det har vært vanskelig, om ikke umulig å konkludere at det fremmer eller hemmer læring. En viktig faktor er hvordan læreren klarer å inkludere digitale verktøy i sin undervisning. For å kunne bruke det på en god måte kreves både didaktisk og digital kompetanse. Ordet digital kompetanse har blitt innført i nyere tid og etter innføringen av Kunnskapsløftet, LK06 har det blitt regnet som en egen basisferdighet (som har blitt videreført ved LK20). En del av fokuset innenfor denne kompetansen har vært på den tekniske biten, mens den sosiale biten innenfor denne kompetansen har til en viss grad vært oversett (Svanberg & Wille, 2013)

### *2.2.1 Variabler og algebra*

Et sted vi møter variabler i tillegg til programmering er i algebra. Algebra var i sin tid et stort skritt i sin for fremragende matematikere, og det er ikke rart at det kan være utfordrende for

---

<sup>4</sup> Fra min projektskisse

dagens elever å forstå algebra. Den algebraen vi bruker i grunnskole og videregående skole består hovedsakelig av likninger, funksjoner og formler (Hinna, Rinvold & Gustavsen, 2016).

Variabler kan forstås på mange måter. En måte å tenke på variabler er at de representerer et unik, men ikke ukjent mengde. Eller at variabler kan representere mengder som kan endre seg. En misoppfatning elever kan ha om variabler er at variabelen, for eksempel en bokstav, tilsvarer en bestemt mengde, og ikke at en variabel kan representere flere eller uendelige mengder (Van de Walle, Karp & Bay-Williams, 2015).

Mange elever møter en utfordring når de skal tolke tekstoppgaver knyttet til algebra. Dette kan grunnes at elevene misforstår eller overser ord og fraser. En annen grunn kan være at de gjør en feil i prosessen med å løse oppgaven som fører til et feil svar, eller at de gjør en feil i vurderingen av prosessen for å løse oppgaven (Jupri & Drijvers, 2016).

Flere forskere er i dag inspirert av Vygotskij og anbefaler elevene for mulighet til å lære algebra som språk og tankeredskap. Algebra som språk innebærer at eleven bruker symbolene i algebra for å uttrykke meningsfulle matematiske påstander. Algebra som tankeredskap innebærer at symbolene i algebra brukes av elevene til å tenke (Hinna et al., 2016)

### *2.2.2 Variabler i programmering og Python*

Om man ønsker å bruke en verdi i et Python-program så kan denne verdien lagres i en variabel. I Python lages en variabel når en verdi er gitt til variabelen. Når den er lagd kan den brukes senere i programmet med andre variabler eller verdier. For å lage en variabel så gir man den et navn etterfulgt av likhetstegn og en verdi. Python skiller mellom store og små bokstaver i variabler, det vil si at hvis du lager en variabel med stor forbokstav og vil bruke denne senere, men skriver med liten forbokstav så vil ikke Python kjenne igjen variabelen. Det er lurt å gi variabelen et beskrivende navn som forteller noe om verdien som er lagret i variabelen (Sweigart, 2015). Videre så må en variabel i Python starte med en bokstav eller understrek, den kan ikke starte med tall. Man kan kun bruke alfanumeriske tegn og understrek, samt at variabelen ikke kan være Pythons nøkkelord (operatorer og lignende)

Flere tekstbaserte programmeringsspråk krever kunnskap om å definere variabler før disse kan brukes i programmet. Python er et objektorientert programmeringsspråk der alt behandles som et objekt, som for eksempel funksjoner, som gjør at når du gir en funksjon et navn og en verdi, så har du definert variabelen (Pilgrim, 2009). Dette gjør at Python er et relativt enkelt programmeringsspråk å lære.

Ifølge Karlsen er det som er aller mest bruk i programmering variabler. Variabler brukes i alle typer programmering. For å visualisere hva en variabel er foreslår hun å sammenligne det med en boks hvor man kan putte hva som helst oppi (Karlsen, A. i Sund, 2023)

Hovedforskjellen på variabler i algebra som elever jobber med på skolen og variabler i Python er at i algebra så skal vi ofte finne verdien til en variabel i oppgaver, mens i Python gir vi en verdi til variabelen for å bruke denne senere.

Blokkprogrammeringen som ble brukt i prosjektet er Python-basert og følger samme regler som beskrevet ovenfor

### ***2.2.3 Tidligere studier om programmering og algebra***

Selv om programmeringsspråk har sitt opphav i algebra, så medfører disse programmeringsspråkene også et nytt system av representasjoner og syntaks-regler som skiller seg fra de naturlige språkene. For å lære disse nye språkene må vi lære både språkets oppbygging, syntaksen, og semantikken, språkets innhold. Det er en forskjell på hvordan elever skaper forståelse mellom algebra og programmering, der algebra er mer dynamisk mens programmering er mer statisk. Elevene kan få en større forståelse for likhet og ulikhet i algebra gjennom kontrasterende eksempler som viser forskjellen mellom algebra og programmering, men denne læringen kan hindres om læreren ikke er bevist på elevenes erfaring i møte med disse forskjellene. Når programmering innfører et nytt sett med meninger så vel som en ny syntaks kan dette føre til misoppfatninger og ha en uheldig innvirkning på læring av algebra (Bråting & Kilhamn, 2021).

Det å kombinere to ulike fagområder kan gi elevene en høy kognitiv belastning. Fellestrekk mellom algebra og programmering kan gi læring hos elevene. Lærere bør skaffe seg kunnskap om variabelens mange fasetter og didaktiske utfordringer (Høyland, 2021).

Bruk av blokkbasert programmering for å lære matematikk viser seg å være engasjerende og oppmuntrer til diskusjoner elevene mellom. Feilsøking i fellesskap er en god måte å lære et tema i fellesskap. Dette indikerer at algoritmisk tankegang er en sosial prosess og en samarbeidsprosess når dette er basert på å lære blokkprogrammering som en integrert del av matematikkundervisningen. Fra et algoritmisk tankesett ståsted er det tydelig at klassediskusjonen ofte består av konsepter som løkker og variabler. Når elevene jobber med blokkprogrammering er det tydelig at samtalen omhandler sentrale matematiske prosesser (Andersen, 2022)

## 2.3 Oppgaver, motivasjon og samarbeid

### 2.3.1 Oppgaver

Oppgavene som ble brukt i perioden var tekstoppgaver i form av problemløsningsoppgaver. Problemløsningsoppgaver består hovedsakelig av to deler, en analysedel og en syntesedel. Analysen består av å ta oppgaven fra hverandre, bryte den ned i mindre bestanddeler og lager en plan på å løse oppgaven. Syntesedelen består av å sette oppgaven sammen igjen basert på analysedelen og løse oppgaven (Hana, 2014). Oppgavene økte gradvis i vanskelighetsgrad utover prosjektperioden.

### 2.3.2 Kort om motivasjon og samarbeid

Kompetanse i matematikk handler for eleven om å utvikle forståelse og ferdighet, samt opplevelse av mestring i matematikkfaget. Autonomi innebærer å handle ut ifra egne interesser og verdier. I skolen kommer elevenes handling fra en ytre kilde som for eksempel læreren, men elevens handling er autonom om den bunnar i egne interesser og verdier. Tilhørighet handler om å føle seg trygg i et fellesskap. Det fremheves ofte at elevene må være trygge på hverandre for å kunne jobbe sammen med hverandre, men det kan også gå andre vei, om elevene samarbeider så blir de trygge på hverandre.

Tilhørighet gir elever tryggheten de trenger for å være autonome. Når elevene opplever autonomi mens de arbeider med oppgaver gir dem økt følelse av kompetanse. Følelsen av kompetanse gir elevene den selvtilliten de trenger for å føle seg akseptert og oppleve tilhørighet. Om elevene får positive erfaringer i å mestre disse utfordrende oppgaver kan de få en positiv innstilling til å jobbe med matematikk. Om elevene ikke får utfordringer kan de over tid miste interessen for matematikk (Wæge & Nosrati, 2018).

Kunnskapsmobilitet (Knowledge Mobility) er når kunnskapen går fra elev til elev, og fra gruppe til gruppe. For å øke denne kunnskapsmobiliteten er det viktig at elevene ikke jobber i selvvalgte grupper. Selvvalgte grupper fungerer som en barriere for at kunnskap skal kunne flytte rundt iblant elevene, da disse gruppene ofte er de samme og fungerer som sosiale barrierer i tillegg. Veksling i gruppene eller læringspar er med på å bryte ned begge barrierene, den sosiale og barrieren for kunnskapsmobilitet. Denne mobiliteten har tre former, elever fra en gruppe låner ideer fra en annen gruppe (1), elever fra en gruppe går og sammenligner sine løsninger med en annen gruppe (2), to eller flere grupper diskuterer forskjellige løsninger (Liljedahl, 2021).

### *2.3.2 Skolens og lærerens plass og rolle*

Elevene kommer ikke som blanke tavler inn i skolen (tabula rasa), men bringer med seg egne interesser og erfaringer. Noen av disse hverdags erfaringene er like for elevene, mens andre er ulike. Elevenes hverdags erfaringer møter skolens struktur i klasserommet. Utfordringen i skolen er å gjøre undervisningen relevant for elevenes hverdagsliv (Erstad & Smette, 2017).

Skolen gjennomgår reformer og det innføres nytt pensum. Disse reformene kan bunne i en ytre påvirkning som samfunnsendringer der skolen må tilpasse seg samfunnets behov. En utdanningsreform skal ideelt sett innebære framskritt i form av implementering av ny teknologi, ny kunnskapsorientering osv. (Postholm, Haug, Munthe & Krumsvik, 2017).

George Pólyas firestegs problemløsningsprosess opptrer ofte i mange lærebøker. Det å lære disse stegene til elevene eksplisitt kan øke elevenes kompetanse på problemløsning. Disse fire stegene er kort fortalt:

- 1) Forstå problemet: Først må du finne ut hva problemet omhandler og identifisere hvilke spørsmål eller problem som blir stilt
- 2) Lag en plan: Du må tenke på hvordan løse problemet. Vil du gjøre det skriftlig med utregning eller vil du løse problemet med konkrete?
- 3) Gjennomfør planen: I dette steget gjennomfører du planen du valgte.
- 4) Se tilbake: Dette siste steget er kanskje det viktigste steget som ofte blir oversett. Det er her vi ser om svaret i steg 3 svarer på spørsmålet eller problemet i steg 1. Om den

ikke svarer på dette, gå tilbake til steg 2 og velg en annen strategi for å løse problemet, eller gå tilbake til steg 3 for å fikse noe med din strategi.

Pólyas problemløsningssteg kan og bør implementeres i mange typer problemløsning fra enkel utregning til autentiske problemer med mange steg (Van de Walle et al., 2015).

For å kunne gi en god undervisning innenfor alle disipliner i matematikk, programmering inkludert, trenger læreren selv god kunnskap innenfor feltet. Et verktøy for å reflektere over hvordan denne kunnskapen kommer til uttrykk i undervisningen er kunnskapskvartetten. Denne kan deles inn i fire brede dimensjoner: foundation, transformation, connection og contingency.

- Foundation er grunnlaget læreren har i form av sin kunnskap som læreren har tilegnet seg gjennom blant annet sin utdanning. Dette er en kunnskap læreren har selv om det ikke blir brukt til andre formål. I denne kunnskapen ligger det en potensiale til å påvirke pedagogiske valg og strategier. Kunnskap og forståelse for matematikk, samt kunnskap om forskningsbasert mattedidaktikk hører hjemme her.
- Transformation er hvordan læreren «re-representerer» lærestoff for elevene. Dette kan ta form av eksempler, aktiviteter, demonstrasjoner og spørsmålsstilling. Denne kategorien omhandler handlinger som er rettet mot elever og er basert på grunnleggende kunnskap (foundation).
- Connection handler om å se sammenhenger og helhet i matematikkfaget. Denne delen knytter sammen de adskilte delene innenfor faget som for eksempel læringen av en prosedyre.
- Contingency handler om hvordan en lærer håndterer uforutsette innspill og lærerens beredskap og evne til å svare på disse.

For matematikklærere er kunnskapskvartetten i hovedsak et refleksjonsverktøy for å reflektere over egen eller andres undervisning (Gustavsen, Hinna, Borge & Andersen, 2014).

Lærere må kunne faget de skal undervise i. Dette er kanskje det mest grunnleggende i en lærers kompetanse. Lærere trenger denne kunnskapen til å hjelpe elever lære faginnholdet. Samtidig er ikke fagkunnskap alene nok for å undervise. Læreren må kunne matte på en slik måte at den er nyttig for elevene og gir matematisk mening for elevene. Det å kunne avansert matematikk vil ikke være nok for å gi en god læring for elevene. Det viktigste er å forstå og være i stand til å undervise i matematikk. (Loewenberg Ball, Thames & Phelps, 2008)

## 3 Metode

Metodedelen består av hovedkapitlene 3.1 Vitenskapsteoretisk paradigme. 3.2 Kvalitativ tilnærming. 3.3 Datainnsamling. 3.4 Analyse. 3.5 Utvalg. 3.6 Gjennomføringen. 3.7 Validitet, reliabilitet og overførbarhet. 3.8 Et kritisk blikk på studien og 3.9 Etske betrakninger

### 3.1 Vitenskapsteoretisk paradigme

Vår tolkning av datamateriale er ikke en avspeiling av et fenomen, men heller en tolkning av fenomenet. Konstruktivismen påpeker at våre observasjoner baserer seg på tidligere erfaringer (Bukve, 2021, s31). Konstruktivismen baserer seg på at vi ikke kan skille mellom det som studeres og den som gjennomfører studien og vi kan aldri med full sikkerhet si at det vi studerer faktisk er slik. Mitt ståsted er i sosialkonstruktivismen som sier at mennesker ikke skaper sin oppfatning alene, men i samhandling med andre mennesker. Alle konstruktivistiske epistemologiene har et felles utgangspunkt: Verden er ikke objektiv, men er noe vi mennesker mer eller mindre aktivt konstruerer (Postholm & Jacobsen, 2018, s49-51).<sup>5</sup>

Ontologiske spørsmål kan sies at handler om hvordan virkeligheten ser ut, mens epistemologiske spørsmål handler om hvordan man kan tilegne seg kunnskap om virkeligheten. Fortolkning vil være elementært i samhandlingen mellom mennesker (Johannesen, Tufte & Kristoffersen, 2016).

### 3.2 Kvalitativ tilnærming

I min studie brukte jeg en kvalitativ tilnærming for å få et innblikk i elevenes opplevelse av å bruke variabler i programmering. For å få en forståelse for dette brukte jeg observasjon og intervju. Kvalitativ tilnærming er bruk av ord mens kvantitativ tilnærming er å bruke tall (Merriam, 2009). Ved en kvalitativ tilnærming henter jeg inn informasjon gjennom ord og språk (Postholm & Jacobsen, 2018). Med bruk av kvalitativ tilnærming undersøker jeg en liten gruppe tett, mens om jeg hadde valgt en kvantitativ tilnærming ville jeg ha undersøkt en langt større gruppe.

---

<sup>5</sup> Fra min prosjektskisse

### 3.3 Datainnsamling

Dette var en observasjons- og intervjustudie der det ble gjort observasjoner i klasserommet underveis i studien, og intervju av to informanter når studien var over. Den ene informanten hadde programmert med tekstbasert programmering og den andre med visuell programmering.

#### 3.3.1 Observasjon

Observasjonen ble gjort ved hjelp av et observasjonsskjema som jeg fylte inn underveis i timen ved hjelp av stikkord. Den hadde noen faste punkter, men og et stort felt for å kunne gjøre frie notater. I etterkant av timen ble disse skrevet inn i et Word-dokument med hele setninger. Ingen av observasjonene inneholdt noe som kunne være personidentifiserende som for eksempel kjønn eller initialer.

Observasjon har blitt sett på som den mest grunnleggende metoden for datainnsamling. Observasjonene skjer i en naturlig setting og ikke inne på et laboratorium og kan derfor kalles naturalistisk. Gjennom bruk av observasjon kan forskeren se nærmere på både menneskelig aktivitet og på omgivelsen dette skjer i (Postholm & Jacobsen, 2018). I min observasjon så observerte jeg på samhandling i klasserommet som elevene til vanlig bruker, så omgivelsene observasjonene pågikk i var elevenes vante omgivelser. Derfor var ikke omgivelsene en faktor som påvirket observasjonene.

Observasjon som metode er velegnet når man vil ha direkte tilgang til det man undersøker, eksempler på hva som kan undersøkes ved observasjon er samhandlinger. Observasjon er best egnet innenfor et begrenset geografisk område som for eksempel et klasserom (Christoffersen & Johansen, 2018). Det å drive med observasjon krever en nærhet til den eller de som observeres, derfor er observasjon best egnet på et begrenset geografisk område. Det må være oversiktlig for observatøren hva som foregår for å få et mest mulig helhetlig bilde av de som observeres

I min observasjon kommer jeg tett på elevene da det er elever jeg underviser ukentlig i matematikk, så min avstand fra deltakerne i min forskning er liten. Samtidig er jeg den som gjennomfører undervisningsopplegget i forskningen og har dermed stor deltakelse. Dette gjør min observatørrolle til en fullstendig deltaker (Postholm & Jacobsen, 2018). Fordelen er at jeg



kommer tett på deltakerne, men ulempen er at jeg kan påvirke resultatet og at min tolkning av data kan bli påvirket av denne relasjonen. For å gjøre observasjonen enklere benytter jeg meg av et observasjonsskjema med kategorier jeg kan fylle ut, men om noe utenom skjemaet oppstår så skrives denne observasjonen ned i etterkant. I min observasjon noteres ingen personopplysninger ned.<sup>6</sup> En annen faktor som påvirker min mulighet til å gjøre funn under en absolutt deltakende observatørrolle er at mitt fokus vil noen ganger være på å hjelpe å støtte elever i sin læring og kan derfor gå glipp av viktige observasjoner som har innvirkning på funnene.

Det at det ikke ble introdusert noe nytt i klasserommet i form av observatører eller andre elementer enn det elevene er vant til gjorde at observasjonen ble gjort i en naturlig setting som gjorde at funnene ble så realistisk som mulig (Cohen, Manion, Morrison & Bell, 2011). Faktorene i min observasjon var ikke knyttet til undervisningsområder eller undervisningsmateriell. Disse var faktorer elevene møter i sin skolehverdag og er derfor ikke noe ukjent som trekkes inn i situasjonen der observasjon pågår. Dette gjorde at jeg i stor grad kunne fokusere min observasjon på mitt forskningsområde som var bruk av variabler knyttet til forskjellige metoder for å programmere i matematikkfaget.

Svakheten i observasjonen var at jeg min observatørrolle var absolutt deltakende, og jeg måtte støtte og hjelpe elever underveis i prosjektperioden. Dette kan være med på å gjøre at jeg ikke var i stand til å observere i en like stor grad som en absolutt observatør ville kunnet være i stand til å observere (Christoffersen & Johansen, 2018). Styrken i min observasjon var som sagt at det ikke var andre elementer i undervisningen enn det elevene allerede var vant til og at ingen ukjente elementer kom inn i undervisningsrommet som kan påvirke elevenes væremåte. Derfor fikk jeg mulighet til å gjøre observasjoner på mine forskningsspørsmål i stedet for andre påvirkninger elevene kan møte på i en observasjonssammenheng.

### **3.3.2 Intervju**

For å innhente data gjennomfører jeg et fenomenologisk intervju etter at jeg har gjennomført prosjektet. De er to informanter, en fra niende klasse og en fra tiende klasse. Intervjuet er et semistrukturert intervju med intervjuguide. Dette gir mulighet for oppfølgingsspørsmål og nye

---

<sup>6</sup> Fra orosjektskissen

temaer kan dukke opp i intervjuet (Postholm & Jacobsen, 2018). Fordelen med denne formen for intervju er at informanten selv kan komme med viktige betraktninger om temaet det intervjues om. Dette gir rom for informanten å fokusere på det som de opplevde som relevant knyttet til prosjektet. Jeg forsøkte å lage intervjuguiden åpen nok til at informantens syn i størst mulig grad kunne komme fram, men fokusert nok til at intervjuet svarte på forskningsspørsmålene.

Intervju er velegnet for å samle inn kvalitative data, da informanten har større frihet til å uttrykke seg enn det et strukturert spørreskjema gir. Oppfatninger og erfaringer til mennesker kommer best fram om informanten får være med å bestemme hva som blir tatt opp i intervjuet. Sosiale fenomener er komplekse og et kvalitativt intervju gir mulighet for å få fram nyanser og kompleksitet (Christoffersen & Johansen, 2018). Ved bruk av forhåndsbestemte svar så har informantene liten mulighet til å få fram akkurat hva de tenker, svarene må passe inn i en forhåndsbestemt kategori. I bruken av et semistrukturert intervju så for informantene mulighet til å gi svar som stemmer overens med egen oppfatning av situasjonen. Dette er en fordel for forskeren og, da det gir rom til å få fram nyanser knyttet til opplevelser og kompleksiteten som finnes i et samfunn, eller i mitt tilfelle et klasserom.

Relasjonen mellom intervjuer og informant er viktig for kvaliteten på informasjonen som kommer fram i intervjuet. Settingen for intervjuet handler om hvor intervjuet foregår (Christoffersen & Johannessen, 2012). Jeg har en god relasjon til informantene så dette kan være med på å sikre kvaliteten i informasjonen som kom fram i intervjuet. Settingen var på et kjent rom som informantene jevnlig bruker, et grupperom ved skolen så intervjuet foregikk i kjente omgivelser. Dette kan bidra til å skape ro og trygghet i en intervjusituasjon som gjør at informantene får mulighet til å reflektere over sine svar underveis i intervjuet. Det ble spurt hos informasjonen om det var greit å gjøre lydopptak under intervjuet når intervjuet startet og begge informantene samtykket i dette. I etterkant ble intervjuet transkribert og informantene fikk lese gjennom transkripsjonen for å se om de hadde blitt forstått rett. Lydopptaket ble slettet etter transkripsjonen.

I utvelgelse av informanter til intervjuet er første betingelse at elever og foresatte har skrevet under på et samtykkeskjema, kun de elevene med informert samtykke er med i denne prosessen. En strategi for tilfeldig utvalg er å trekke ut for eksempel hver tiende elev fra en

liste, men dette krever en lang liste (Christoffersen & Johannessen, 2018). I mitt utvalg har jeg en relativ liten gruppe, så mitt tilfeldige utvalg gjennomføres med loddtrekning. Betingelsen var at det skulle trekkes en tilfeldig fra gruppen som programmerte med tekst, og en tilfeldig fra gruppen som programmerte visuelt. En kritisk refleksjon for min utvalgsstrategi er at det er ganske få elever, og at hver elev derfor har stor sannsynlighet for å bli valgt gjennom trekning og da kan man spørre seg om dette er tilfeldig nok med trekningen.<sup>7</sup> Likevel valgte jeg en tilfeldig trekning da dette ga meg størst sjans for å få svar på spørsmålene uten påvirkning fra meg som forsker.

Det ble gjort et pilotintervju for å sjekke om spørsmålene var forståelig og ga informantene mulighet til å gi så riktige og presise svar som mulig. Det å stille gode spørsmål er en nøkkelfaktor for å finne svar på det man forsker på, og et pilotintervju vil kunne teste spørsmålene (Merriam, 2009). Pilotintervjuet ble testet på en elev som ikke var trukket ut til selve intervjuet slik at selve intervjuet var informantens første møte med spørsmålene. Dette ble gjort slik at begge informantene skulle ha samme utgangspunkt i selve intervjuet, og at den ene informanten ikke skulle kunne møte mer forberedt til å svare på spørsmål enn den andre informanten. Selve pilotintervjuet ble gjennomført etter at jeg hadde trukket informanter til hoved intervjuet. Deltakeren i pilotintervjuet var en elev som var del i prosjektet og hadde gitt samtykke til å delta i prosjektet.

### *3.3.3 Teknologiske hjelpemidler i studien*

I intervjuet ble det brukt en lydopptaker for å samle data fra informantene. Begge informanter samt deres foresatt var informert om dette og hadde gitt samtykke til dette. Fordelen med å bruke lydopptaker er at det gir en bedre og en mer nøyaktig gjengivelse av det som blir sagt i et intervju sammenlignet det å bruke egne notater, for eksempel i form av stikkord (Yin, 2014). I etterkant ble lydopptaket transkribert og selve opptaket slettet.

Jeg ønsket i utgangspunktet å benytte meg av NVivo i analysearbeidet, men opplevde det som svært arbeidskrevende å bruke og det var derfor lettere for meg å gjøre analysen mer manuelt. Dette var lettere for meg, særlig da jeg har få informanter, enn å sette meg grundig inn i

---

<sup>7</sup> Fra prosjektskissen

programmet i prosjektperioden samtidig som jeg var i full jobb. Ved en større studie, og med større kjennskap til programmet vil det i stor grad være lettere å ha oversikt over hva informantene tenker når man deler svarene deres i meningsblokker.

### 3.4 Analyse

Intervjuet ble transkribert i etterkant slik at jeg kunne analysere mine funn. I analyseprosessen brukte en fenomenologisk tilnærming for å lettere kunne sortere data i meningsenheter som gjør videre analyse enklere. Selve analysen var en tematisk analyse som gir en teoretisk fleksibel måte å analysere kvalitative data. Med en slik tilnærming til analyseprosessen blir jeg ikke bundet til å se kun etter mønster eller tematiske funn, noe jeg kan bli med en ren fenomenologisk analyse, men kan få en større forståelse for hva deltakerne i studiet har erfart i løpet av forskningsperioden.

I praksis betyr det at delte transkripsjonen inn i meningsenheter for å sortere dataene, men kunne også se nærmere på transkripsjonen som helhet, deler av transkripsjonen, eller en detaljert lesing av transkripsjonen.<sup>8</sup>

#### 3.4.1 Fenomenologisk analyse

Giorgio (1985) beskriver en tretrinnsprosess for den fenomenologiske analysen

- 1) Lese for å få en oppfatning av helheten
- 2) Utvikling av meningsenheter
- 3) Transformerings av deltakernes uttalelser til psykologiske uttrykk

Steg 1 er å lese gjennom hele transkripsjonen for å få et helhetsinntrykk. Steg to er å dele transkripsjonen inn i meningsenheter, disse må samsvare med psykologiske fenomenologiske perspektivet. I forskning i skolen betyr det at pedagogiske teorier kan være forskerens utgangspunkt. Steg tre er å ta utgangspunkt i meningsenheten for å kunne presentere en andre-ordensbeskrivelse. Selve strukturen til andre-ordensbeskrivelsen er meningsenhetene, som er relatert til teori. Her brukes ikke teorien til å tolke materialet, men for å sette ord på funnene samt abstrahere til den disiplinen forskningen er rettet mot, i mitt tilfelle pedagogikken

---

<sup>8</sup> Fra prosjektskissen

(Postholm & Jacobsen, 2018, s.160-161).<sup>9</sup> Etter at intervjuene ble transkribert ble de gjennomlest for å gi meg et helhetlig inntrykk av det informantene kom med under intervjuet. Det jeg opplevde som en utfordring var å dele opp funnene i meningsenheter. Mine valg av disse enhetene er påvirket av at jeg ikke har så mange års erfaring innenfor forskningsfeltet så en mer erfaren forsker villet kanskje kommet opp med en annen og mer grundig måte på å dele opp disse meningsenhetene. Når det er sagt så har jeg delt opp meningsenhetene på en slik måte at de lar seg sammenligne mellom informantene for å finne likheter og ulikheter i funnene. Det tredje steget består av å bruke gjeldende teorier til å sette ord på funnene og ikke tolke funnene. Jeg har i stor grad forsøkt å bruke gjeldende teori og tidligere forskning til å kunne ordlegge mine funn.

### *3.4.2 Tematisk analyse*

Tematisk analyse presenteres innenfor hermeneutisk fenomenologi. Det å analysere livserfaringer er en kompleks og kreativ prosess, og at det ikke er en regelbunden prosess for å utvikle og formulere en tematisk forståelse (Van Manen, 2016, i Postholm & Jacobsen, 2018). Forskeren kan ta i bruk tre ulike tilnærminger for å frembringe tema og innsikt:

1. En helhetlig lesetilnærming der forskeren leser hele transkripsjonen og utformer en setning som dekker meningsinnholdet.
2. En selektiv tilnærming der forskeren leser transkripsjonen mange ganger for å finne hvilke utsagn som er essensielle eller gir uttrykk for fenomenet eller erfaringen som blir beskrevet.
3. En detaljert tilnærming der forskeren retter oppmerksomheten mot hver enkelt setning eller klaser av setninger og spør hva som uttrykkes eller beskrives av fenomenet gjennom disse. For å få frem meninger knyttet til erfaringer prøver forskeren å identifisere tematiske uttrykk, uttalelser eller narrative paragrafer.

Dermed kan en erfaring eller et fenomen a) forskningsdeltakers beskrivelse tas opp og transkriberes, b) bli skrevet om til en anekdote, c) gjennom helhetlig, selektiv eller linje for linje-tematisering bli bearbeidet, d) dette blir forskerens utgangspunkt til skrivingen (Postholm & Jacobsen, 2018, s. 161-162). I min gjennomlesing av transkripsjonen for å få et helhetsinntrykk så skrev jeg kort om meningsinnholdet i teksten. Jeg leste gjennom transkripsjonen med et mer selektivt blikk for å se om det var noen utsagn som sto fram som viktige, eller ga et godt bilde av informantens erfaringer. Deretter så jeg nærmere på setninger

---

<sup>9</sup> Fra prosjektskissen

eller beskrivelser som kunne fortelle meg noe om informantens opplevelse av fenomenet programmering ved bruk av variabler.

Tematisk analyse er en fleksibel måte å analysere data på som gir et stort spenn av analytiske muligheter, noe som gir et bredt rom for å tolke et datasett på. Dette kan være en fordel for å hente ut informasjon fra et bredt spekter. Men kan også være en ulempe da det kan vanskeliggjøre utviklingen av spesifikke retningslinjer for en høyere-ordens analyse i et større forskningsprosjekt, som for eksempel et forskningsprosjekt med mange deltakere. Det kan være overveldende for en forsker å avgjøre hvilke aspekter av datasettet som skal fokuseres på. Tematisk analyse har en begrenset tolkningsmulighet utenom en enkel beskrivelse av et fenomen om den ikke brukes innenfor eksisterende teoretiske rammeverk å forankre beskrivelsen av fenomenet i (Braun & Clarke, 2006, s. 97). Begrensingen for meg som forsker ved å bruke denne tilnærmingen for analyse er at mine tolkningsmuligheter av et mer komplekst fenomen er begrenset om jeg ikke kan støtte meg på tidligere etablerte teoretiske rammer. Dette gjør at denne måten å analysere på er godt egnet om man forsker på et fenomen som er beslektet eller knyttet til tidligere forskning, men er uegnet der man er i et helt nytt forskningsfelt og driver nybrottsarbeid. Denne analysemetoden er velegnet i bachelor- og masteroppgaver hvor man gjerne utforsker fenomener knyttet til tidligere forskningsarbeid, men vil kunne hindre en erfaren forsker å gjøre oppdagelser i et helt nytt felt.

Så en tematisk analyse er en fleksibel analysemetode, men det er viktig å være bevisst på mulige fallgruver og begrensninger som metoden frembringer. Dette er gjeldende for alt analysearbeid og alle analysemetoder.<sup>10</sup> All metode for datainnsamling og analyse har sine fordeler og ulemper. Derfor må man gjøre en overveielse om hvilke metoder for datainnsamling og hvilke metoder for analyse gir flest fordeler og færrest ulemper knyttet til det aktuelle forskningsprosjektet.

### ***3.4.3 Bearbeiding i analysen***

Som nevnt tidligere ønsket jeg først å bruke programmet NVivo i mitt analysearbeid. Dette viste seg å være for tidkrevende og vanskelig da jeg hadde liten forkunnskap i hvordan bruke

---

<sup>10</sup> Fra prosjektskissen

programmet. Videre så hadde jeg kun to informanter å analysere og sammenligne svarene mellom, så de var lettere for meg å skrive ut svarene og merke meningsblokker med markørtusjer i forskjellige farger, grønn for positiv, gul for nøytral og rød for negativ. I en større studie ville det vært enklere og sortert disse meningsblokkene og sammenlignet svarene fra de forskjellige informantene. En annen faktor som ville gjort bruken av NVivo enklere er min kunnskap om programmet. Midt i en hektisk prosjektperiode så var det dessverre liten tid til å sette seg inn i et analyseprogram når fokuset mitt var å øke min kompetanse innenfor programmering. Analyse med markering for hånd egner seg ikke til større prosjekter da dette vil føre til dårlig oversiktighet og mye arbeid.

### 3.5 Utvalg

Utvalget var elever fra niende. Og tiende trinn ved en skole. Betingelsen for å delta i prosjektet var samtykke fra elev og foresatte. Prosjektet skulle opprinnelig kun involvere en tiendeklasse, men utfordringen var å få nok deltakere til prosjektet, derfor ble det rekruttert deltakere fra niende klasse ved den samme skolen.

Ofte er det utfordringer å identifisere og få tilgang til informanter innenfor utdanningsforskning. Ikke bare må forskeren få tilgang til informanter, men studien må også være gjennomførbart (Christoffersen & Johannessen, 2018). Dette opplevde jeg i forbindelse med min studie, utfordring med å få nok informanter til min forskning. Min utfordring knyttet til utvalg av deltakere i prosjektet gjorde at jeg måtte velge ut fra to forskjellige klassetrinn. Måten dette kan påvirke funnet er at elever ved tiende trinn gjerne har en større erfaring i oppgaveløsning enn det elever i niende trinn har. Derfor kan elevene i tiende klasse ha flere strategier for oppgaveløsning som gjør at funnene mine favoriserer gruppen i tiende trinn som jobbet med tekstbasert programmering. Valget jeg gjorde var at elever fra niende trinn skulle programmere visuelt mens elever fra tiende trinn skulle programmere med tekst. Dette baserte seg på min vurdering om at tekstbasert programmering er mer krevende enn visuell programmering, men for å få et så godt bilde som mulig knyttet til forskningsspørsmålene mine, så kunne jeg valgt å mikse klassene slik at klassetrinn ikke ble en faktor i forskningen.

### 3.6 Gjennomføringen

Hvordan studien gjennomføres påvirker resultatet, derfor skal vi se nærmere på hvordan min studie ble gjennomført slik at denne kan ettergås med et kritisk blikk. En viktig faktor er

elevenes forkunnskap i forkant av studien. Mine elever hadde en viss forkunnskap, og funnene kan ha vært annerledes med en elevgruppe som ikke har noen forkunnskap, eller en elevgruppe med en større forkunnskap. En annen faktor som påvirker funnene er selve gjennomføringen av studien så denne skal jeg ta for meg, både studieperioden og hvordan enkelttimer ble organisert. I delkapittelet ovenfor denne står det om utvalg som også vil ha en innvirkning som ble gjort i studieperioden, og studiens overførbarhet til andre elever.

### *3.6.1 Elevenes forkunnskap*

For å gi et så helhetlig bilde som mulig så er det viktig å vite elevenes forkunnskap om programmering. Elevene i studien har tidligere erfaring med programmering, så dette er ikke deres første møte med programmering.

Samtlige deltakere har jobbet med programmering før studien. Tre måneder før studien startet hadde elevene et programmeringsverksted med MicroBit der de programmerte en BitBot til å kjøre rundt. Dette var en visuell programmering i nettleser på nettstedet <https://makecode.microbit.org/>. Programmeringsverkstedet var to klokketimer.

Elevene hadde programmert i skoleåret før studien tok plass med både visuell programmering i Scratch (<https://scratch.mit.edu/>) og tekstbasert programmering i Python (<https://trinket.io/python3>). To år før studiestart hadde alle deltakere i studien programmert med BitBot i form av visuell programmering. Tre år før studiestart hadde samtlige elever i studien litt teori om programmering, samt at de hadde programmert hverandre i form av lapper de skrev der andre elever skulle følge instruksene på lappen.

For å innhente denne informasjonen spurte jeg lærerne som underviste elevene i matematikk årene før om elevene hadde kjennskap til programmering for å innhente informasjon. Jeg var selv med på programmeringsverkstedet som var tre måneder før studien.

Så denne studien omfatter elever med tidligere kjennskap til programmering, og dette kan gjøre at funnene ville vært annerledes i en elevgruppe uten forkunnskaper, men også i en gruppe med enda større forkunnskaper.

Dette er en viktig betraktning å ha i bakhodet i gjennomgangen av denne studien.



### *3.6.2 Prosjektets oppbygging*

Kravet for utvelgelse av deltakere i studien var at elevene selv ønsket og delta og at foresatte ga sitt samtykke til deltakelse. Prosjektet hadde tolv deltakere som hadde gitt samtykke i å delta i prosjektet. Disse tolv ble delt i to grupper på seks hver, der den ene gruppen programmerte med tekstbasert og den andre gruppen programmerte med visuell programmering. Det var niende trinn som programmerte visuelt, mens tiende trinn programmerte tekstbasert. Årsaken til denne delingen var min vurdering at tiende klasse hadde mer erfaring med programmering enn niende klasse og at denne delingen ville jevne ut litt av forskjellene mellom klassetrinnene.

Hver gruppe jobbet en time på mandag, en time på onsdag og en time på fredag i to uker, totalt seks timer hver i prosjektperioden. Det ble vekslert på hvem som fikk første time de gjeldende dagene av elevene som programmerte med tekst og elevene som programmerte visuelt. Vekslingen ble gjort slik at begge gruppene fikk mulighet til å ha et ferskt blikk fra lærer på dagens gjennomføring.

Vi programmerte i nettleser på nettsiden <https://trinket.io/python3> for å programmere med Python, og nettsiden <https://trinket.io/blocks> for å programmere med blokker. Det ble i begge tilfeller brukt trinkets nettsted slik at funksjonene i programmeringen skulle være så lik som mulig, der det eneste som skilte var tekstbasert eller visuell programmering. Underveis i prosjektet ble det tydelig at muligheten for å lagre prosjektene uten å opprette en brukerkonto skilte disse to løsningene, dette vil jeg belyse nærmere videre i oppgaven.

Fordelen med å bruke trinket er at det er en gratis løsning og programmet som lages kjøres på server-siden slik at man kan programmere på nettbrett, mobil og PC. Det er uavhengig av egen maskinvare så selv eldre datautstyr kan brukes til programmering. Da det er gratis å bruke har alle skoler uavhengig av skolens økonomi mulighet til å benytte seg av denne løsningen. Det at selve programkjøringen skjer på server-siden gjør at skolene ikke trenger å bruke penger på moderne datautstyr om økonomien er trang. Så dette er en løsning som alle skoler kan benytte seg av.

### *3.6.3 Gjennomføring i timen*

Deltakerne ble i starten av timen satt i læringspar. Det ble rullert hver time på hvem som jobbet i lag slik at alle skulle få mulighet til å samarbeide med alle på gruppen. Elevene fikk

utdelt oppgaver som de løste i læringspar og på slutten av timen gikk vi igjennom løsningene i felleskap. På slutten av timen gjorde vi også endringer i de forskjellige variablene i en løsning for å se hvordan svaret endret seg. Det var et hovedoppgaveark for hver time, samt at jeg hadde ekstraoppgaver om noen ble ferdig før timen var omme. Elevene tok skjermbilder av oppgavene underveis og de som programmerte med tekstbasert programmering kopierte i tillegg koden over i en skriveblokk som de lagret. Elevene la inn kommentarer i programmet for å forklare hva de forskjellige variablene var slik at det ble tydelig for meg hva de hadde tenkt mens de løste oppgavene. Måten elevene delte oppgavene med meg var at de sendte skjermbilder av oppgavene til meg på e-post. Dette gjorde at jeg var i stand til å gjøre en observasjon av selve oppgaveløsningen fra samtlige elever i etterkant av undervisningen. Dette lettet på noe av tidspresset som jeg hadde i klasserommet som både underviser og observatør.

### 3.7 Validitet, reliabilitet og overførbarhet

Validitet og reliabilitet er vanskelig i all forskning, særlig kvalitativ: «To attain absolute validity and reliability is an impossible goal for any research model» (Le Compte & Goetz, 1982, i Gustavsen, Hinna, Borge & Andersen, 2014). Vi kan aldri oppnå en absolutt validitet eller reliabilitet da all metode innenfor forskningen har sine begrensninger.

#### 3.7.1 Validitet

I forskningen så må validiteten til forskningen vurderes ut ifra hvilke begrensninger det er knyttet til egen forskning (Postholm & Jacobsen, 2018). Jeg har et relativt lite utvalg, men prosjektet kan fortelle noe om hva elevene opplever i møte med programmering i matematikk. Min forskning er rettet mot niende og tiende trinn, så min forskning vil kunne si noe om elever ved niende og tiende trinn, men den kan kanskje overføres til andre klassetrinn. Svakheten her er at det kun var elever ved niende trinn som programmerte med visuell programmering og kun elever fra tiende trinn som programmerte med tekstbasert programmering. Validiteten kunne vært styrket om jeg mikset elevene slik at klassetrinn ikke hadde kunnet påvirke funnene. En annen måte å styrke validiteten på er å gjenta prosjektet en stund etter at den første gjennomføringen er gjort.

Det er betryggende å få like resultater om vi gjentar en måling med like betingelser og for like resultater. En slik måling er repeterbart og reproducerbart, noe vi uttrykker som reliabilitet

(Pripp, 2018). Dette gjelder og funn i kvalitative studier. Om funnene samsvarer over en viss tidsperiode så kan man med større sikkerhet si at de opprinnelige funnene hadde en større grad av validitet (Postholm & Jacobsen 2018). Begrensingen for mitt prosjekt var tidsbegrensning, så jeg hadde ikke den muligheten til å gjenta prosjektet etter at den opprinnelige studieperioden var over.

### **3.7.2 Relabilitet**

For at andre skal kunne ettergå min forskning skal den være så transparent som mulig. Dette er så andre kan reprodusere eller reteste mine resultater. Innenfor kvalitativ forskning kan en slik reproduksjon av studiet gi en annen resultat enn tidligere studier. Dette betyr ikke nødvendigvis at tidligere forskning feiltolket resultater, men i forskning på mennesker kan betingelsene endres. Slike betingelser kan være samfunnsbasert eller forskerens påvirkning på deltakere. Samhandlingen mellom forskere og deltakere kan utarte ulikt fra studie til studie og påvirke resultatet. I kvalitativ forskning knytter vi relabilitet til refleksjon over hvordan utførelse av studiet og forskeren selv kan påvirke funnene, og dette krever to ting: Forskeren reflekterer over sin egen påvirkning, og at forskeren synliggjør forskningsprosessen slik at andre kan reflektere over dette (Postholm & Jacobsen, 2018, s. 223-224).<sup>11</sup> I min studie har jeg som forsker også vært elevenes faste matematikklærer gjennom skoleåret, noe som kan ha påvirket deltakerne. En annen faktor er at betingelsene kan endres over tid. Nå er vi helt i startfasen med å bruke programmering i matematikkfaget, men om noen år er det en innarbeidet del av undervisningen. Så dette gjør at funnene i framtiden kan være annerledes enn funnene som ble gjort i min studieperiode, en samfunnsmessig endring der elevene er mer vant til å bruke programmering innenfor matematikkfaget. Innenfor kvalitativ forskning så vil funnene kunne endre seg da både mennesker, betingelser og samfunn endrer seg over tid.

### **3.7.1 Overførbarhet**

Dette er en liten studie med få deltakere i et begrenset geografisk område. Dette gjør at studien ikke kan regnes som en statistisk generalisering som vil kreve at det er gjort et utvalg basert på tilfeldig trekning blant alle som inngår i en populasjon (Postholm & Jacobsen, 2018) og mine funn kan derfor ikke sies å være allmenngyldige. For at funnene skulle kunne

---

<sup>11</sup> Fra prosjektskissen

generaliseres og regnes som allmenngyldige så ville det vært nødvendig å gjøre en forskning på et tilfeldig utvalg i en populasjon. Mitt utvalg var elever som hadde gitt samtykke ved en enkelt skole.

Funnene kan derimot ha en gyldighet innenfor naturalistisk generalisering i form at mine funn kan være gjenkjennbart for den som leser studien (Postholm & Jacobsen, 2018). Dette betyr at studien min ikke er en allmenngyldig beskrivelse av fenomenet jeg har forsket på, men kan ha en overføringsbetydning for lærere eller forskere som studerer samme eller lignende fenomen.

I min studie tar jeg for meg programmering, men det finnes andre teknologiske hjelpemidler i matematikkundervisningen som kan ha lignende betingelser, som for eksempel programmet GeoGebra der vi kan bruke variabler. Om man velger å bruke denne studien må man derimot være bevisst på begrensingene studien har i overføringsverdi for å bruke studien på en kritisk og god måte.

### 3.8 Et kritisk blikk på studien

Min for forståelse av det jeg skal studere påvirker meg i min datainnsamling. Når det gjøres observasjoner, vil dette være påvirket av mine tidligere erfaringer og forståelse. En annen forsker ville kanskje gjort andre observasjoner enn det jeg gjorde i klasserommet (Christoffersen & Johannesen, 2012). Da jeg var alene om å gjøre observasjoner er alle observasjoner gjort fra mitt ståsted og vil derfor påvirke funnene. En mulighet for å omgå noe av denne påvirkningen kunne være å ha flere observatører slik at flere øyne hadde observert hva som skjer i klasserommet. Ulempen med en slik løsning ville være introduksjon av noe nytt i læringsrommet som kunne påvirke elevenes væremåte under studien, særlig i starten før elevene hadde blitt vant til observatøren i klasserommet.

Forskningens kvalitet er avhengig av en dialog om substansen i funnene som er gjort. I tillegg må forskeren reflektere over to forhold:

- a) Hvilke begrensninger er knyttet til forskningen?
- b) Hvordan har forsker påvirket resultatet med gjennomføringen av forskningen?

Punkt a omhandler forskningens gyldighet, det betyr, hvilke konklusjoner har egentlig forskeren dekning for å trekke ut ifra datamaterialet. Punkt b handler om forskningens pålitelighet. I hvor stor grad kan vi stole på funnene i forskningen? Er forskningen gjort på en

god måte? Kan vi stole på det informanter sier i et intervju? Og har forskeren fått med seg all viktig informasjon? (Postholm & Jacobsen, 2018). Om gyldigheten i min forskning så må det påpekes at datamateriale er begrenset da det var få deltakere i studien, så konklusjonen kan ikke sies gi en statistisk generalisering om temaet oppgaven tar for seg. På forskningens pålitelighet så kan vi først se på informantene. Begge informanter har jeg en god relasjon til noe som kan være med på å gi sannferdige svar fra informantene. For å kunne se om forskningen er gjort på en god måte så har jeg lagt fram det som er relevant for studien, metode, teori og valg av analyse. På spørsmålet om jeg som forsker har fått med seg all informasjon så var det en begrensning at min observatørrolle var absolutt deltakende og måtte noen ganger rette fokuset på å hjelpe elevene.

En annen faktor som påvirker funnene, er tidsaspektet. Prosjektet varte i to uker så alle funn ble gjort innenfor en kort tidsramme. Dette gir en svakhet i studien da elevene i prosjektperioden møtte noe nytt og spennende og var positiv av nyhetens interesse. Kanskje ville elevenes engasjement og motivasjon endret seg ved en lengre periode med programmering. Om studien hadde blitt gjentatt med noen ukers mellomrom fra første studien, og funnene hadde samsvart ville studien ha hatt større relabilitet (Christoffersen & Johannesen, 2012). For at jeg skulle kunnet dykke dypere i materien som jeg forsker på så ville det krevdes en lengre forskningsperiode. Min begrensning i tid ga meg et lite rom for å samle inn data, så alle funn er basert på kun to uker. En forskning som hadde gått over en lengre periode ville krevd mere resurser og en erfaren forsker for å kunne gjennomført en såpass krevende studie.

Prosjektet hadde få deltakere noe som gjør at det er mindre datamateriale å basere seg på. Dette gjør at funnene kan slå litt skjevt ut om det er noen ytterpunkter, en større gruppe ville ha jevnet ut funnene og gjort at man kunne med større sikkerhet generalisert funnene. For å kunne ha hatt en statistisk generalisering måtte jeg ha valgt ut tilfeldige deltakere fra en populasjon (Postholm & Jacobsen, 2018) noe som ville krevd en langt større studie med flere forskere. Derfor må mine funn sees i lys av at det er en geografisk begrenset studie med få deltakere.

En annen viktig betraktning er at jeg måtte få deltakere fra to forskjellige klassetrinn for å få nok deltakere til studien. En tiendeklassing har et år mer skolegang enn en niendeklassing og

kan derfor ha mer erfaring i oppgaveløsning. En annen svakhet er at det var kun niendeklassinger som hadde visuell programmering og kun tiendeklassinger som hadde tekstbasert programmering. Dette kan også påvirke mine funn. Da særlig med tanke på at elever i en tiende klasse har et år mer erfaring med oppgaveløsning enn det en niende klasse har. Da oppgavene var lik for begge grupper hadde gruppen som arbeidet med tekstprogrammering et fortrinn sammenlignet med de som drev visuell programmering. Dette kan ha påvirket mine funn.

Målet med denne kritiske refleksjonen er å påpeke at det finnes svakheter i min studie, dette gjør jeg for å være så transparent som mulig rundt min forskning.

### 3.9 Etske betraktninger

Jeg må være bevisst min rolle som forsker og som lærer. Jeg er fast matematikklærer i klassene jeg skal forske på og må derfor være bevisst på hvordan dette spiller inn. Når jeg forsker på mine egne elever, er de i kjente omgivelser med de normale undervisningsrammene. Dette kan være en fordel da det ikke introduseres andre elementer inn i undervisningen. Ulempen er at jeg som forsker kan oppleve utfordring med objektivitet, det kan være mer utfordrende å se på deltakersvar med objektive briller.

De forskningsetiske retningslinjene kan i hovedsak sammenfattes i tre typer hensyn forskeren må ta: Informantens rett til selvbestemmelse og autonomi, forskerens plikt til å respektere informantens privatliv og forskerens plikt til å unngå skade (Nerdrum, 1998, i Christoffersen & Johannessen, 2018).

Deltakerne i studien skriver under informert samtykke som opplyser hva prosjektet går ut på og at det er frivillig å delta. Prosjektet er meldt til DNS og det er søkt om å få ta lydopptak. Det tas hensyn til personvernet til deltakerne gjennom anonymisering av transkripsjonen av intervju. Deltakerne skal få lese gjennom transkripsjonen for å om de er forstått rett. (Postholm & Jacobsen, 2018). Deltakerne kan når som helst trekke samtykket.

Etske overveielser i observasjonen er: Hva skal observeres? Hvilken hensikt har observasjonen? Hvor skal det observeres? Hvordan skal det observeres? I hvilken utstrekning skal det observeres? Hvordan skal observasjoner brukes og oppbevares? (Bjørndal, 2017).

Med tanke på disse overveielsene skal jeg observere elever som jobber med programmering i et klasserom. Hensikten er å lære hva elever opplever i dette arbeidet og det skal brukes et

observasjonsskjema med observasjonskategorier. Det blir ikke notert ned noe som kan være personidentifiserende. Disse observasjonene skal oppbevares slik at andre ikke kan lese den direkte observasjonen jeg har gjort, og i etterkant skal disse analyseres og originalobservasjonene makuleres.<sup>12</sup>

Det er også gjort en betraktning om personvern knyttet til bruk av teknologiske løsninger som elevene har brukt i prosjektperioden (Bergsjø et. al., 2020). Jeg tok en avgjørelse om å ikke opprette bruker for elevene ved bruk av trinkets løsning da det krevde navn og e-postadresse. Dette gjorde at det ikke var mulig å lagre arbeidet, men jeg fant en akseptabel løsning som gjorde at vi kunne ta vare på arbeidet uten å lage en brukerprofil hos trinket.

---

<sup>12</sup> Fra min projektskisse

## 4. Funn

Mine funn fra elever baserer seg på observasjoner i klasserommet og intervju etter endt studie. Funnene er delt inn i to hovedkategorier, 4.1 Observasjoner, og 4.2 Intervjuer. Et annet funn som ikke handlet om elever, men min egen kompetanse og digital etikk baserer seg på observasjon og refleksjon og går under 4.3 Andre funn.

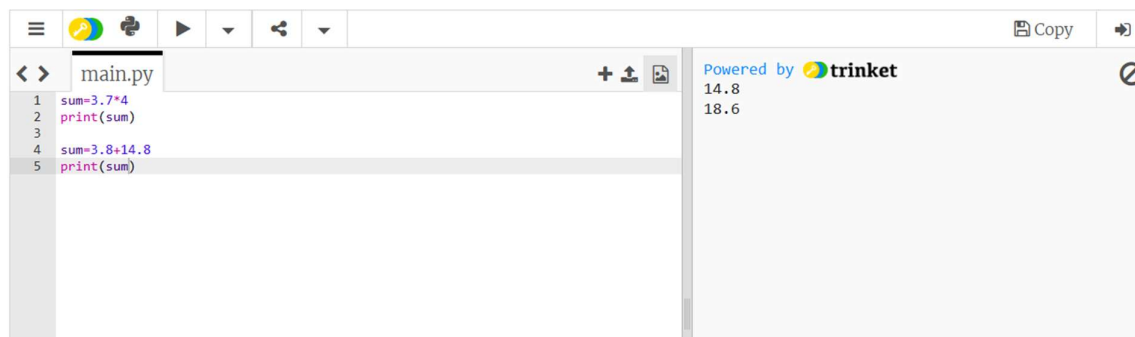
### 4.1 Observasjoner

Observasjonene ble skrevet ned med hjelp av et observasjonsskjema der jeg skrev ned stikkord. Straks etter endt time ble disse skrevet ned i et Word-dokument med hele setninger.

Funnene fra observasjon er delt inn i underkategoriene: *Fra kalkulator til variabler, Samarbeid i og på tvers av grupper, Bruk av kommentarer i programmet, Feilsøking, Motivasjon og Tidsbruk, en forskjell på Python og blokk*

#### 4.1.1 Fra kalkulator til variabler

Første time av prosjektet startet jeg med gruppen som skulle programmere med tekst. Første oppgave var: «Jan kjøpte 3,7 hekto druer, Thomas kjøpte fire ganger så mye druer som Jan. Hvor mye kjøpte de til sammen?». Elevene løste dette med å regne steg for steg der de brukte Python som en kalkulator der de selv regnet ut stegene (se bilde under). Da de hadde besvart oppgaven endret jeg oppgaven til at «Jan kjøpte 5.3 hekto druer, hvor mye blir det da til sammen?». Elevene brukte samme metode for å løse denne oppgaven også.



```
main.py
1 sum=3.7*4
2 print(sum)
3
4 sum=3.8+14.8
5 print(sum)
```

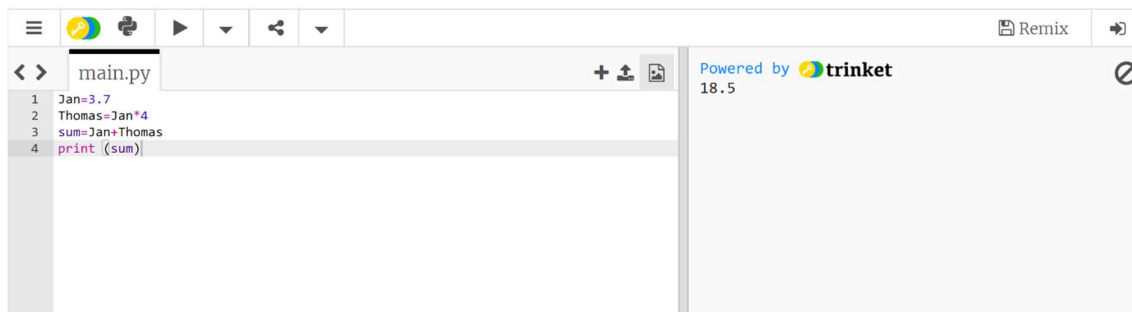
Powered by trinket  
14.8  
18.6

Da spurte jeg om det er en annen måte å programmere på slik at vi kan endre verdien uten å måtte regne ut hele oppgaven på nytt.

Etter en samtale elevene mellom kom det forslag om å bruke variabler, men elevene husket ikke hvordan lage variabler i Python og spurte meg om de kunne få en oppfriskning på bruk



av variabler. Vi tok en kort repetisjon på variabler i Python og elevene ga uttrykk for at det var mye enklere å gjøre endring av verdier ved bruk av variabler.

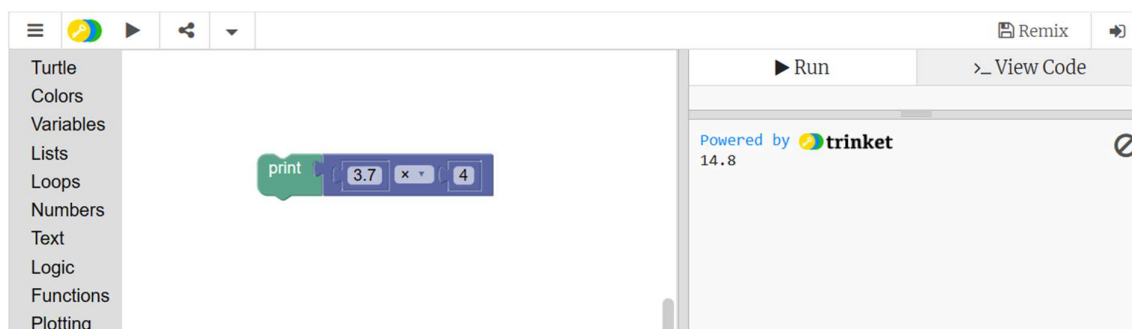


```
1 Jan=3.7
2 Thomas=Jan*4
3 sum=Jan+Thomas
4 print (sum)
```

Powered by trinket  
18.5

Fordelen med Python er at det er enkelt å lage en variabel. Det er bare å skrive for eksempel et navn etterfulgt av likhetstegn og gi den en verdi. Utover i prosjektperioden brukte Python-gruppen variabler aktivt.

Gruppen som programmerte med visuell programmering fikk samme oppgave: «Jan kjøpte 3,7 hekto druer, Thomas kjøpte fire ganger så mye druer som Jan. Hvor mye kjøpte de til sammen?».

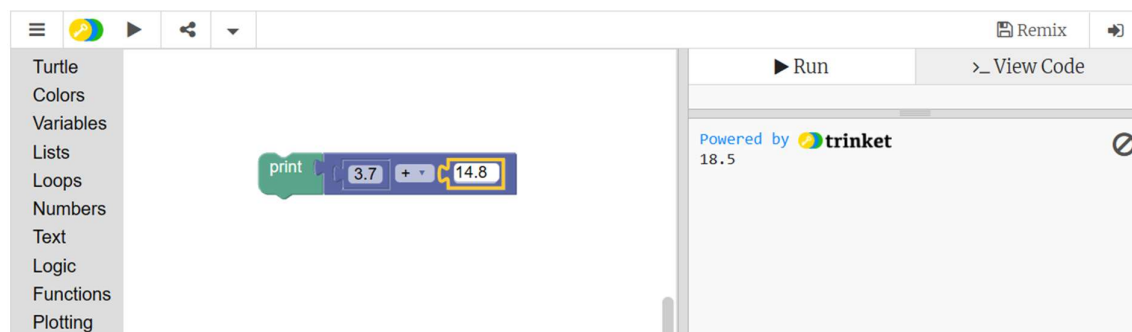


Turtle  
Colors  
Variables  
Lists  
Loops  
Numbers  
Text  
Logic  
Functions  
Plotting

print 3.7 × 4

Powered by trinket  
14.8

De kjørte koden og fikk 14,8. Så brukte de de samme blokkene, men endret tallene og gangetegn med plusstegn (man kan velge tegn ved å trykke på nedpilen ved plusstegnet).



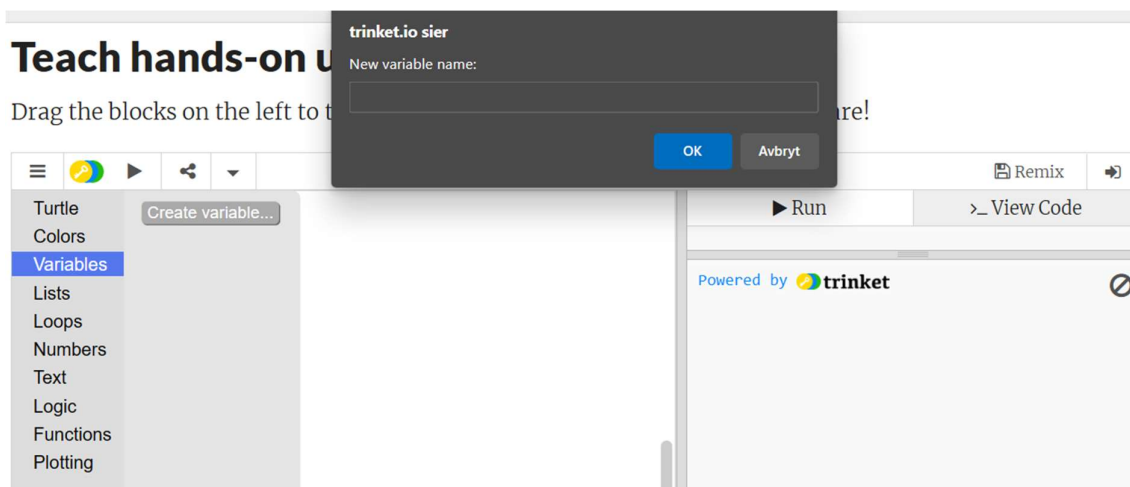
Turtle  
Colors  
Variables  
Lists  
Loops  
Numbers  
Text  
Logic  
Functions  
Plotting

print 3.7 + 14.8

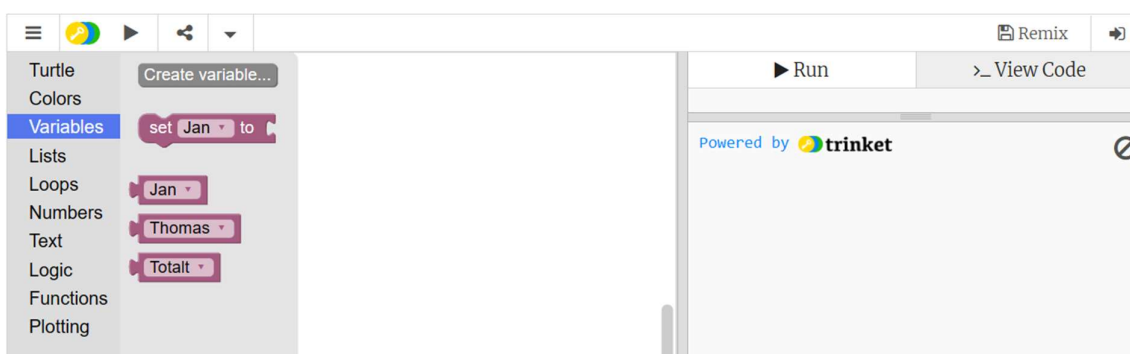
Powered by trinket  
18.5

Jeg endret oppgaven til at Jan kjøpte 5.3 hekto druer. Elevene løste dette med å endre tallene i blokken og regnet ut på samme måte. På spørsmål om det er andre måter vi kan programmere dette slik at vi kan endre verdien uten å gjøre hele regnestykket kom denne gruppen også fram til at vi kan bruke variabler. Elevene var usikre på hvordan lage variabler i blokkprogrammering og vi hadde en kort felles gjennomgang på hvordan lage variabler.

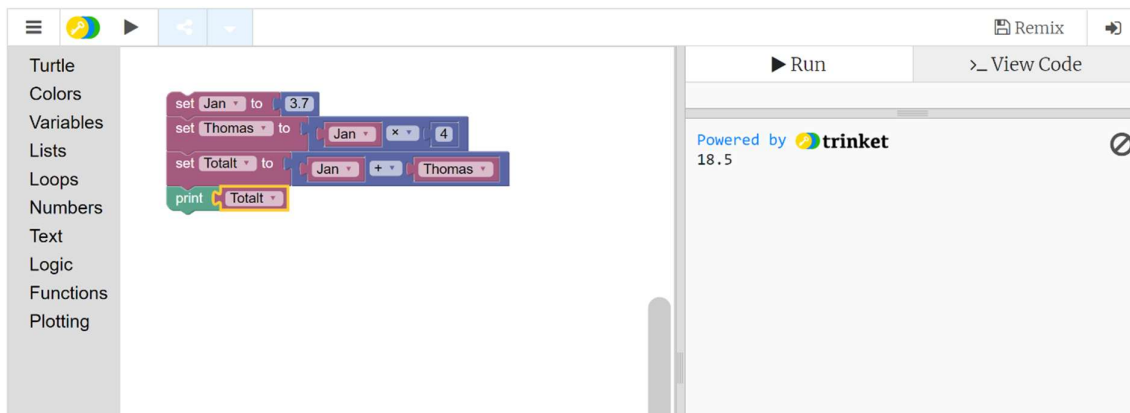
Det å lage variabler i blokkprogrammering (i trinkets løsning, kan variere fra utvikler til utvikler) er mer arbeid sammenlignet med Python. Først må det lages en variabel for hver variabel som skal brukes.



Når variablene er lagd kan disse trekkes inn i programmeringsfeltet og gis en verdi. For å endre «set\_to» blokken fra Jan så trykker man på pilen ved navnet og får opp alle variablene man har laget.



Elevene kom fram til denne løsningen på oppgaven

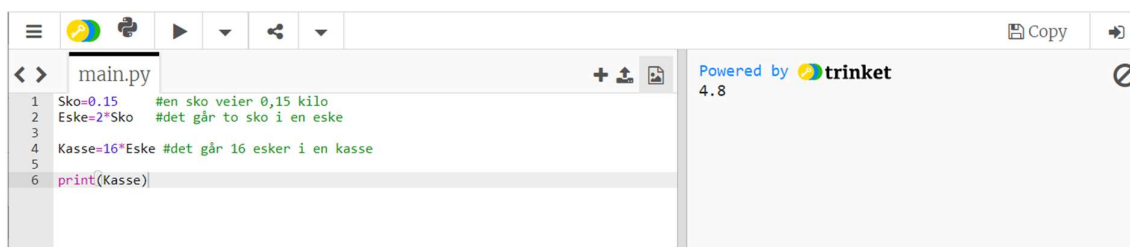


Elevene i blokkprogrammeringsgruppa ga uttrykk for at det var litt tungvint å bruke variabler og at det var lettere å bare endre verdi i en utregningsblokk slik de hadde gjort først. Utover i prosjektperioden ble elevene vant til å lage variabler og syntes det ble lettere å bruke når de først hadde blitt trygg på det, men ga uttrykk for at det ofte var enklere å regne stegene i en regneblokk.

#### 4.1.2 Bruk av kommentarer i programmet

Underveis i prosjektet ble elevene bedt om å legge inn kommentarer på framgangsmåten deres slik at det kom tydelig fram hvordan de tenkte når de gikk fram for å løse oppgaven.

En vesentlig forskjell mellom tekstbasert og visuell programmering er mulighet for bruk av kommentarer I Python er det enkelt å sette inn kommentarer ved å bruke tegnet # (firkant/hashtag/sharp). Om man ønsker å sette inn en kommentar, for eksempel forklaring på de forskjellige stegene så er det bare å skrive inn # etterfulgt av kommentaren. Kommentaren er ikke en del av programmet og har derfor ikke innvirkning på selve programmet, men gjør det lettere å forklare tanken bak stegene samt bruke tidligere programmer som en mal for å lage nye programmer (se bildet under).



I visuell programmering (trinkets løsning, andre leverandører kan ha andre løsninger) så lot det seg ikke gjøre å sette inn kommentarer slik som i Python. Elevene løste dette med å sette inn en tekstblokk som ikke var koblet til en programkjøringsblokk (print-blokk etc.). Når tekstblokken er frittstående (grønn blokk med anførselsteg) så er den ikke en del av programmet og påvirker derfor ikke kjøringen av programmet, men gir elevene mulighet til å sette inn kommentarer.



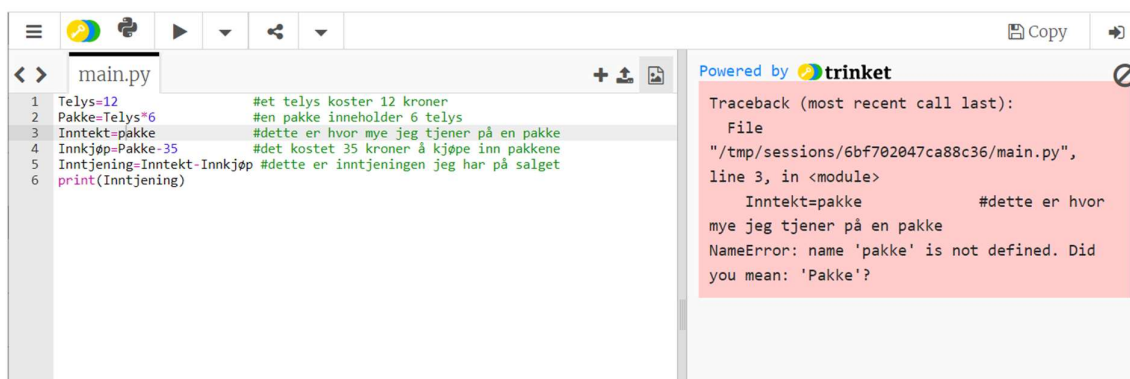
Dette gjorde at det var lettere å kommentere stegene i Python enn i blokkprogrammering.

### 4.1.3 Feilsøking

Som vist i tabell 1 (Bocconi, S., Chiocciariello, A. og Earp, J., 2018) så er feilsøking en del av algoritmisk tankegang. Her ble det observert en forskjell mellom de to metodene for programmering.

I visuell programmering så er det ferdige blokker med ferdige kommandoer, og operatorene passer kun inn i korrekt rekkefølge som puslebrikker. Dette gjorde at elevene i denne gruppen ikke hadde feil i syntaksen i koden (syntax error). Syntaksen er språkoppybyggingen og rekkefølgen i programmet og i visuellprogrammering passer syntaksen inn kun på en måte som gir riktig syntaks. Utfordringene de møtte på var om de hadde gjort de forskjellige operasjonene i riktig rekkefølge, det vil si forstått og brutt ned oppgaven slik at rekkefølgen på utregningene ble riktig. I Andersens (2022) funn der elevene opplevde elevene en del feilsøking, gjerne grunnet at små detaljer var oversett (Andersen, 2022). Forskjellen i mine funn kontra Andersens funn bunner muligens i oppgavedesignet, der Andersens legger opp til flere mulige feilkilder og dermed gir elevene flere muligheter til feilsøking i blokkprogrammering.

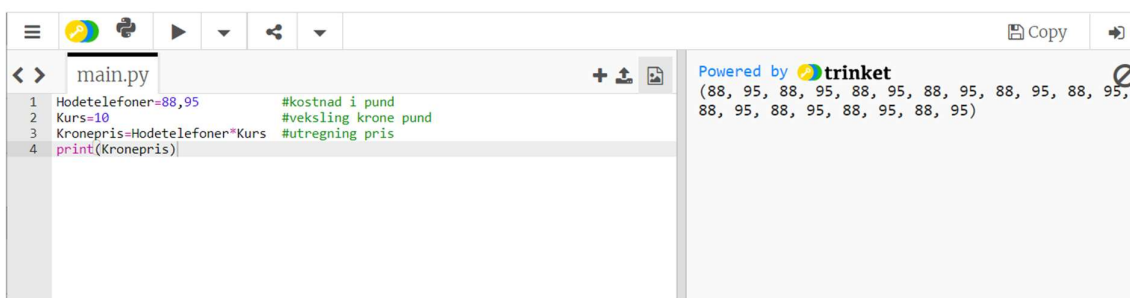
Tekstbasert programmering ga flere utfordringer med feil i koden. Her møtte elevene feil med syntaksen (syntax error) og uventete tallrekker når programmet kjørte. Den vanligste feilen elevene møtte på var feilstaving av variabler, det vil si at de hadde definert en variabel i en kodelinje og når de skulle bruke denne variabelen i en annen kodelinje så var det forskjell på stavingen. Den feilen som gikk igjen mest var at de definerte en variabel i ene linjen med stor forbokstav, men når denne skulle brukes i en annen del av programmet skrev de variabelen med liten forbokstav (se eksempelbildet under, feilen er i linje 3).



```
1 Telys=12 #et telys koster 12 kroner
2 Pakke=Telys*6 #en pakke inneholder 6 telys
3 Inntekt=pakke #dette er hvor mye jeg tjener på en pakke
4 Innkjøp=Pakke-35 #det kostet 35 kroner å kjøpe inn pakkene
5 Inntjening=Inntekt-Innkjøp #dette er inntjeningen jeg har på salget
6 print(Inntjening)
```

Traceback (most recent call last):  
File  
"/tmp/sessions/6bf702047ca88c36/main.py",  
line 3, in <module>  
Inntekt=pakke #dette er hvor  
mye jeg tjener på en pakke  
NameError: name 'pakke' is not defined. Did  
you mean: 'Pakke'?

En annen vanlig feil elevene møtte på var uventede tallrekker når de kjørte programmet sitt. Årsaken til denne feilen var at elevene brukte komma når de skulle skrive desimaltall. I Python brukes punktum for å skille enerplass fra tidelsplass, mens komma er brukt for å skille tallene i en liste av tall. Derfor fikk elevene en lang liste med tall som gjentok seg når de brukte komma i verdien på en desimaltallsverdi på en variabel (se bilde under). Denne feilen dukket opp kun når det tallet desimaltallet skulle multipliseres med var et heltall, om begge tallene var skrevet som desimaltall med komma dukket det opp en feilmelding. Denne feilen var derfor mest fremtredende på de oppgavene med lavere vanskelighetsgrad



```
1 Hodetelefoner=88,95 #kostnad i pund
2 Kurs=10 #veksling krone pund
3 Kronepris=Hodetelefoner*Kurs #utregning pris
4 print(Kronepris)
```

(88, 95, 88, 95, 88, 95, 88, 95, 88, 95, 88, 95, 88, 95, 88, 95, 88, 95, 88, 95, 88, 95)

I starten av prosjektet trengte elevene en del hjelp fra lærer for å finne ut hva som var feil i programmet, men lærte etter hvert å lese feilkoden og finne ut hvilken linje koden lå i. De ble også klar over at uventede tallrekker var et resultat av bruk av komma i stedet for punktum i en variabel.

#### ***4.1.4 Tidsbruk, en forskjell på Python og blokk***

En forskjell som jeg observerte i perioden, var hvor mange oppgaver de to gruppene klarte å løse på en klokke time. Begge gruppene løste samme oppgaver, men gruppen som programmerte i Python løste flere oppgaver i løpet av timen enn gruppen som programmerte i blokk. Python-gruppen løste 1-3 oppgaver mer enn blokk-gruppen i løpet av en time. Særlig oppgaver med flere variabler tok mer tid å løse i blokk. Det var enkelt for meg å sjekke antall løste oppgaver underveis da elevene delte skjermbilder av oppgavene med meg på e-post.

#### ***4.1.5 Samarbeid i og på tvers av læringspar***

Gjennom observasjon så kom det tydelig fram at det var mye samtaler innad i læringsparene om hvordan de kunne løse oppgavene. De kom med ideer og prøvde ut flere løsninger.

Det var også mye samtaler og samarbeid på tvers av læringsparene. Elevene gikk rundt og sammenlignet løsningene de hadde kommet fram til og førte samtaler om de forskjellige metodene for å løse oppgaven. Dette gjaldt for både tekstbasert og visuell programmering. På slutten av timen sammenlignet vi løsninger i plenum og hadde en diskusjon med hele gruppen om de forskjellige løsningene. Samtlige elever bidro i samtaler og faglige diskusjoner

#### ***4.1.6 Motivasjon***

Gjennom hele prosjektperioden ga elevene i begge gruppene uttrykk for at de syntes programmering var spennende å jobbe med, dette kom fram i samtaler med elevene underveis i perioden. Gjennom observasjon så jeg at elevene hadde godt samarbeid med en god tone seg imellom hele prosjektperioden. Alle jobbet godt i lag uavhengig av hvem de hadde som læringspartner. Det var god arbeidsinnsats gjennom hele perioden og elevene i begge grupper spurte om vi kunne jobbe mer med programmering i løpet av året.

## 4.2 Intervjuer

Det ble gjennomført intervju med en informant fra hver gruppe, en som programmerte med tekst, og en som programmerte visuelt. Intervjuet var semistrukturert med en intervjuguide. På mange av spørsmålene hadde informantene sammenfallende svar. Jeg vil ha størst fokus på spørsmålene der svarene ikke var sammenfallende, eller hadde en grad av forskjell. Jeg vil gjengi spørsmålene etterfulgt av funnene fra intervjuene. For å gjengi svarene, samt skille mellom informantene så kaller jeg eleven som hadde programmert med Python for Elev Tekst, og den eleven som hadde programmert med visuell programmering for Elev Blokk.

Se intervjuguide for spørsmålene som ble stilt (vedlegg), noen av svarene førte til oppfølgingsspørsmål. Nederst i avsnittet ligger det en tabell som viser funn.

### 4.2.1 *Generelle spørsmål om programmering*

Først stilte jeg noen innledningsspørsmål om programmering (generell del)

Begge informanter beskrev prosjektperioden som spennende og lærerikt. De gir positive beskrivelser av perioden prosjektet har pågått. De opplever det motiverende å jobbe med programmering og begge ønsker mer programmering i løpet av skoleåret. Informantene sitter igjen med et positivt inntrykk av å bruke programmering i matematikkfaget og tenker at det er nyttig for læring innenfor matematikk. Begge svarer at algebra er et tema hvor programmering er et godt læringsverktøy. Begge har en positiv opplevelse av å drive med programmering.

Begge informanter svarer at programmering kan brukes til å løse matematiske oppgaver, mens Elev Blokk i tillegg beskriver hvordan programmering kan brukes til styring av systemer, robot og lignende, hvor bruk av variabler er nyttig.

Begge informanter synes at programmering hører hjemme i matematikkfaget og ønsker mer programmering i undervisningen framover.

### 4.2.2 Programmering og variabler

På spørsmålene som omhandler bruk av variabler i programmering (variabel del) var det noen forskjeller i svarene fra informantene.

På spørsmål om hvordan det er å lage variabler når vi programmerer Svarte Elev Tekst at det var lett å lage variabler, man skriver bare et navn, likhetstegn og en verdi.

Elev Blokk svarte at det var litt tungvint. Først må man lete variabel i menyen og velge ny variabel å gi det navn.

På spørsmål om hvordan det er å bruke variabler svarte Elev Tekst at det var enkelt å bruke når man først har lært seg det. Det er viktig å passe på at man skriver riktig, hvis man har brukt stor forbokstav i variabelen må man huske å bruke stor forbokstav når man skal gjøre utregning. Variabler gjør at det er lett å endre verdiene. Går mye raskere å programmere når man bruker variabler, effektivt.

Elev Blokk svarte at det var greit når det er få variabler. Det blir uoversiktlig hvis man har mange variabler. Det er lett å bruke feil variabel når man må velge hvilke variabler som skal stå i blokken (*variabler velges ved en rullegardinmeny der man velger hvilket navn som skal stå i variabelblokken*). Det er ofte enklere å regne ut stegene selv i en regneblokk.

### 4.2.3 Motivasjon og samarbeid i programmering

På spørsmål om hvordan motivasjonen har vært i prosjektperioden svarte Elev Tekst at motivasjonen har vært god. Det har vært spennende, litt utfordrende å jobbe med programmering. I starten var det noen ganger litt frustrasjon når programmet ikke ville virke, men etter hvert så ble det lettere.

Elev Blokk svarte at det har vært motiverende å jobbe med programmering. Det har vært morsomme mattetimer mens vi har jobbet med programmering.

På spørsmål om hva som har gjort perioden motiverende svarte Elev Tekst at det er gøy å lære noe nytt og få det til. Alle i gruppen har vært positive og hjulpet hverandre. Det var motiverende når man fikk programmet til å fungere som det skulle.

Elev Blokk svarte at det er fint å gjøre noe annet enn det vi gjør til vanlig. Fint med variasjon fra vanlig matteundervisning. Så er det fint å jobbe på PC



På spørsmål samarbeid i prosjektperioden svarte begge informantene at de opplevde godt samarbeid i prosjektperioden. De opplevde at alle bidro til å løse oppgaver og syntes det var fint å variere hvem som samarbeidet i læringspar.

#### ***4.2.4 Knytte programmering til annen matematikk***

På spørsmål om de har lært noe matematikk i prosjektperioden svarte Elev tekst at selve matten vi brukte kan jeg fra før, så det ble mer en repetisjon på det som vi har lært tidligere. Det er greit å få repetert matte med litt programmering. Programmering er jo litt praktisk matte, og praktisk matte er morsomt.

Elev Blokk svarte at det har vært læring. Mye kunne jeg fra før, men noe av matten var litt glemt og det var nyttig å få repetert litt. Programmering er mye som matte, man må gjøre ting steg for steg.

På oppfølgingsspørsmål om hvor vi bruker variabler i matematikk svarte Elev Tekst: Algebra, der har vi brukt variabler. Vi regnet med bokstaver og skulle finne ut hvor mye det var verdt. Elev Blokk svarte: Vi har jo jobbet med variabler i algebra der vi regnet med A og B og sånt.

På spørsmål om de har bedre forståelse for variabler i algebra etter prosjektperioden svarte Elev Tekst ja, jeg har lært om variabler på en litt annen måte enn da vi jobbet med algebra. Jeg skjønner bedre hvorfor vi skal kunne algebra.

Elev Blokk svarte ja. Forstår variabler bedre etter å ha programmert. Syns jeg har lært det bedre med programmering enn når vi jobbet i boka.

#### ***4.2.5 Tabell***

Tabellen viser funnene i intervjuet og rangerer svarene som positiv, nøytral eller negativ, samt en kommentar i tabellen. Merk at jeg ikke gjengir spørsmålene, men jeg har delt opp spørsmålene i temaer og meningsblokker, så noen av spørsmålene er slått sammen. Jeg har valgt ut noen sitater fra intervjuene og lagt inn i tabellen.

	Elev blokk	Elev Tekst	Kommentar
Opplevelse av programmering generelt	Positiv «Spennende, morsomt»	Positiv «Gøy, interessant, noe nytt»	Samsvarer
Motivasjon under prosjektperioden	Positiv «Motiverende»	Positiv «Spennende»	Samsvarer
Samarbeid under prosjektperioden	Positiv «Alle bidro»	Positiv «Godt samarbeid på alle sammen»	Samsvarer
Bruk av variabler i programmering	Nøytral «Litt tungvint.» «Blir uoversiktlig med mange variabler. Er ofte enklere å regne ut stegene selv»	Positiv «Enkelt å bruke når man først har lært det»	Differensierer. Blokkløsningen oppleves som tungvint, særlig med mange variabler.
Bruk av kommentarfunksjonen	Positiv «Var litt knotete, men er fint å skrive ned hva man gjorde så man kan bruke skjermbildet til hjelp for andre programmer»	Positiv «Kan være nyttig for å huske hva de forskjellige stegene gjorde»	Samsvarer Blokkløsningen var litt knotete
Lære matte gjennom programmering	Positiv «Programmering er mye som matte, man må gjøre ting steg for steg»	Positiv «Programmering er jo litt praktisk matte, og praktisk matte er morsomt»	Samsvarer Begge opplever programmering som nyttig i læring av matematikk
Knytte variabler i programmering til annen matematikk (obs, kun svar, ikke positivt, nøytral eller negativ)	«Vi har jo jobbet med variabler i algebra»  «Forstår variabler bedre etter å ha programmert»	«Algebra, der har vi brukt variabler»  «Jeg skjønner bedre hvorfor vi skal kunne algebra»	Begge knytter variabler opp mot algebra

### 4.3 Andre funn

I løpet av min studie gjorde jeg noen andre funn enn det som gikk på elevenes opplevelser. Disse funnene mener jeg er viktig å ta med da disse kan ha hatt en innvirkning på funnene jeg gjorde blant elevene. Disse funnene baserer seg på min egen kompetanse innenfor programmering, og en hindring elever som programmerte med visuell programmering møtte på knyttet til det å kunne ta vare på arbeidet de hadde gjort.

### 4.3.1 Egen kompetanse

Underveis i prosjektperioden gjorde jeg et annet viktig funn som ikke gjaldt elever, men meg som underviser. En utfordring for meg var min egen kompetanse på programmering. I løpet av min studietid ble LK20 innført, og programmering var ikke på dagsorden i studien min de første to årene. På tredje studieår hadde vi litt om programmering, men mest om visuell programmering. Dette gjorde at jeg i forkant av prosjektet måtte sette meg inn i Python å lære meg mye grunnleggende funksjoner, og sette meg inn i trinkets løsning for visuell programmering. Dette er noe de fleste som nå jobber i skole blir å møte på som en utfordring. Det må settes av tid til at lærere skal få utviklet sin kompetanse innenfor programmering, men hensyn til tid og økonomi kan være et hinder for at dette gjennomføres i stor nok grad (Vinnervik & Bungum, 2022). I kunnskapskvartetten så er utgangspunktet for de andre påfølgende punktene *Foundation*, som er lærerens egen kunnskap og forståelse i matematikk. For å kunne lære videre til elevene så må eget kunnskapsgrunnlag være på plass. Da kan man gjennom *Transformation* videreformidle denne kunnskapen ved hjelp av eksempler eller spørsmålsstillinger. Videre så kan man trekke paralleller mellom de forskjellige delene av matematikken gjennom *Connection*. Og når man er trygg på det man skal undervise kan man på en god måte møte det uventede gjennom *Contingency*. I undervisningen kan det komme uventede spørsmål fra elever som man ikke kan forutse, men være forberedt på å møte med en god kunnskap om det som skal undervises i (Gustavsen et al., 2014)) Jeg brukte en del tid i prosjektperioden på å sette meg grundigere inn i de forskjellige funksjonene etter hvert som nye problemstillinger oppsto, eller elevene stilte spørsmål som jeg ikke kunne svare på der og da. Dette var utfordrende med tanke på studien da en del av tiden min gikk til å øke min egen kompetanse samtidig som jeg drev en undersøkelse.

### 4.3.2 Digital etiske utfordringer kan påvirke funn

Som nevnt tidligere så oppdaget jeg en utfordring knyttet til personvern i forkant av at prosjektet startet (Bergsjø et. al., 2020). Dette gjorde at det ikke ble opprettet brukerprofiler for elevene, noe som gjorde at det ikke var mulig å lagre arbeidet til elevene underveis. Med en brukerprofil ville elevene enkelt ha kunnet lagre arbeidet de gjorde underveis og bruke dette senere. For å forsøke å løse dette på en god måte så valgte jeg å be elevene ta skjermbilder av oppgavene. De som programmerte med tekst tok også og kopierte teksten i

programmet over i et tekstdokument. Dette gjorde at det var enklere for elever som programmerte med tekst å bruke tidligere arbeid som de hadde lagret da de kunne kopiere direkte fra tekstdokumentet over til Python. De som programmerte visuelt, måtte se på skjermbildet og rekonstruere det de hadde gjort tidligere i et tomt programmeringsfelt. Det innebar at de måtte på nytt lage alle variabler som de tidligere hadde brukt, noe som gjorde det mer arbeidsomt for denne gruppen. Dette aspektet er viktig å ha med seg når vi ser på de andre funnene som tyder på at det å bruke variabler i visuell programmering er vanskeligere enn tekstbasert programmering. Elevene som programmerte med tekst kunne enklere gjenbruke variabler enn elever som programmerte med visuell programmering. Dette kunne vært annerledes om det ble opprettet brukerprofil på trinkets løsning for å programmere, eller bruke andre løsninger enn trinket i programmeringen.

## 5 Drøfting

Forskningsspørsmålene jeg forsøker å finne svar på er

- 1) Er det forskjell på å jobbe med variabler i visuell programmering og tekstbasert programmering?
- 2) Trenger elevene mer støtte fra lærer når de arbeider med den ene metoden for programmering, tekstbasert eller visuell programmering?
- 3) Er det forskjell på å lære variabler i visuell programmering og tekstbasert programmering?

Underveis i hvert delkapittel er det en drøfting basert på teori. I slutten av hvert delkapittel er det en litt dypere drøfting der jeg forsøker og samle trådene.

### *5.1.1 Forskjell på å bruke variabler i visuell og tekstbasert programmering*

Det første forskningsspørsmålet mitt er: Er det forskjell på å jobbe med variabler i visuell programmering og tekstbasert programmering?

Ifølge Karlsen er variabler det mest brukt innenfor all programmering variabler (Karlsen, A. i Sund, 2023). Bruk av variabler er en naturlig del av algoritmisk tankegang når vi jobber med programmering. Det gjør det enklere å bryte ned en problemstilling til enklere bestanddeler og behandle disse. I norsk skole legges det opp til både tekstbasert og visuell programmering (Bocconi, S., Chiocciariello, A. og Earp, J., 2018) og det er derfor er det interessant å se nærmere på hvordan det er å jobbe med dette i de forskjellige metodene for programmering.

En vesentlig forskjell på å bruke variabler i tekstbasert programmering og visuell programmering er hvordan man lager variablene. I Python så kan man skrive navnet på variabelen etterfulgt av likhetstegn og angi verdi (Pilgrim, 2009). Elevene var raskt i gang med å bruke variabler og syntes det var greit å bruke variabler. De opplevde en del feil i programmet, gjerne forskjellig skrivemåte på en variabel fra en del av programmet til en annen (Zweigart, 2015), en annen feil som ofte dukket opp var bruk av komma når de skrev desimaltall. I Python brukes punktum for desimaltall og komma brukes for å lage en liste eller talrekke. Gjennom intervju kom det fram at informanten som programmerte med tekst tenker

at bruken av variabler var effektivt og gjorde det enklere å løse oppgaver, men man må passe på å bruke samme skrivemåte på variabelen hver gang den skal brukes.

I blokkprogrammering derimot må man først lage variabelen, trekke inn variabelblokken inn i programmeringsfeltet og gi den en verdi ved bruk av andre blokker. Gjennom observasjon så ble det observert at elevene som programmerte visuelt syntes det var tidkrevende og vanskelig å bruke variabler, særlig i starten, noe som ble bekreftet gjennom intervju. Programmer med mange variabler var ifølge informanten litt frustrerende og det er ikke alltid like lett å holde styr på de forskjellige variablene. Dette kan tyde på at bruk av variabler er mer effektivt og mindre arbeidskrevende i Python sammenlignet med blokkprogrammering.

En observasjon var forskjellen på bruk av kommentarer for å forklare fremgangsmåten i tekstbasert og visuell programmering. I Python er det en innebygd kommentarfunksjon, noe som ikke var i blokkprogrammering. Dette var noe informantene også kom med innspill på. Det var lettere å bruke kommentarer i Python sammenlignet med blokk da kommentarer i Python lages ved å skrive tegnet `#`(firkant/hashtag/sharp) etterfulgt av kommentaren, mens kommentarer i visuell programmering måtte løses ved å bruke en frittstående tekstblokk som ikke var påkoblet noen funksjonsblokker. Bruk av kommentarfunksjon gjør det lettere for lærer å se hvordan elevene har tenkt når de bryter ned oppgaven til mindre bestanddeler for å løse oppgaven, som er en del av algoritmisk tankegang (Bocconi, S., Chiocciariello, A. og Earp, J., 2018), samt at det fungerer som en notis som elevene senere kan bruke for å løse andre oppgaver da det forklarer stegene. Begrensningen for å kunne si noe generelt om dette er at jeg kun brukte løsninger fra en leverandør, trinket, og dette kan derfor variere fra andre leverandørers løsninger.

Som nevnt ovenfor så var det noen forskjeller på å jobbe med variabler i visuell programmering og tekstbasert programmering. En forskjell var å lage og bruke variabler, her opplevde elevene som programmerte med tekstbasert programmering variabler som enklere å bruke. Dette viser at forskjellige teknologiske løsninger påvirker hvordan det er å jobbe med i dette tilfelle variabler. Elevene som programmerte med visuell programmering opplevde bruk av variabler som mere tungvint enn å regne ut stegene selv. Dette kan føre til at de ikke i like stor grad ser nytteverdien av å bruke variabler i et program på samme måte som elevene som programmerte med tekstbasert programmering. Andre teknologiske løsninger som var mer tungvint i visuell programmering var bruken av kommentarer. Det var ikke noen innebygd

funksjon for dette i løsningen vi brukte, men elevene kom opp med en kreativ løsning på dette, de la inn en løs tekstblokk. Bruk av kommentarer kan være nyttig som en notis på hva de forskjellige variablene gjør, noe elevene kan overføre til andre oppgaver i ettertid. Begge gruppene brukte kommentarer i sine program etter at de fikk i oppdrag å bruke dette i innleveringen. Et annet aspekt med de forskjellige løsningene for programmering var feilsøking. I visuell programmering passer blokkene i lag som puslespillbrikker og det er ikke mulig å sette inn blokker som ikke passer inn i syntaksen (rekkefølgen på kommandoer). Blokkene kobler seg ikke på om de ikke er på rett sted. Derfor er feilkilden i visuell programmering i stor grad tolkning av oppgaven der elevene kan ha misforstått rekkefølgen i utregningen. I tekstbasert programmering så er det mange feilkilder i selve programmet. Det kan være feil i syntaksen, skrivemåter på variabler der de har brukt stor forbokstav i variabelen, men liten bokstav senere i programmet når de skal bruke variabelen. Elevene opplevde også feil knyttet til desimaltall der de brukte komma i stedet for punktum. Dette gjør at programmering med tekst gir mange flere muligheter for feilsøking underveis i programmeringen. Dette kan tyde på at tekstbasert programmering gir flere læringsmuligheter enn visuell programmering og viser at det er en forskjell på å jobbe med, i dette tilfellet variabler, mellom disse to formene for programmering.

For å konkludere så er det en forskjell på å jobbe med de to forskjellige måtene å programmere på. Det var både det å lage og å bruke variabler som skilte de to og tekstbasert programmering opplevdes som enklest å bruke. Elevene som programmerte visuelt manglet en innebygd kommentarfunksjon, men løste dette på en kreativ måte. Det var mest feilsøking i tekstbasert programmering.

### ***5.1.2 Lærer støtte i visuell og tekstbasert programmering***

Det andre forskerspørsmålet jeg stiller er: Trenger elevene mer støtte fra lærer når de arbeider med den ene metoden for programmering, tekstbasert eller visuell programmering?

I begge former for programmering trengte elevene opplæring i hvordan lage og bruke variabler når man programmerer, så dette var likt i begge grupper. Det var elevene selv som spurte om opplæring i hvordan lage og bruke variabler. I forkant av prosjektperioden måtte jeg selv sette meg grundig inn i hvordan man lager og bruker variabler i begge former for programmering, noe som var tidkrevende. Dette kan være en utfordring for mange lærere ut i

skolen da det ofte er lite tid til egenutvikling i en hektisk skolehverdag (Vinnervik & Bungum, 2022). En viktig faktor som påvirker læring gjennom digitale verktøy, er lærerens egen kompetanse og i hvor stor grad læreren klarer å inkludere disse i sin undervisning (Svanberg & Wille, 2009).

I gruppen som programmerte med tekst trengte elevene lærerstøtte i forbindelse med feilsøking i starten av prosjektet mens elevene som programmerte visuelt ikke fikk opp feilmeldinger. Dette gjorde at gruppen som jobbet med tekst fikk muligheten til å drive feilsøking i syntaksen eller uttrykk i programmet, mens elevene som jobbet med visuell programmering ikke fikk denne muligheten (Bocconi, S., Chiocciariello, A. og Earp, J., 2018). Feilsøking er en del av algoritmisk tankegang, så her har tekstbasert programmering en fordel framfor visuell programmering. Det gir flere læringsmuligheter for feilsøkingsdelen av algoritmisk tankegang. I den norske skole er det et stort fokus på fremtidsrettede ferdigheter som problemløsning, kreativitet og digitale ferdigheter (Vinnervik & Bungum, 2022). Elevene som programmerte med tekst lærte seg å lese feilkoden for å finne i hvilke linje av programmet feilen lå i, og deretter se om det var stavefeil. En feil som ikke ga feilkode, men et uventet svar var bruk av komma i stedet for punktum i desimaltall. Denne feilen ga en tallrekke i stedet for et enkelt tall. Elevene lærte raskt hva denne feilen skyldtes og hvordan rette opp i feilen. I sin artikkel beskriver Andersen (2022) problemløsning som en fellesøvelse når man programmerer i blokk. Andersens elever opplevde en del feilsøking i sitt arbeid med blokk. Dette kan skyldes to ting, vi brukte to forskjellige løsninger for blokkprogrammering, og en annen mulighet er at det var forskjell i oppgaveutformingen som la opp til at Andersens elever i større grad fikk mulighet til å jobbe med feilsøking.

Noe som gikk igjen i begge gruppene var at de enten hadde misforstått oppgaven, eller gjort en feil i rekkefølgen av oppgaveløsingen. Dette kan skyldes at de hadde gjort en feil i enten analysedelen eller i syntesedelen av oppgaveløsningen (Hana, 2014). I begge gruppene så skjedde det at elevene trengte støtte til å bryte ned oppgaven til mindre bestanddeler for å kunne løse oppgaven. For å gi elevene mulighet til å løse oppgaven selv ble det gitt hint om hva som kunne være årsaken til feilen i tolkingen av oppgaven i stedet for å instruere elevene (Skaalvik & Skaalvik, 2018). Mange elever opplever utfordringer knyttet til tolkning av tekstoppaver når de jobber med algebra. En av grunnene kan være at eleven misforstår ord eller overser ord og fraser i oppgaven. En annen feil kan være knyttet til løsning av oppgaven



som gir et feil svar eller at de gjør en feil når de vurderer prosessen for å løse oppgaven (Jupri & Drijvers 2016). Vi kan gi elevene et verktøy de kan bruke for å kunne forstå oppgaven bedre før de løser den, og sjekke om løsningen stemmer når oppgaven er løst. Dette verktøyet er Pólias firestegs problemløsningsprosess som består av å først forstå problemet, så legge en plan for å løse problemet, etterfulgt av å gjennomføre planen og når oppgaven er løst, se tilbake på løsningen (Van de Walle et al., 2015). Videre så kan man som lærer møte uventede svar eller feile svar om *Contingency* fra kunnskapskvartetten er på plass, men det fordrer at læreren har god kunnskap om det som skal undervises (Gustavsen et al., 2014)

Oppgavene var utformet slik at de gradvis økte i vanskelighetsgrad slik at elevene skulle ha mulighet til å mestre oppgavene, samt møte utfordringer slik at oppgavene ikke ble kjedelige. Om elevene ikke opplever utfordring i matematikkundervisning kan de etter hvert miste interesse for faget (Wæge & Nosrati, 2018). Jeg forsøkte i stor grad å utforme oppgavene på en slik måte at det ble elevstyrt læring (Jahhnke & Nordberg, 2013, i Johanson & Karlsen, 2018). Jeg forsøkte også å gi elevene støtte i form av hint framfor og instruere elevene hvordan de kunne løse oppgavene (Postholm i Steen-Olsen & Postholm, 2015). Oppgavene var tekstopp-gaver i form av problemløsningsoppgaver slik at elevene skulle få mulighet til problemløsning i oppgaveløsningen og ikke kun problemløsning på den tekniske biten av programmering. Problemløsning er et satsningsområde i norsk skole som fokuserer på framtidsrettede ferdigheter (Vinnervik & Bungum, 2022).

Samarbeidet blant elevene var som sagt svært godt. Årsaken til dette kan være at elevgruppen er trygg på hverandre. Denne tryggheten gir elevene en følelse av tilhørighet, noe de trenger for å være autonome. Om elevene opplever autonomi når de jobber med oppgaver så gir det dem en økt følelse av kompetanse, mens opplevelsen av kompetanse gir elevene selvtilliten de trenger for å føle seg akseptert og oppleve tilhørighet. Dette skaper en positiv opplevelse av matematikkfaget (Wæge & Nosrati, 2018). Dette var en elevgruppe som allerede er trygge på hverandre, så det er vanskelig å si om det å jobbe med variabler innenfor programmering var utslagsgivende for samarbeid elevene mellom. Det å skape en samarbeidskultur er noe som man gjør uavhengig av programmering, det jobbes med i all undervisning. En faktor man kan se nærmere på er at programmering er relativt nytt for samtlige elever som var med i prosjektet og hadde derfor en noenlunde samme forforståelse for programmering. Dette kan

bidra til at alle stiller likt og at alle føler at de kan bidra med noe i oppgaveløsningen, noe som gir en følelse av kompetanse som igjen kan skape en større følelse av tilhørighet. Dette gjelder nå i innføringsfasen av LK20, men etter hvert vil vi få elever opp i ungdomsskolen med en stor kompetanse innenfor programmering, så i framtiden kan det gi større skille i kompetansen innen programmering.

I begge gruppene var det godt samarbeid og elevene var flinke til å hjelpe hverandre. Det at elevene støttet hverandre gjorde at det var mindre behov for lærerstøtte. Så det kan sies at mye av læringen som skjedde var elevstyrt. Elevene kommuniserte aktivt og lærte gjennom en sosial setting (Postholm i Steen-Olsen & Postholm, 2015). Dette samarbeidet, sammen med at de vekslet på læringspar førte til en kunnskapsmobilitet i form av at de spurte de andre læringsparene og at de sammenlignet sine løsninger. En annen faktor for denne kunnskapsmobiliteten var at vi i slutten av timen gikk igjennom oppgavene og samtalte over de forskjellige løsningene (Liljedahl, 2021). Elevene jobbet mye på tvers av læringsparene og bidro til at de kunne lære hverandre, og lære fra hverandre. Samtaler i og på tvers av læringspar var framtrepende og kommunikasjonen i klasserommet var positivt ladet. Det at elevene brukte samtalen, og dermed språket aktivt mens de jobbet med et teknologisk hjelpemiddel var et godt eksempel på digital didaktikk (Johanson & Karlsen, 2018). Noe som kan bidra til å skape en slik setting er å gi oppgaver som kan utfordre elevene uten å bli uoverkommelige slik at elevene kan ligge i flyt-sonen (Lyngsnes & Rismark, 2017). Bruken av språk gjennom kommunikasjon i en sosial setting, som her er samarbeid om oppgaven bidrar til læring på individnivå (Postholm i Steen-Olsen & Postholm, 2015). Oppgavene var utformet slik at vanskelighetsgraden økte, men i prosjektperioden så klarte elevene på en god måte å løse oppgavene. De lå ikke utenfor elevenes nærmeste utviklingszone (Skaalvik & Skaalvik, 2018). Min rolle ble i stor grad å støtte elevene under læringsprosessen med hint om de sto fast og tilbakemeldinger på løsningene de kom fram til. Videre så kan det påpekes at vi kan forstå digital kompetanse bredere enn kun den tekniske gjennomføringen, den inkluderer og den sosiale biten av digital læring (Svanberg & Wille, 2009). Elevene syntes også at programmering er nyttig noe som gjorde det relevant for dem å lære (Erstad & Smette, 2017).

Flere forskere er i dag inspirert av Vygotskij og mener at algebra bør læres som et språk og et tankeverktøy. Språket er å bruke algebraiske symboler til å formulere meningsfylte

matematiske påstander og tankeredskapet er at symbolene brukes til å tenke (Hinna et al., 2016). Her kan bruk av programmering være med på å introdusere algebra som språk. Under hele prosjektperioden var det mange matematiske samtaler i begge gruppene. De diskuterte løsninger og brukte et matematisk språk for å forklare tanken bak sin egen løsning. Det at vi jobbet med variabler i programmering kan være med på å knytte variabler fra programmering til variabler i algebra. Det å jobbe med programmering fremmet elevenes medvirkning og samarbeid på en god måte. Språk var aktivt brukt i og på tvers av læringspar. Elevene formulerte presist hva en variabel er og hva den gjør i programmet, så dette kan ha en overføringsverdi til å konsist beskrive symboler i algebra. Det at variablene i programmet hadde et beskrivende navn gjorde det lettere for elevene å kommunisere variabler. Det å jobbe med programmering kan også være med å utvikle algebra som et tankeverktøy. Når vi jobber med variabler i programmering så gir vi dem ofte et meningsfullt navn som elevene kanskje lettere knytter til en tallverdi. Dette kan være et utgangspunkt for videre utvikling av elevenes tankeverktøy.

Noe av motivasjonen og påfølgende samarbeidet med bruk av variabler i programmering kan ha årsak i at programmering er nytt og spennende for elevene og de opplever derfor en motivasjon basert på dette. Etter hvert som programmering blir en mer integrert del av skolehverdagen kan det føre til at programmering blir en normal hverdag og mister statusen som nytt og spennende. Når det er sagt så vil elevene oppleve motivasjon om de føler mestring, samarbeid og autonomi som nevnt ovenfor uavhengig av hva vi jobber med i faget. Derfor er det vanskelig å si noe konkret om programmering i seg selv fremmer motivasjon, særlig på grunn av at elevene har forskjellige interessefelt i utgangspunktet, noe som kan føre til at elever med stor teknologisk interesse kan bli motivert av programmering i seg selv, mens elever som ikke har denne interessen kanskje ikke opplever samme motivasjon. For øyeblikket er programmering relativt nytt i norsk skole, men på sikt vil det bli en vanlig del av undervisningen som kan gi utslag på elevenes motivasjon da dette ikke lengre er nytt og spennende. Det at elevene ser nytteverdien av å lære programmering er også en viktig faktor for deres motivasjon og interesse for å tilegne seg denne ferdigheten. For mange elever er programmering relevant og de ser på det som en viktig og fremtidsrettet ferdighet. Dette kan være med å opprette en motivasjon innenfor læring av programmering i matematikk.

Lærerrollen og lærerstøtten i programmering vil være til stede i framtiden også, og da må vi ha fokus på hvordan vi i størst mulig grad kan hjelpe elevene til å bli autonome og føle økt kompetanse, samt samarbeide med andre elever. Oppgavene må som i alt av matematikkundervisning legge til rette for at elevene skal ha mulighet til å mestre disse innenfor sin nærmeste utviklingszone, men være utfordrende nok til at disse ikke blir kjedelig. For om elevene ikke møter utfordringer så kan de miste motivasjonen for matematikkfaget. Lærerstøtten bør i følge Vygotskij basere seg på hint framfor instruksjoner, slik at elevene selv kan finne løsningen på oppgaven. For å kunne lage oppgaver og gi gode hint til elevene så kreves det at lærerne får utviklet sin kompetanse innenfor programmering som nevnt i delkapittelet 4.3 Andre funn. Det sosiale spiller inn i møtet med ny teknologi i undervisningen og kan sies å være en like viktig del av digital kompetanse som det tekniske. Denne sosiale biten fører til mer kommunikasjon mellom elevene og kan gi elevene et språk i algebra. Et vanlig problem som elevene kan møte på når de løser tekstopp-gaver er at de misforstår eller overser ord eller fraser i tekstopp-gaven. Dette vil føre til at de kan gjøre en feil når de skal dele opp-gaven opp i mindre bestanddeler og dette vil igjen føre til følgefeil som påvirker svaret. I programmering er dette relevant da en feil verdi på en variabel eller en manglende variabel vil påvirke hele programmet. Elevene kan si seg fornøyd med svaret om de ikke gjennomgår programmet sitt og ser om den er bygd opp korrekt og følger opp-gavebeskrivelsen som gis. Et verktøy vi kan gi elevene for å forstå opp-gaven i første omgang, og gjennomgå løsningen i etterkant er Pólyas fire-steps problemløsningsprosess (Gustavsen et al., 2014).

En kort oppsummering: Elevene trengte lærerstøtte i starten av prosjektet for å lære å bruke variabler. Elevene som programmerte med tekst trengte og i starten lærerstøtte i forbindelse med feilsøking. Utformingen av opp-gavene var slik at elevene selv skulle kunne mestre disse og samarbeidet mellom elevene gjorde behovet for lærerstøtte mindre da de støttet hverandre. Elevene viste stor motivasjon i prosjektperioden, men noe av denne motivasjonen kan skyldes at programmering er noe nytt og spennende.

### 5.1.3

Det tredje forskerspørsmålet mitt er: Er det forskjell på å lære variabler i visuell programmering og tekstbasert programmering?

Som nevnt ovenfor i avsnittet «5.1.1 Forskjell på å bruke variabler i visuell og tekstbasert programmering» så var det forskjell på hvordan variabler lages, og derfor en forskjell i å lære hvordan lage variabler i programmering. Bruk av variabler følger samme regler i begge tilfellene. Forskjellen er mengden arbeid det er å bruke disse variablene og derfor så elevene som programmerte med tekst mer nytte i å bruke variabler. Dette kan ha innvirkning på i hvor stor grad elevene lærer bruk av variabler. Elever som programmerte med tekstprogrammering så på bruk av variabler som effektivt, mens elevene som programmerte med visuell programmering opplevde det som enklere å regne ut stegene selv, så de så ikke samme nytteverdi i å bruke variabler.

Det som vi skal se nærmere på nå er om elevene klarer å knytte bruk variabler fra programmering til annen matematikk, da særlig algebra. Begge informanter kunne knytte bruk av variabler til algebra på spørsmål om hvor i matematikken vi møter på variabler. Begge mente at programmering ga dem en større forståelse for variabler når de jobbet med det i programmering. Det kan gi større læringsutbytte når vi lærer ved samarbeid og ved hjelp av teknologi. Vi lærer ikke av teknologien, men ved hjelp av teknologien (Johanson & Karlsen, 2018). En svakhet med dette funnet er at den baserer seg på elevenes egen oppfatning om de har en større forståelse for variabler i algebra etter å ha brukt variabler i programmering. Svakheten i dette funnet er at det er ikke gjort noen test før og etter prosjektperioden for å måle dette objektivt.

I de nordiske landene er det en bred forståelse hva programmering innebærer. Et nøkkelkonsept er algoritmisk tankegang som består av modellering, abstrahering og feilsøking. I Norden, Norge inkludert så ses algoritmisk tankegang som en tankeprosess og er derfor ikke teknologiavhengig, men er en kompetanse som overføres på tvers av fag og disipliner (Bocconi, S., Chiocciariello, A. og Earp, J., 2018). Dette fører til at det å jobbe med variabler innenfor programmering kan overføres til algebra og gi en større forståelse for variabler. Videre så gir det elevene et til verktøy for å løse matematiske problemer.

Når det er sagt finnes det en fallgrube når vi jobber med variabler i programmering kontra det å jobbe med det i algebra spesifikt. Bruk av variabler i programmering kan være med på å forsterke en misoppfatning om at en variabel er en plassholder for en bestemt verdi i stedet for at en variabel kan ha flere eller uendelig mange verdier (Van de Walle, Karp & Bay-Williams, 2015). Grunnen til dette er at vi gir en bestemt verdi til en variabel når vi programmerer (særlig når vi driver grunnleggende programmering). For å unngå å forsterke denne misoppfatningen kan vi endre verdiene i spørsmålet, eller drive med programmering med variabler samtidig som man jobber med algebra utenom programmering. En annen fare er at elevene lærer å knytte variabler til meningsfylte navn og kan senere slite med å forstå at et tall eller et symbol også kan representere variabler.

Elevene knyttet bruk av variabler innenfor programmering med variabler i algebra og ga uttrykk for at de fikk en bedre forståelse for variabler i algebra etter å ha programmert. Basert på elevsvar så gir programmering med variabler en læring innenfor algebra. Svakheten med dette funnet er at det kun baserer seg på elevenes opplevelse av læringen og ikke et målbart resultat i form av en test før og etter prosjektperioden. Forskjellen i læringen av variabler mellom de to metodene for programmering var arbeidsmengden for å bruke variabler, noe som gjorde at elevene som drev med visuell programmering ga uttrykk for at det var lettere å regne ut stegene selv. Dette kan føre til at elever som programmerer med visuell programmering ikke ser samme nytteverdi av å bruke variabler som elever som jobber med tekstbasert programmering. Dette kan igjen påvirke motivasjonen til å lære variabler hos de elevene som programmerer i visuell programmering, noe som kan påvirke læringen om variabler. I de nordiske landene har vi en bred forståelse om hva programmering er noe som gjør at kompetanse kan føres på tvers av disiplinene i matematikk, i dette tilfellet vi variabler kunne overføres fra programmering til algebra, men og fra algebra til programmering.

Det er en del likheter med variabler i både programmering og i algebra, men bruken av variabler i programmering kan også forsterke misoppfatningen at en variabel er en plassholder for en bestemt verdi da vi gir variabelen en verdi når vi programmerer. I algebra kan en variabel ha flere eller uendelig mange verdier, så det er viktig å være bevisst på denne forskjellen når vi jobber med variabler så vi unngår å forsterke en misoppfatning. Jeg forsøkte å unngå denne fellen med å endre verdiene i oppgaven som elevene allerede hadde løst slik at de kunne se at variabelen kunne ha flere verdier. Selv om programmering bygger på algebra

så er det forskjeller mellom algebra og programmering. Ved programmering så innføres nye språkelementer som en annen syntaks og forskjeller i meningsinnhold. Elevene må da i møte med programmering lære seg et nytt språk som skiller seg fra de naturlige språkene. Dette i seg selv kan være en barriere for å lære algebra ved hjelp av programmering da språk er en viktig del av elevenes læring. Videre så kan algebra skape misoppfatninger om hva algebra er om læreren ikke er bevist på elevenes opplevelser når de møter programmering. Som nevnt tidligere kan en av disse misoppfatningene være at en variabel er en plassholder for en bestemt verdi. Algebra er gjerne dynamisk av natur, mens programmering er mer statisk. Så i algebra kan vi få en bredere forståelse av hva en variabel er enn det vi for med å kun bruke programmering.

For å konkludere: Bruken av variabler i programmering kan være til hjelp for å lære variabler i algebra om læreren er bevisst på forskjellene mellom disse to formene for variabler. Dette er viktig å vite om for å unngå å skape eller forsterke misoppfatninger. Forskjellen på å lære variabler fra visuell programmering og tekstbasert programmering er hovedsakelig mengden arbeid som elevene må gjøre for å lage og bruke variabler. Tekstbasert programmering oppleves av elevene som enklere enn visuell programmering med tanke på variabler.

## 5.2 Andre betraktninger

Både i forkant av prosjektet og underveis i prosjektet ble det gjort observasjoner, betraktninger og refleksjoner som er relevant i undervisning med hjelp av digitale plattformer. Det kreves kunnskap om personvern og man må gjøre noen etiske betraktninger når man jobber med digitale løsninger. En annen ting som kreves er at lærere for tid og mulighet til å utvikle sin egen kompetanse på teknologiske løsninger slik at de i størst mulig grad har mulighet til å hjelpe elevene med å lære å jobbe med disse løsningene.

### 5.2.1 *Utfordringer som underviser*

Utfordringen jeg møtte på i forkant og under prosjektperioden var min egen kompetanse innenfor programmering, da særlig i Python. Jeg måtte sette meg inn i grunnleggende funksjoner, kommandoer og syntaksen til programmeringsspråket. Løsningen som ble brukt i visuell programmering var også ukjent for meg da jeg startet prosjektplanleggingen, men min tidligere erfaring med blokkprogrammering gjorde denne overgangen noe lettere. For å

forberede meg best mulig til å hjelpe og støtte elevene så prøvde jeg å løse oppgavene elevene skulle få utdelt slik at jeg kunne støtte de når de skulle programmere. Dette kan ha påvirket studiens utforming og gjennomføring, med en større kompetanse hadde jeg hatt mer forkunnskap om programmering og kunne ha hatt flere betraktninger om studiens oppbygning. Når det er sagt så har prosjektet vært lærerikt og jeg sitter igjen med en større forståelse for programmering som ferdighet og hvordan jeg kan undervise elever i programmering.

I løpet av min studietid ble LK20 innført med programmering i matematikkfaget. Mange lærere som nå er i skoleverket vil møte på denne utfordringen, de må få tid til å få utvikle sin kompetanse innenfor programmering, men ofte er det lite tid til utviklingsarbeid eller dårlig økonomi i skolen som kan hindre læreres mulighet til dette (Vinnervik & Bungum, 2022). Utdanningsinstitusjoner for lærere må i framtiden ha større fokus på programmering i matematikkfaget slik at nye lærere stiller med den nødvendige kompetansen innenfor programmering. Det er og viktig å få hevet kompetansen om programmering hos lærere som er ute i arbeid, men tid og økonomi kan sette en stopper for denne kompetansehevingen.

Kunnskap om det man skal undervise i er selve grunnmuren for lærere, kunnskapen og forståelsen de har i matematikk og fagdidaktikk, eller Foundation. Transformation er hvordan læreren fremlegger fagstoff. Det kan for eksempel være aktiviteter, demonstrasjoner og spørsmålsstilling eller eksempler. Connection kunne dra paralleller på tvers av matematikkfaget Contingency er hvordan man håndterer og evner til å svare på uventete spørsmål, kommentarer eller svar i undervisningen (Gustavsen et al., 2014). Det at lærere skal kunne faget de underviser i er noe av det mest grunnleggende for en lærers kompetanse. Læreren trenger fagkunnskap for å kunne hjelpe elevene til å lære faget. Men kunnskap i matematikk er ikke nok alene, en lærer må kunne forstå matematikk på en slik måte at det er nyttig for eleven og gir mening for eleven (Loewenberg Ball, Thames & Phelps, 2008).

Digital kompetanse krever både didaktisk kompetanse, men også en teknisk kompetanse. Tidligere har fokuset innenfor digital kompetanse hatt fokus på den tekniske biten, men framover må vi også se på den sosiale biten av dette begrepet (Svanberg & Wille, 2009). I min prosjektperiode så var den sosiale biten fremtredende uten at jeg direkte hadde planlagt for dette. Det kan være at det å jobbe med en teknologisk løsning er en naturlig form for



elevene å sosialisere seg. Dagens elever har vokst opp med teknologi og de er svært vant til å bruke teknologi i sin hverdag. Boken til Svanberg og Wille kom ut i 2009 og det har vært en stor utvikling på teknologifronten siden da. Når det er sagt må all undervisning ha fokus på den sosiale biten og lærere skal fremme et godt klassemiljø.

Skolen gjennomgår reformer og det innføres nytt pensum med jevne mellomrom. Samfunnsendringer er et eksempel der skolen må tilpasse seg nye behov. En utdanningsreform består ofte av framskritt i form av implementering av ny teknologi og ny kunnskapsorientering (Postholm, Haug, Munthe & Krumsvik, 2017). Både lærere og elever må sette seg inn i mange nye teknologier i skolen i dag. Fra før av har vi regneark og dynamiske graftegnere i matematikkfaget, og nå kommer programmering for fullt inn i matematikkfaget.

For å kunne gi en god undervisning må lærere ha den nødvendige kompetansen som kreves for å drive undervisning. Når læreplaner endres, kommer det ofte inn nye aspekter å sette seg inn i for en lærer. Disse endringene i læreplaner bunner gjerne i nye behov for kompetanse i yrkesliv og samfunn generelt. Dette medfører gjerne innføring og bruk av ny teknologi i skoleverket. Ved innføringen av LK20 kom det i matematikkfaget en helt ny teknologi å sette seg inn i for mange lærere, programmering. Tidspress og økonomiske utfordringer kan hindre lærere i å få utviklet sin kompetanse innenfor programmering, noe som kan være et hinder for at programmering implementeres på en god måte i matematikkfaget. Lærerens egen kompetanse og kunnskap om det som skal undervises i er selve grunnmuren for undervisningen. Vårt kunnskapsgrunnlag avgjør i hvor stor grad vi kan gjøre denne kunnskapen tilgjengelig for våre elever. Det avgjør også om i hvor stor grad vi klarer å trekke linjer mellom de forskjellige grenene i matematikk, i denne studien, paralleller mellom variabler i algebra og variabler i programmering. Et godt kunnskapsgrunnlag gjør oss også bedre rustet til å møte på det uventete som kan skje i klasserommet, som for eksempel uventete spørsmål fra elever knyttet til temaet vi for øyeblikket underviser i. Mange universiteter tilbyr nå lærerstudentene programmering som en del av lærerstudiet. For studenter som nå starter på lærerutdanningen vil nok programmering være en naturlig del av utdanningen på lik linje med andre teknologiske hjelpemidler som for eksempel GeoGebra. Dette gjør at de har den nødvendige kompetansen når de starter i læreryrket. Vi må forstå

digital kompetanse på en bred måte som ikke kun ser på det tekniske ved å bruke teknologi, men også hvordan den sosiale delen spiller inn når elevene utvikler sin digitale kompetanse. I mitt utdanningsløp så opplevde jeg at LK20 ble innført og de første årene av studien var basert på LK06 som ikke hadde programmering i matematikkfaget. Derfor har jeg i forkant og underveis i prosjektperioden måtte satt meg inn i både det å programmere med Python og blokkprogrammering. Det har vært utfordrende å både gjøre en studie, samt jobbe med min egen kompetanse innenfor programmering, men det har samtidig vært svært nyttig og lærerikt. Det har også gjort meg enda mer bevisst på nødvendigheten for å stadig utvikle min egen kompetanse innenfor programmering, kunnskap og fag didaktiske kompetanse for å kunne gi en god undervisning.

### *5.2.2 Etiske utfordringer med digitale plattformer*

I bruken av digitale plattformer møter vi nye utfordringer knyttet til personvern. Digital etikk er nyttig for å begrense disse utfordringene. Her gjelder det å identifisere og begrense risiko i møtet med teknologi (Bergsjø et al., 2020). I forkant av prosjektet opprettet jeg en bruker på trinket for å se hvordan lagring av programmer jeg lager fungerte. Etter å ha gått igjennom prosessen for å lage en bruker vurderte jeg det til at dette ikke var noe jeg ville mine elever skulle gjøre. Prosessen krevde navn og brukernavn, samt at man måtte akseptere datainnsamling i form av cookies. Datainnsamling fra nettleverandører omtales som den nye oljen. Nettselskaper tjener store summer på å samle inn data i form av brukeropplysninger for videresalg til annonsører (Bergsjø et al., 2020). Derfor er det viktig å gjøre en etisk refleksjon om det er hensiktsmessig å opprette en bruker for elevene. I mitt tilfelle kunne det unngås å lage en bruker da vi kunne lagre arbeidet i form av skjermbilder for blokkprogrammering, og i tillegg til skjermbilder kunne vi kopiere teksten fra Python-programmet over i et tekstdokument.

Det finnes mange forskjellige løsninger for å programmere både visuelt og tekstbasert. En del av disse løsningene er et program som kan installeres på datamaskinens harddisk. Det viktige med all bruk av teknologi er å sjekke hva som samles inn av personopplysninger i forkant av bruk av teknologien.

Som lærere så må vi i den grad det er mulig å prøve å være i forkant når vi møter ny teknologi. En stor utfordring er hensynet til personvernet til elevene. Mange teknologiske løsninger vi kan bruke er gratis, men de må jo tjene penger på et vis. En måte å tjene penger på er å selge personopplysninger til annonsører slik at de kan ha målrettet reklamer til brukere. Derfor er det viktig at lærere gjør en undersøkelse og en etisk refleksjon i forkant av å bruke nye teknologiske løsninger.

En viktig betraktning knyttet til funnet i min studie er hvordan det at vi ikke lagde elevbrukere kan ha påvirket andre funn som ble gjort om bruk av variabler i programmering. Elevene som programmerte med tekst kunne lett bruke tidligere variabler med at de hadde lagret programmet i et tekstdokument, noe som ikke var mulig i visuell programmering. De elevene som programmerte visuelt, måtte ta skjermbilder av arbeidet sitt og senere gjenskape programmet i et tomt programmeringsfelt. Da måtte de lage variablene på nytt og sette variablene inn igjen i programmet. Dette kan ha bidratt til at elevene opplevde variabler innenfor visuell programmering mer tungvint enn det i utgangspunktet trengte å være. Dette kunne vært unngått med å lage en brukerkonto som ville tillat lagring i trinket, eller ved bruk av andre løsninger for programmering. Likevel kan jeg konkludere at det var vanskeligere å lage og å bruke variabler ved visuell programmering. Dette baserer jeg på observasjoner med nye oppgaver som hadde andre variabler enn de variablene elevene hadde brukt fra før, men funn på intervju kan i noen grad ha blitt påvirket av denne delen. Dette er noe jeg belyser slik at mine funn skal være så transparent som overhodet mulig og at andre kan se på dette funnet og gjøre sin egen vurdering av denne.

### 5.3 Tidligere forskning og veien videre

Min studie var en kvalitativ studie med observasjon og intervju, og ikke en dokumentanalyse. Derfor har jeg ikke søkt spesifikt etter dokumenter som omhandler tidligere forskning på feltet, men har kommet over noen tidligere og interessante artikler om temaet. Det er en god del forskning innenfor lignende temaer som har en overføringsverdi til det å jobbe med programmering i skolen.

Blokkbasert programmering i matematikkfaget er engasjerende for elevene og oppmuntrer til diskusjoner seg imellom. Mye av denne klassediskusjonen består av begrep som variabler og

løkker, so m er en del av algoritmisk tankegang. En annen viktig del av diskusjonene er at elevene samtaler om sentrale matematiske prosesser. En god måte å utvikle samarbeid på er å bruke feilsøking i felleskap. Det er indikasjoner på at algoritmisk tankegang er en sosialprosess som skaper samarbeid når det å programmere er integrert i undervisningen på en god måte (Andersen, 2022). Elevenes engasjement var noe jeg og fant i min studie, samt at det å jobbe med programmering, både tekstbasert og visuell førte til diskusjoner mellom elevene. Elevene i min studie diskuterte mye om variabler, men andre sentrale matematiske prosesser kom også fram i diskusjonen.

I artikkelen «Exploring the intersection of algebraic and computational thinking» så ser forfatterne på læring av algebra gjennom programmering og påpeker at programmering har sitt utgangspunkt i algebra, men skiller seg fra algebra i og med at programmering er et annet språk med annen meningsinnhold og språkoppbygging enn de naturlige språkene. Så elevene må i tillegg til å lære seg algebra tilegne seg kunnskap om et helt nytt språk. Et annet viktig skille er at algebra er mer dynamisk, mens programmering er mer statisk, og når programmering i tillegg innfører nye språkregler kan dette skape misoppfatninger om algebra. Om elevene får se eksempler som viser disse forskjellene kan de få en større forståelse for algebra, men denne læringen krever at læreren er bevist på elevenes erfaringer, ellers så kan læring hindres (Bråting & Kilhamn, 2021). For elevene som programmerte i Python så var det en del nye ord eller meninger de måtte lære for å kunne lage et program. Syntaksen kunne skape forvirring hos elevene og de måtte drive en del feilsøking. For elevene som programmerte med blokk var både meninger ved blokken, funksjonen den hadde, og syntaksen i språket mer selvforklarende. Etter prosjektet ble avsluttet så ble det ikke gjort noen undersøkelser på om elevene hadde noen misoppfatninger om algebra, men dette kan være et interessant felt å se nærmere på i framtiden.

Med at vi kombinerer to ulike fagområder kan elevene oppleve en høy kognitiv belastning. Fellestrekk mellom algebra og programmering kan gi læring hos elevene. Lærere bør skaffe seg kunnskap om variabelens mange fasetter og didaktiske utfordringer (Høyland, 2021). I prosjektperioden jobbet vi kun med programmering og ikke algebra ved siden av. Dette førte til at elevene ikke opplevde en høy kognitiv belastning, med det er to store felt så det å jobbe med begge kan gi en høy belastning på elevene. Informantene i min studie ga uttrykk for at de

fikk en større forståelse for variabler algebra gjennom programmering. Her er det flere fellestrekk ved variabelen og dens funksjon. Men det er og noen forskjeller og dette fordrer at læreren har god kunnskap om variabler innenfor både algebra og programmering.

Skillet mellom programmering og algebra er noe vi må være bevisst på når vi jobber med programmering. For å ha den nødvendige kompetansen for å kombinere disse to feltene så krever det kunnskap fra både programmering og algebra. En utfordring for elevene er at de i tillegg til å lære matematikk så må de lære seg et helt nytt språk som er ulik fra de naturlige språkene. Når vi kombinerer matematikkundervisning med noe som i praksis er språkundervisning kan det føre til høy kognitiv belastning hos elevene. Variablene som vi bruker innenfor begge felt har en del likheter, men også noen vesensforskjeller. Algebra er mer dynamisk av natur enn programmering som er mer statisk. Om vi lærer variabler kun gjennom programmering kan dette skape en misoppfatning at en variabel tilsvare en bestemt verdi. Men om læreren er bevisst på denne fallgraven og har kunnskap om variabelens mange sider så kan kombinasjonen av fagfeltet føre til økt forståelse for algebra, og i videreføringen variabelen. I min studie så baserte funnet på om elevene hadde en større forståelse for variabler i algebra etter å ha programmert seg på elevens subjektive oppfatning av dette og ikke basert på tester som kan bekrefte dette. De utviklet en større forståelse for variabler i forbindelse med programmering, men om dette hadde en direkte overføringsverdi til algebra er vanskelig å si noe om.

Det har vært forsket på bruken av blokkprogrammering, eller visuell programmering i matematisk sammenheng og funnene der er at det er motiverende for elevene å jobbe med visuell programmering, noe jeg også gjorde funn på i min studie. Et annet funn som sammenfaller mellom min studie og Andersens (2022) studie var at de var mye klasediskusjon når det arbeides med blokkprogrammering. Videre så tar Andersen for seg variabler og løkker var noe som ble diskutert i læringsfellesskapet og som er en del av algoritmisk tankegang. Min studie ser på variabler spesifikk. Et punkt hvor Andersen hadde et annet funn enn meg knyttet til blokkprogrammering var feilsøking. Min studie tyder på at det var lite feilsøking i blokkprogrammering, mens Andersen kommer med eksempler på feilsøking i felleskap. En av årsakene kan være at vi brukte to forskjellige løsninger på blokkprogrammering, men en annen årsak kan være utformingen av oppgavene, der mine

oppgaver kanskje ikke la like godt opp til muligheter for feilsøking sammenlignet med Andersens oppgaver

Veien videre vil være å utforske hva som kjennetegner en god undervisning av variabler gjennom bruk av både algebra og programmering. En større studie som har mer ressurser enn denne begrensede studien vil kunne gi lærere et godt verktøy for å inkludere programmering i algebraundervisningen. Matematikk er mer enn bare algebra så en mer generell studie på hvordan man kan implementere programmering i matematikkfaget og organisere undervisningen vil gi lærere en større mulighet for å drive med programmering i alle deler av matematikkfaget.

Forslag til forskningsspørsmål er

1. Hva kjennetegner god undervisning av variabler gjennom algebra og programmering?
2. Hvordan bruke programmering for å styrke elevenes læring i matematikk?
3. Hvordan organisere undervisning av programmering i matematikk?

Programmering i matematikk er per dags dato et relativt nytt felt i skolen, og en større studie tar tid å gjennomføre. Derfor vil vi se mange publikasjoner om programmering i matematikk i tiden framover. Det er viktig at disse studiene blir tilgjengelig for lærere i skolen slik at de kan bruke kunnskapen som kommer fram i sin hverdag.

Framover må også programmering bli en del av matematikkutdanningen for lærerstudenter slik at framtidens lærere stiller med den nødvendige kompetansen når de skal ut i skolen. Flere utdanningsinstitusjoner tilbyr allerede dette. Et annet punkt er å kurse lærere som allerede jobber i skolen slik at de også for utviklet sin kompetanse.

## 6 Konklusjon

Det er noen forskjeller å jobbe med variabler i tekstbasert og visuell programmering, blant annet hvordan man lager en variabel og bruk av kommentarfunksjoner. I starten opplevde elever som programmerte med tekstbasert programmering en del utfordringer knyttet til bruk av variabler. Disse utfordringene besto ofte av forskjellige stavemåter i en variabel fra et sted i programmet til en annen, eller bruk av komma når de skrev desimaltall. I Python brukes punktum for å skrive desimaltall og komma brukes for å lage en liste med tall. Elevene som drev med tekstbasert programmering, måtte oftere drive med feilsøking i programmet sitt enn de som programmerte visuelt. Feilsøking er en del av algoritmisk tenkemåte (Bocconi, S., Chiocciariello, A. og Earp, J., 2018) og elever som drev med tekstbasert programmering mer av denne biten enn de som programmerte visuelt. Elever som programmerte visuelt, opplevde bruk av variabler som tungvint og uoversiktlig og syntes det som oftest var enklere å gjøre utregningen steg for steg. Dette kan påvirke motivasjonen til å lære variabler. Her var det noen feilkilder som for eksempel hvor mye enklere det er å lagre variabler og bruke dem senere i tekstbasert programmering når man ikke oppretter en brukerprofil hos trinket.

I begge former for programmering trengtes det en del lærerstøtte i starten av prosjektet, særlig når de skulle lære hvordan lage og bruke variabler, men elevene ble etter hvert mer autonome. Elevene som programmerte med tekst lærte seg å lese feilmeldinger i Python for å finne ut hvor feilen i programmet lå og klarte etter hvert å løse disse uten lærerstøtte. Den støtten jeg i stor grad forsøkte å bruke var å gi hint til elevene (Postholm i Steen-Olsen & Postholm, 2015). Samarbeidet blant elevene i begge gruppene var god, og elevene hjalp hverandre med oppgaveløsningen som førte til mindre behov for lærerstøtte. Mye av læringen skjedde gjennom kommunikasjon elever imellom (Postholm i Steen-Olsen & Postholm, 2015). Noe som var med på å gi elevene mulighet til selvstyrt læring var utformingen av oppgavene slik at de lå i elevenes nærmeste utviklingszone (Skaalvik & Skaalvik, 2018). I hovedsak skilte behovet for lærerstøtte seg til feilmeldinger som kun elever som programmerte med tekstbasert programmering.

Som nevnt ovenfor i avsnittet «5.1.1 Forskjell på å bruke variabler i visuell og tekstbasert programmering» så var det forskjell på hvordan variabler lages, og derfor en forskjell i å lære hvordan lage variabler i programmering. Bruk av variabler følger samme regler i begge tilfellene. Forskjellen er mengden arbeid det er å bruke disse variablene og derfor så elevene

som programmerte med tekst mer nytte i å bruke variabler. Dette kan ha innvirkning på i hvor stor grad elevene lærer bruk av variabler. I Norge har vi en bred forståelse for hva programmering er, da særlig algoritmisk tankegang som ses på som en tankeprosess uavhengig av teknologi og er en kompetanse som går på tvers av matematiske disipliner (Bocconi, S., Chiocciariello, A. og Earp, J., 2018). Dette kan føre til at læring av variabler innenfor programmering kan gi en større forståelse for variabler i algebra og vice versa.

Begge informanter knyttet bruk av variabler til algebra når de ble spurt om hvor i matematikken vi møter på variabler. De opplevde at programmering ga dem en større forståelse for variabler i algebra. Læringsutbytte når vi lærer ved samarbeid og ved hjelp av teknologi. (Johanson & Karlsen, 2018). Det finnes en fallgrube når vi jobber med variabler i programmering enn med algebra spesifikt. Variabler i programmering kan være med på å forsterke en misoppfatning om at en variabel er en plassholder for en bestemt verdi i stedet for at en variabel kan ha flere eller uendelig mange verdier (Van de Walle, Karp & Bay-Williams, 2015). Grunnen til dette er at vi gir en bestemt verdi til en variabel når i programmering. For å unngå å forsterke denne misoppfatningen kan vi endre verdiene i spørsmålet, eller drive med programmering med variabler samtidig som man jobber med algebra utenom programmering. En svakhet med dette funnet er at det er basert på elevenes opplevelse av læring og ikke en test i forkant og etterkant av studien. Studien gir noen indikasjoner på læring av variabler i algebra, men kan ikke si noe konklusivt.

Et hinder for å undervise programmering i matematikkfaget på en god måte er lærernes kompetanse. Det kreves tid og økonomi for å utvikle denne kompetansen blant lærere som jobber i skolen. Både tid og økonomi er en utfordring, noe som kan hindre utviklingsarbeid innenfor programmering (Vinnervik & Bungum, 2022). Lærerutdanningen må ha større fokus på programmering i matematikkfaget slik at nyutdannede lærere har den nødvendige kompetansen innenfor programmering.

I forkant av all bruk av teknologi så må man gjøre en digital etisk refleksjon over fordeler og ulemper med bruk av teknologien (Bergsjø et al., 2020). Det er viktig å være bevisst på innsamling og bruk av personopplysninger fra teknologileverandører.



Veien videre er å gjøre flere større forskningsprosjekt knyttet til matematikk i tiden framover. Dette vil gi lærere en større mulighet for å drive en god forskningsbasert undervisning i programmering i matematikkfaget.

## 7 Referanseliste

- Andersen, R. (2022). Blokkbasert programmering og algoritmisk tenkning i en samarbeidslæringskontekst: En case-studie av programmering integrert i et matematikkfag. *Acta Didactica Norden*, (16:4). <https://doi.org/10.5617/adno.9169>
- Bergsjø, L. O., Eilifsen, M., Tønnesen, K. T. & Vik, L. G. V. (2020). *Barn og unges digitale dømmekraft. Verdiløft i barnehage og skole*. Universitetsforlaget
- Bjørndal, C. R. P. (2017). *Det vurderende øyet. Observasjon, vurdering og utvikling i pedagogisk praksis*. (3. utg.). Gyldendal Akademiske
- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group. <https://doi.org/10.17471/54007>
- Braun, V. & Clarke, V. (2008). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 2006, (3:2). 77-101. <https://doi.org/10.1191/1478088706qp063oa>
- Bukve, O. (2021). *Forstå, forklare, forandre*. (2. utg.). Universitetsforlaget
- Bråtind, K. & Kilhamn, C. (2021) Exploring the intersection of algebraic and computational thinking, *Mathematical Thinking and Learning*. (23:2). 170-185. <https://doi.org/10.1080/10986065.2020.1779012>
- Christoffersen, L. & Johannessen, A. (2018). *Forskningsmetode for lærerutdanningene*. (2. utg.). Abstrakt forlag
- Cohen, L., Manion, L., Morrison, K. & Bell, R. C. (2011). *Research methods in education* (7. utg.). Routledge.
- Ersted, O. & Smette, I. (red). (2017). *Ungdomsskole og ungdomsliv: Læring i skole, hjem og fritid*. Cappelen Damm Akademiske
- Gustavsen, T. S., Hinna, K. R. C., Borge, I. C. & Andersen, P. S. (red). (2014). *QED 5-10. Matematikk for grunnskolelærerutdanningen Bind 2*. Cappelen Damm Akademiske
- Hana, G.M. (2014). *Matematiske tenkemåter: Metamatematikk for lærerutdanningen*. Caspar Forlag
- Høyland, J. (2021). *Algebra fra Scratch: En studie av elevers arbeid med programmering i algebraundervisning, med fokus på generalisering og variabler*. [Masteroppgave]. Norges teknisk-naturvitenskapelige universitet, Fakultet for samfunns- og utdanningsvitenskap Institutt for lærerutdanning

- Johannessen, A., Tufte, P. A., & Kristoffersen, L. (2016). *Introduksjon til samfunnsvitenskapelig metode*. Abstrakt forlag
- Johanson, L. B. & Karlsen, S. S. (red), (2018). *Restart: Å være digital i skole og utdanning*. Universitetsforlaget
- Jupri, A., & Drijvers, P. (2016). Student Difficulties in Mathematizing Word Problems in Algebra. *Eurasia Journal of Mathematics, Science and Technology Education*, (12:9), 2481-2502. <https://doi.org/10.12973/eurasia.2016.1299a>
- Kunnskapsdepartementet (2019a). Læreplan i matematikk (MAT01-05). Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. Hentet fra <https://www.udir.no/lk20/mat01-05/om-faget/fagets-relevans-og-verdier>
- Kunnskapsdepartementet (2019b). Læreplan i matematikk (MAT01-05). Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. Hentet fra <https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv14>
- Liljedahl, P. (2021). *Building Thinking Classrooms in Mathematics: 14 Teaching Practices for Enhancing Learning: Grades K-12*. Corwin Press, Inc.
- Loewenberg Ball, D., Thames, M. H., & Phelps, G. (2008). Content Knowledge for Teaching. *Journal of Teacher Education*, (59:5), 389–407. <https://doi.org/10.1177/0022487108324554>
- Merriam, S. B. (2009). *Qualitative research: A guide to design and implementation* (3. utg.) San Francisco: Jossey-Bass.
- Pilgrim, Mark. (2009). *Dive into Python 3*. Apress Berkeley, CA <https://doi.org.mime.uit.no/10.1007/978-1-4302-2416-7>
- Postholm, M. B., Haug, P. Munthe, E. & Krumsvik, R. (red). (2017), *Lærere i skolen som organisasjon*. Cappelen Damm Akademiske
- Postholm, M. B. & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm akademisk.
- Pripp, A. H. (2018). Validitet. *Tidsskrift for den Norske Lægeforening*. <https://doi.org/10.4045/tidsskr.18.0398>
- Scratch Foundation. (2019). *Scratch 3.0*. Hentet fra <https://scratch.mit.edu/>
- Skaalvik, E. & Skaalvik, S. (2018). *Skolen som læringsarena: Selvoppfatning, motivasjon og læring*. (3. utgave). Universitetsforlaget
- Steen-Olsen, T. & Postholm, M. (2015). *Å utvikle en lærende skole: Aksjonsforskning og aksjonslæring i praksis*. Høyskoleforlaget

- Sund, S. (2023, 27 januar). Vil gjøre det enklere å forstå programmering. *Skolemagasinet*.  
<https://skolemagasinet.no/nyheter/vil-gjore-det-enklere-a-forsta-programmering/>
- Svanberg, R. & Wille, H. (red). (2013). *La stå!: Læring – på veien mot den profesjonelle lærer*. Gyldendal Akademiske
- trinket. «2018). Blocks. Hentet fra <https://trinket.io/blocks>
- trinket. (2018). *Python 3*. Hentet fra <https://trinket.io/python3>
- Vinnervik, P. & Bungum, B: (2022). Computational thinking as part of compulsory education: How is it represented in Swedish and Norwegian curricula?. *Nordic Studies in Science Education*, 2022. (18:3). 384-400. <https://doi.org/10.5617/nordina.9296>
- Wæge, K & Nosrati, M. (2018). *Motivasjon i matematikk*. Universitetsforlaget
- Yin, R. K. (2014). *Case study research: Design and methods* (5. utg.). Calif: Sage Publications
- Zweigart, Al. (2015). *Automating the boring stuff with Python: Practical Programming for Total Beginners*. No Starch Press, Incorporated <https://ebookcentral-proquest-com.mime.uit.no/lib/tromsoub-ebooks/detail.action?docID=4503140>

## 8 Vedlegg

### *Vedlegg 1 Intervjuguide*

#### Generelle spørsmål

1. Hvordan har prosjektperioden vært?
2. Hva tenker du om programmering?
3. Hva kan vi bruke programmering til?
4. Hvordan har samarbeidet vært?
  
5. Er det motiverende å jobbe med programmering?
6. Bør programmering være en del av matematikkfaget?
7. Er programmering til hjelp for å lære matematikk?
8. Har du noen tanker om hvilke matematiske temaer som er godt egnet til å bruke programmering som et verktøy
9. Har du lært noe matematikk i denne perioden?
10. Hvor i matematikken bruker vi variabler?

#### Spørsmål om variabler

11. Hva er en variabel?
12. Hvordan er det å lage variabler i programmering?
13. Hvordan er det å bruke variabler i programmering?
14. Er det noen utfordringer ved bruk av variabler i programmering?
15. Har du lært mer om variabler i algebra etter å ha brukt det i programmering

## Vedlegg 2 Samtykkeskjema

Vil du delta i forskningsprosjektet

***«Hva opplever elevene gir en lavere inngangsterkel til programmering av visuell programmering eller tekstbasert programmering?»»***

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å finne ut hvordan elever opplever programmering i matematikkfaget. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

### Formål

Formålet med prosjektet er å finne ut hva elever tenker og opplever i forbindelse med programmering i matematikkfaget. Prosjektet er knyttet til en masteroppgave i matematikdidaktikk, og selve undervisningsprosjektet med påfølgende intervju vil vare i to uker fra 9. januar til 20. januar. Problemstillingen i prosjektet er «Hva opplever elevene gir en lavere inngangsterkel til programmering av visuell programmering eller tekstbasert programmering?»»

### Hvem er ansvarlig for forskningsprosjektet?

UIT Fakultet for humaniora, samfunnsvitenskap og lærerutdanning er ansvarlig for prosjektet.

## Hvorfor får du spørsmål om å delta?

Du får spørsmål om å delta da du og dine foresatte har underskrevet på samtykkeskjema. Det er trukket fire informanter til intervju gjennom tilfeldig trekning, og du er en av dem.

## Hva innebærer det for deg å delta?

Hvis du velger å delta i prosjektet, innebærer det at du deltar i et intervju. Det vil ta deg ca. 45 minutter. Spørreskjemaet inneholder spørsmål om hva du tenker om programmering og om du opplever dette som en hjelp til å lære matematikk. Dine svar blir tatt opp med en lydopptaker og i etterkant transkriberes dette og lydopptaket slettes. Jeg blir også å gjøre observasjoner i klasserommet, men i disse observasjonene skrives ikke det ned personopplysninger

## Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

## Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrevet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. Lydopptaket fra intervjuet skrives ned i anonymisert form og lydopptaket slettes i etterkant.

## Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?

Prosjektet vil etter planen avsluttes når oppgaven er godkjent 20.05.2023. Lydopptaket transkriberes og selve lydopptaket slettes 20.02.2023. Transkriberingen er anonymisert slik at ingen kan identifiseres.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra UIT Fakultet for humaniora, samfunnsvitenskap og lærerutdanning har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

## Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

Hvis du har spørsmål til studien, eller ønsker å vite mer om eller benytte deg av dine rettigheter, ta kontakt med: UIT Fakultet for humaniora, samfunnsvitenskap og lærerutdanning ved Dag-Martin Berg på e-post [dbe@uit.no](mailto:dbe@uit.no) og telefon [REDACTED], eller veileder ansatt ved UIT Steinar Thorvaldsen på e-post [steinar.thorvaldsen@uit.no](mailto:steinar.thorvaldsen@uit.no) og telefon [REDACTED]

Vårt personvernombud: Joakim Bakkevold ved UIT. E-post [personvernombud@uit.no](mailto:personvernombud@uit.no) eller telefon [REDACTED]

Hvis du har spørsmål knyttet til Personverntjenester sin vurdering av prosjektet, kan du ta kontakt med:

Personverntjenester på epost ([personverntjenester@sikt.no](mailto:personverntjenester@sikt.no)) eller på telefon: [REDACTED]

Med vennlig hilsen

Dag-Martin Berg

(Forsker)



---

## Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «Hvordan opplever elever bruk av programmering i matematikkundervisningen», og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i intervju

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

---

(Signert av prosjektdeltaker, dato)

Jeg samtykker til at mitt barns opplysninger behandles frem til prosjektet er avsluttet

---

(Signert av forelder/foresatt, dato)

## Vedlegg 3 Godkjenning fra NSD



[Meldeskjema](#) / [Programmering i matematikkfaget](#) / Vurdering

# Vurdering av behandling av personopplysninger

**Referansenummer**

780915

**Vurderingstype**

Standard

**Dato**

23.01.2023

**Prosjekttittel**

Programmering i matematikkfaget

**Behandlingsansvarlig institusjon**

UIT Norges Arktiske Universitet / Fakultet for humaniora, samfunnsvitenskap og lærerutdanning / Institutt for lærerutdanning og pedagogikk

**Prosjektansvarlig**

Steinar Thorvaldsen

**Student**

Dag-Martin Berg

**Prosjektperiode**

09.01.2023 - 31.01.2023

**Kategorier personopplysninger**

Alminnelige

**Lovlig grunnlag**

Samtykke (Personvernforordningen art. 6 nr. 1 bokstav a)

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 31.01.2023.

[Meldeskjema](#) 

**Kommentar**

Personverntjenester har vurdert endringen i prosjektslutt dato.

Vi har nå registrert 31.01.2023 som ny slutt dato for behandling av personopplysninger.

Vi vil følge opp ved ny planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Kontaktperson: Line Raknes Hjellvik

Lykke til videre med prosjektet!