

Nikolai Ellingsen

Hva legger et utvalg matematikklærere i begrepet programmering og hvordan underviser de i programmering på 5-7 trinn?



Universitetet i Sørøst-Norge
Fakultet for humaniora, idretts- og utdanningsvitenskap
Institutt for matematikk og naturfag
Postboks 235
3603 Kongsberg

<http://www.usn.no>

© 2022 Nikolai Ellingsen

Denne avhandlingen representerer 45 studiepoeng

Sammendrag

I denne masteroppgaven har jeg benyttet meg av en kvalitativ forskningsmetode med semi-strukturert intervju. Denne metoden ble brukt for å undersøke problemstillingen: Hva legger et utvalg matematikklærere i begrepet programmering og hvordan underviser de i programmering på 5-7 trinn. De seks informantene har ulike erfaringer med programmering og ulik opplæring i programmering. Etter intervjuene ble det brukt innholdsanalyse for å analysere dataene. Dette var med på å systematisere og gjøre analysearbeidet konsekvent. Etter analysearbeidet satt jeg igjen med noen resultater som jeg presenterte med utgangspunkt i kategoriene og kodene som jeg definerte.

De fleste av informantene mine forbinder programmering med noe som foregår på en datamaskin, når de selv skal definere programmering. Selv om en datamaskin er en sentral del av det å programmere, så viste også informantene i intervjuet at de forbinder programmering med ferdigheter og arbeidsmåter som inngår i den algoritmiske tenkeren. Programmering kan slik en av informantene påpeker, ses på som en prosess som foregår over flere steg. Fra datamaterialet viser det tydelig at programmering er et rikt begrep som inneholder logikk, feilsøking, samarbeid og mer, og at programmering er mer komplekst enn bare å skrive programkode på en datamaskin.

Når det gjelder hvordan utvalget mitt underviser i programmering så er det variasjon. Informantene viser til flere måter å undervise på, samt ulike verktøy som kan benyttes i programmeringsundervisningen. Informantene var blant annet opptatt av å ha utforskende undervisning hvor elevene fikk prøve og feile, slik at de fikk gjort erfaringer knyttet til programmering. Det som er viktig å påpeke er at lærerne har ulik opplæring og erfaring knyttet til programmering. Dette vil ha stor innvirkning på hva de legger i et begrep og hvordan de underviser.

Abstract

In this master's degree I have used a qualitative research method with a semi-structured interview. This method was used for exploring the issue: What does a selection of mathematics teachers put into the concept of programming and how do they teach programming in grade 5-7? The six informants have different experiences with programming and different training in programming. After the interviews, content analysis was used to analyze the data. This helped to systematize and make the analysis work consistent. After the analysis work, I was left with some results that I presented based on the categories and codes that I defined.

Most of my informants connect programming to something that takes place on a computer when they are defying programming themselves. Although a computer is a central part of programming, the informants in the interview also showed that they connect programming with skills and working methods that are part of the computational thinking. As one of the informants points out, programming can be seen as a process that takes place over several steps. From the data material, it's clear that programming is a rich concept that includes logic, debugging, collaboration and more, and that programming is more complex than just writing code on a computer.

When it comes to how my selection teaches programming, there is some variation. My informants refer to several ways of teaching, as well as various tools that can be used in teaching programming. Among other things, the informants were having inquiry-based teaching where the students could try and fail, so that they could gain experience related to programming. What is important to point out is that teachers have different training and experience related to programming. This will have a big impact on what they put into a concept and how they teach.

Forord

Jeg vil først starte med å takke veileder Reiar Kravik for gode og konstruktive tilbakemeldinger gjennom hele prosessen. Jeg setter stor pris på at du har tatt deg god tid til å svare på e-post og satt av tid til veiledning. Jeg vil også takke andre veiledere og studenter i ulike masterseminarer for gode tilbakemeldinger og oppmuntringer i arbeidet med oppgaven. Takk til Erlend Stabæk og Håkon Sanden for gode tips og samtaler i forbindelse med masteroppgaven. Takk til familie og venner som har vært interesserte og spurt om oppgaven, og for å få meg til å tenke på andre ting. Til slutt en stor takk til alle lærerne som har stilt opp som informanter i denne oppgaven. Arbeidet med masteroppgaven har vært lærerikt, tidkrevende, til tider frustrerende, men alt i alt en erfaring som jeg vil få godt nytte av. Jeg er snart ferdig med fem års utdanning til et yrke som jeg gleder meg til å bli en del av. Det å være lærer er et givende og fint yrke, og jeg ser frem til å bruke kunnskapene jeg har til å bidra til barn og unges utvikling.

Haugesund, mai 2022

Nikolai Ellingsen

Innholdsfortegnelse

Sammendrag	iii
Abstract.....	iv
Forord	v
1.0 Innledning.....	4
2.0 Teori	5
2.1 <i>Hva er programmering?</i>	6
2.2 <i>Problemløsningsstrategier</i>	8
2.2.1 <i>Algoritmisk tenkning</i>	9
2.3 <i>Programmering i skolen</i>	13
2.3.1 <i>Hvorfor programmering i skolen</i>	14
2.3.2 <i>Hvordan har programmering utviklet seg i den norske skolen opp gjennom årene?</i>	15
2.4 <i>Programmering i fagfornyelsen</i>	17
2.5 <i>Tidligere forskning</i>	18
2.6 <i>Undervisningsverktøy</i>	21
2.6.1 <i>Analog programmering</i>	21
2.6.2 <i>Blokkprogrammering</i>	22
2.6.3 <i>Tekstprogrammering</i>	22
3.0 Metode.....	22
3.1 <i>Kvalitativ forskningsmetode</i>	23
3.1.1 <i>Det kvalitative forskningsintervju</i>	23
3.1.2 <i>Fordeler og begrensninger ved kvalitativ metode med semi-strukturert intervju</i>	24
3.1.3 <i>Forskningsdesign: En fenomenologisk tilnærming</i>	26
3.2 <i>Intervjuguide</i>	27
3.2.1 <i>Kategori 1: Holdninger</i>	29
3.2.2 <i>Kategori 2: Undervisningspraksis</i>	30
3.2.3 <i>Kategori 3: Opplæring</i>	31
3.2.4 <i>Kategori 4: Individuell og kollektiv kompetanse</i>	31
3.3 <i>Datainnsamling</i>	32

3.3.1 Utvalg.....	32
3.3.2 Gjennomføring av datainnsamling.....	33
3.3.3 Etterarbeid etter datainnsamling (analyse)	36
3.4 Forskningsetikk.....	41
3.4.1 Samtykkeerklæring	42
3.4.2 Konfidensialitet.....	42
3.5 Validitet og reliabilitet.....	43
3.5.1 Validiteten til denne masteroppgaven	44
3.5.2 Reliabiliteten til denne masteroppgaven.....	46
4.0 Resultater	48
4.1 Del 1 Presentasjon av hver informant.....	48
4.1.1 Informant 1.....	48
4.1.2 Informant 2.....	49
4.1.3 Informant 3.....	50
4.1.4 Informant 4.....	50
4.1.5 Informant 5.....	51
4.1.6 Informant 6.....	51
4.2 Del 2 Resultatet av analysen.....	52
4.2.1 Begrepet programmering	57
4.2.2 Undervisningspraksis	61
5.0 Drøfting	66
5.1 Del 1: Hva svarer informantene på spørsmålene som er direkte knyttet til problemstillingen	67
5.1.1 Hva legger et utvalg matematikklærere i begrepet programmering.....	67
5.1.2 Hvordan underviser et utvalg lærere på 5-7. trinn i programmering	68
5.2 Del 2: Drøfting av resultatene fra kategoriseringen av dataene	69
5.2.1 Begrepet programmering	69
5.2.2 Undervisningspraksis	73
5.3 Konklusjon	75
5.3.1 Hva legger et utvalg matematikklærere i begrepet programmering.....	75
5.3.2 Hvordan underviser et utvalg lærere i programmering på 5-7. trinn	77
5.4 Muligheter for videre forskning.....	79
6.0 Litteraturliste.....	80

7.0 Vedlegg.....	85
7.1 Vedlegg 1: Intervjuguide.....	85
7.2 Vedlegg 2: Samtykkeskjema	88
7.3 Vedlegg 3: Godkjenning fra NSD.....	92

1.0 Innledning

I den nye læreplanen er programmering blitt en sentral del av matematikken, men også i andre fag. I tillegg er programmering en kompetanse som trengs for å leve og jobbe i dagens samfunn, men også i det samfunnet som er når barna vokser opp (Sevik & m.fl., 2016, s. 10). I artikkelen til Griffin & Care (2014) skrives det om en rekke ferdigheter som trengs for å leve i det 21 århundre som defineres inn i 4 litt større og bredere kategorier: «ways of thinking», «ways of working», «tools for working» og «living in the world» (Griffin & Care, 2014, s. 11). Her finnes det en rekke ferdigheter som kan øves på gjennom bruk av programmering. Ferdigheter som kritisk tenkning, problemløsning, kreativitet og lignende er ferdigheter som artikkelen påpeker er viktig for fremtiden, og som da kan læres ved bruk av blant annet programmering (Griffin & Care, 2014, s. 246). Ved å lære og programmere går elevene fra å være brukere av en datamaskin og dens programmer til og også kunne skape ting på den. Vi lever i en verden som digitaliseres mer og mer, og det trengs dermed at mennesker får hevet den teknologiske kompetansen for å møte de ferdighetene som trengs i det samfunnet vi lever i nå, og vil leve i fremtiden.

Programmering er noe som skal jobbes med i flere fag, slik at elevene utvikler de ferdighetene og kunnskapene som trengs i programmering, som også kan overføres til andre fag og andre deler av livet. Målet med studien er å forske på hva et utvalg matematikklærere legger i begrepet programmering og hvordan de underviser i programmering på 5 til 7 trinn.

Med tanke på at jeg skal undervise i programmering når jeg er ferdig med utdannelsen min, så ønsker jeg å fordype meg i dette, slik at mine fremtidige elever vil få et bedre utbytte av undervisningen. Det finnes mange lærere ute i skolen, og helt sikkert lærere som ikke har fått opplæring og noen som har fått opplæring i programmering. I tillegg finnes det lite forskning på hvordan lærere faktisk underviser i programmering, noe som jeg kommer mer innpå i teorikapittelet mitt.

En av hovedgrunnene til at jeg vil se på hvilke praksiser som finnes blant lærere er at jeg finner lite informasjon om dette i litteraturen. Jeg finner i litteraturen eksempler som

forteller om at det er lite kunnskaper blant lærere i programmering, at de trenger opplæring, og at de vil møte på utfordringer, men veldig lite om hvordan lærere faktisk underviser og hvilke kunnskaper de faktisk har (Dolonen et al., 2019) (Bocconi et al., 2018) (Dagiené et al., 2022) (Johansen, 2020) (Forsström & Kaufmann, 2018) (Stenseth et al., 2019) (Stylianides, 2007). Dette ønsker jeg derfor å finne mer ut av, og finne konkrete eksempler på hvordan lærere har undervist i programmering så langt. Programmering er nytt på læreplanen og det vil derfor være ulike praksiser blant lærerne. Ved å kartlegge hvordan lærere underviser vil jeg også få en oversikt over hvilke kunnskaper som finnes ute i skolen i dag, sammenlignet med hvor det er ønskelig at lærerne skal være. Hensikten med denne oppgaven er dermed å utforske hvilken kunnskap lærerne har.

Studien skal være en kvalitativ studie hvor lærere som underviser i matematikk på 5-7 trinn vil bli intervjuet. Disse intervjuene vil i ettertid analyseres og gått i dybden på, for å kunne svare på problemstillingen som jeg har laget til denne oppgaven. Problemstillingen jeg har definert er: Hva legger et utvalg matematikklærere i begrepet programmering og hvordan underviser de i programmering på 5-7 trinn?

2.0 Teori

I kapittel 2.1 vil jeg fortelle om hva som legges i programmeringsbegrepet og hvordan begrepet har utviklet seg. Kapittel 2.2 tar jeg for meg problemløsningsstrategier, og da videre om algoritmisk tenkning som står sentralt i den nye læreplanen. I kapittel 2.3 ser jeg på programmering i skolen, hvorfor det er kommet inn og hvordan programmeringsundervisningen og dataundervisningen har utviklet seg, siden det kom inn i skolen og frem til i dag. I kapittel 2.4 ser jeg på programmering i den nåværende læreplanen og hvilke fag som skal ha programmering som en del av undervisningen. I kapittel 2.5 går jeg gjennom tidligere forskning når det kommer til programmering som begrep og hvordan lærere har undervist i programmering. I det siste kapittelet 2.6 legger jeg frem noen undervisningsverktøy som kan brukes i programmeringsundervisningen, og da med fokus på analog programmering, blokkprogrammering og tekstprogrammering.

2.1 Hva er programmering?

Hva som menes med begrepet programmering har endret seg opp gjennom årene (Blackwell, 2002, s. 204). Alt fra hva som legges i selve begrepet og til ferdighetene som mennesker har utviklet innenfor programmering, og da gjerne programmering digitalt, har gjort at definisjonen av programmering har utviklet seg og endret seg. Å programmere forbindes gjerne med å gjøre noe digitalt, men programmering er mye mer enn å bare bruke en datamaskin. Senter for IKT i utdanningen skrev en rapport i 2016 som handler om programmering i skolen. De har definert begrepet programmering på denne måten:

Programmering, slik det er brukt i dette dokumentet, omfatter mer enn å bare skrive programkode som kan kjøres på en datamaskin, det inkluderer også prosessen med å komme fram til denne koden. Det vil si prosessen fra å identifisere et problem og tenke ut mulige løsninger på problemet, til å skrive kode som kan forstås av en datamaskin, og å feilsøke og kontinuerlig forbedre denne koden (Sevik & m.fl., 2016, s. 9).

Vanligvis har begrepet programmering blitt brukt for å definere det å lage programkode og det å lage instruksjoner til en datamaskin (Sevik & m.fl., 2016, s. 9). Selv om programmering gjerne forbindes med noe som foregår digitalt, så er det også viktig å presisere at det er ikke datamaskinen som gjør jobben for deg. Uten et individ som gir instruksjoner eller setter de ulike reglene, så er datamaskinen ubrukelig. Datamaskinen gjør bare det du forteller den det den skal gjøre. Datamaskinen er derfor et viktig verktøy for å kunne programmere, og er med på å gjøre jobben for oss mennesker enklere. Samtidig så må ikke programmering foregå digitalt. En kan også programmere analogt ved bruk av penn og papir eller praktisk ved bruk av kroppen, og analog programmering ved bruk av penn og papir eller kroppen, kan være til god hjelp med å se sammenhenger innenfor programmering (Statped, 2021a).

Sanneutvalget var en ekstern arbeidsgruppe som skrev en rapport på vegne av Utdanningsdirektoratet tilbake i 2016, og skulle gjøre en faggjennomgang av teknologi i grunnopplæringen. Dette var med et særlig fokus på programmering og bruk av teknologi i skolen. Programmering ble i denne rapporten definert slik: «Programmering vil si å bryte et

gitt problem ned i et sett av kommandoer og så få en datamaskin til å utføre disse kommandoene» (Sanne et al., 2016, s. 18). Sanneutvalget i likhet med rapporten fra senter for IKT i utdanningen beskriver programmering som noe mer enn det å skrive kode inn på datamaskinen. Det viser, slik Blackwell skriver, at programmering har utviklet seg opp gjennom årene (Blackwell, 2002, s. 204). Det har gått fra å være og skrive programkode på en datamaskin til å gjelde hele prosessen rundt, både før, under og etter skrivingen av programkoden.

Programmering handler om, slik det står i rapporten til senter for IKT i utdanningen, om alt som også foregår før og etter koden er skrevet inn. Det å finne et problem, tenke ut løsninger, skrive koden, og vurdere og forbedre koden er en del av prosessen til å programmere (Mannila, 2017, s. 63). Det å programmere er tett knyttet til å løse problemer, og i arbeidet med å løse problemer trenger en forskjellige problemløsningsstrategier. Det å programmere er en måte å jobbe med problemløsningsoppgaver. I læreplanen til matematikk i kjerneelementene er det problemløsningsstrategien algoritmisk tenkning som det skal særlig øves på (Utdanningsdirektoratet, 2020). «Lage og programmere algoritmer med bruk av variabler, vilkår og løkker», «bruke variabler, løkker, vilkår og funksjoner i programmering til å utforske geometriske figurer og mønstre» «bruke programmering til å utforske data i tabeller og datasett» er kompetansemål etter henholdsvis 5, 6 og 7 trinn i matematikk (Kunnskapsdepartementet, 2019b). Disse målene skal det jobbes med og øves på ved bruk av algoritmisk tenkning.

Mannila (2017) beskriver programmering i samme åndedrag som problemløsning (Mannila, 2017, s. 63). Hun skriver om at samme hvilke problem som vi møter på, så må en finne fremgangsmåter for å løse problemet, og da trengs det å planlegge, fundere og finne en fremgangsmåte for å løse problemet, og slik er det også med programmering (Mannila, 2017, s. 63). Programmering handler altså om prosessen med å komme frem til riktig svar. Det er ikke bare sluttproduktet som betyr noe, men også alle de stegene en må gjennom for å komme frem til et sluttprodukt (Mannila, 2017, s. 63). Det er ikke bare det å skrive inn selve programkoden som er programmering, men altså hele prosessen fra start til det endelige sluttproduktet.

2.2 Problemløsningsstrategier

I prosessen med å løse et problem finnes det flere forskjellige fremgangsmåter (Persson, 2014, s. 7). Problemløsningsstrategier kan ses på som fremgangsmåten en velger å benytte når en møter på problemer eller oppgaver, som krever at du bruker store deler av de kunnskapene som du besitter. En kan også argumentere for at en problemløsningsstrategi er hva du gjør, når du ikke vet hva du skal gjøre (Liljedahl, 2021, s. 19). Med det så kan en tenke at problemløsningsoppgaver er så komplekse oppgaver, at du ikke klarer å løse den slik den blir presentert. Derfor kan det være nødvendig å ta i bruk en problemløsningsstrategi for å ha muligheten til å løse problemet.

Den problemløsningsstrategien som er sentral i denne masteroppgaven er algoritmisk tenkning, og algoritmisk tenkning gir en muligheten til å løse problemer som en kanskje ikke ville ha klart å løse alene (Wing, 2006, s. 33). Denne problemløsningsmetoden blir benyttet av Utdanningsdirektoratet, og kommer i tillegg til uttrykk i kjerneelementene i fagfornyelsen i matematikk, som elevene selv skal øve på (Utdanningsdirektoratet, 2020). Selv om det finnes flere problemløsningsstrategier så er det også veldig små forskjeller mellom mange av strategiene, og noen av de har også flere likhetstrekk som gjør forskjellene enda mindre (Persson, 2014, s. 7).

Teorien bak å løse et problem er forholdsvis enkel, mens selve utførelsen er mer kompleks. Matematikeren Polya (1957) gav fire steg for å løse et problem (Polya, 1957, s. xvi). I det første steget må man forstå hva som er problemet, hva er det ukjente og hvilke data har man tilgjengelig (Polya, 1957, s. xvi). Det andre steget er å planlegge, og har du løst et lignende problem tidligere, kan du bruke resultater fra tidligere problemer eller vet du om teoremer som kan hjelpe deg med å løse problemet (Polya, 1957, s. xvi). Steg tre er å utføre og løse selve problemet, og det siste steget er å evaluere resultatet og sjekke om argumentene er korrekte (Polya, 1957, s. xvi). De fleste problemløsningsmetoder baserer seg på prinsippene til Polya, og hans problemløsningsmetode har flere likhetstrekk med algoritmisk tenkning som jeg kommer nærmere inn på i neste kapittel (Persson, 2014, s. 7).

2.2.1 Algoritmisk tenkning

Algoritmisk tenkning er det norske oversettelsen av ordet computational thinking. Det engelske begrepet ble brukt av Seymour Papert i sin bok fra 1980, revidert i 1990, som handler om barn og datamaskiner (Papert, 1990, s. 182). Computational thinking er ikke et nytt fenomen, men det er nylig oversatt til norsk og fått sin plass i den norske læreplanen fra 2020 (Wing, 2006, s. 33). I den forrige læreplanen fra 2006 er algoritmisk tenkning ikke med. Algoritmisk tenkning er dermed et nytt fenomen i den norske skolen, som det nå konkret står i læreplanen, at elevene skal øve på. Det finnes ingen universell modell for computational thinking, og det er dermed laget en modell til algoritmisk tenkning på norsk (Mannila, 2017, s. 40).



Figur 1: Den algoritmiske tenkeren. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/> (Utdanningsdirektoratet, 2019).

«Algoritmisk tenkning er en problemløsningsmetode. Algoritmisk tenkning innebærer å tilnærme seg problemer på en systematisk måte, både når vi formulerer hva det er vi ønsker å løse og når vi foreslår mulige løsninger» (Utdanningsdirektoratet, 2019). Algoritmisk tenkning er viktig i arbeidet med å kunne bruke strategier og fremgangsmåter for å løse problemer som en møter på. I algoritmisk tenkning brytes problemet ned i mindre

delproblem som igjen kan løses på en systematisk måte (Csizmadia et al., 2015, s. 8). Det å tenke algoritmisk er noe som elevene skal øve på gjennom hele skolegangen sin. Det handler blant annet om å vurdere hvilke steg som en må gå gjennom for å løse problemer, og videre det å kunne bruke den teknologiske kompetansen en besitter, for å få datamaskinen til å løse deler av eller hele problemet (Utdanningsdirektoratet, 2019). Det å programmere på en datamaskin er med andre ord et viktig verktøy for å øve på nettopp algoritmisk tenkning.

Selv om algoritmisk tenkning kan øves på ved bruk av programmering og en datamaskin, så er det ikke bare på denne måten elevene kan øve på algoritmisk tenkning. Det skal også øves på i andre deler av matematikken. Algoritmisk tenkning er ikke noe som er særegent for programmering, men programmering er da en måte å øve på algoritmisk tenkning.

Algoritmisk tenkning er et sammensatt begrep som inneholder flere viktige begreper som tar for seg prosessen med å tenke algoritmisk, og det å løse problemer ved å bruke algoritmisk tenkning. Disse begrepene er logikk, algoritmer, dekomposisjon, mønster, abstraksjon og evaluering (Utdanningsdirektoratet, 2019). Disse begrepene er nært knyttet til de fire stegene til Polya (Polya, 1957, s. xvi). I begge problemløsningsstrategier skal en forstå problemet, forutse, løse problemet og evaluere. Foruten disse nøkkelbegrepene så er det også andre ting som er viktige i arbeidet med algoritmisk tenkning.

Den algoritmiske tenkeren må være systematisk og analytisk i sitt arbeid, men det er minst like viktig å være skapende, eksperimenterende og åpen for alternative løsninger. Det krever en nysgjerrig og utforskende tilnærming for å formulere og løse problemer. Å gjøre feil underveis er en viktig del av prosessen, og den algoritmiske tenkeren må ha strategier for å oppdage at noe er feil og rette feilene. Dette krever en «kognitiv kondis» for å ikke gi opp, og det er viktig å trene denne ved å jobbe med kontinuerlig forbedring underveis i prosessen. Gode løsninger oppstår ikke et vakuum, og samarbeid og deling er derfor sentrale arbeidsmetoder for den algoritmiske tenkeren (Utdanningsdirektoratet, 2019).

Det er som nevnt mye mer enn de nøkkelbegrepene som ligger til grunn for algoritmisk tenkning. En må også ha god utholdenhet, ettersom at svaret på problemet ikke nødvendigvis vil komme med en gang, og det kan også hende at en må feilsøke og rette på

ting en har gjort, og endre disse til det bedre (Utdanningsdirektoratet, 2019). Det er dermed en prosess som foregår over flere ledd. Algoritmisk tenkning er et sammensatt begrep som inneholder mye og hvor det kreves mye. Dette er en problemløsningsstrategi som ikke læres over natten, men noe som elevene trenger å gjenta og øve på, slik at de skal utvikle de ferdighetene som trengs for å kunne løse problemer ved bruk av algoritmisk tenkning.

Logikk er det første av de seks nøkkelbegrepene. I algoritmisk tenkning handler begrepet logikk om å forutse og analysere dataene eller viktig informasjon i et problem (Csizmadia et al., 2015, s. 6). For å løse et problem ved programmering kreves det at en har en plan for hvordan en skal løse problemet, og at en da kan analysere og forutse hva som kommer til å skje (Mannila, 2017, s. 81). For å kunne analysere og forutse et problem så må det ligge et fundament i bunn hos den som skal løse problemet. For at elevene skal kunne løse et problem i skolen så kreves det at de har lært seg grunnleggende ferdigheter og har et fundament, slik at de kan få muligheten til å løse problemet.

Algoritmer er det andre av seks nøkkelbegreper i algoritmisk tenkning, og det handler om å lage en plan for hvordan en ønsker å løse problemer som en møter på (Utdanningsdirektoratet, 2019). En algoritme er en nøyaktig beskrivelse av hvordan en løser noe steg-for-steg, og kan sammenlignes med det å følge en oppskrift (Mannila, 2017, s. 53). En algoritme er riktig dersom den løser og gir deg det riktige svaret, men en algoritme som gir deg feil svar kan også, noen ganger, være til nytte (Cormen et al., 2009, s. 6). I algoritmisk tenkning så handler det om å finne ut hvilke fremgangsmåter og hvilke steg som må til. En er altså på utkikk etter å finne fremgangsmåter, kjente eller ukjente, som gir deg muligheten til å løse problemet.

Dekomposisjon er det tredje av seks nøkkelbegreper og handler om å bryte ned et problem i mindre del-problem (Csizmadia et al., 2015, s. 8). Dette er for å gjøre problemet mer overkommelig ettersom at et problem kan i seg selv være stort og uoversiktlig. Da kan det å dele problemet inn i mindre deler sørge for at en klarer å løse problemet. Det å dele et problem i mindre deler kan også gjøre muligheten for samarbeid mellom personer enklere, ettersom at arbeidet kan deles mellom personer (Mannila, 2017, s. 81).

Mønster er det fjerde av seks nøkkelbegreper i algoritmisk tenkning. «Et mønster er noe som fortsetter, eller som kunne fortsette etter et forutsigbart system. Å kjenne igjen og beskrive slike systemer er hva denne delen av matematikken handler om» (Pind, 2011, s. 265). Ved å se at et mønster fortsetter så handler det om i algoritmisk tenkning, å klare og se tilbake til tidligere problemer som en har løst. Dette kan være til hjelp med å løse det nåværende problemet. Det å kunne se sammenhengen mellom det nåværende problemet, og et eller flere tidligere problem, kan være med på å gjøre løsningen av det nåværende problemet til en enklere og en mer effektiv prosess (Mannila, 2017, s. 81).

Abstraksjon er det femte nøkkelbegrepet. «Abstaktion innebär att minska komplexiteten genom att gömma irrelevanta detaljer och fokusera på de viktigaste delarna. Vi förenklar alltså det vi arbetar med för att göra arbetet lättare att hantera» (Mannila, 2017, s. 82). Abstraksjon handler altså om å forenkler problemet med å fjerne detaljer og informasjon som ikke er nødvendige for å løse det gitte problemet.

Evaluering er det siste av nøkkelbegrepene og handler om å gjøre vurderinger i arbeidet med problemer (Utdanningsdirektoratet, 2019). Dette innebærer å se på om løsningen som en har kommet frem til er gyldig og løser det den skal løse. Evaluering handler også om hvordan en kan videreutvikle løsningen, og vurdere hvilken løsning som er best for den konteksten en befinner seg i (Mannila, 2017, s. 83).

Feilsøking er som de neste fire begrepene ikke en del av de seks nøkkelbegrepene som ligger i algoritmisk tenkning. De er derimot viktige begreper som er med på å forklare hvordan en jobber med algoritmisk tenkning, men også hvordan en kan jobbe med programmering i undervisningen. Disse begrepene ligger derimot under arbeidsmåter i algoritmisk tenkning, og viser til hvordan en kan jobbe med problemløsningsstrategien (Utdanningsdirektoratet, 2019). Det å gjøre feil er en naturlig del av programmering. I arbeidet med programmering vil elevene flere ganger møte på feil i sin kode, og må da starte arbeidet med å finne hvor de har gjort feil og hvilken type feil. Dette kan være en tidkrevende prosess, men til gjengjeld noe de vil ha et læringsutbytte av (Haraldsrud et al., 2020, s. 35).

Samarbeid er en annen viktig del av programmering, ettersom at i arbeidslivet programmerer folk sjelden helt alene. De aller fleste programmer skapes gjennom samarbeid og arbeid med andre mennesker (Mannila, 2017, s. 85). Det å samarbeide og det å kunne samarbeide når en jobber med programmering, er noe som elevene vil ha stort utbytte av i nåværende skolegang, og i sitt fremtidige yrkesliv. Det finnes flere måter å samarbeide på, men en vanlig metode innenfor programmering er parprogrammering (Haraldsrud et al., 2020, s. 171). «Ved parprogrammering samarbeider to og to om å skrive en kode på en maskin. En har til enhver tid en programmerer og en kommenterende part» (Haraldsrud et al., 2020, s. 171).

«När vi skal lösa ett problem eller förverkliga en idé med hjälp av programmering behöver vi möjlighet att utforska, experimentera och pröva oss fram» (Mannila, 2017, s. 82). I programmering får elevene mulighet til å drive med utforsking, hvor elevene må være på jakt etter løsninger. I den algoritmiske tenkeren går dette under arbeidsmåten å fikle (Utdanningsdirektoratet, 2019). Det er ikke alltid slik at løsningen på et problem kommer enkelt for elevene, og at de dermed må utforske og prøve seg frem. Det å ha utforskende undervisning kan sammenlignes med det å ha problemløsningsoppgaver, og er dermed nært knyttet til algoritmisk tenkning, noe som igjen gjør at det å jobbe utforskende vil ha innvirkning på det å jobbe med algoritmisk tenkning. Dette er noe som kommer til uttrykk i lov om grunnskolen og den videregående opplæringen (Opplæringslova, 1998, § 1-1).

Det å ha utholdenhet i programmering handler om at elevene må øve seg på å stå i oppgavene, altså evnen til å kunne jobbe med en og samme oppgave over lengre tid (Utdanningsdirektoratet, 2019). Det kreves at elevene utvikler evnen til å ikke gi opp. For at elevene skal få øvd på dette, så trengs det at elevene opplever oppgavene som meningsfulle, og at de kjenner at de eier problemet de møter på (Mannila, 2017, s. 84).

2.3 Programmering i skolen

Programmering har i en eller annen form eksistert mange år i skolen. I de neste delkapitlene vil jeg presentere hvorfor det skal undervises i programmering samt vise til utviklingen fra

1966 frem til i dag. Programmering har særlig vært en del av den videregående skolen i mange år, mens grunnskolen har undervist mindre i dette.

2.3.1 Hvorfor programmering i skolen

Programmering har vært en del av skolen som valgfag på ungdomsskolen og videregående skole. Det som er nytt nå er at det er blitt obligatorisk i matematikkundervisningen, samt i naturfag, musikk og kunst og håndverk, som det kommes innpå senere. Det å kunne programmere ses på som en nødvendig ferdighet for å kunne leve i dagens samfunn, og i samfunnet som vil være når dagens unge skal ut i arbeidslivet (Sevik & m.fl., 2016, s. 10). I en verden som stadig digitaliseres kreves det at elevene får de ferdighetene som trengs for å kunne mestre det livet som kommer om x-antall år (Mannila, 2017, s. 39). I et dokument publisert av EU i 2014 kom de frem til at i løpet av 2015 ville 90% av alle jobbene kreve grunnleggende IT-kunnskaper (Communication, 2014, s. 3). Etersom en ikke vet hvordan verden vil se ut i fremtiden, så må skolen tilpasse seg omverden å gi elevene de ferdighetene som en tror vil være viktige i deres voksne liv. En tror altså at flere og flere jobber vil kreve noen grunnleggende IT-kunnskaper eller grunnleggende programmeringskunnskaper.

Foruten at elevene skal lære å programmere og at programmering ses på som en viktig ferdighet nå og i fremtiden, så vil programmering også bidra positivt inn på andre områder i skolen. I fagfornyelsen står begrepet dybdelæring sterkt. Dette begrepet er ikke et nytt fenomen, men har i fagfornyelsen fått sin plass. I overordnet del i læreplanen defineres dybdelæring slik: «Dybdelæring i fag innebærer å anvende kunnskaper og ferdigheter på ulike måter, slik at elevene over tid kan mestre ulike typer faglige utfordringer individuelt og i samspill med andre» (Kunnskapsdepartementet, 2020). Programmering kan være med på å bidra til dybdelæring hos elevene, ettersom at programmering gjør at vi kan utforske kjente konsepter på flere måter, men også ukjente konsepter på måter som ikke er utforsket ennå (Haraldsrud et al., 2020, s. 14). «Dessuten gir programmering oss metoder og teknikker som kan brukes på tvers av fag, noe som kan bidra til en dypere forståelse av problemløsningsstrategier og bestemte arbeidsmåter» (Haraldsrud et al., 2020, s. 14). Programmering er altså en mulighet for å la elevene utforske i undervisningen, noe som også

elevene har krav på (Opplæringslova, 1998, § 1-1). Det å programmere er altså en måte å jobbe med flere elementer i skolen som det er ønskelig at elevene skal øve på.

Programmering gir også muligheter i skolen som ikke har vært der tidligere. «Med programmering er det enklere å arbeide ut fra et modelleringsperspektiv der elevene kan utforske ulike sammenhenger og modeller uten å ta for mye hensyn til matematiske begrensninger. Dette gjør at vi kan flytte fokuset til drøfting, analyse og en dypere forståelse av de ulike modellene» (Haraldsrud et al., 2020, s. 14). I kjerneelementene i matematikk står mange av de nevnte begrepene sentralt (Utdanningsdirektoratet, 2020). I matematikken skal elevene kunne bruke matematiske modeller for å beskrive deres liv og samfunnet generelt, og kunne vurdere og analysere disse på en kritisk måte. Ved å bruke programmering kan en også tilnærme seg problemer i matematikken på en annen måte, samt tilnærme seg problemer som en ikke har hatt mulighet til å tilnærme seg tidligere (Haraldsrud et al., 2020, s. 14). Dette gjør at spekteret i matematikken blir større og elevene vil få en bredere kompetanse i matematikken, ettersom at ved å bruke programmering kan en fokusere på drøfting, analyse og utforske, uten å måtte ta hensyn til matematiske begrensninger (Haraldsrud et al., 2020, s. 14).

2.3.2 Hvordan har programmering utviklet seg i den norske skolen opp gjennom årene?

Selv om programmering er nytt i læreplanen av 2020, så har data og informasjonsteknologi vært i skolen i en eller annen form siden 1966 (Bostrøm et al., 2008, s. 3). Det har derimot utviklet seg ulikt i grunnskolen og i den videregående skolen, og det vil trekkes frem det viktigste som har skjedd i utviklingen frem til i dag.

I 1968 ble EDB (elektronisk databehandling) undervist ved Forsøksgymnaset, hvor det i ettertid var lærere som fikk opplæring i EDB (Bostrøm et al., 2008, s. 3). EDB er et begrep som er utdatert og som kan sammenlignes med de mer kjente begrepene IT og IKT (Busch, 2019). IT, IKT og EDB er ganske mye bredere begreper enn programmering, tre begreper som har mye av det samme meningsinnholdet, og er et «samlebegrep på teknologi som brukes til å bearbeide, lagre og formidle informasjon...» (Kolbjørnsen, 1998, s. 7). I 1981 ble det innført et obligatorisk 10-timers kurs for alle, som skulle være et prøveprosjekt over 3 år. Dette

gjorde at alle fikk en liten innføring i grunnleggende kunnskaper til data. Det samme året var det også debatt om bruken av teknologien i skolen, og viktigheten av å få det raskt inn i skolen (Bostrøm et al., 2008, s. 4). I skolereformen av 1994 ble faget informasjonsteknologi opprettet som er det samme faget som eksisterer i den videregående skolen i dag (Bostrøm et al., 2008, s. 8). Faget har derimot utviklet seg og endret seg i tråd med de ulike læreplanene som har vært i skolen siden den tid.

I 1995 ble det utarbeidet en rapport på bruken av EDB i skolen. Her kom det frem at det var få skoler som hadde fått det inn i sin undervisning på en god måte, men det var mange som hadde prøvd det i sin undervisning (Rudlang, 1995, s. 15). Særlig norsk og matematikk kom godt ut, hvor skolene hadde brukt det henholdsvis 83 og 75 prosent i disse fagene (Rudlang, 1995, s. 15). Hvilke PC-er som var tilgjengelig og hvor mange PC-er som var tilgjengelig på skolene spilte en stor rolle på disse tallene, ettersom det var mange elever per PC. I 2002 ble det pålagt at alle grunnskoler skulle bruke IT der det var hensiktsmessig (Bostrøm et al., 2008, s. 11). Fire år senere i 2006 ble digitale ferdigheter regnet som en grunnleggende ferdighet som elevene skal lære (NOU 2013: 2, 2013, s. 100). I den samme NOU-en fra 2013 kom det frem at elever har ingen eller få muligheter til å lære programmering i grunnskolen, men at de gjennom informasjonsteknologi på videregående får møtt litt på programmering der (NOU 2013: 2, 2013, s. 105).

Programmering er nytt i læreplanen av 2020, men har vært i skolen i mange år i en eller annen form, og det har vært i skolen som valgfag på ungdomsskolen siden 2012 (Dæhlen & Eriksen, 2015, s. 9). Programmering, datakunnskaper og dataferdigheter har eksistert mange år i skolen, men det har foregått en utvikling likt som ellers i samfunnet. I litteratursøk gjort om programmering som valgfag så har det blitt funnet en rapport fra 2015 som handler om evaluering av valgfagene på ungdomsskolen (Dæhlen & Eriksen, 2015). Her hadde NOVA på oppdrag fra Utdanningsdirektoratet å kartlegge om hensikten med valgfagene var innfridd (Dæhlen & Eriksen, 2015, s. 3). I starten ble programmering som valgfag kalt for teknologi i praksis. I rapporten kommer det frem at det var få elever som likte måten dette faget var lagt opp på, og elevene var skuffet over hvor utdatert dette faget var, og de ønsket å ha enda mer programmering og data inn i faget (Dæhlen & Eriksen, 2015, s. 75). Tidligere het faget teknologi og praksis før det byttet navn til programmering.

Foruten den rapporten er det også skrevet en masteravhandling om programmering som valgfag. Den handler om hvilke ulike praksiser som eksisterer for faget programmering i ungdomsskolen. «Blant funnene i arbeidet kan det nevnes at det er en forståelse for viktigheten av å jobbe med programmering i skolen, men at det er uenighet om faget skal være et eget fag eller om det skal fortsette å være et valgfag. Andre funn viser at det er få lærere som har utdanning og erfaring i koding...» (Thorland, 2018, s. iv).

Nå har det altså gått fra at elever kunne selv velge å ha programmering, til at alle elever opp til og med videregående skal ha programmering i et eller flere fag. Dette er i tråd med det som er tenkt at elevene må ha av ferdigheter i fremtidens arbeidsliv. Programmering er ikke lenger noe som elevene kan velge eller ikke velge, men noe som vil påvirke helle skolegangen deres i flere fag.

2.4 Programmering i fagfornyelsen

Sanneutvalget ser på teknologi som en så sentral del av det samfunnet vi lever i nå, og det samfunnet som vil komme. Viktigheten av teknologi og programmering presiseres på det sterkeste, og de skriver at «Teknologi som fagområde bør stå på egne ben i læreplaner og i undervisning. Erfaring fra Norge og andre land viser at det ikke er hensiktsmessig å lære om teknologi ved å integrere det i andre fag» (Sanne et al., 2016, s. 76). Sanneutvalget anbefalte å lage et nytt fag i skolen som skulle omhandle teknologi og programmering. I fagfornyelsen og slik det er blitt løst i skolen er programmering inn i fagene matematikk, naturfag, musikk og kunst og håndverk. Dette er ikke godt nok for å gi elevene de ferdighetene og kunnskapene som trengs i følge Sanneutvalget (2016), og de frykter da at en konsekvens av dette vil være at faget blir nedprioritert (Sanne et al., 2016, s. 76).

Programmering skal være en del av matematikkundervisningen i grunnskolen. I de grunnleggende ferdighetene om digitale ferdigheter står det om at programmering skal brukes for å utforske og løse problemer som de møter på (Kunnskapsdepartementet, 2019b, s. 5). I læreplanmålene er programmering nevnt eksplisitt i læreplanmålene fra 5. til 10. klasse (Kunnskapsdepartementet, 2019b). Det betyr derimot ikke at programmering ikke skal

startes på før elevene begynner i 5. klasse. I 4. klasse skal elevene lære om algoritmer, og kunne uttrykke disse ved bruk av variabler, løkker og vilkår (Kunnskapsdepartementet, 2019b, s. 8). Når det gjelder opp til 3 klasse så står det ikke noe spesifikt om ferdigheter eller kunnskaper som er knyttet direkte til programmering, men det kan fremdeles undervises i programmering på disse trinnene. Kompetansemålene kan oppnås for elevene ved å gå veien via programmering. Elevene på de laveste trinnene skal også læres i digitale ferdigheter, så de minste elevene skal også få muligheten til å utforske og løse matematiske problem ved bruk av programmering og algoritmisk tenkning.

Programmering i skolen er ikke særegent for matematikken, men det skal også undervises i programmering i fagene musikk, naturfag og kunst og håndverk som det står om i læreplanen til disse fagene, i digitale ferdigheter under grunnleggende ferdigheter (Kunnskapsdepartementet, 2019c, s. 5) (Kunnskapsdepartementet, 2019d, s. 5) (Kunnskapsdepartementet, 2019a, s. 5). Elevene skal altså møte på programmering i flere fag noe som gjør at de vil få muligheten til å løse et variert utvalg av problem ved bruk av programmering. Dette gjør også at elevene kan bruke kunnskaper de lærer i forskjellige fag inn i løsningen av problemene i andre fag, noe som er ønskelig med fokuset på dybdelæring som er kommet inn i skolen.

2.5 Tidligere forskning

Når det gjelder hvordan lærere underviser i programmering, så er programmering noe som er nytt i den norske læreplanen, men også i læreplanene internasjonalt. For å se hvordan undervisningen i programmering skal foregå så er det nødvendig å se på hvordan læreplaner og land har jobbet med programmering, ettersom læreplanene legger føringer for hvordan undervisningen skal foregå. Selv om det er nytt i læreplaner, så er det allikevel flere som har argumentert for programmering i skolen før 2000-tallet. I en bok fra 1990 (første bok om samme tema gitt ut i 1980) skrev Seymour Papert om at dataprogrammering kunne bidra til at barn ble bevisste om deres egen tenking og læring (Papert, 1990, s. 19). I en artikkel om hans liv refereres det til en av hans bøker, og om hva som kan oppnås ved at barn programmerer: «the child programs the computer and, in doing so, both acquires a sense of

mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building» (MIT, 2016). Han gjennomførte forskning på 1980-tallet, før forskningen etter hvert uteble på 1990-tallet på grunn av manglende resultater (Dolonen et al., 2019, s. 3). I Skolverket nevnes det også at programmering har i varierende grad vært en del av den svenske skolen siden slutten av 1960-tallet, men at en først på 1980-tallet fikk de første erfaringene ved å undervise i programmering (Helenius et al., 2018, s. 2).

På starten av 2000-tallet bestemte flere utdanningspolitikere i en rekke land, at det å få programmering og kunnskaper om dette inn i skolen, som noe nødvendig (Dagiené et al., 2022, s. 2). Landene brukte forskjellige begreper om hva de skulle ha inn i læreplanene, men omtaler det her som programmering. Ettersom programmering ble sett på som en viktig ferdighet i det 21. århundre, var det viktig for utdanningspolitikerne å få dette inn på læreplanen.

I 2006 publiserte Jeanette Wing en artikkel som heter «computational thinking» som pekte på viktigheten av å lære «computational thinking», som til norsk oversettes til algoritmisk tenkning. I artikkelen til Jeanette Wing skriver hun «Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability» (Wing, 2006, s. 33). Dette begrepet fikk raskt fotfeste i flere land, og da særlig i USA og landene i Europa, og disse landene begynte å anerkjenne viktigheten av å få dette inn på læreplanene (Dagiené et al., 2022, s. 2).

Uzunboylu et al (2017) tar for seg de første landene som fikk integrert programmering i læreplanen, og Estland ble i 2012 det første landet som fikk programmering inn i læreplanen, og begynte å lære elevene i programmering fra de var sju år (Uzunboylu et al., 2017, s. 785). I den samme artikkelen kommer det også frem at antall artikler om programmering i Tyrkia har økt betraktelig etter hvert som programmering har blitt en del av læreplanene (Uzunboylu et al., 2017, s. 786). Det vil si at jo mer det undervises i programmering, jo mer vil det også forskes på temaet, ettersom dette gjør mulighetene for forskning større.

I september 2014 var England det andre landet som fikk programmering inn på læreplanen. Dette var med på å skape et momentum i resten av Europa og USA, som bidro til endring i læreplaner, og da læreplaner som fokuserte på programmering (Bocconi et al., 2018, s. 1). I 2016 hadde flere land utviklet nye læreplaner og/eller fått programmering inn i læreplanen. Australia utviklet en læreplan i 2016 som het «Digital Technologies». Det ble laget et nytt fag som ble gjort obligatorisk for alle som gikk i barnehagen og til de var ferdige med 8. klasse, og etter 8. klasse var det valgfritt om de ville ha det videre (Dagiené et al., 2022, s. 5). Også i Finland fikk de i 2016 programmering inn på læreplanen, selv om de vedtok allerede i 2014 at de skulle få det inn, så det tok to år før det ble en realitet (Bocconi et al., 2018, s. 13).

Ettersom programmering er fremdeles ganske nytt, så vil også forskningen på dette være mindre. Jeg har derimot funnet ut hvordan de nordiske landene arbeider med forskjellige typer programmering, og da særlig i opplæringen av lærere, som de igjen tar med ut i klasserommene (Bocconi et al., 2018, s. 22). Det som går igjen i de nordiske landene er bruken av programmet Scratch, når det gjelder å visualisere programmering. Når det gjelder tekstprogrammering så brukes Python i alle landene bortsett fra Danmark, som ikke har oppført noe spesifikt. Ved programmering av fysiske objekter så er det blant annet brukt Micro:bit og Lego Mindstorms. Til slutt når det gjelder programmering av dataprogrammer så er det litt ulik praksis i de nordiske landene (Bocconi et al., 2018, s. 22).

I artikkelen til Bocconi et al (2018) kommer det også frem at det å introdusere algoritmisk tenkning og programmering til lærere er en kompleks prosess, når læreplaner endres og lærere må læres opp (Bocconi et al., 2018, s. 13). Det kreves tid til at lærere kan forberede seg samt at nye læreverk må publiseres (Bocconi et al., 2018, s. 13). I Norge og de andre nordiske landene er det laget kurs og studier som skal være med på å forberede lærere til å undervise i algoritmisk tenkning og programmering, og i Norge er det satt opp spesifikke kurs for å lære opp lærere (Bocconi et al., 2018, s. 23). Resultatene fra opplæringen av lærerne er det for tidlig å si noe om, men at det blir satt fokus på at dette er viktig (Bocconi et al., 2018, s. 24).

Som en kan se med bakgrunn i litteratursøk, så er programmering noe nytt for mange land, skoler og for lærere, og ganske nytt for mange å undervise i. Dermed vil det også være forskjellig praksis ute blant lærere. Etter hvert som lærere får mer utdanning og mer praksiserfaring knyttet til programmering, så vil også kvaliteten heves. Derfor vil jeg i denne masteroppgaven finne ut hvordan lærere underviser nå og hva de legger i begrepet programmering nå i dag. Definisjonen de har på programmering og hvordan de underviser kan være annerledes i dag sammenlignet med hvordan de vil gjøre det om noen år. Som nevnt er programmering noe som har vært snakket om i mange tiår allerede, men utviklingen både i skolen og på det teknologiske har vært stor, noe som gjør at hvordan lærere underviser i programmering har utviklet seg siden den tid. I tillegg vil gjerne undervisningspraksisen blant lærere variere i dag, ettersom programmering fortsatt er noe nytt for mange, og det er gjerne noen som har få erfaringer med å undervise i dette.

2.6 Undervisningsverktøy

En del av problemstillingen er som nevnt å se på hvordan lærere underviser i programmering på 5-7. trinn. Når det kommer til hvordan lærere underviser vil en vesentlig del av programmeringsundervisningen være å se på hvilke verktøy de bruker. I dette kapittelet vil det defineres tre måter å programmere på som er analog programmering, blokkprogrammering og tekstprogrammering. På Udir er det ikke lagt noen føringer for hvilke undervisningsverktøy som skal benyttes i undervisningen. Her står lærerne ganske fritt, og det er da kompetansemålene som en skal ta utgangspunkt i. Det er da opp til lærerne å vurdere hvilke(t) undervisningsverktøy som egner seg best for å nå det gitte målet.

2.6.1 Analog programmering

Analog programmering er programmering som foregår uten en datamaskin (Berg, 2021, s. 42). Det å programmere uten en datamaskin kan være med på å lære elevene sammenhenger innenfor programmering, slik at de får jobbet med forståelsen av hvorfor og hvordan ting skjer (Statped, 2021a). «Visualisering av fagstoffet ved å bruke konkrete og

bilder er også viktig for mange elever når de skal introduseres for noe helt nytt» (Statped, 2021a). Det å programmere ved å bruke konkreter kan altså være en fin inngang for elevene.

2.6.2 Blokkprogrammering

Blokkprogrammering er programmering som foregår ved bruk av blokker. I blokkprogrammering dras det inn blokker som passer, og som utfører bestemte handlinger (Haraldsrud et al., 2020, s. 87). Blokkprogrammering er noe mindre komplisert enn tekstprogrammering, og også dette kan være en naturlig inngang til programmeringsundervisningen i skolen, slik også analog programmering kan være (Statped, 2021b). Det finnes mange programmer for å drive med blokkprogrammering, og lærere har en rekke verktøy å benytte seg av.

2.6.3 Tekstprogrammering

«Med tekstprogrammering menes språk der du skriver koden ved hjelp av tekst. Tekstprogrammering er generelt sett vanskeligere enn blokkprogrammering og benyttes gjerne for de eldste elevene» (Statped, 2021b). En del tekstprogrammeringsspråk kan være kompliserte og inneholder syntaks, som kan gjøre at en må skrive mye kode for å løse et lite problem (Mannila, 2017, s. 188). Dette gjør at tekstprogrammering er lite egnet til nybegynnere, selv om de også kan forstå noe av det (Mannila, 2017, s. 188). Det finnes flere forskjellige tekstprogrammeringsspråk, noe som gjør at utvalget innen programmering med tekst er rikt.

3.0 Metode

I kapittel 3.1 tar jeg for meg kvalitativ forskningsmetode som jeg har benyttet meg av. Jeg vil også her komme nærmere innpå kvalitativt intervju som jeg har benyttet samt forskningsdesignet for oppgaven. I kapittel 3.2 skriver jeg om intervjuguiden som jeg har konstruert. Kapittel 3.3 handler om datainnsamling, og prosessen før, under og etter datainnsamlingen, og konkret hva jeg har gjort i de forskjellige stegene (analyse). Kapittel 3.4

handler om forskningsetikk og de forskjellige valgene jeg har tatt knyttet til det etiske i masteroppgaven. Det siste kapitlet 3.5 handler om masteroppgaven sin validitet og reliabilitet gjennom alle oppgavens faser.

3.1 Kvalitativ forskningsmetode

Metode velges med utgangspunkt i hva som gir best mulighet for å undersøke problemstillingen. I denne masteroppgaven skal det anvendes en kvalitativ forskningsmetode. Der kvantitative metoder baserer seg på at informasjonen om menneskers meninger og antagelser formidles gjennom tall, så vil kvalitative metoder innhente informasjon om menneskers meninger og antagelser gjennom bruk av ord og språket (Postholm & Jacobsen, 2021, s. 89). Både kvalitative og kvantitative metoder har sine fordeler og begrensninger, som gjør at ingen av metodene er ufeilbarlige (Postholm & Jacobsen, 2021, s. 89). Dette er viktig for forskere å tenke gjennom på forhånd før en velger metode, slik at en velger den metoden som egner seg best for å undersøke problemstillingen.

«Når en bruker den kvalitative metoden, er intensjonen å forstå og beskrive hva spesifikke mennesker gjør i sitt hverdagsliv, og hvilken mening disse handlingene har for dem. Beskrivelse, forståelse og mening er sentrale begreper i en tekst som presenterer en kvalitativ studie» (Postholm & Jacobsen, 2021, s. 95). I denne masteroppgaven skal det ses på hvordan et utvalg lærere underviser i programmering og hva de legger i begrepet programmering. For at informantene skal kunne uttrykke seg, og at meningene til informantene skal komme frem, ettersom det er det som er interessant for oppgaven, vil en kvalitativ metode dermed være den mest hensiktsmessige å bruke.

3.1.1 Det kvalitative forskningsintervju

I denne masteroppgaven skal det gjennomføres noen dybdeintervju. «Formålet med det kvalitative forskningsintervju er å forstå sider ved intervjupersonens dagligliv, fra hans eller hennes eget perspektiv. Forskningsintervjuets struktur er likt den dagligdagse samtalen, men

som et profesjonelt intervju involverer det også en bestemt metode og spørreteknikk» (Kvale & Brinkmann, 2010, s. 43).

Det finnes flere måter å strukturere et intervju på, men i denne masteroppgaven benyttes det et semi-strukturert intervju (Postholm & Jacobsen, 2021, s. 120). Å undersøke mennesker meninger og antagelser er sentralt i arbeidet med å undersøke problemstillingen, og ved å bruke et semi-strukturert intervju vil det igjen øke mulighetene for å gjøre nettopp dette (Kallio et al., 2016, s. 2959). I et semi-strukturert intervju vil det være en intervjuguide i bunn som har som hensikt å styre samtalen mellom intervjuer og informant. Intervjuguiden vil være med på å strukturere samtalen mellom intervjuer og informanten, slik at intervjuguiden blir den røde tråden i samtalen. For å skape en god interaksjon mellom intervjuer og informant, så har et semi-strukturert intervju vist seg å være godt egnet for nettopp dette (Kallio et al., 2016, s. 2955).

Et semi-strukturert intervju vil gi intervjuer muligheten til å stille oppfølgingsspørsmål, der informanten har andre interessante betraktninger som intervjuguiden ikke belyser (Kallio et al., 2016, s. 2955). Dette krever at intervjuer er godt forberedt, samtidig som en er kapabel til å stille oppfølgingsspørsmål. Samtidig som en stiller oppfølgingsspørsmål, bør en også prøve å stille de samme spørsmålene i intervjuene, slik at det blir mulig å sammenligne svarene til informantene (Kvernmo, 2005, s. 73).

Hvis en ønsker å bruke et semi-strukturert intervju kreves det i utgangspunktet et gitt nivå av kunnskap innenfor forskningsområde, ettersom at spørsmålene i intervjuguiden skal baseres på tidligere forskning og nåværende kunnskap (Kallio et al., 2016, s. 2955). Semi-strukturerte intervju benyttes ofte i ulike studiesituasjoner (Kvernmo, 2005, s. 73). Spørsmålene som er utformet i denne masteroppgaven er nøye utformet og vurdert av både meg og veileder, noe som skal være med på å kvalitetssikre intervjuguiden og intervjuet.

3.1.2 Fordeler og begrensninger ved kvalitativ metode med semi-strukturert intervju

I tidsskriftet til Barriball & While (1993) om bruken av semi-strukturert intervju som datainnsamlingsmetode listes det opp fem fordeler ved bruk av personlig intervju, som det

heter i denne artikkelen (Barriball & While, 1993, s. 329). Dette er et intervju hvor informantene intervjues en og en, slik det er gjort i denne oppgaven. Det har potensiale til å overkomme den lave deltagelsen til en spørreundersøkelse, fungerer bra for å undersøke holdninger, verdier og meninger, det gir muligheten til å observere kroppsspråket til informantene, det gjør det enklere å sammenligne svarene til informantene ettersom en er da sikker på at det er informantene som faktisk svarer, og til slutt sørger et intervju for at informantene ikke får hjelp fra omverdenen (Barriball & While, 1993, s. 329). Ved å gjennomføre et intervju får en dermed oppleve hva informantene selv mener om de forskjellige temaene som det intervjues om.

En annen fordel med et semi-strukturert intervju er at det gjør det enklere for intervjuer å stille oppfølgingsspørsmål, sammenlignet med for eksempel spørreundersøkelser. En kan i intervju få stilt oppfølgingsspørsmålet med en gang, noe som vil gjøre at en kan ta tak i det interessante informantene presenterer med en gang (Kvale & Brinkmann, 2010, s. 147). Den siste fordelen som vil listes opp her, er at i intervjuene vil det bli tatt lydopptak som gjør at intervjuene kan bearbeides og jobbes mer med i etterkant. Dette gjør at jeg vil kunne gi informantene min fulle oppmerksomhet under hele intervjuet. Dette er ikke en fordel med semi-strukturert intervju i seg selv, men en faktor som legges til i gjennomføringen av disse intervjuene.

Når det gjelder begrensninger ved semi-strukturerte intervju, er det viktig å nevne disse også, for denne måten å intervju på er ikke ufeilbarlig. Et semi-strukturert intervju er tidkrevende og kan sørge for at en ikke rekker like mye som en kunne ha ønsket (Opdenakker, 2006, s. 4). Dette kan igjen føre til at utvalget av informanter ikke blir stort, og som igjen gjør det vanskeligere å generalisere. En må derfor sørge for at utvalget som en har i forskningen er stort nok til å kunne generalisere.

Ved å gjennomføre semi-strukturert intervju som forskningsmetode, må en være bevisst på å ikke lede informantene i den retningen du selv ønsker. Hvordan en opptrer ved bruk av kroppsspråk og hvordan en stiller spørsmålene er aspekter ved et intervju som er viktige å tenke gjennom på forhånd (Opdenakker, 2006, s. 3). Den siste begrensningen er noe som Wengraf kaller for dobbel oppmerksomhet (double attention), og med dette mente han at

du må både høre på hva informantene sier og hva de ønsker å formidle, samtidig som du må huske på dine egne behov for å sikre at alle spørsmålene dine blir besvart (Wengraf, 2001, s. 194).

3.1.3 Forskningsdesign: En fenomenologisk tilnærming

Denne studien har fenomenologi som forskningsdesign, og for å forstå fenomenologien bedre, er det naturlig å definere hva en konstruktivistisk tilnærming er, ettersom fenomenologien har sitt utspring fra konstruktivismen. I en konstruktivistisk tilnærming hvor en ser på sosiale forhold eller handlinger mennesker foretar seg, er det viktig å være bevisst på at den sosiale virkeligheten alltid vil være tidsbegrenset og begrenset til den perioden personene lever i nå. Det vil derfor ikke være mulig å trekke slutninger som vil vare over lengre perioder (Postholm & Jacobsen, 2021, s. 49). Når sosiale fenomener studeres så er det også viktig å påpeke at det er umulig å skille forskeren fra objektet (informanten), og forskeren og objektet vil i samtale kunne påvirke hverandre, og kunnskapen skapes sammen i møte mellom forsker og objektet (Postholm & Jacobsen, 2021, s. 49).

«En konstruktivistisk tilnærming tar også utgangspunkt i at det er umulig å skille mellom objekt som studeres, og den som studerer. Immanuel Kant hevdet i *Kritikk av den rene fornuft* at mennesker aldri med full sikkerhet kan si at dette er slik objektet jeg studerer, virkelig er. Det eneste vi som mennesker kan si noe om er, hvordan vi oppfatter fenomenet» (Postholm & Jacobsen, 2021, s. 49). Med dette menes det at vi konstruerer en gjengivelse av objekter, og ikke nødvendigvis slik objektene faktisk er, altså objektene forklares ut fra det utvalget en har. Dermed vil den oppfattelsen jeg sitter igjen med i denne masteroppgaven ikke være virkeligheten i seg selv, men som skrevet vil det være min oppfattelse av virkeligheten (Postholm & Jacobsen, 2021, s. 49).

I denne masteroppgaven er det en fenomenologisk tilnærming, som er en ytterliggående form for konstruktivisme. «Utgangspunktet er at det vi kan studere, ikke er verden som den er, men slik den oppfattes av den enkelte. I et fenomenologisk perspektiv er alle oppfatninger av verden «sanne» for den enkelte. Det interessante er nettopp å forstå den enkeltes oppfatning» (Postholm & Jacobsen, 2021, s. 50). Denne tilnærmingen vil være

nyttig når det er menneskers oppfatninger av virkeligheten som er en masteroppgaves fokus, slik det også er i dette tilfelle. Det vil generaliseres med utgangspunkt i hvordan informantene selv opplever virkeligheten og hvilke «sannheter» de har. I fenomenologien så forsker en på andre menneskers opplevelser og meninger om noe (Høgheim, 2020, s. 141).

En fenomenologisk tilnærming beskriver altså meningen til flere, og ønsker å generalisere med utgangspunkt i det gitte utvalgets meninger, ved å sammenligne informantenes meninger (Creswell & Poth, 2016, s. 57). Det som er typisk for en slik metode er at problemstillingen starter med «hva er ...», og når det kommer til metode i en fenomenologisk tilnærming, så er et semi-strukturert intervju noe som gjerne brukes (Høgheim, 2020, s. 141). Dette er fordi en ønsker å ta fatt i informantenes opplevelser og meninger, og dette er en styrke i et semi-strukturert intervju, som vil bidra til å svare på problemstillingen (Høgheim, 2020, s. 141).

3.2 Intervjuguide

I gjennomføringsfasen av et semi-strukturert intervju kreves det også en intervjuguide som er utformet på forhånd. «En intervjuguide skal være til hjelp for forskeren slik at man underveis blir minnet om de temaer en ønsker data om, og samtidig bidra til at samtalene ikke får for mange sidespor. Derfor blir det først viktig å bestemme seg for hvilket fokus intervjusamtalene skal ha» (Kvernmo, 2005, s. 55). Spørsmålene ble utformet med utgangspunkt i tidligere forskning, og i artikler om forskningstemaet (Kallio et al., 2016, s. 2955). Å utvikle en intervjuguide som er nøyaktig og grundig gjennomarbeidet vil være med å sørge for og øke validiteten til den kvalitative forskningen (Kallio et al., 2016, s. 2962). I utviklingen av denne intervjuguiden har jeg benyttet meg av de fem stegene som Kallio et al har utviklet.

Det første steget handlet om å identifisere hvordan en bruker et semi-strukturert intervju, og om dette er passende for å forske på min problemstilling (Kallio et al., 2016, s. 2959). Fordeler og begrensninger med semi-strukturert intervju har det blitt skrevet om i kapittel 3.1.2, som viser da hvilke utfordringer en kan møte på i et semi-strukturert intervju, samt de fordeler et slik intervju kan medføre. Det neste steget handler om å få en oversikt over mitt

forskningstema (Kallio et al., 2016, s. 2959). Dette ble gjort ved å se på tidligere forskning, og den tidligere forskningen er da med på å legge et grunnlag for hvordan spørsmålene i intervjuguiden ble utformet (Kallio et al., 2016, s. 2959). Det tredje steget var å lage en første versjon av intervjuguiden. Intervjuguiden er et verktøy som skal brukes for å samle inn data, og denne ble testet ut i det fjerde steget som var et pilotintervju (Kallio et al., 2016, s. 2959).

Før intervjuene med informantene, så ble det gjennomført et pilotintervju for å teste intervjuet som metode og spørsmålene i intervjuguiden. Dette gjorde at spørsmålene i intervjuguiden kunne endres, fjerne spørsmål som viste seg å ikke være nødvendige å stille, samt hvis det var spørsmål som burde ha blitt med i intervjuguiden. Ved å gjennomføre et pilotintervju får en testet om metoden en bruker gir den informasjonen som en er på jakt etter, og om spørsmålene er forståelige for de som deltar (Høgheim, 2020, s. 165). Dette intervjuet er ikke tatt med i datasamlingen i denne masteroppgaven. «Uansett hvilken tilnærming til forskning man bruker, metode eller design, så bør man pilotere instrumentet man skal bruke ... Når man skal pilotere, er det selvfølgelig en fordel å få mulighet til å prøve ut instrumentet på et utvalg som kommer fra den populasjonen man skal forske på, men dette er absolutt ikke en forutsetning for å få til en vellykket pilotering» (Høgheim, 2020, s. 164). Pilotintervjuet ble gjennomført på en matematikklærer på mellomtrinnet, som gjorde at intervjuguiden ble prøvd ut på en person fra det samme utvalget som skulle være med i de senere intervjuene.

«Har man aldri gjennomført en spørreundersøkelse, et intervju eller en observasjon tidligere, er det ikke intuitivt hvordan man skal oppføre seg, hva man skal si, eller hvordan rollen som forsker egentlig kan se ut» (Høgheim, 2020, s. 165). Et pilotintervju vil gi forskeren gode erfaringer, samt god trening i det å utføre intervju før selve datainnsamlingen starter. Det kan også være en styrke å gjennomføre flere pilotintervju, slik at intervjuet til selve datainnsamlingen blir enda mer finjustert (Barriball & While, 1993, s. 334). Ved å gjennomføre et pilotintervju får en mulighet til å modifisere intervjuguiden slik at spørsmålene blir bedre, samt endre på spørsmål som blir overflødige. Det å endre på intervjuguiden etter pilotintervjuet er det femte og siste steget (Kallio et al., 2016, s. 2961).

Hvordan spørsmålene blir utformet vil ha stor betydning på hvilke svar informantene presenterer. Det finnes mange måter å stille spørsmål på, men det er viktig at spørsmålene ikke blir ledende eller manipulerende, slik at intervjuer får de svarene en selv ønsker. Det er også viktig at spørsmålene ikke blir for kompliserte, at spørsmålene blir korte og enkle for informanten å reflektere rundt (Kvale & Brinkmann, 2010, s. 146). Spørsmålene som er laget i intervjuguiden vil være de som det i utgangspunktet fokuseres på, men det vil også kunne stilles oppfølgingsspørsmål til informantene. «Oppfølgingsspørsmål er spørsmål som har til hensikt å innhente forklaringer knyttet til tema, begreper eller hendelser som forskningsdeltakeren introduserer i løpet av intervjuet» (Postholm & Jacobsen, 2021, s. 122).

Det kan også være aktuelt å stille oppklarings spørsmål til informantene. Dette er spørsmål som brukes dersom informantene er tvetydige eller litt uklare i sine svar, hvis intervjuer ikke er helt sikker på hva informantene mener (Postholm & Jacobsen, 2021, s. 125).

Intervjuguiden ble delt inn i ulike kategorier som representerte temaer som jeg ønsket å stille spørsmål i. Under disse kategoriene ble det laget spørsmål som skulle stilles til informantene. Helt til slutt i intervjuguiden ble det gitt mulighet for informantene å tilføye noe, hvis det var noe de følte at de ikke fikk sagt i intervjuet. Avslutningen av intervjuet med eventuelle oppfølgingsspørsmål går det ikke videre inn på i kategoriene når de defineres i intervjuguiden, da dette spørsmålet ikke krever noen videre forklaring. I intervjuet blir de ikke vist den algoritmiske tenkeren. Dette er for å se hva informantene selv vet om programmering, og om de er introdusert for, eller kan ferdigheter og begreper knyttet til den algoritmiske tenkeren.

3.2.1 Kategori 1: Holdninger

I starten av intervjuet eller før intervjuet starter er det viktig å skape tillit mellom intervjuer og informant. Da kan et virkemiddel være å stille spørsmål som defineres som biografiske spørsmål, som handler om arbeidserfaring, tidligere utdanning og etter- og videreutdanning (McCracken, 1988, s. 34). Disse biografiske spørsmålene er ikke skrevet ned i intervjuguiden og ikke med i selve intervjuet, da dette var spørsmål som ble stilt før intervjuet begynte. Sammen med informantene ble det gjennomført en liten uformell samtale for å skape tillit,

og en liten relasjon før selve intervjuet begynte. Spørsmålene i denne kategorien kan knyttes direkte opp mot problemstillingen til masteroppgaven. I denne kategorien går spørsmålene direkte inn på hva lærere legger i programmering samt hva de forbinder med programmering i skolen.

Det første spørsmålet som ble stilt informantene handlet om hvordan fagfornyelsen hadde påvirket deres lærerhverdag og undervisning. Dette er et spørsmål og et tema som påvirker samtlige lærere uavhengig av erfaring og kunnskap. Dette ble også en naturlig overgang videre til programmering, ettersom dette er noe nytt som står eksplisitt på den nåværende læreplanen at samtlige elever skal få kunnskaper om. Videre i denne kategorien handler spørsmålene om hva de legger i begrepet programmering, om hvilke tanker de har rundt at programmering har blitt en del av undervisningen og hvordan dette har påvirket undervisningen deres så langt.

3.2.2 Kategori 2: Undervisningspraksis

I kategori 2 handlet spørsmålene om undervisning i programmering, og hvilke undervisningspraksiser informantene selv har innenfor programmering. Spørsmålene i denne kategorien kan knyttes direkte opp mot problemstillingen. I denne kategorien stilles det spørsmål som handler om hvordan lærerne underviser og hvorfor de underviser slik de gjør.

I starten av denne kategorien stilles det et ja/nei spørsmål til informantene, hvor det ble spurt om de hadde startet med å inkludere programmering i undervisningen sin. Svarte informantene ja gikk det videre til kolonnen med ja og stilte de spørsmålene som var der. Svarte informantene nei gikk det videre til kolonnen med nei og stilte spørsmålene som var i den kolonnen. Dette ble gjort for å være forberedt på at informantene kunne ha ulik bakgrunn og erfaring når det kommer til undervisning i programmering. Ved å stille dette ja/nei spørsmålet og ved å forberede spørsmål til om de svarte ja eller nei, gjør det at det raskt kunne stilles nye spørsmål som informantene kunne reflektere rundt.

I denne kategorien var jeg interessert i å finne ut hvilke programmer informantene bruker i undervisningen, hvorfor de bruker disse programmene, hvordan de underviser og hvordan de hjelper elevene videre i faget enten om de mestrer programmering eller om de synes at programmering er vanskelig. Det var også ønskelig at informantene skulle reflektere rundt om de underviser annerledes i programmering sammenlignet med resten av temaene i matematikk, og sammenlignet med de andre fagene de underviser i. I tillegg til konkrete eksempler på hvordan de underviser var det også interessant å se om informantene ser overføringsverdi fra programmering til resten av matematikken, og til andre fag i skolen.

3.2.3 Kategori 3: Opplæring

I den tredje kategorien ønskes det å kartlegge hvilken opplæring som lærerne hadde fått i programmering, eller om de ikke hadde noen formell opplæring enten via kurs eller gjennom studier. Spørsmålene i denne kategorien treffer ikke problemstillingen ordrett slik som kategori 1 og 2 gjør. Selv om spørsmålene ikke kan knyttes direkte opp mot problemstillingen, så vil spørsmål som omhandler opplæringen lærerne har innenfor programmering, ha noe å si for hvordan de selv underviser, og hva de legger i begrepet programmering. Det er dermed interessant å undersøke hvilken opplæring de har, ettersom dette vil ha en direkte påvirkning på undervisningen og begrepsforståelsen deres.

Også denne kategorien startet med et ja eller nei spørsmål som informantene ble stilt. Her ble informantene spurt om de hadde fått noen opplæring knyttet til programmering. Hvilket svar de kom med la dermed grunnlaget for hvilke spørsmål som skulle stilles videre i intervjuet. Svarte de ja så ønsket så var det ønskelig å finne ut hvilken type opplæring de hadde fått, omfanget av opplæringen og hvilken nytteverdi de selv så i opplæringen. Hvis de derimot svarte nei på at de ikke hadde fått opplæring, så var det ønskelig å finne ut hvorfor de ikke har noen opplæring. Var det på grunn av manglende tilbud, tid eller om de selv føler at de ikke har behov for det.

3.2.4 Kategori 4: Individuell og kollektiv kompetanse

Den fjerde og siste kategorien i intervjuguiden handlet om informantenes individuelle kompetanse og kompetansen til kollegaene til informanten. Spørsmålene i denne kategorien, i likhet med kategori 3, så kan heller ikke denne knyttes direkte opp mot problemstillingen. Selv om den ikke finnes ordrett i problemstillingen vil individuell og kollektiv kompetanse ha noe å si for undervisningen og begrepsforståelsen til den enkelte læreren. Hvordan skolene arbeider eller ikke arbeider med programmering vil ha påvirkning på læreres undervisning og begrepsforståelse. Hvis skoler har fått i gang gode kollektive læreprosesser kan dette ha påvirkning på utviklingen til den enkeltes lærer sin kompetanse. Dette kan igjen gjøre at undervisningen og begrepsforståelsen til en lærer kan ha kommet lengre i utviklingen, sammenlignet med en lærer som jobber på en skole hvor de har arbeidet lite med programmering. For de som ikke har studert programmering eller hatt kursing i dette, kan den kollektive kompetansen og arbeidet på skolen ha noe å si for hvordan lærerne på den skolen underviser og deres begrepsforståelse.

I disse spørsmålene ønskes det å finne ut hvordan informanten selv plasserer seg sammenlignet med resten av kollegaene når det kommer til kompetanse. Det ønskes også å høre om informantenes tanker rundt kollegaers holdninger knyttet til programmering, og om de føler seg forberedt på å undervise i dette. Til slutt er det interessant å høre hvordan de har jobbet med programmering på arbeidsplassen, på tvers av trinn og på tvers av fag. Dette for å finne ut om det er ulike praksiser blant skolene når det kommer til arbeidet med programmering.

3.3 Datainnsamling

3.3.1 Utvalg

I denne masteroppgaven har jeg valgt å forske på hva et utvalg matematikklærere legger i begrepet programmering og hvordan de underviser i programmering på 5-7 trinn. I utgangspunktet vil dette gi meg mange lærere å forske på, da programmering står beskrevet i flere fag på læreplanen enn matematikk. Det skal derimot kun forske på matematikklærere på 5-7 trinn. Hvem som velges som informanter er en kritisk faktor i dette

forskningsarbeidet, og i jakten på å få troverdige svar i det som det ønskes å forske på (Kvernmo, 2005, s. 55). Jeg ønsket også å finne lærere som jobbet på ulike skoler, for å finne et bredest mulig utvalg, og for å gjøre det enda tydeligere i å sammenligne de forskjellige informantene.

I arbeidet med forskningsmetoder så kan utvalget være for lite, og at det dermed medfører at det ikke blir mulig å generalisere. Altså en kan ikke trekke slutninger ettersom utvalget ikke representerer stor nok andel av det som faktisk finnes. Det er derfor viktig å påpeke at slutningene blir tatt ut ifra det utvalget en selv har forsket på, samtidig må en passe på at utvalget faktisk blir stort nok til å drive forskning. Dette har noe å si for validiteten til oppgaven som det skrives mer om senere i kapittelet.

Utvalget mitt består av seks informanter som alle jobber på 5-7 trinn. De underviser alle i matematikk. Et slikt utvalg kan være med på å gi indikasjoner på hvordan ting kan være for en større gruppe mennesker. Det er viktig å presisere at det generaliseres på bakgrunn av mitt utvalg.

3.3.2 Gjennomføring av datainnsamling

I gjennomføringen av datainnsamlingen valgte jeg å benytte meg av lydopptak under intervjuene. Dette var for å sikre at informantene fikk min fulle oppmerksomhet gjennom hele intervjuet. Ved å benytte lydopptak gjorde det at intervjuet kunne jobbes mer med i etterkant, og gå enda mer i dybden på intervjuet, slik at analysearbeidet skulle bli enklere. I denne masteroppgaven er det benyttet semi-strukturert intervju noe som gir muligheten til å stille oppfølgingsspørsmål. Hvis all oppmerksomhet går til å notere ned det informantene sier, så kan en gå glipp av interessante betraktninger som en kan grave enda mer i. Det er vanskelig å få med seg alt som blir sagt hvis en noterer, og lydopptak kan være til hjelp i gjennomføringen av intervju (Postholm & Jacobsen, 2021, s. 131).

«For at deltakerne skal oppleve intervjuet som en mest mulig trygg situasjon, er det formålstjenlig at det er de som velger stedet hvor intervjuet skal foregå» (Postholm & Jacobsen, 2021, s. 132). Konteksten kan påvirke informantene og intervjuene i en

masteroppgave (Høgheim, 2020, s. 167). Derfor fikk informantene selv velge hvor de ville ha intervjuene. Det var satt av 45 minutt til å gjennomføre selve intervjuet, men alle brukte mindre tid enn dette. Informantene ble også intervjuet en og en. I tillegg til miljøet intervjuene foregår i, er det viktig at væremåten til intervjuer er behagelig og trygg for de som skal intervjues. Hvordan en møter på informantene og hvordan en ordlegger seg i intervjuet, vil ha påvirkning på hvordan kvaliteten på intervjuet blir.

To av intervjuene måtte gjennomføres over internett på grunn av geografiske avstander, og på grunn av koronasituasjonen i Norge. Det å gjennomføre intervjuene over internett har både fordeler og begrensninger. I intervju over internett kan den mellommenneskelige kontakten bli svakere, og viktige aspekter som kroppsspråk og informanternes opplevelse av intervjuet kan bli påvirket (Thunberg & Arnell, 2021, s. 6). Dette kan medføre at kvaliteten på intervjuet og kvaliteten på datainnsamlingen blir svakere (Thunberg & Arnell, 2021, s. 6). Samtidig var det en stor fordel at intervjuene kunne gjennomføres på nett. Ved å kunne gjennomføre intervju over nett gjorde dette at antall informanter ble høyere enn det kanskje ville vært (Thunberg & Arnell, 2021, s. 5). Det argumenteres også for at å gjennomføre intervju over internett vil ha like stor nytteverdi som intervju der en møtes fjes til fjes, og at intervju over internett ikke nødvendigvis er en begrensning (Thunberg & Arnell, 2021, s. 6).

Det er også interessant å nevne at i et intervju så er det en samtale mellom to som ikke er likeverdige. Med dette menes det at det er en intervjuer som styrer samtalen og som styrer hva det snakkes om. Selv om et intervju skal være en interaksjon mellom mennesker, og hvor mennesker deler synspunkter, vil gjerne et forskningsintervju bare fokusere på hva den som intervjues har å si og mener (Kvale & Brinkmann, 2010, s. 23). Dette er også naturlig når det er ønskelig fra intervjuers side å innhente informasjon om meninger og antagelser fra de som intervjues. Da vil meningene til den som intervjuer ikke være interessante. Selv om det skal være en samtale mellom to likeverdige parter vil det samtidig ikke være det, og situasjonen kan oppleves som litt kunstig.

I boken det kvalitative forskningsintervju står det blant annet at: «Forskeren ønsker at intervjuet skal være så dyptgående og utforskende som mulig, med risiko for å krenke personen, men ønsker på den annen side å respektere den intervjuede personen så mye

som mulig – dermed risikerer han eller hun å få empirisk materiale som bare skraper overflaten» (Kvale & Brinkmann, 2010, s. 184). I intervjusituasjoner kan man dermed gå glipp av relevant og nyttig informasjon. En må derfor være bevisst de prosessene som foregår i et intervju, og være bevisst på hvordan spørsmålene blir stilt.

«Kvaliteten på det originale intervjuet er avgjørende for kvaliteten på den senere analyseringen, verifisering og rapportering av intervjuet» (Kvale & Brinkmann, 2010, s. 174). Hele masteroppgaven vil ta utgangspunkt i intervjuene som gjennomføres, og intervjuene vil derfor være ekstremt viktige for masteroppgaven. Kvaliteten på intervjuene vil spille direkte inn på kvaliteten på masteroppgaven. Det er derfor viktig å være god forberedt før, under og etter intervjuene, for å sørge for at prosessen i og rundt intervjuene blir av så høy kvalitet som overhodet mulig.

Kvale & Brinkmann (2010) nevner seks kvalitetskriterier for et intervju. Selv om kriteriene kan ses på som uoppnåelige, kan de likevel fungere som retningslinjer for hvordan et godt intervju kan foregå (Kvale & Brinkmann, 2010, s. 175). Jeg var særlig opptatt av i mine intervju å verifisere det intervjupersonene sa, og å være flink til å følge opp informantene sine svar, i tråd med et semi-strukturert intervju. De seks kvalitetskriteriene handler blant annet om hvordan spørsmålene intervjuer stiller, hvordan taletiden blir mellom partene og verifiseringen av meningene til informanten (Kvale & Brinkmann, 2010, s. 175).

Et annet viktig aspekt av kvaliteten på intervjuet, er den som intervjuer og den som blir intervjuet. Hvordan de som blir intervjuet opptrer i intervju, om de er samarbeidsvillige, motiverte og kunnskapen de sitter inne med, vil ha stor betydning på intervjuene (Kvale & Brinkmann, 2010, s. 175). Det er også viktig at svarene de gir på de ulike spørsmålene er konkrete og relevante. Også intervjueren vil ha stor påvirkning på kvaliteten på intervjuene. Intervjueren må, særlig i et semi-strukturert intervju, foreta raske avgjørelser på hva som det skal spørres ekstra om og hvordan. Intervjuer må også som nevnt tidligere ha gode kunnskaper i det temaet som undersøkes, dette igjen for å vite hva som er relevant å spørre mer om, og ikke (Kvale & Brinkmann, 2010, s. 176).

Et intervju som form er allerede en setting der det vil være et asymmetrisk forhold mellom partene, ettersom den ene stiller spørsmål og den andre svarer. Intervjuet er også en instrumentalisering av samtalen, hvor den gode samtalen ikke er målet, men et virkemiddel for forskeren for å få frem betraktningene til den som intervjues (Kvale & Brinkmann, 2010, s. 52). Det kan også være et asymmetrisk maktforhold med tanke på at den ene parten har store kunnskaper innenfor det praktiske, og den andre kan ha store kunnskaper innenfor det vitenskapelige. Dette kan også medføre usikkerhet blant begge parter, hvis partene ikke gjør seg forstått av hverandre.

Kvaliteten på intervjuene som gjennomføres vil som nevnt ha stor betydning på resten av oppgaven. I Kvale & Brinkmann (2010) sin bok listes det opp ti kvalifikasjonskriterier for intervjueren, altså kvaliteter som intervjueren selv bør ha i sitt ferdighetsregister (Kvale & Brinkmann, 2010, s. 177). Alle ti kriteriene er i seg selv viktige for en intervjuer og som det i denne oppgaven har blitt fulgt etter beste evne i gjennomføringsfasen av intervjuene, men også i forberedelsene av intervjuene.

De ti kvalifikasjonskriteriene ble brukt aktivt både før og under gjennomføringen av intervjuene. Som førstegangsforsker så jeg på det som en fordel å ha noen konkrete kriterier å forholde seg til i gjennomføringen av intervjuene. «Intervju kvalifikasjoner som dem som er skissert i boks 9.2, og som er formulert i sammenheng med et fenomenologisk livsverdenintervju, kan føre til gode intervjuer i den forstand at de produserer rikholdig kunnskap og etisk sett skaper en positiv situasjon for intervjupersonene» (Kvale & Brinkmann, 2010, s. 178). Selv om disse kvalifikasjonskriteriene er laget for den som intervjuer vil det også påvirke de som blir intervjuet. Ettersom at intervjuene er avgjørende for kvaliteten på det videre arbeidet, så jeg på det som en styrke å ha teori på hvordan en skal gjennomføre intervju på en best mulig måte, både for informantene og for meg som intervjuer.

3.3.3 Etterarbeid etter datainnsamling (analyse)

Når en analyser skal en gå fra en sammensatt helhet som skal brytes ned i ulike deler til å undersøke delenes mening og budskap (Høgheim, 2020, s. 199). Gjennom analysearbeidet

ønsker en å kunne svare på problemstillingen, og hvilken analysemetode en benytter må tas på bakgrunn av hvilken metode som vil gi deg best mulighet til å svare på problemstillingen. Etter intervjuene ble det gjennomført analyse av intervjuene. I analysen ble det tatt i bruk innholdsanalyse. I en innholdsanalyse ønsker en å tolke innholdet i tekstdataene ved bruk av systematisk koding og kategorisering av temaer og mønstre (Høgheim, 2020, s. 202).

I innholdsanalysen har det blitt benyttet fem steg. De fem stegene var å bli kjent med dataene, lage koder, lage kategorier, analysere og til slutt trekke slutninger (Høgheim, 2020, s. 202). I analysearbeidet ble det også brukt NVivo som kan brukes inn i det analytiske arbeidet. Dette programmet er med på å sortere dataene på en oversiktlig måte, og som gjør det enklere å endre og redigere på koder og kategorier.

Det som er viktig i arbeidet med å analysere data er at en er konsekvent og bruker den samme metoden gjennom hele analysearbeidet. For å få dette til kan det lages en oversikt over de ulike stegene som det går gjennom, altså lage en logg, for å huske hva som blir gjort og hvorfor det ble gjort. Dette har blitt brukt i denne oppgaven. Dette blir gjort for at analysearbeidet skal være konsekvent gjennom hele analysen, og for å huske hva som ble gjort tidligere når arbeidet ble hentet opp igjen.

Før innholdsanalysen startet så ble intervjuene transkribert. Transkripsjon handler om oversettelsen fra talespråk til skriftspråk (Høgheim, 2020, s. 133), og dette er en prosess som foregår for at intervjuene skal kunne analyseres enklere (Kvale & Brinkmann, 2010, s. 186). Det er enklere å analysere et intervju som er transkribert sammenlignet med å analysere samtidig som en hører på lydopptakene. I en transkripsjon bør en også ha med ordlyder og uttrykk, slik at transkripsjonen blir så nære det aktuelle intervjuet (Høgheim, 2020, s. 133). Selv om en bør ha med ordlyder er det også mulig å skrive transkripsjonen uten ordlydene. Hva transkripsjonen skal brukes til vil spille inn på hvordan en skal transkribere (Kvale & Brinkmann, 2010, s. 190). Velger en derimot å skrive om på setningene slik at de ikke blir ordrett, så må en være forsiktig, ettersom at informantene sine meninger kan endres eller tolkes annerledes (Kvale & Brinkmann, 2010, s. 192). Dermed kan viktige betraktninger fra informantene gå tapt eller at betraktningene tolkes annerledes enn det som var ment fra informantenes side. Også tegnsetting vil påvirke meningene til informantene, så det er viktig

å være bevisste de valgene som eksiterer når en skal transkribere. I denne masteroppgaven er det blitt transkribert ordrett slik at betraktningene til informantene blir mest mulig likt, slik de selv formidlet det i intervjuene.

Ettersom det er forskjeller mellom talespråket og skriftspråket gjør det at transkripsjon ikke bare foregår uten problem. I en muntlig samtale vil kroppsspråk, stemmeleie, tempo og ironi ha påvirkning på det som det snakkes om. Dette kan medføre at det foregår abstraksjon fra intervjuet til transkripsjonen, men allerede ved lydopptaket skjer den første abstraksjonen (Kvale & Brinkmann, 2010, s. 187). «Lydopptaket av intervjuet innebærer en første abstraksjon fra de samtalende personers direkte fysiske tilstedeværelse ... Transkripsjonen av intervjusamtalen til en skriftlig form innebærer enda en abstraksjon ... Transkripsjon er kort sagt svekkede, dekontekstualiserte gjengivelser av direkte intervjusamtaler» (Kvale & Brinkmann, 2010, s. 187). I en transkripsjon må en skrive så nøyaktig som mulig fra lydopptaket. En bør prøve å sørge for at transkripsjonen blir så nære det reelle intervjuet som ble gjennomført. Dette er derimot vanskelig å få til da viktige aspekter ved intervjuet blir borte i lydopptaket og ved selve transkriberingen.

«Når intervjuene transkriberes fra muntlig til skriftlig form, blir intervjusamtalene strukturert slik at de er bedre egnet for analyse. Når materialet struktureres i tekstform blir det lettere å få oversikt over det, og struktureringen er i seg selv en begynnelse på analysen» (Kvale & Brinkmann, 2010, s. 189). Transkribering er en tidkrevende prosess, men den vil være med på å bidra til at forskeren setter seg enda mer inn i intervjuene. Det vil også gjøre det enklere å analysere, samtidig som at det er en begynnelse på selve analysen som blir det neste steget etter gjennomført datainnsamling.

I det første steget i analysearbeidet skal en gjøre seg kjent med dataene. Her ble alle transkripsjonene lest tre ganger hver, foruten alle gangene transkripsjonene ble lest gjennom senere i analysen. Grunnen til dette var for å bli ekstra godt kjent med hvert intervju, for en kan ikke begynne med selve analysearbeidet uten å være godt kjent med intervjuene en har gjennomført (Høgheim, 2020, s. 203). Transkripsjonene ble først lest gjennom en gang. Deretter ble et og et avsnitt lest og kommentert. Til slutt ble transkripsjonen lest en gang til og kommentarene som var skrevet til transkripsjonene. Det å

bli godt kjent med dataene sine er et nødvendig og et viktig steg i forkant av det analytiske arbeidet.

Den neste fasen som ble gjennomført i innholdsanalysen var å lage koder. En kode er et ord eller en kort frase som er beskrivende for den representative delen (Rossman & Rallis, 2011, s. 282). I denne fasen skal en videre organisere dataene, og en vil til slutt ha flere deler fra de ulike informantene som ligger under et og samme ord eller frase. I kodingen er det viktig å være bevisst meningsfortettingen, som handler om å korte ned på informantens uttalelser til kortere formuleringer, med andre ord, dataene skal bli mindre, men ikke meningene (Kvale & Brinkmann, 2010, s. 212) (Høgheim, 2020, s. 204). Det var i denne fasen Nvivo ble tatt i bruk, slik at prosessen skulle bli mer organisert og systematisert. Nvivo kunne også ha blitt brukt i transkriberingen av intervjuene, men i transkriberingen ble det brukt Microsoft Word. Etter å ha kodet ferdig ble kodene lest gjennom og det som var plassert inn i dem. Det ble deretter tatt en ny vurdering på det som var plassert i de forskjellige kodene. Til slutt var det en tredje runde hvor dokumentet ble lest gjennom og kodene som var plassert inn i dokumentet, før en ny vurdering ble tatt på de ulike kodene og det som var plassert inn i dem.

I arbeidet med steg 2, så er det lurt å lage en kodebok. «En kodebok er en skildring av kodearbeidet man gjør, hvilken strategi man bruker, betydningen av enkelte koder som kan være vanskelig å forstå senere, og ikke minst memoer om hvorfor og hvordan man har tenkt i kodearbeidet» (Høgheim, 2020, s. 205). Selv om det anbefales å beskrive enkelt koder, så ble alle kodene i denne oppgaven beskrevet. Dette var for å sørge for å ha en klar beskrivelse av de enkelte kodene. Hvis det var usikkerhet på hvor dataene skulle plasseres, så kunne kodeboken tas i bruk. Dette gjorde at kodingen av alle dataene ble konsekvent gjennom hele analysen.

I kodingen har det blitt brukt in vivo-koding som da handler om at det er dataene som skaper kodene, og bruker da språket i dataene til å lage kodene (Høgheim, 2020, s. 204). Selv om det ble brukt in vivo-koding så har det også blitt brukt deskriptive koder. Dette var fordi at av og til inneholdt det informantene sa ikke ord eller fraser som var beskrivende nok, eller som kunne plasseres i koder som eksisterte fra før. Derfor ble det også laget noen koder som

beskrev dataene uten at det nødvendigvis kom fra dataene selv (Høgheim, 2020, s. 204). Det har derfor blitt benyttet både in vivo-koding og deskriptive koder i fasen med å lage koder.

Steg 3 i analysearbeidet var å lage kategorier. Etter å ha delt dataene ned til mindre bestanddeler, så skal man remontere dataene sammen igjen i kategorier, og kategorier er ord eller fraser som er felles eller beskrivende for en gruppe koder (Høgheim, 2020, s. 206). Etter å ha kodet dataene satt jeg igjen med et ganske stort utvalg med koder. For å organisere og systematisere det videre ble det laget kategorier som oppsummerte flere koder. Også i arbeidet med kategorier ble det kodeboken benyttet. Dette var for å ha en beskrivelse av hva som ble lagt i de ulike kategoriene, slik at en til enhver tid kan være konsekvent samt at det hele tiden er klart hvordan en har tenkt i de ulike fasene.

I arbeidet med å lage kategorier ble det i utgangspunktet tatt i bruk induktiv analyse. Det handler om å lage kategoriene på bakgrunn av innholdet i kodene (Høgheim, 2020, s. 207). Induktiv analyse er også særlig egnet når en har fenomenologisk forskningsdesign, slik det er brukt i denne oppgaven (Høgheim, 2020, s. 207). Dette ble benyttet på flere av kategoriene. En av kategoriene fikk også det samme navnet som en kategori i intervjuguiden. Dette var ikke noe som var planlagt, men som ble naturlig i arbeidet med å lage kategoriene. Det i seg selv er likt deduktiv analyse som handler om å ha forhåndsbestemte kategorier (Høgheim, 2020, s. 208). Dette var noe som ikke var forhåndsbestemt at kategorien undervisningspraksis i intervjuguiden og i analysen skulle være de samme, men det ble naturlig da disse var mest passende å bruke i kategoriseringen. Det ble også laget et tema som igjen var beskrivende for alle kategoriene.

Det fjerde steget blir kalt for analyse. I dette steget skal en ta for seg kodene, kategoriene og temaet en har laget for å besvare problemstillingen (Høgheim, 2020, s. 211). En skal videre se på alt en har laget for å finne tekstdata som inneholder betraktningene til informantene, men også finne mønster og se sammenhenger mellom informantene (Høgheim, 2020, s. 211). For å få til dette har det blitt brukt en konvensjonell innholdsanalyse, og målet med denne metoden er å beskrive et fenomen eller flere, i dette tilfelle, for gitte subjekter (Høgheim, 2020, s. 212).

Denne måten å analysere på har likhetstrekk til tematisk analyse, og benyttes også ofte i arbeidet med fenomenologi. Som nevnt tidligere har oppgaven en fenomenologisk tilnærming, som vil si at det skal forskes på læreres opplevelser og meninger om programmering og programmeringsundervisning, og målsetningen vil da være å trekke ut det som er essensen i et fenomen som det forskes på (Høgheim, 2020, s. 212).

Det siste steget i analysen er å trekke slutninger. «Etter hvert som koder, kategorier og eventuelt også temaer vokser fram fra datamaterialet, må man tenke på slutningene man skal trekke. Dette er konklusjonene man skal trekke på bakgrunn av analysene man har gjort om tendenser, perspektiver eller meninger som finnes i datamaterialet» (Høgheim, 2020, s. 214). I arbeidet med å trekke slutninger må en være bevisst på hva som er viktig og hva som ikke er viktig. Slutningene som trekkes ut og dataene som brukes skal være med på å besvare problemstillingen. Som nevnt tidligere, har det blitt brukt en induktiv analyse. Dette brukes for å forstå eller beskrive noe (Høgheim, 2020, s. 215). Når det ble trukket slutninger så måtte det trekkes slutninger som var med på å beskrive begrepet programmering, og slutninger som var med på å beskrive hvordan lærere underviser i programmering.

3.4 Forskningsetikk

Når en forsker på mennesker er det mange etiske hensyn å ta og en må være bevisst de forskningsetiske retningslinjene som eksisterer (Høgheim, 2020, s. 86). Den som intervjuer skal ikke spre dataene og informasjon om informantene til andre, om det er via e-post eller at andre får tilgang til dataene på andre måter. Det er også viktig at dataene er sikret enten med passordbeskyttelse på egen datamaskin, eller at det er lagret på en annen enhet. I denne masteroppgaven blir informantene navngitt ved informant 1, informant 2 og så videre. Det er ikke mulighet for noen å finne ut hvilken person som har sagt hva, ettersom at oversikten på dette vil holdes avskilt fra min personlige datamaskin. Forskningsetikk er noe som påvirker hele forskningen på flere forskjellige måter, og vil spille en rolle gjennom hele forskningen (Høgheim, 2020, s. 86).

«Forskerens rolle som person, forskerens integritet, er avgjørende for kvaliteten på den vitenskapelige kunnskap og de etiske beslutninger som treffes i kvalitativ forskning» (Kvale &

Brinkmann, 2010, s. 92). Min rolle som forsker og hvilke etiske beslutninger som blir tatt vil være avgjørende for hvilken kvalitet forskningen vil få til slutt. Det er dermed viktig at en forsker er bevisst de prosessene som ligger bak en forskningsoppgave. Videre vil det presenteres noen etiske beslutninger og prosesser som en forsker må tenke gjennom i arbeidet med forskning, og som har blitt gjort i denne oppgaven.

3.4.1 Samtykkeerklæring

I denne masteroppgaven har informantene fått tildelt en samtykkeerklæring som de skrev under på før intervjuet startet. Dette for å vise til hvilke rettigheter de har som deltakere i denne masteroppgaven. I samtykkeerklæringen står det om formålet til masteroppgaven, hvem som er ansvarlige for prosjektet, informantenes personvern og rettigheter som informantene har (vedlegg 2). Informantene skal dermed ha tilgang til informasjonen, men som forsker har en også ansvar for at informantene har forstått informasjonen på samtykkeerklæringen (Postholm & Jacobsen, 2021, s. 249). Det er også viktig at informantene som er med på forskningen har takket ja til dette selv, og at dette er frivillig (Postholm & Jacobsen, 2021, s. 248).

Det er viktig i en samtykkeerklæring at informantene får informasjon om de rettighetene de har og formålet med forskningen, samtidig er det viktig å påpeke at det å informere om for mye kan være med på å skade selve datainnsamlingen. «Å informere om en undersøkelse krever nøye avveining mellom på den ene siden å informere for mye og for utførlig, og på den andre å utelate sider ved designen som kan ha betydning for deltakerne» (Kvale & Brinkmann, 2010, s. 89). Ved å gi for mye informasjon om selve intervjuprosessen gir dette informantene muligheten til å forberede seg. En må derfor gjøre en avveining over hvor mye informasjon deltakerne skal få om selve intervjuprosessen. Som forsker skal en derimot sørge for at informantene får vite om alle rettighetene de har som deltakere i denne masteroppgaven til enhver tid, og at de når som helst har mulighet til å trekke seg.

3.4.2 Konfidensialitet

«Konfidensialitet i forskningen innebærer at private data som identifiserer deltakerne ikke avsløres» (Kvale & Brinkmann, 2010, s. 90). I denne masteroppgaven vil deltakerne ikke kunne gjenkjennes med navn, arbeidssted og bosted for å nevne noe. Personopplysninger om informantene er ikke med i denne masteroppgaven, og er heller ikke å oppdrive noe sted. Det er dermed umulig å finne ut hvem som har sagt hva i denne masteroppgaven, og informantene er anonymisert (Høgheim, 2020, s. 90). Informantene har fått hvert sitt nummer, og det nummeret og navnet deres blir oppbevart på forskjellige enheter. Disse dokumentene er passordbeskyttet noe som gjør at det bare er jeg som har tilgang til denne informasjonen. I denne masteroppgaven er personopplysninger ikke relevant for oppgaven, og informantene har dermed ikke trengt å oppgi dette ved intervjustart. Dette gjør at informantene til enhver tid vil være anonymisert i denne masteroppgaven. I tillegg til at informantene anonymiseres så har også forsker taushetsplikt. Det betyr at informasjon om informantene eller det de sier skal ikke deles med noen andre enn de som det er avtalt med på forhånd, det vil i dette tilfellet kun være veileder for masteroppgaven som får tilgang til informantenes betraktninger (Høgheim, 2020, s. 90).

Også i selve analysen av dataene er det viktig å være bevisst konfidensialiteten i forskningen. I arbeidet med å analysere dataene kan det dukke opp problemstillinger rundt hvordan dataene fremstilles. «I den grad det er mulig, skal vi forsøke å gjengi resultater fullstendig og i riktig sammenheng. Sitater som er tatt ut av en større sammenheng, kan ofte få en helt annen mening enn hvis de settes inn i en større kontekst» (Postholm & Jacobsen, 2021, s. 251). I arbeidet med analysen av dataene og skrivingen av analyse og drøfting må en være bevisst på dette. En skal sørge for at data blir presentert så fullstendig som mulig, og at data ikke blir tatt ut av kontekst, slik at informantene blir gitt andre meninger enn det de selv ga uttrykk for. En skal derfor presentere betraktningene til informantene slik de selv gjorde. Det å presentere betraktningene til informantene innebærer også at forsker ikke skal forfalske dataene og resultatene som blir funnet i analysen av dataene (Postholm & Jacobsen, 2021, s. 252).

3.5 Validitet og reliabilitet

I dette delkapittelet vil jeg knytte opp begrepene validitet og reliabilitet til denne masteroppgaven. Jeg vil her vise til hvordan valgene jeg har tatt som forsker påvirker validiteten og reliabiliteten til denne masteroppgaven. Validitet kan ses på som en uttalelses sannhet, riktighet, gyldighet og styrke, mens reliabilitet handler om forskningsresultatenes troverdighet og pålitelighet (Kvale & Brinkmann, 2010, s. 250).

I arbeidet med en masteroppgave er det viktig å være bevisst de utfordringene som er der allerede fra starten av. Forskning på atferd og sosiale fenomener, slik det er gjort i denne oppgaven, vil aldri være verdinøytralt. Dette gjelder både kvalitativforskning og kvantitativ (Postholm & Jacobsen, 2021, s. 92). Dette er for min del, og for andre som benytter seg av kvalitative og kvantitative forskningsmetoder viktig å huske på. «Valg av teori og design, valg av metode, samt egen bakgrunn vil alltid medføre at forskere studerer et utsnitt av virkeligheten. Hvilket utsnitt som studeres, hvordan det studeres, og hvordan det fortolkes og fremstilles, er dermed i seg selv verdiladet» (Postholm & Jacobsen, 2021, s. 92). Alt av valg en gjør i arbeidet med kvalitativ forskningsmetode vil påvirke hvordan masteroppgaven blir seende ut til slutt. Det er dermed viktig å dokumentere gjennom hele prosessen for å sørge for validiteten og reliabiliteten til masteroppgaven. Selv om en masteroppgave skal være objektiv, så vil den aldri være det. Dette er fordi at metode, analyse og drøfting blir gjort på subjektive antagelser. Det vil derfor aldri være helt objektivt, og en må derfor sørge for at forskningen dokumenteres slik at alle valgene en gjør blir synliggjort for de som skal lese masteroppgaven (Postholm & Jacobsen, 2021, s. 93).

3.5.1 Validiteten til denne masteroppgaven

Validitet handler som nevnt om forskningens sannhet, riktighet gyldighet og styrke (Kvale & Brinkmann, 2010, s. 250). I forskning handler validitet blant annet om at metoden som er benyttet egner seg til å undersøke det som den skal undersøke (Kvale & Brinkmann, 2010, s. 250). For å undersøke hva lærere legger i et begrep, å hvordan de underviser i programmering, så ville det være hensiktsmessig å intervjuere lærere. Å bruke spørreundersøkelse ville gitt informantene mulighet til å forberede seg og samtidig gjøre det vanskeligere å stille oppfølgingsspørsmål, dersom de hadde noe annet som kunne vært interessant å spørre mer om.

Det kunne også vært brukt andre metoder innenfor kvalitativ forskning, eller en kombinasjon av flere metoder. Observasjon kunne vært brukt for å se hvordan lærere underviser, og kunne vært med på å gi enda mer konkrete beskrivelser av hvordan de underviser i programmering. Dette ville derimot vært veldig tidkrevende, og det hadde også vært utfordrende å få nok informanter og samtykke fra alle elevene/foreldrene i klassene. For å få et størst mulig bilde av hvordan lærere underviser og hva de legger i begrepet, så vil kvalitativ forskning med intervju være det som gir et best mulig svar på problemstilling, slik at forskningen skulle bli valid. Selv om det er brukt kvalitativforskningsmetode med intervju, så vil ikke det si, som nevnt tidligere, at den er ufeilbarlig. Når lærerne ble intervjuet, så handlet det om deres meninger om programmering og deres erfaringer om undervisning i programmering. Selv om jeg ikke har fått bekreftet dette på noen måte, så betyr ikke det at de lyver, men det er en begrensning ved oppgaven og metoden (Høgheim, 2020, s. 139).

Validitet spiller også en rolle i transkripsjonen. Hva som er en valid transkripsjon eller ikke kan være vanskelig å avgjøre. Hva som er korrekt transkripsjon og hva er nyttig transkripsjon er spørsmål som kan være naturlige å stille seg selv (Kvale & Brinkmann, 2010, s. 194). Ved å ikke ha med pauser, lyder og lignende vil en kunne påføre informantene andre meninger. Også innenfor transkripsjon vil ingen transkripsjoner være objektive, så en må også her dokumentere og forklare hva en gjort for å kunne vise til hvilke valg en har tatt (Kvale & Brinkmann, 2010, s. 194).

«Validitet handler ikke bare om metodene som blir benyttet; forskeren som person, inklusive hans eller hennes moralske integritet og særlig det vi i våre diskusjoner har kalt praktisk klokskap, er avgjørende for evalueringen av den vitenskapelige kunnskapen som blir produsert» (Kvale & Brinkmann, 2010, s. 253). Som nevnt er det derfor viktig å sørge for at oppgaven blir valid at det dokumenteres og at du kan redegjøre for alle de valgene som er tatt.

Kvale og Brinkmann (2010) skriver om standardinnvendinger mot kvalitative forskningsintervjuer, og det å ha intervju blir ikke sett på som en gyldig metode, ettersom det er subjektive meninger som det forskes på (Kvale & Brinkmann, 2010, s. 181). «Å

intervjue er et personlig håndverk, og kvaliteten avhenger av forskerens håndverksmessige dyktighet. Validering blir her et spørsmål om forskerens evne til hele tiden å kontrollere, problematisere og tolke resultatene teoretisk» (Kvale & Brinkmann, 2010, s. 181). For å sørge for at masteroppgaven blir valid, må resultatene til enhver tid tolkes opp mot det teoretiske. Ettersom det er subjektive meninger som kommer frem i intervju, så er det viktig at de behandles som nettopp dette. Det vil si at når det generaliseres så generaliseres det på bakgrunn av de informantene en har fått tak i, og at det er deres meninger som det generaliseres ut ifra.

Når man generaliserer så generaliserer en som nevnt med utgangspunkt bare i sin gruppe av informanter. Noen hevder at det å generalisere ikke er målet med kvalitativ forskning, men at det derimot handler om at en kan trekke slutninger om flere enn de en har forsket på (Høgheim, 2020, s. 154). Dette kalles for ytre validitet. Ytre validitet i min oppgave vil være gjeldende dersom det kan overføres til andre mennesker, og kan være betegnende for også de.

3.5.2 Reliabiliteten til denne masteroppgaven

Reliabilitet handler som nevnt om forskningsresultatenes troverdighet (Kvale & Brinkmann, 2010, s. 250). «Reliabilitet behandles ofte i sammenheng med spørsmålet om hvorvidt et resultat kan reproduseres på andre tidspunkt av andre forskere» (Kvale & Brinkmann, 2010, s. 250). Det vil si at funnene i oppgaven og det som hentes ut av analysen skal kunne hentes ut av andre mennesker ved bruk av de samme dataene. Det vil også si at personer som gjennomfører intervju ved bruk av samme intervjuguide, skal få de samme svarene av informantene (Kvale & Brinkmann, 2010, s. 250). Ved å bruke et semi-strukturert intervju vil det være noen spørsmål som ikke er skrevet ned på forhånd og som stilles til alle informantene. Hvis en annen person skulle gjennomført denne intervjuguiden vil kanskje den personen ha kommet på andre spørsmål som blir spurt der og da. Dette vil igjen ha innvirkning på hvilke resultat og analyser en vil ha i forskningen. Derfor vil det i et semi-strukturert intervju aldri bli helt likt fra gang til gang, og vertfall hvis intervjuet gjennomføres av en annen person. En kan derfor stille spørsmål ved reliabiliteten til alle semi-strukturerte intervju, uten at det betyr at reliabiliteten til oppgaven er svak. For har en et for stort fokus

på reliabilitet vil det kunne være ødeleggende for variasjon og kreativ tenkning, samt muligheten for intervjuere til å improvisere (Kvale & Brinkmann, 2010, s. 250). Det er også verdt å nevne at ledende spørsmål, vil ha en negativt effekt på reliabiliteten til oppgaven, med mindre de er en del av intervjuteknikken som brukes (Kvale & Brinkmann, 2010, s. 250).

Også i transkriberingsprosessen vil reliabilitet ha noe å si. Da særlig i arbeidet med selve transkriberingen. Hvordan blir det informantens sier transkribert, utelater du ord eller skriver du ordrett, vil ha stor betydning på hvordan det informantens sier blir fremstilt. I transkribering i denne oppgaven er det til enhver tid skrevet ordrett det informantens sier. Det er også blitt tatt med tenkepauser og når informantens sier lyder og lignende, som vil ha påvirkning på hvordan det informantens sier blir fremstilt i tekstform. I transkriberingen forsvinner kroppsspråk og holdninger, så for å få det så autentisk som mulig er alt med i transkribering. Skriver en ikke ordrett eller fjerner ord, kan det informantens sier miste sin mening.

Det knyttes også spørsmål til reliabilitet i arbeidet med analysen. Her finnes det strategier for å jobbe med reliabiliteten til egen masteroppgave. I denne oppgaven har det blitt benyttet konstant kodesammenligning som er en metode som brukes når en analyserer alene. Konstant kodesammenligning er en strategi der det handler om å hele tiden sammenligne og analysere koder i analysearbeidet (Høgheim, 2020, s. 216). Denne metoden brukes mest innenfor induktiv analyse, slik som er brukt i denne oppgaven. Gjennom hele analysen skal en ta for seg et og et dokument, lese det, kode det, formulere kategorier. Dette gjentar en med neste dokument, men må da sammenligne nye koder og kategorier som dukker opp, og må hele tiden videreutvikle og formulere nye koder (Høgheim, 2020, s. 216). Det er da hensiktsmessig, å ha en kodebok slik at du enklere kan være konsekvent i arbeidet, og som gjør det enklere å se hva som ligger i de ulike kodene. «Det er viktig å kjenne til hensikten med reliabilitet. Dette er ikke en måte man jobber for objektiv analyse, men heller for konsekvent analyse» (Høgheim, 2020, s. 216). Ved å være åpen og ærlig gjennom hele masteroppgaven vil det bli enklere for utenforstående å sette seg inn i oppgaven for å se hva som er blitt gjort, og hvordan det har blitt gjort. Et viktig aspekt ved reliabilitet når det kommer til analyse er at alle data behandles likt, slik at det blir konsekvent gjennom hele oppgaven (Høgheim, 2020, s. 217).

4.0 Resultater

Resultatkapittelet har jeg delt inn i to deler. I disse to delene vil jeg presentere resultater som jeg i drøftingskapittelet vil drøftes opp mot problemstillingen min: I del 1 vil jeg ta for meg hver informant. Først vil jeg se på hva de legger i begrepet programmering, før jeg tar for meg hvordan de underviser ved å se på direkte spørsmål knyttet til dette. Jeg vil innenfor hver informant presentere kort det som informantene selv sier om spørsmålene knyttet direkte til problemstillingen. I del 2 av resultatkapittelet vil jeg presentere resultatene fra analysen av hele intervjuet. Her vil jeg se på kategoriene og hva som ligger inni disse og presentere dette. Dette for å vise at de kanskje legger andre ting i programmering og underviser annerledes, sammenlignet med det de svarer på spørsmål direkte knyttet til problemstillingen. Under har jeg laget en tabell som viser litt generell informasjon om de ulike informantene.

Tabell 4-1: Viser hvilke lærere som har startet med undervisning i programmering og hvilken opplæring de har.

	Informant 1	Informant 2	Informant 3	Informant 4	Informant 5	Informant 6
Startet med undervisning i programmering	Ja	Ja	Ja	Nei	Ja	Ja
Hvilken type opplæring	Kurs	Studier 30 stp.	Kurs	-	Studier 15 stp.	Kurs

4.1 Del 1 Presentasjon av hver informant

4.1.1 Informant 1

«Da skriver du på PC-en din. Trykker inn noe, også styrer du». Informanten forbinder programmering med noe som foregår på en PC. Dette forteller informanten om begrepet programmering på direkte spørsmål om dette i intervjuet. I samme sekvens stilles det et oppklarings spørsmål for å oppsummere informantens definisjon. «Ja, så programmering for deg er på en måte å gi beskjeder og få den til å gjøre et eller annet?», noe som informanten bekrefter.

Undervisningen har foregått i Minecraft og da med et tverrfaglig utgangspunkt med religionsfaget i skolen. Informanten kommer med opplegget som gikk ut på å bygge religiøse bygg i Minecraft, og påpeker at dette rammer også konstruksjon og måling i matematikkfaget. På spørsmål om hvordan informanten hjelper elevene videre i faget, så forteller informanten om at elevene hjelper seg selv, og at de opptre som «hjelpelærere».

4.1.2 Informant 2

Informanten forteller at programmering kan ses på som en prosess, og trekker frem Linda Manila. Informanten gjentar flere ganger at programmering er mer enn selve kodingen, og at det er mye bredere enn bare selve kodingen. Informanten trekker frem flere elementer som en finner i algoritmisk tenkning, og trekker også frem at det å jobbe med problemløsning er en sentral del av programmeringsbegrepet.

Når det gjelder undervisningen så er den mer løsrevet sammenlignet med andre fag informanten underviser i. Her får de øvd på å stå i oppgavene og løse problemer. Programmeringsundervisningen har vært preget av at elevene skal bli vant til hvordan de ulike programmeringsverktøyene fungerer, og undervisningen har derfor vært lite styrt av læreplanen. Informanten påpeker at elevene lærer mye av hverandre, og at samarbeid er sentralt i undervisningen. Informanten har i undervisningen laget egne opplegg, men også benyttet seg av kidsakoder, som har ferdige opplegg som kan benyttes. Informanten ser på programmering som noe som kan trekkes inn i flere fag. Det blir også trukket fram at elevene må tørre å gjøre feil, og at det er mye læring der.

Informanten har benyttet seg av micro:bit, bit:bot og Scratch. Informanten trekker frem micro:bit og bit:bot som et pluss i programmeringsundervisningen. Undervisningen har blitt påvirket av blokkprogrammeringsprogrammer, og har tatt et bevisst valg om å foreløpig ikke bruke tekstprogrammering.

4.1.3 Informant 3

«Jeg ville forklart at programmering er å innstille en maskin til å utføre noe du ønsker den skal utføre, så kan kalle det for det. Og da må den få noen instruksjoner». Informanten ser på PC som en sentral del av programmeringsbegrepet, og videre å gi den instruksjoner.

Informanten ser ikke noen begrensninger for hvordan programmeringsundervisning skal foregå. Informanten legger opp til at elevene jobber alene eller sammen. Informanten sier at en kan bidra med lesehjelp, og hjelpe så langt en klarer på det nivået informanten selv befinner seg på. I intervjuet forteller informanten om mulighetene for å bruke de elevene som kan mye programmering som «hjelpelærere». Nevner også at det er vanskelig å få elever til å marsjere i takt å få de til å være på nøyaktig samme sted i undervisningen. Derfor vil det egne seg å la de jobbe selvstendig eller sammen to og to. Informanten forteller at det handler om å legge til rette på best mulig måte for at elevene skal lære programmering.

Informanten har brukt micro:bit og Scratch. Undervisningen i programmering startet med Scratch, før de etter hvert har prøvd seg på micro:bit, og de mest ivrige har fått prøvd seg på tekstprogrammering i Python. Informanten trekker frem at det er fine aspekter ved bruk av micro:bit, blant annet det å oppdage og følge instruksjoner slik at programmene læres. Nevner flere ganger at programmering er fin lesetrening for elevene. Informanten har også benyttet seg av kidsakoder.

4.1.4 Informant 4

Denne informanten forbinder programmering med EDB undervisning som informanten hadde på ungdoms- og videregående skole. Informanten forbinder programmering med å

skrive kommandoer inn på en PC, som igjen da vil utføre en handling. Trekker frem flere eksempler i forklaringen av begrepet, men at det handler om få PC-en eller andre ting til å utføre noe.

Selv om informanten ikke har startet med undervisning i programmering, så har informanten fått med seg at Minecraft kan brukes i programmeringsundervisningen.

4.1.5 Informant 5

«Det handler jo på en måte om å skrive algoritmer, ehm instruksjoner som en datamaskin kan følge da». Informanten definerer programmering som noe som omhandler en PC.

Informanten ser på programmering som å skrive instruksjoner inn på en PC som utfører disse handlingene.

På arbeidsplassen til denne informanten har de laget et eget programmeringsfag for hele skolen, hvor de har lagt inn litt naturfag, musikk, kunst og håndverk og matematikk, slik at alle fagene som har programmering blir representert. Denne informanten har ansvar for all undervisning i faget. Dette gjør ifølge informanten, at elevene blir godt kjent med ulike programmeringsprogrammer, og de får også muligheten til å ha prosjekter som går over lengre perioder, når de har en halvtime til en time fast i uka med programmering.

Informanten har også vært opptatt av at undervisningen skal være utforskende. I undervisningen ønsker informanten at elevene skal undre og være nysgjerrige, og øve seg på å finne ut ting selv før de tester koden.

Informanten har benyttet seg av micro:bit og bit:bot. MakeCode arcade er et annet verktøy som informanten har benyttet seg av, hvor de har laget pixel art spill. Scratch og Scratch junior har også blitt brukt.

4.1.6 Informant 6

«Det å se mønster og finne mønster, og finne. Være en liten detektiv, sant. Altså, det er ikke noe mer enn det, men det er og matte». Informanten definerer programmering som å se og finne mønster.

Programmering prøver informanten å ikke gjøre for komplisert for elevene, slik at alle skal få det til. Informanten sier selv at både elevene og læreren lærer underveis i timene, og er derfor opptatt av at elevene skal lære å stå i det. I undervisningssituasjonene er informanten opptatt av at elevene skal få minst mulig hjelp, slik at de øver seg på å løse problemene selv. I undervisningen har informanten latt seg inspirere av den nye læreplanen med tanke på at elevene skal utforske. Informanten har knyttet programmering til andre fag.

Informanten har benyttet seg av bit:bot og micro:bit. Poengterer i intervjuet at bruk av bit:bot og micro:bit er enkelt for elevene. Informanten har også brukt Scratch, og har knyttet det opp mot matematikken med koordinatsystem, og at elevene har laget sine egne multiplikasjonsspill. Informanten har tatt et bevisst valg om å hovedsakelig bruke blokkprogrammering i undervisningen, ettersom det er dette som er enklest for elevene. Informanten har testet tekstprogrammering sammen med elevene, men dette opplevde informanten som vanskelig for elevene og seg selv.

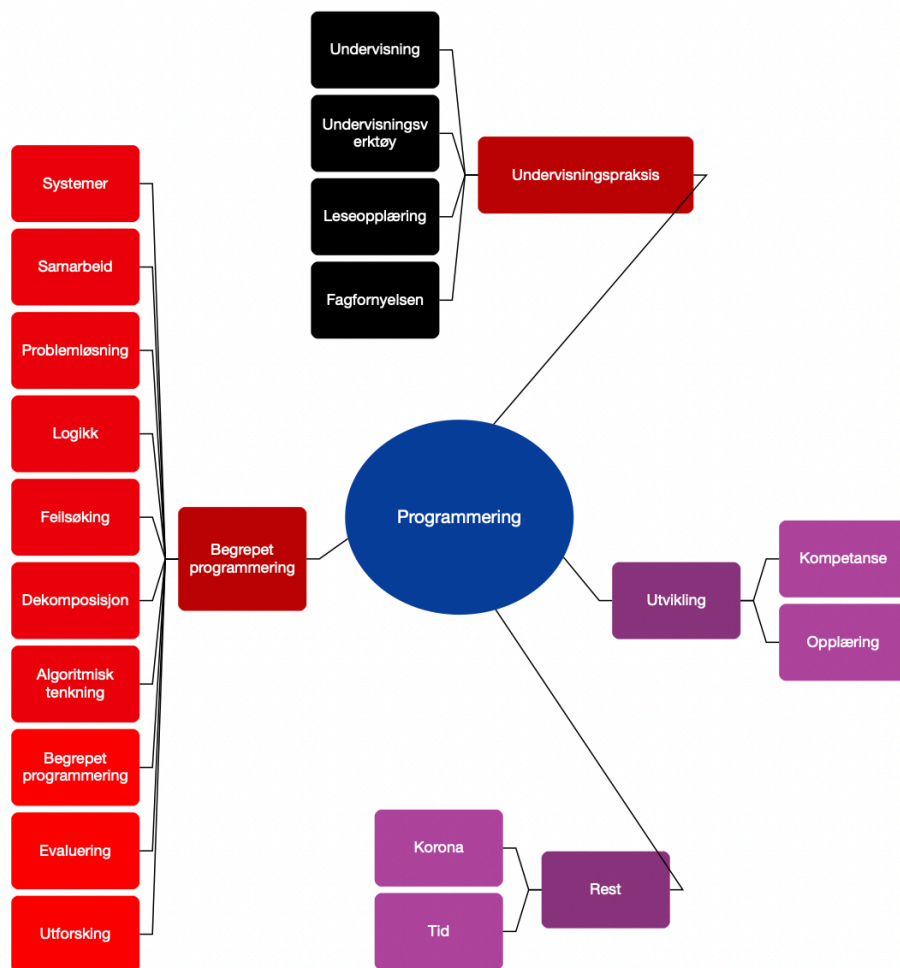
4.2 Del 2 Resultatet av analysen

I denne delen vil jeg presentere resultatene av analysearbeidet med dataene. Her vil jeg ta utgangspunkt i kategoriene jeg har laget og kodene som ligger under kategoriene. Disse vil jeg bruke for å belyse problemstillingen min. I neste delkapittel drøfter jeg resultatene i analysearbeidet mitt. I steg 3 i analysearbeidet arbeidet jeg med å kategorisere dataene mine. Her lagte jeg kategorier som skulle være med på å systematisere og beskrive kodene jeg hadde laget. Jeg vil først se på kategorien «begrepet programmering» før jeg tar for meg kategorien «undervisningspraksis».

Jeg har laget en modell (se figur 4-2) for å vise hvordan temaet, kategoriene og kodene henger sammen. Som en kan se så har noen av kategoriene fått flere koder lagt inn under seg. Kategoriene er de som er knyttet til tema programmering, og koder henger igjen på

kategorier. For eksempel undervisningspraksis er en kategori, og undervisning, undervisningsverktøy og leseopplæring er koder som er festet til kategorien undervisningspraksis.

Figur 4-2: Modellen viser hvordan tema, kategorier og kodene fra analysen min henger sammen.



De forskjellige kodene og kategoriene har ulikt meningsinnhold. Kodene under er de som jeg har benyttet meg av i analysearbeidet og forklaringene som jeg benyttet meg av i kodeboken. Fire av kodene som fagfornyelsen, kompetanse, undervisning og undervisningsverktøy har igjen underkategorier.

- **Algoritmisk tenkning:** Informantene tar selv opp begrepet algoritmisk tenkning og snakker om begrepet.
- **Begrepet programmering:** Hva informantene selv legger i begrepet programmering på direkte spørsmål om hva de legger i programmering.
- **Dekomposisjon:** Når informantene snakker om dekomposisjon og det å bryte ting ned i mindre deler.
- **Evaluering:** Handler om når informantene snakker om evaluering i programmering, om å vurdere løsningene sine og videreutvikle dem.
- **Fagfornyelsen:** Informantenes arbeid med fagfornyelsen og hvordan den har påvirket deres lærerhverdag og undervisning, og da også noen begreper som kan knyttes til fagfornyelsen.
 - Digitalisering: Handler om at informantene jobber mer digitalt og at hverdagen er blitt mer digitalisert enn tidligere, og som grunnleggende ferdighet.
 - Dybdelæring: Handler om når informanten snakker om dybdelæring i forbindelse med fagfornyelsen eller i andre sammenhenger.
 - Kompetansemål: Handler om det som har med kompetansemål i den nye læreplanen å gjøre.
 - Programmering inn i læreplanen: Handler om hva informantene synes om at programmering er kommet inn i læreplanen, holdningene deres knyttet til dette, enten om det er positivt eller negativt. Handler også om hvorfor de mener at programmering skal inn i læreplanen.
 - Tema: Handler om at informantenes fokus på temabasert undervisning.
- **Feilsøking:** Handler om når informantene tar opp feilsøking i arbeidet med programmering.
- **Kompetanse:** Handler om informantenes kompetanse. Da sin vurdering av seg selv og hvor de plasserer seg selv sammenlignet med resten av sine kollegaer. Handler også om kompetansen som finnes på arbeidsstedet når informanten selv beskriver det.
 - Høy: En av de som kan mest.
 - Middels: Midt på treet.
 - Lav: En av de som kan minst.

- **Korona:** Handler om ting som har med Corona å gjøre, ettersom dette har vært en stor del av hverdagen til folk de siste to årene.
- **Leseopplæring:** Handler om når informantene snakker om programmering og leseopplæring.
- **Logikk:** Informantene trekker frem logikk som noe som kan øves på gjennom programmering, og dette som har med logisk tenkning å gjøre.
- **Opplæring:** Her snakker informantene om sin opplæring i programmering. Hvilken opplæring de har hatt enten om det er formell kompetanse i form av kursing eller studier, eller gjennom fellesøkter på skolen. De forteller også om hvordan de opplevde opplæringen de fikk og om hvilken opplæring de skulle ønske de hadde/om de ikke ønsker mer opplæring. Her snakkes det dermed også om arbeidsstedet har prioritert programmering i tirsdagstiden ettersom dette har noe med opplæringen å gjøre.
- **Problemløsning:** Handler om når informantene snakker om problemløsning i forbindelse med programmering, og når de ser sammenhenger mellom programmering og problemløsning.
- **Samarbeid:** Handler om når informanten snakker om samarbeid i forbindelse med programmering.
- **Systemer:** Handler om når informantene snakker om systemer i forbindelse med programmering.
- **Tid:** Handler om tidsaspektet i skolen.
- **Undervisning:** Handler om arbeidsmåter, hvordan de legger opp undervisningen, underviser de annerledes sammenlignet med andre fag, hvordan de har undervist i programmering om det er som noe eget eller kombinert med andre fag. Handler også om hvordan de hjelper elevene videre i faget, og hvordan de opplever elevene i faget.
 - Arbeidsmåter: Handler om arbeidsmåtene de bruker i undervisningen av programmering.
 - Elever: Handler om hvordan informantene opplever elevene i undervisningen av elevene. Handler også om hvordan elevene arbeider og informantenes oppfatning av hvordan elevene arbeider.

- Startet med undervisning i programmering: I intervjuet har jeg stilt informantene om de har startet med programmering. Her lages det en oversikt for å se hvem som har startet med programmering og ikke.
- Veiledning av elever: Hvordan hjelper informantene elevene videre i programmering enten om de mestrer faget eller om de sliter.
- Vurdering av elever: Handler om hvordan informanten vurderer læringen til elevene sine i programmering.
- **Undervisningsverktøy:** Handler om når informantene snakker om programmer som de bruker i undervisningen. Inneholder også konkrete verktøy som kan brukes i programmeringsundervisningen. Handler også om hvorfor de bruker de programmene de gjør. Handler også om de har benyttet seg av analogprogrammering, blokkprogrammering og tekstprogrammering.
 - BitBot_Microbit: Handler om når informantene snakker om BitBot og Microbit. Ettersom disse er nært knyttet sammen i undervisningen, så er de slått sammen til en kode.
 - Blokkprogrammering: Når informanten snakker om bruk av blokkprogrammering i undervisningen sin.
 - Egenkomponerte opplegg: Dersom informanten har konstruert egne opplegg som informanten har gjennomført i undervisning.
 - Kidsakoder: Når informanten snakker om Kidsakoder og bruken av det i undervisningen.
 - Makecode arcade: Handler om når informantene snakker om dette programmet.
 - Minecraft: Handler om det som informantene snakker om når det gjelder Minecraft.
 - Scratch: Når informantene snakker om bruken av Scratch i undervisningen sin.
 - Teknolab: Handler om bruk av Teknolab med blant annet Lego.
 - Tekstprogrammering: Handler om når informantene snakker om for eksempel koding i python og generelt om tekstprogrammering.
- **Utforskning:** Handler om når informantene snakker om utforskning i programmering og i undervisningen sin.

Hvorfor data har blitt plassert i den gitte koden og ikke en annen kode, er vurderinger som må gjøres. Ved å ha en kodebok så hjelper dette med å holde kodingen konsekvent. I et av

intervjuene sier informant 3: «Hvorfor går ikke den katten til høyre når musen går til venstre opplegg. Hva er det som går feil, så den der analysebiten som jeg ser, det er kanskje den største likheten, men det er logiskmatematisktenking». Dataene her kan plasseres i koden som handler om feilsøking, plasseres i logikk og deles opp og plasseres i både logikk og feilsøking. Jeg tok valget om å legge den i logikk. Ettersom at informanten selv definerte dette som logiskmatematisktenkning valgte jeg å legge det i koden logikk. Hvis informantene selv definerer et begrep så valgte jeg å legge det inn i den koden de definerte det som, selv om innholdet kunne plasseres inn i andre koder også.

Jeg kunne også ha laget en kode som het algoritmer og en som het mønster. Det er to informanter som snakker om algoritmer og en som snakker om mønster. Dette snakker de om i forbindelse med et spørsmål som handler direkte om programmering, og hva de legger i det begrepet. I kodingen av dataene mine tok jeg et valg om å skille mellom det de svarte direkte på spørsmål om programmering og det som kom frem senere i intervjuet. Derfor ble det som de tre informantene sa om algoritmer og mønster lagt inn i koden begrepet programmering.

Det som ligger som meningsinnhold i kategoriene mine er:

- **Undervisningspraksis:** Handler om alt som har med programmeringsundervisningen til informantene å gjøre. Alt fra didaktiske valg til verktøy de bruker i undervisningen.
- **Begrepet programmering:** Handler om alt informantene sa som knyttes til begrepet programmering.
- **Utvikling:** Handler om hvilken kompetanse og opplæring informantene har fått, men også om hvilken kompetanse og opplæring som har foregått på arbeidsteden til informantene.
- **Rest:** Jeg lagde også en rest kategori som handler Korona og tid. Dette er koder som jeg ikke har funnet plass til i de andre nevnte kategoriene.

4.2.1 Begrepet programmering

I kategorien «begrepet programmering» ligger det ti koder med forskjellig meningsinnhold. Denne kategorien er med på å hjelpe meg med å se på del 1 av problemstillingen. Den første koden som trekkes frem, er «algoritmisk tenkning». Det er to informanter som refererer konkret til begrepet algoritmisk tenkning. Informantene er innom 5 av 6 begreper i algoritmisk tenkning, og de kommer selv ikke innom abstraksjon. Informant 6 definerer som nevnt tidligere programmering som det å se mønster, som en viktig del av det, som for øvrig er et av begrepene som brukes for å forklare algoritmisk tenkning.

Det å se mønster og finne mønster, og finne. Være en liten detektiv, sant (Informant 6).

Det er også to informanter som nevner ordet algoritme i forbindelse med definering av begrepet programmering som er et annet begrep i algoritmisk tenkning.

Programmering vil jeg si er på en måte mer enn bare selve kodingen, altså følge algoritmene (Informant 2).

Begrepet programmering, ehm ja. Det handler jo på en måte om å skrive algoritmer, ehm instruksjoner som en datamaskin kan følge da. (Informant 3).

I koden «logikk», som er et underbegrep til algoritmisk tenkning, så er det tre informanter som snakker om logiskmatematisktenkning og logisk resonnering når det senere i intervjuet snakkes om sammenhengen mellom matematikk og programmering. En av informantene ser på logikk som noe som er konkret og viktig for elevene:

Jeg synes logikk da. Gjør det, det er veldig konkret. Altså, du får veldig raskt respons på det du gjør ... Logisk tenkning er noe elevene trenger å øve mye på (Informant 1).

To av informantene peker også på logiskmatematisktenkning i sammenheng med å skrive kommandoer, og informant 3 knytter dette i tillegg opp mot feilsøking.

Hvorfor går ikke den katten til høyre når musen går til venstre opplegg. Hva er det som går feil, så den der analysebiten som jeg ser, det er kanskje den største likheten, men det er logiskmatematisktenking (Informant 3).

Det å forstå hvordan ting henger sammen (Informant 3).

Ja, det er logiskresonnering, og altså det er jo matematisk, ehm. De må jo gjøre, de må sette inn rett kommandoer for at den skal gjøre det som du ønsker ... (Informant 4).

Dekomposisjon er det to informanter som er innom i sine intervju.

Dette med å sitte med litt sånn dekomposisjon, mindre deler (Informant 2).

Altså det å bryte ned (Informant 3).

Det er også en av informantene som snakker om evaluering i intervjuet.

Ja, jeg er mer opptatt av prosessen enn selve resultatet. Så jeg tenker oppgaven i seg selv, ehm, greit nok at hvis de løser oppgaven, men så tenker jeg, ja hva nå? Også videreutvikler jeg, sant, så jeg ser ikke noen begrensninger den veien heller. Altså, det er ikke to streker under svaret, ferdig. Jeg tenker og kan vi bruke det mer, hva mer kan vi putte på, å ja, vi kan forbedre dette her (Informant 2).

Samarbeid er et begrep som kommer opp under arbeidsmåte i den algoritmiske tenkeren. Dette gjelder også de neste kodene i resultatdelen. To av informantene er innom samarbeid og samarbeidsoppgaver, og viktigheten av dette. Informant 2 nevner samarbeid sammen med en rekke andre ting som er viktig å øve på i arbeidet med programmering. Informant 3 sier flere ting om samarbeid, men sier også at det er bra for de introverte menneskene som liker å holde på for seg selv.

Det er knallbra samarbeidsoppgaver, for disse her introverte menneskene som liker å tutle på for seg selv og skape resultat, knallbra. Ja, og det er også, altså apropos programmering, det er en veldig fin ting å samarbeide, vel å merke to og to, og vel å merke kanskje to og to på et relativt likt nivå, for hvis det er et veldig stort sprik enten i maktforhold, altså hakkeorden innad i mellom de to, da vil alltid en dominere og den andre bare godta, og hvis den ene er mye flinkere til å kode enn den andre, så vil den bare gjøre det, og den andre ikke brøyte seg vei for, for at han er opptatt av å lære, han vil bare bli underdanig, så sånn sett, det er vertfall min erfaring, like barn leker best. Ehm, men, hvis, hvis du finner en god match så er det faktisk, da får de en fellesskapsfølelse av å klare ting sammen (Informant 3).

Det er to av informantene som er innom feilsøking i intervjuet, og sier blant annet dette:

Feilsøking for eksempel, dette her, dette med å forutse hva kommer til å skje, hva tror jeg kommer til å skje (Informant 2).

... når du finner feil må du ta å feilsøke (Informant 3).

Det er tre av informantene som er innom det å ha utforskende undervisning, eller dette med å fikle, som er et ord som også kan brukes om dette. I intervjuet har informantene ikke blitt vist eller introdusert for den algoritmiske tenkeren av meg i intervjuet. Informant 2 sier i det neste sitatet: som du har med. Da snakker informanten om den algoritmiske tenkeren selv og refererer da til den algoritmiske tenkeren som «du».

At de skal tørre å fikle ..., fikle å holde på. Udir har jo denne algoritmiske tenkeren, sant, som, ehm, som du har med. Du har arbeidsmåter, og går gjennom disse punktene (Informant 2).

Det har vi kanskje brukt litt mer utforskende, sånn at de må finne ut og prøve seg frem selv og teste ut koden sin ... Så der har det vært litt mer utforskning (Informant 5).

La de få forske ut og bruke den tida (Informant 6).

Fire av informantene kommer innpå at programmering egner seg til å arbeide med problemer. Både informant 3 og 6 nevner problemløsning og det å arbeide med problemer, videre sier informant 2 og 5 dette:

Jeg tenker at når du holder på med programmeringen så sitter du og, du sitter og. Problemløsning er et stikkord som kommer igjen her (Informant 2).

Også er det, er det jo en fin måte å lære seg å løse problemer på (Informant 5).

4.2.2 Undervisningspraksis

Kategorien «undervisningspraksis» er med på å hjelpe meg med å svare på den andre delen av problemstilling som handler om hvordan underviser lærere i programmering på 5-7. trinn. Undervisningspraksis inneholder 4 koder med ulikt meningsinnhold. 3 av disse kodene hadde igjen underkoder med ulikt meningsinnhold.

Undervisning er en kode som har noen underkoder. Disse handler om arbeidsmåter, veiledning av elever, om de har startet med undervisning, elevers oppfatning av faget og vurdering av elever. Fire av informantene snakker om det å knytte programmering opp mot andre fag, og informant 1 trekker frem et eksempel med å ha brukt programmering i et tverrfagligprosjekt.

Ehm, vi kan holde på med masse forskjellig innen dette, det er mange flere muligheter og det stenger ikke så veldig mye. Det er derfor jeg sier at du kan dra det inn i veldig mange forskjellige fag og holde på (Informant 2).

Men altså, ikke bare matte og programmering, det er ikke der. Programmering hører til i alt (Informant 6).

På skolen til informant 5 så har de laget et eget programmeringsfag, hvor alle elevene får en halvtime til en time fast i uka med programmering.

Og da gir det jo meg muligheter for å jobbe mer tverrfaglig og kanskje mer helhetlig med det egentlig og. At det ikke blir sånn avgrenset at nå programmerer vi matte, nå programmerer vi musikk, og nå programmerer vi kunst og håndverk, men det blir, blir en helhetlig pakke av det da. De bruker jo elementer fra flere forskjellige fag når vi holder på (Informant 5).

Nei, jeg har ikke knyttet det opp mot andre fag ... så jeg har undervist programmering som programmering i seg selv (Informant 3).

Det er også en av informantene som trekker frem at undervisningen har vært mer fri sammenlignet med andre fag i skolen som de underviser i:

Jeg er nesten helt løst. Jeg har aldri hatt noe som på en måte så er så lite, gjerne så går jeg bare litt gjennom i begynnelsen hvilke begreper vi skal lære oss litte granne og oppsummerer litte granne, og det er veldig løse oppgaver, altså, og hvis vi på en måte, hånden kommer opp, så er det på en måte, de får ikke noen løsning når de sitter og fikler og trykker og prøver i lag, og vi. Vi prøver på en måte å få inn litt, litt tankegangen rundt hvordan vi gjør det med programmering, dette med, disse tingene vi var innom i sted, vi prøver på en måte å snu litt at de også jobber litt sånn. Og det tenker jeg at det må ha kjempenytteverdi i for eksempel matematikken (Informant 2).

Ettersom at ikke alle lærerne har fått like mye opplæring så er det tre informanter som trekker frem aspektet med at elevene kan brukes som hjelpelærere i undervisningen, og flere bruker dette aktivt i undervisningen. De trekker også frem dette med at elevene lærere samtidig som lærer:

De hjelper seg selv, de, de har jo lært meg. Ehm, de synes det er veldig, veldig stas (Informant 1).

Men, jeg tenker jo vi lærer jo ofte litt i lag, og mange elever kan jo litte granne fra før (Informant 2).

men jeg, jeg må legge ting til rette for at de selv kan finne ut av det. Og så kan når, for det at, det er elever som blir trigget, og når noen blir trigget og driver dette videre, og så kan jeg bruke de som lærere, så da sender jeg andre i klassen, så når de står fast og jeg ikke kan hjelpe de, så sender jeg de til de elevene som jeg vet kan, også sier jeg spør om de kan hjelpe (Informant 3).

Selv om informant 4 ikke har begynt med undervisning i programmering ser også denne informanten nytteverdien av å bruke elevene som hjelpelærere:

Også må en ikke være redd for at elevene kanskje kan mer enn deg. La elevene få sleppe til, elevmedvirkning, ikke være redd for å la de vise og forklare andre, og bruke de som en ressurs (Informant 4).

Informantene har brukt flere undervisningsverktøy i undervisningen, og forklarer også hvorfor de har brukt akkurat disse, og hvorfor de eventuelt har utelatt andre.

Bit:bot og micro:bit er det fire informanter som har benyttet seg av i undervisningen. Informant 3 og 5 snakker om at de har brukt det litt i undervisningen sin og prøvd det ut sammen med elevene.

Jeg har brukt micro:bit. Jeg har jobbet litte granne ut ifra det. Ehm, micro:bit har vi egentlig brukt veldig mye. Det som er greit med micro:biten er jo at, vi har jo klasesett på skolen ... Jeg har jo litt tro på at de får sittet med hver enkelt sin (Informant 2).

Og når de kommer inn i det der, når de sitter med bit:bottene og lar de gå, og microchiper, da, da tror veldig mange at det blir komplisert. Men når de ser at det ikke er det, så kommer de på igjen (Informant 6).

Informant 1 har kursing i dette, men har ikke benyttet seg av dette enda, men det er noe som kan tas frem og brukes når det skulle bli aktuelt:

Vi fikk med oss noen roboter, og vi har noen sånne, mus eller noe sånt noe. Så det står slik at vi skal kunne bruke det (Informant 1).

Fire av informantene har også benyttet seg av Scratch, og informant 2 og 3 sier dette:

Jeg har jo brukt programmert i Scratch ... Scratch er jo veldig sånn nybegynneraktigprogram (Informant 2).

Ja, jeg har brukt ganske mye Scratch ... det var det vi begynte med, eller før det igjen så begynte vi med denne her ballen som ruller høyre, venstre, opp, ned. Ehm, det var på Ipad i sin tid. Også begynte vi å kode i Scratch (Informant 3).

En av informantene har også benyttet seg av et program som heter Makecode Arcade. Dette er det ingen av de andre som har testet ut, men informanten sier dette:

Også har vi brukt et program som heter Makecode arcade. Der lager vi sånn pixelart spill da, også lager vi noen skikkelige enkle versjoner av det. Så det har vi gjort der da ... så vi har holdt på veldig mye med koordinatsystemet og ehm, negative tall og sånn når vi har holdt på i den Makecode arcade appen, og laget sånn pixelartspill i de større trinnene og. Og da har jeg bedt dem om å finne ut om de må ha et negativt tall eller et positivt tall for å få figuren til å hoppe, og sånne ting (Informant 5).

Informant 1 har brukt Minecraft Education i sin undervisning, og programmeringsmuligheten du kan benytte deg av i det programmet. Informant 4 som ikke har brukt programmering i undervisningen enda er klar over at Minecraft kan brukes i programmeringsundervisningen.

Det er også to av informantene som har benyttet seg av de ferdige oppleggene som en kan finne på Kidsakoder:

alt i fra å bruke dette litt sånn Kidsakoder og liksom ferdige opplegg (Informant 2).

Den sida med Kidsakoder er kjempebra, for der ligger det mye ferdig opplegg, og de oppleggene med instruksene, og det er jo klart at når ungene har gjennomført 10 sånne forskjellige så har ungene fått et bitte litt lite grunnlag til å kunne kode videre på (Informant 3).

Fire av informantene har også tatt bevisste valg når det kommer til blokkprogrammering og tekstprogrammering. Når det gjelder blokkprogrammering sier informantene dette:

da har jeg valgt bevisst å bruke på en måte, ehm, litt sånn blokkprogrammering. Jeg følte at det var det mest naturlige å begynne med (Informant 2).

Det er lett for elevene, sant, blokkprogrammering er ekstremt lett. Det gjør jeg. Det er derfor jeg bruker det, for det er lett tilgjengelig (Informant 6).

Når det gjelder tekstprogrammering så sier informantene at det foregår på et nivå som er for høyt for sin elevgruppe, samtidig som det er på et litt høyere nivå enn det informantene selv er på:

så tenker jeg det blir for mye med disse ehm Python og alt dette her. Jeg tenker, ehm, vi får se litte granne, men jeg tenker at vi er ikke kommet helt der enda. Og jeg heller er ikke kommet helt der at jeg er trygg nok og ta dette helt inn i klasserommet enda (Informant 2).

Nei, vi har ikke holdt på med noe tekstprogrammering ... for går man over til tekstprogrammering vertfall i barneskolen så er det jo bare på engelsk, og da blir det mange som detter fra. Og det er mye å holde styr på, med å få riktig syntaks og semikolon og alt på riktig plass (Informant 5).

Det er også to informanter som har prøvd tekstprogrammering i klassen, eller latt de elevene som har hatt et ønske om det få teste det ut:

... også var det han, han som dro det lengst ble veldig interessert i å begynne å skrive kode i Python. Det kan jeg ingenting om, så da sa jeg at du må finne ut av det selv (Informant 3).

Nei, altså jeg prøvde meg og elevene. Og nå var det en stund siden jeg hadde hatt python da, og det gikk ikke så veldig bra. Så vi måtte gi oss lite grann, men det var greit (Informant 6).

Rest-kategorien handler om koder som jeg ikke har funnet plass til i andre kategorier. I denne kategorien har jeg plassert kodene tid og Korona. Selv om denne kategorien blir presentert i resultatet, så har den hatt påvirkning på lærerhverdagen de siste par årene. Korona har påvirket innføringen av den nye læreplanen, noe som igjen har påvirket tiden lærere har hatt til rådighet og disponeringen av den.

5.0 Drøfting

Drøftingskapittelet er lagt opp likt som resultatkapittelet. Først vil jeg drøfte resultatene i del 1 som handler om det som informantene svarer direkte på spørsmålene jeg stiller om begrepet programmering og undervisning, altså del 1 i resultatkapittelet. I del 2 vil jeg drøfte rundt kategoriseringen og kodingen av dataene mine, og hva som er resultatene av dette arbeidet. Til slutt trekker jeg noen konklusjoner, før jeg også kommer med muligheter til videre forskning innenfor dette temaet.

5.1 Del 1: Hva svarer informantene på spørsmålene som er direkte knyttet til problemstillingen

5.1.1 Hva legger et utvalg matematikklærere i begrepet programmering

Blackwell (2002) skriver om programmeringsbegrepet og at det har endret seg opp gjennom årene (Blackwell, 2002, s. 204). Sevik & m.fl. (2016) definerer programmering som noe mer enn bare det som skjer digitalt med å skrive kode, men også selve prosessen som ligger til grunn i arbeidet med å lage kode, enten om det er digitalt eller analogt (Sevik & m.fl., 2016, s. 9). Sanneutvalget (2016) definerer begrepet slik: «Programmering vil si å bryte et gitt problem ned i et sett av kommandoer og så få en datamaskin til å utføre disse kommandoene» (Sanne et al., 2016, s. 18). Mannila (2017) skriver om at programmering handler om prosessen med å komme frem til riktig svar (Mannila, 2017, s. 63). I teorien så finnes det ulike definisjoner, samtidig er det likhetstrekk mellom definisjonene. Det som går igjen i definisjonene er bruk av datamaskin, og at det ikke nødvendigvis er svaret som er det viktigste, men hele prosessen som ligger bak. Dette kan en også se i mitt utvalg av informanter. Sevik og Mannila har likhetstrekk i sine definisjoner med at programmering er mer enn bare koden, og at det handler om prosessen som ligger bak. Sevik og Sanneutvalget har også med datamaskinen i sine definisjoner, noe som Mannila ikke har. Fire av seks informanter definerer programmering med å skrive inn eller gi datamaskinen instruksjoner som den skal utføre. Dette i likhet med Sanneutvalget (2016) som definerer programmering som noe som skjer på en datamaskin (Sanne et al., 2016, s. 18).

Det er videre interessant å se informant nummer 2 sine betraktninger rundt definisjonen av programmering. Informanten holder selv på med studier innenfor programmering, og ser i likhet som Sevik og Mannila på programmering som noe mer enn selve kodingen. I intervjuet henviser informant nummer 2 til Linda Mannila og hennes definisjon av programmering, at programmering kan ses på som en prosess og ikke bare som selve kodingen (Mannila, 2017, s. 63). Videre refererer informanten til algoritmisk tenkning som Utdanningsdirektoratet selv har på sine planer (Utdanningsdirektoratet, 2020). At programmering er mer enn selve kodingen kommer også flere informanter inn på senere i intervjuet, men når de blir stilt

spørsmål direkte om hva de legger i begrepet programmering, så står datamaskinen og det å skrive programkode sentralt. Informant 2 har en bredere forståelse knyttet til programmeringsbegrepet. Dette kan henge sammen med at informanten gjennomgår studier i datainnsamlingsperioden og at informanten dermed har ny og fersk kunnskap på dette temaet, noe som Bocconi et al (2016) poengterer at lærere trenger en grundig opplæring for å få gode begreper og kunnskaper på plass (Bocconi et al., 2016, s. 7).

I del 2 i drøftingskapittelet vil jeg drøfte rundt det som kom frem senere i intervjuet, for å vise at informantene kanskje legger noe annet i begrepet, enn det de gir uttrykk for når jeg stilte spørsmålet om hva de legger i begrepet programmering.

5.1.2 Hvordan underviser et utvalg lærere på 5-7. trinn i programmering

I resultatkapittelet så kommer det frem at informantene har flere måter å undervise i programmering. Fem av seks informanter har startet med programmering, men det er variasjon i hvor mye de har prøvd det ut. På grunn av pandemien som har påvirket samfunnet de siste par årene har det medført at eventuell opplæring har utgått, siden lærere ikke har kunnet samles på et sted. Dette forteller informantene om i sine respektive intervjuer. Ting er blitt satt på vent under Koronaen, og andre ting enn programmering har blitt prioritert, men at det gjerne blir prioritert nå som samfunnet går tilbake til normale tilstander. Nå som algoritmisk tenkning er kommet inn i skolen og hverdagen blir mer normalisert, så trengs det mye opplæring og opplæring av flere lærere. Dette kommer til uttrykk i en rapport fra EU (2016) om computational thinking, ettersom at innføringen av algoritmisk tenkning tar tid (Bocconi et al., 2016, s. 7).

Når det gjelder undervisningsverktøy så er det flere som kan tas i bruk ifølge informantene mine. Undervisningen til informantene mine har i all hovedsak vært preget av blokkprogrammering, hvor to av informantene har testet tekstprogrammering. Fire av seks informanter har benyttet seg av micro:bit og Scratch, som er blokkbaserte programmer. Disse programmene blir brukt ettersom det er programmeringsprinsippene som står i fokus, istedenfor detaljer og syntaks, slik det er i tekstprogrammering (Statped, 2021b). Ettersom at Scratch er laget spesielt for barn og unge, så kan dette være et naturlig program å bruke i

undervisningen (Mannila, 2017, s. 130). Selv om det ikke er konkrete kompetansemål før etter 4 trinn som konkret handler om programmering, så skal det øves på algoritmisk tenkning før 4 trinn (Kunnskapsdepartementet, 2019b). Det er likevel mulighet å innføre dette og tilpasse dette til kompetansemål før 4 trinn, og bør læres så tidlig som mulig (Hsu et al., 2018, s. 298). Da kan en inngang være bruk av nettopp blokkprogrammering, ettersom dette gir mye visuell støtte for elevene (Statped, 2021b).

Tre av informantene har tatt bevisste valg om å starte med blokkprogrammering, ettersom disse blir sett på som enklere for elevene. Dette kommer også til uttrykk i litteraturen (Statped, 2021b) (Mannila, 2017, s. 188). Både i litteraturen og ifølge informantene mine, kan det å starte med blokkprogrammering være en måte å starte med undervisning i programmering for elevene, før de etterhvert går over til tekstprogrammering (Statped, 2021b) (Mannila, 2017, s. 189).

Det er både likheter og ulikheter i undervisningsverktøyene til informantene mine. Det flere måter å undervise på, og ved bruk av ulike verktøy så gir dette muligheter for individuelle tilpasninger (Statped, 2021b). Jeg vil drøfte enda mer i kapittel 5.2.2, ettersom jeg i dette delkapittelet kun drøfter det som er presentert i del 1 i resultatet. I del 2 vil jeg komme mer inn på drøfting av programmeringsverktøy, didaktiske valg og organisering av undervisningen.

5.2 Del 2: Drøfting av resultatene fra kategoriseringen av dataene

5.2.1 Begrepet programmering

Det som har kommet ut av analysen er at informantene legger andre ting i begrepet programmering enn de svarer når de blir spurt direkte spørsmål om begrepet programmering. Her drøftes kategorien undervisningspraksis med de underliggende kodene som er presentert i resultatkapittelet. Algoritmisk tenkning og begrepene innenfor algoritmisk tenkning er det flere informanter som er innom i sine intervjuer. Det er to som refererer eksplisitt til algoritmisk tenkning, men også flere som refererer til begreper som er

sentrale innenfor algoritmisk tenkning. Informantene kommer innom fem av seks begreper innenfor algoritmisk tenkning, men det er ikke slik at alle informantene er innom fem av seks begreper. Det varierer fra informant til informant hvor mange begreper de er innom fra den algoritmiske tenkeren. Informantene er innom begrepene logikk, algoritmer, dekomposisjon, mønster og evaluering som er fra algoritmisk tenking (Utdanningsdirektoratet, 2019).

Algoritmisk tenkning og programmering er nært knyttet sammen, ettersom at programmering kan bidra til en dypere forståelse av problemløsningsstrategier (Haraldsrud et al., 2020, s. 14). Det å programmere handler ikke bare om å skrive kode og forstå datamaskiner, men også å løse problemer, og da trengs det en problemløsningsstrategi (Sevik & m.fl., 2016, s. 13). Algoritmisk tenkning er en problemløsningsstrategi som kan brukes for å løse problemer (Wing, 2006, s. 33). Det er denne problemløsningsstrategien som det er ønskelig at elevene skal lære seg i den norske skolen (Utdanningsdirektoratet, 2020). Programmering er en måte å øve på algoritmisk tenkning, og kan gjøre at en kan tilnærme seg problemer på en annen måte enn tidligere (Haraldsrud et al., 2020, s. 14).

Logikk og logisk resonnering er det tre informanter som snakker om i intervjuene som de kobler til programmering. Logikk er et av seks nøkkelbegreper i algoritmisk tenkning (Utdanningsdirektoratet, 2019). Logikk handler om å forutse og analysere dataene eller viktig informasjon i et problem (Csizmadia et al., 2015, s. 6). Som informant 3 sier så handler det om å forstå hvordan ting henger sammen. Det å kunne analysere og forutse hvordan en skal løse problemer og hvordan en skal programmere for å løse problemet, er avgjørende. Hvis en ikke klarer å analysere problemet, så vil det gjøre det vanskelig og i det hele tatt komme i gang. For å løse problemet trengs det en plan for hva en skal gjøre, og hva som kommer til å skje (Mannila, 2017, s. 81). For å løse problemer med programmering trengs det ferdigheter i å analysere og forutse hva som vil skje (Csizmadia et al., 2015, s. 6). Logisk resonnering er også viktig for å kunne feilsøke (Csizmadia et al., 2015, s. 6). Feilsøking er en arbeidsmåte innenfor algoritmisk tenkning (Utdanningsdirektoratet, 2019).

Dekomposisjon er et annet begrep som informantene forbinder med begrepet programmering. Informant 2 og 3 snakker om det å bryte noe ned i mindre deler, og dekomposisjon er et annet av de seks nøkkelbegrepene i algoritmisk tenkning (Csizmadia et

al., 2015, s. 8). Dekomposisjon handler om å bryte et større problem ned i mindre del-problem (Mannila, 2017, s. 81). I programmering så er det ofte mange deloperasjoner som skal gjennomføres før en kommer frem til et resultat. Det å ha evnen til å dele problemet opp i mindre deler, kan gjøre det enklere å komme frem til et resultat (Csizmadia et al., 2015, s. 8). Dekomposisjon og logikk henger litt sammen. For å kunne bryte et problem ned i mindre deler, så trengs det også å kunne analysere og forstå problemet en møter på. Som informant 3 snakker om, så handler det om å bryte ned, analysere og forstå hvordan ting henger sammen.

Evaluering er et begrep som informantene selv ikke nevner, men som er det siste av seks nøkkelbegreper i algoritmisk tenkning (Utdanningsdirektoratet, 2019). Informant 2 er den eneste informanten som er innom dette i intervjuet. Det handler om å vurdere løsningene som en kommer frem til, men også å videreutvikle løsningene (Csizmadia et al., 2015, s. 7). Informant 2 snakker om i intervjuet at prosessen er viktigere enn selve resultatet, og informanten er mer opptatt av hvordan de jobber og hvordan de videreutvikler løsningene sine, noe som også Mannila (2017) skriver om (Mannila, 2017, s. 83). I arbeidet med programmering er det mulig å komme frem til flere forskjellige løsninger (Mannila, 2017, s. 82). En må derfor kunne vurdere løsningene en kommer frem til og vurdere hvilken som er best til den konteksten som en befinner seg i (Mannila, 2017, s. 82). Dette er også i tråd med Utdanningsdirektoratet sitt ønske med fokus på hvilke løsninger som er gyldige, og det er noe som skal gå gjennom hele matematikkundervisningen (Utdanningsdirektoratet, 2021).

Programmering er noe som, særlig i arbeidslivet, sjelden gjennomføres alene (Mannila, 2017, s. 85). I analysen kommer det frem at samarbeid blir brukt for å forklare programmering. I tillegg blir samarbeid sett på som en viktig ferdighet i det 21 århundre (Sevik & m.fl., 2016, s. 15). I intervjuene er det to av informantene som snakker om samarbeid, og informant 3 presiserer to og to som samarbeider. Parprogrammering er når to personer jobber sammen (Mannila, 2017, s. 85). Det finnes flere positive sider ved å ha parprogrammering (Lewis, 2011, s. 107). Lewis (2011) lister opp seks positive sider ved parprogrammering som handler om at elever har ulike kompetanser, og ved samarbeid gjør det at elevene kan løse oppgaver som de ikke hadde klart alene, men også at ved å samarbeide gjør det at elever må forklare til hverandre hvorfor de gjør de valgene de gjør

(Lewis, 2011, s. 107). I programmering er det også mulig å la elevene samarbeide om løsninger, selv om de ikke jobber med samme del, og dette er fordi at elevene kan dele oppgaver seg imellom ved å dekomponere problemet (Mannila, 2017, s. 81).

Hvilke elever som samarbeider og ikke, er problemstillinger som lærere må ta høyde for (Liljedahl, 2021, s. 41). Liljedahl (2021) skriver at elever vet hvorfor de blir plassert i en gruppe, og vet hvilken rolle de skal ha i gruppa, og at dette kan medføre at elevene bruker liten tid på å faktisk tenke på hvordan oppgaven skal løses (Liljedahl, 2021, s. 42). Det er derfor viktig å være bevisst rundt nivåinndelingen på gruppene, og at målene til både lærer og elever er tydelige (Liljedahl, 2021, s. 40). Samarbeid kan være med på å løfte elever som ville ha strevd for seg selv, og elevene kan sammen hjelpe hverandre videre (Imsen, 2014, s. 141). Hvis elevene er på ulikt nivå, så kan de ha forskjellig inngang til hvordan de ser problemet (Liljedahl, 2021, s. 53). Programmering består av mange elementer, og samarbeid kan være med på å hjelpe elevene i å løse oppgavene de møter på. Forskning viser at ved å la elever samarbeide, og da med parprogrammering, har økt elevens kompetanse (Lewis, 2011, s. 105).

I arbeidet med programmering så kommer alle elevene til å måtte feilsøke i koden sin. Ved å feilsøke så må en lete i koden, og dette er en prosess som kan være tidkrevende (Haraldsrud et al., 2020, s. 35). Dette vil gjøre at utholdenheten til elevene vil settes på prøve, og det å stå i oppgaven er noe som elevene dermed får øvd på. Det å stå i det nevner informant 2 og 6 i sine intervjuer. Hvis elevene klarer å stå i det, så kan læringsutbytte for elevene potensielt være stort (Haraldsrud et al., 2020, s. 35).

I resultatet er informantene innom det å utforske og det å jobbe med problemløsning når det snakkes om programmeringsbegrepet. Mannila (2017) skriver om at når et problem skal løses ved bruk av programmering så må det gis mulighet til å utforske (Mannila, 2017, s. 82). At elevene skal jobbe med algoritmisk tenkning og utforske er noe som kommer til uttrykk på Utdanningsdirektoratets sider, og i loven om grunnskolen og den videregående opplæringen til elevene (Utdanningsdirektoratet, 2020) (Opplæringslova, 1998, § 1-1). I intervjuet er tre av informantene innom utforsking og fire informanter innom problemløsning. I skolen skal det jobbes med algoritmisk tenkning, som er en

problemløsningsstrategi som det skal øves på ved bruk av programmering (Utdanningsdirektoratet, 2019). Ved å løse problemer må elevene ofte ha en utforskende tilnærming, hvor de prøver og feiler, før de til slutt kommer frem til en løsning (Mannila, 2017, s. 82). Det å jobbe med problemer og utforske, så må også elevene her ha god utholdenhet.

5.2.2 Undervisningspraksis

Det første jeg ser på er koden undervisning som handler om valg informantene gjør i undervisningen og hvordan de underviser i programmering. Programmeringsundervisningen kan både undervises som noe eget, men også knyttes opp mot andre fag. Programmering er noe som skal undervises i matematikk, musikk, kunst og håndverk og naturfag (Kunnskapsdepartementet, 2019b) (Kunnskapsdepartementet, 2019c) (Kunnskapsdepartementet, 2019a) (Kunnskapsdepartementet, 2019d). Fire av informantene mine har brukt det i et eller flere av disse fagene, men også koblet det opp mot andre fag i tverrfaglige prosjekter, og i artikkelen til Hsu et al (2018) kommer det frem at algoritmisk tenkning kan bli knyttet opp mot flere fag (Hsu et al., 2018, s. 299).

Undervisningen i programmering kan også være av typen utforskende, som vil si at elevene skal få eksperimentere og prøve seg frem (Mannila, 2017, s. 82). I programmeringsundervisningen er det tre av informantene som påpeker at programmering er en fin måte å jobbe utforskende på. Mannila (2017) skriver om at når et problem skal løses ved bruk av programmering så må det gis mulighet til å utforske (Mannila, 2017, s. 82). Ved å ha en utforskende tilnærming, så er det flere ting elevene får øvd seg på, og elevene får blant annet trene opp sin utholdenhet, og det å stå i det (Utdanningsdirektoratet, 2019). Elevene har krav på å utforske i undervisningen, så dette er noe samtlige elever skal få prøvd seg på (Opplæringslova, 1998, § 1-1).

Informant 1 og 3 bruker elevene sine aktivt i undervisningen som «hjelpelærere». Dette henger sammen med at det er flere elever som har foreldre som jobber med programmering, eller elever som har dette som hobby. Dette gjør at elevene kan opptre som «hjelpelærere» i undervisningen. Det å bruke elevene aktivt i undervisningen vil også føre til

at elevene må samarbeide mer i undervisningen, og da i par, som kan bidra til økt læring, slik Lewis (2011) skriver (Lewis, 2011, s. 107). Det er da viktig at elevene som hjelper andre elever ikke bare gjør oppgavene for de andre, men at elevene som er «hjelpelærere» faktisk lærer opp elevene. To av informantene peker på at undervisningen er preget av samarbeidsoppgaver, men også at de elevene som vil jobbe individuelt må få lov til det.

Når det gjelder hvordan undervisningen til informant 5 foregår, så er det et interessant moment med undervisningen til denne informanten. På skolen til denne informanten har de valgt å lage et eget programmeringsfag. Her får elevene mellom en halvtime og en time i uken med programmering. Dette gir elevene mulighet til å jobbe jevnt og trutt med programmering, enn at det blir sporadisk. Ved å ha faste timer i uka gjør det også at elevene kan ha prosjekter som går over en lengre periode. Dette er som sanneutvalget (2016) skrev i sin rapport at: «...vil ha bedre betingelser ved å være et eget fag i læreplaner, læring og undervisning» (Sanne et al., 2016, s. 51). Å integrere programmering i flere fag er noe som sanneutvalget ikke anbefalte å gjøre (Sanne et al., 2016, s. 76). De argumenterte for, og viste til erfaringer fra andre land, at det å ha det som eget fag ville være hensiktsmessig, og ved å ha det integrert i andre fag vil gjøre at programmering blir nedprioritert (Sanne et al., 2016, s. 76). Ved å ha et eget fag vil det sørge for at elevene får en plass hvor de får øvd seg på programmering og tilegnet seg de kunnskapene de trenger (Sanne et al., 2016, s. 76). Programmering og de ferdighetene som det øves på i programmering trengs i fremtiden til elevene (Sevik & m.fl., 2016, s. 10). Det ville dermed vært interessant å se hva som ville vært effekten av å ha programmering som et eget fag.

Når det gjelder undervisningsverktøyene som informantene har brukt og bruker i undervisningen så har de benyttet seg av noen. Av de som har startet med undervisning i programmering, så har fire av fem informanter vært innom micro:bit og bit:bot. Grunnen til at mange har vært innom dette er at informantene påpeker at dette er en enkel inngang for elevene, noe som også Haraldsrud (2020) påpeker er en enkel måte å introdusere programmering med mikrokontrollere for elevene på (Haraldsrud et al., 2020, s. 257). Dette er fordi at det er konkrete ting som de kan trykke og fikle på. Informantene forteller at micro:bit synliggjør programmeringen, og det blir tydeligere for elevene hva de gjør rett og hva de gjør galt, det gjør programmeringen visuelt for elevene (Statped, 2021b). Fire av

informantene har også vært innom Scratch i sin undervisning. Informant 3 startet derimot undervisningen sin i Scratch, før informanten beveget seg over mot micro:bit, hvor de andre startet motsatt. Blokkprogrammering kan være en fin inngang for elevene (Statped, 2021b).

Det som kjennetegner undervisningsverktøyene som informantene har tatt i bruk er at det i all hovedsak er blokkprogrammering. Dette henger sammen med at blokkprogrammering foregår på et nivå som alle elevene kan være med på, slik informant 2 påpeker. Informant 5 forteller at tekstprogrammering blir sett på som for vanskelig for det nivået de er på nå, og at det foregår som regel på engelsk, noe som gjør at flere elever kan dette av, og som kommer til uttrykk i litteraturen (Statped, 2021b). Ved å ha blokkprogrammering får elevene fokusert på programmeringsprinsippene istedenfor syntaks og andre detaljer (Statped, 2021b). Tekstprogrammering er lite egnet for nybegynnere, men også de kan få et utbytte av det (Mannila, 2017, s. 188). Selv om det kan være vanskelig for elevene, så er det noen av informantene som har testet tekstprogrammering med elevene. Informant 3 lot de ivrigste elevene få teste tekstprogrammering, mens informant 6 har prøvd tekstprogrammering med hele klassen, men merket at dette foreløpig ble vanskelig for klassen og seg selv. Selv om programmeringsundervisningen foregår på 5-7. trinn, så kan det være mulig å innføre tekstprogrammering her. Blokkprogrammering kan også være med på å gjøre overgangen til tekstprogrammering enklere for elevene (Mannila, 2017, s. 189).

5.3 Konklusjon

5.3.1 Hva legger et utvalg matematikklærere i begrepet programmering

Selv om flere informanter definerer programmering som koding på en datamaskin viser informantene at de senere i intervjuet forbinder andre ting med programmering, selv om de ikke sier dette eksplisitt. Det kan også henge sammen med hvordan spørsmålene i intervjuet blir stilt, og at informantene ikke nødvendigvis går i dybden på begrepet når jeg stiller de spørsmålet hva de legger i begrepet programmering eller hvordan de vil definere det. Ettersom at ordlyden i spørsmålene kan ha vært litt ulik i intervjuene, så kan dette også påvirke hva som blir lagt i begrepet av informantene.

Informantene viser i intervjuene og etter analysen at de har en dypere begrepsforståelse for programmering enn de kanskje gir uttrykk for når de selv definerer begrepet programmering. Informantene viser til flere begreper i den algoritmiske tenkeren som de har kunnskaper om, og arbeidsmåter som det kan jobbes med programmering på, som også ligger i den algoritmiske tenkeren (Utdanningsdirektoratet, 2019). Alle ferdighetene og arbeidsmåtene de viser til er med på å forklare programmeringsbegrepet, noe som også Mannila (2017) skriver om (Mannila, 2017, s. 85).

Konklusjonen jeg vil trekke ut ifra mitt datamateriale er at programmering er noe som kan knyttes til en datamaskin, samtidig som at programmering er noe mer enn bare selve kodingen. Det er en informant som snakker om programmering som en prosess som foregår over flere steg, som også kommer til uttrykk i litteraturen (Sevik & m.fl., 2016, s. 9). Ferdighetene og arbeidsmåtene som ligger under algoritmisk tenkning er det flere informanter som snakker om i sine intervjuer. Disse begrepene er først og fremst viktige for å forklare den algoritmiske tenkeren (Utdanningsdirektoratet, 2019). Men ettersom at programmering og algoritmisk tenkning er nært knyttet sammen, så er disse begrepene også viktige for å forklare programmeringsbegrepet (Haraldrud et al., 2020, s. 14). Dette viser at programmeringsbegrepet er et rikt begrep. Informantene mine legger mye forskjellig i begrepene, hvor ferdighetene og arbeidsmåtene som ligger under algoritmisk tenkning kommer igjen i flere av intervjuene, og er noe som de selv refererer til uten at de får noe informasjon om den algoritmiske tenkeren i intervjuene.

Programmering er algoritmer, mønster, logikk, dekomposisjon, evaluering, samarbeid, feilsøking, utforsking, problemløsning, altså ferdigheter og arbeidsmåter som inngår i den algoritmiske tenkeren (Utdanningsdirektoratet, 2019). Det er også viktig å påpeke, slik jeg er inne på i drøftingen av kategoriene mine, så henger disse begrepene sammen. Dette viser, som skrevet i dette avsnittet, at programmering er et rikt begrep som inneholder mye, og det er slik Blackwell (2002) skriver, så eksisterer det flere definisjoner og at begrepet utvikler seg (Blackwell, 2002, s. 204).

At det eksisterer ulike definisjoner av programmering henger nært sammen med at begrepet er noe nytt for mange lærere i skolen. Det er også viktig å huske på at siden dette er nytt for mange i skolen, så eksisterer det også svært store forskjeller i hvordan skolene har jobbet med programmering blant lærerne, og hvilken opplæring de har.

Når det gjelder implikasjoner på praksisfeltet som denne konklusjonen og forskningens resultater vil ha å si, så er det to ting som det har kommet frem fra intervjuene. Når det gjelder begrepsforståelsen så tyder det på at lærere har fått en variert forståelse av begrepet. Oppgaven viser at et par av informantene som er innom flere kategorier og koder, og noen informanter som er innom få. Selv om fem av seks informanter har fått kursing, så viser dette at det fremdeles er en vei å gå. Dette sier også noen av informantene, men at de er på god vei mot å få den ønskede kompetansen. I en rapport fra EU (2016) trekkes det frem at det trengs en grundig opplæring for å få på plass kunnskaper og begreper (Bocconi et al., 2016, s. 7).

En ulempe ved å ha et begrep som det eksisterer så mange ulike definisjoner av, er da hvordan dette kan påvirke undervisningen til elevene. Ved å ha ulike definisjoner av et begrep i skolen, så kan dette medføre at elevene får en veldig ulik opplæring, og at en blir veldig prisgitt den læreren som en får. Hana (2013) peker på tre funksjoner som en definisjon har til hensikt å hjelpe med: «å bidra til enklere og mer konsis språkbruk, sikre felles og forståelig språkbruk og være matematisk fruktbare» (Hana, 2013, s. 56). Elevene kan være prisgitt læreren de får, men en bør gjøre alt en kan for at elevene uavhengig av lærer skal kunne få den opplæringen de fortjener. Ved å ha en felles definisjon av et begrep så kan det gjøre det enklere for lærerne når de skal undervise (Hana, 2013, s. 53).

5.3.2 Hvordan underviser et utvalg lærere i programmering på 5-7. trinn

Undervisningen i programmering blant mine informanter på 5-7. trinn er preget av noe variasjon. Det er variasjon når det kommer til hvordan selve undervisningen er lagt opp og til hvilke undervisningsverktøy som benyttes. Ettersom at programmering er ganske nytt, og ikke minst at det er stor variasjon i hvilken type opplæring de har, så vil også undervisningspraksisen blant lærere være variert. Det er viktig å påpeke at det finnes flere

forskjellige måter å undervise i programmering på, og at det finnes ikke et fasitsvar på hvordan undervisningen skal foregå. Det er derimot viktig, som i alle andre fag i skolen, å ha variasjon i undervisningen slik at elevene får møtt på et rikt antall aktiviteter, ressurser, læringsarenaer og vurderingsformer (Utdanningsdirektoratet, 2022).

Når det gjelder hvordan informantene mine underviser er det i hovedsak tre ting som går igjen. Det er noen likheter i undervisningen samtidig som det er variasjon i hvordan de underviser. Det første er utforskning og da hvordan elevene jobber utforskende i programmeringsundervisningen. I skolen skal elevene jobbe utforskende, og dette er noe som de har krav på (Opplæringslova, 1998, § 1-1). Når et problem skal løses ved bruk av programmering så trengs det at elevene får muligheten til å utforske (Mannila, 2017, s. 82). Dette gjør at programmering er en mulighet for å la elevene utforske, noe som tre av informantene har gjort.

Det andre er samarbeid og mulighetene for å bruke samarbeid i programmeringsundervisningen, og da særlig to og to som jobber sammen. Ettersom at programmering i arbeidslivet sjeldent foregår alene, så får elevene gjort erfaringer med å samarbeide i undervisningen til informantene mine (Mannila, 2017, s. 85). Det finnes flere positive sider ved å ha samarbeid i programmeringsundervisningen, og da særlig parprogrammering (Lewis, 2011, s. 107). Ved å la elevene samarbeide i undervisningen gjør det at elevene får brukt hverandres kompetanser, og kanskje klarer å løse oppgaver som de ikke hatt klart alene (Lewis, 2011, s. 107).

Det siste er bruken av blokkprogrammering i undervisningen til informantene mine. Fem av seks informanter har brukt blokkprogrammering i forskjellige verktøy, hvor micro:bit og Scratch er de mest brukte. Fordelen med disse blokkprogrammeringsverktøyene er at de synliggjør for elevene hva som skjer (Statped, 2021b). Ved å bruke disse verktøyene gjør det at elevene får fokusere på programmeringsprinsippene, og ikke andre unødvendige detaljer (Statped, 2021b). Scratch er blant annet laget spesielt for barn og unge, og det å bruke dette i undervisningen på 5-7. trinn, slik informantene mine har gjort, så kan dette være en naturlig inngang for elevene (Mannila, 2017, s. 130). Det som går igjen i undervisningen til informantene mine er fokuset på utforskning, samarbeid og bruken av blokkprogrammering.

Når det gjelder implikasjoner på praksisfeltet, så er det noen ting som kommer frem fra analysen. Læreryrket er et profesjonsyrke og med en slik tittel kommer det også ansvar og frihet til å ta de valgene en selv mener er best for sin elevmasse. Som skrevet viser denne masteroppgaven at det er flere måter å legge opp programmeringsundervisningen sin på. Dette kan også opptre som en trygghet for lærerne, at det finnes flere veier til mål, og at du kan få elevene dine langt på vei uten mye opplæring selv. Samtidig viser litteraturen at det å starte med blokkprogrammering, kan være en naturlig inngang til programmering (Mannila, 2017, s. 189). Dette er også noe som informantene mine har tatt bevisste valg rundt. Informantene mine påpeker også at du kan lære elevene mye, selv om du ikke kan mye selv, og at du lærer sammen med elevene dine. For lærere som synes programmering er vanskelig, så kan det være trygt for de å vite at det er fullt mulig å lære programmering sammen med elevene sine.

5.4 Muligheter for videre forskning

Når det gjelder muligheter for videre forskning så er det noen ting det kan forskes på. Det første er å gå videre med det som er forsket på i denne oppgaven. Intervjue flere og få et bredere bilde av hvordan tilstanden i norske skoler er når det gjelder programmeringsbegrepet og programmeringsundervisningen. Det andre er å forske videre på undervisning i klasserommet. Her er det å se konkret på undervisningen til lærerne i klasserommet, men også se på elevene i ulike lærings situasjoner i programmeringsundervisningen. Videre kan det å forske på ulike innganger og undervisningsverktøy som egner seg for klasserommet.

Det tredje er å se på hvordan programmeringsundervisningen er organisert. I dag er programmeringsundervisningen lagt inn i flere fag, og sanneutvalget anbefalte å lage et eget programmeringsfag. Det som kunne vært spennende og sett på over en lengre periode er læringsutbytte til de som har fått det som et eget fag, slik en av informantene mine har, sammenlignet med de som får det i flere fag.

6.0 Litteraturliste

- Barriball, K. L. & While, A. (1993). Collecting data using a semi-structured interview: a discussion paper. *Journal of Advanced Nursing-Institutional Subscription*, 18(10), 328-335.
- Berg, T. K. (2021). Analog programming.
- Blackwell, A. F. (2002). What is Programming. <https://www.ppig.org/files/2002-PPIG-14th-blackwell.pdf>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P. & Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*, 68.
- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). *The Nordic approach to introducing Computational Thinking and programming in compulsory education* (Nordic@BETT2018 Steering Group).
<http://www.itd.cnr.it/doc/CompuThinkNordic.pdf>
- Bostrøm, E., Bø, O., Langmyhr, D. & Rydland, T. (2008). *Informasjonsteknologifaget og skoleverket-en bakgrunn og handlingsplan for Norsk Informatikkråd* (Rapport 2008:7). Høgskolen i Østfold. <https://hiof.brage.unit.no/hiof-xmlui/handle/11250/148600>
- Busch, P. A. (2019, 21.10. 2019). *EDB*. Store norske leksikon. <https://snl.no/EDB>
- Communication, E. C.-D.-G. f. (2014). *Digital agenda for Europe : rebooting Europe's economy*. Publications Office. <https://data.europa.eu/doi/10.2775/41229>
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2009). *Introduction to algorithms*. MIT press.
<http://139.59.56.236/bitstream/123456789/106/1/Introduction%20to%20Algorithms%20by%20Thomas%20%20H%20Coremen.pdf>
- Creswell, J. W. & Poth, C. N. (2016). *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). Computational thinking-A guide for teachers.
https://eprints.soton.ac.uk/424545/1/150818_Computational_Thinking_1_.pdf

- Dagienė, V., Jevsikova, T., Stupurienė, G. & Juškevičienė, A. (2022). Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes. *Computer Science and Information Systems*, (00), 1-33.
- Dolonen, J. A., Kluge, A., Litherland, K. & Mørch, A. I. (2019). *Litteraturgjennomgang av programmering i skolen*. <https://www.duo.uio.no/handle/10852/76290>
- Dæhlen, M. & Eriksen, I. M. (2015). " Det tenner en gnist". Evaluering av valgfagene på ungdomstrinnet. <https://oda.oslomet.no/oda-xmlui/handle/20.500.12199/5084>
- Forsström, S. E. & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. <https://hiof.brage.unit.no/hiof-xmlui/bitstream/handle/11250/2599710/ForsstromALiterture2018.pdf?sequence=2>
- Griffin, P. & Care, E. (2014). *Assessment and teaching of 21st century skills: Methods and approach*. Springer.
- Hana, G. M. (2013). *Matematiske byggesteiner*. Caspar.
- Haraldsrud, A. D., Sveinsson, H. A. & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Helenius, O., Misfeldt, M., Rolandsson, L. & Ryan, U. (2018). Om programmering i matematikundervisning. *Skolverket*, 1, 9. https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/0-digitalisering/Gymnasieskola/448_matematikundervisningmeddigitalaverktygII_GY/del_01/Material/Flik/Del_01_MomentA/Artiklar/MA2_Gy_01A_01_omprogrammering.docx
- Hsu, T.-C., Chang, S.-C. & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310.
- Høgheim, S. (2020). *Masteroppgaven i GLU*. Fagbokforlaget.
- Imsen, G. (2014). *Elevenes verden: innføring i pedagogisk psykologi* (5. utg.). Universitetsforlaget.
- Johansen, A.-K. (2020). *Programmering vil bli en utfordring for lærere*. Forskning.no. <https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold-matematikk/programmering-vil-bli-en-utfordring-for-laerere/1711838>

- Kallio, H., Pietilä, A. M., Johnson, M. & Kangasniemi, M. (2016). Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of advanced nursing*, 72(12), 2954-2965.
https://onlinelibrary.wiley.com/doi/pdf/10.1111/jan.13031?casa_token=r8ewQyPV2ksAAAAA:PoYtf22UUIOF13RakItNueSMOkLTeygorQFyXr5tBjAwWBpj8mOO0EI-AApooPBi elkxuJZTajun44
- Kolbjørnsen, P. Ø. (1998). *Statistikk om informasjonsteknologi: status, behov og utviklingsmuligheter*. Statistisk sentralbyrå. https://ssb.brage.unit.no/ssb-xmlui/bitstream/handle/11250/2768700/rapp_199801.pdf?sequence=1
- Kunnskapsdepartementet. (2019a). *Læreplan i kunst og håndverk (KHV01-02)*. Fastsatt som forskrift. <https://www.udir.no/lk20/khv01-02>
- Kunnskapsdepartementet. (2019b). *Læreplan i matematikk (MAT01-05)*. Fastsatt som forskrift. <https://www.udir.no/lk20/mat01-05>
- Kunnskapsdepartementet. (2019c). *Læreplan i musikk (MUS01-02)*. Fastsatt som forskrift. <https://www.udir.no/lk20/mus01-02>
- Kunnskapsdepartementet. (2019d). *Læreplan i naturfag (NAT01-04)*. Fastsatt som forskrift. <https://www.udir.no/lk20/nat01-04>
- Kunnskapsdepartementet. (2020). *Overordnet del av læreplanverket - Kompetanse i fagene*. <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/kompetanse-i-fagene/?lang=nob>
- Kvale, S. & Brinkmann, S. (2010). *Det kvalitative forskningsintervju* (2. utg.). Gyldendal Norsk Forlag AS.
- Kvernmo, G. (2005). Intervju som metode—barn/unge som informanter. *Denne boken er en revisjon av førsteutgaven fra 2005. Revisjonen omfatter korrektur og implementering av ny referanse stil, APA Style. I tillegg er et utvalg av artiklene faglig revidert. Følgende artikler er revidert «Eksperimentell design, med spesiell vekt på ulike typer av N= 1 design» av Erik Arntzen, «Vitenskapelig tenkemåte», 301.*
<https://oda.oslomet.no/oda-xmlui/bitstream/handle/20.500.12199/208/studenten.pdf?sequence=1&isAllowed=y#page=35>
- Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education*, 21(2), 105-134.

- Liljedahl, P. (2021). *Building Thinking Classrooms in Mathematics, Grades K-12*. SAGE Publications Inc.
- Mannila, L. (2017). *Att undervisa i programmering i skolan : varför, vad och hur?* Studentlitteratur AB.
- McCracken, G. (1988). *The long interview* (Bd. 13). Sage.
- MIT. (2016, 01.08.2016). *Professor Emeritus Seymour Papert, pioneer of constructionist learning, dies at 88*. Massachusetts Institute of Technology.
<https://news.mit.edu/2016/seymour-papert-pioneer-of-constructionist-learning-dies-0801>
- NOU 2013: 2. (2013). *Hindre for digital verdiskaping*. K.-o. distriktsdepartementet.
<https://www.regjeringen.no/no/dokumenter/nou-2013-2/id711002/>
- Opdenakker, R. (2006). Advantages and disadvantages of four interview techniques in qualitative research. *Forum qualitative sozialforschung/forum: Qualitative social research*, 7(4). <https://www.qualitative-research.net/index.php/fqs/article/view/175>
- Opplæringslova. (1998). *Lov om grunnskolen og den vidaregåande opplæringa* (LOV-1998-07-17-61). <https://lovdata.no/dokument/NL/lov/1998-07-17-61>
- Papert, S. (1990). *Children, computers and powerful ideas*. I. New York: Basic Books.
<http://kvantti.kapsi.fi/Documents/LCL/mindstorms-chap1.pdf>
- Persson, J. (2014). *Problemløsning*. NTNU.
<https://www.ntnu.no/documents/2004699/1267941227/Problemløsning+-+Persson.pdf/32189e9f-5174-416a-863a-5fcd54f26045>
- Pind, P. (2011). *Håndbok i matematikkundervisningen*. Cappelen Damm AS.
- Polya, G. (1957). *How to solve it - A new aspect of mathematical method*. Princeton University Press.
<https://math.hawaii.edu/home/pdf/putnam/PolyaHowToSolveIt.pdf>
- Postholm, M. B. & Jacobsen, D. I. (2021). *Forskningsmetode for masterstudenter i lærerutdanning*. Cappelen Damm AS.
- Rossmann, G. B. & Rallis, S. F. (2011). *Learning in the field: An introduction to qualitative research*. Sage.
<https://books.google.no/books?hl=no&lr=&id=JgTLGHLhwFAC&oi=fnd&pg=PR1&dq=Learning+in+the+field:+An+introduction+to+qualitative+research&ots=IUjuZvTCMs&>

[sig=B0ki1vrrar6QMbtaUkkHLud2i8qY&redir_esc=y#v=onepage&q=Learning%20in%20the%20field%3A%20An%20introduction%20to%20qualitative%20research&f=false](https://doi.org/10.1080/13645579.2021.1935565)

Rudlang, H. (1995). *Bruk av edb i skolen 1995*. Statistisk sentralbyrå.

https://ssb.brage.unit.no/ssb-xmlui/bitstream/handle/11250/2755832/rapp_199523.pdf?sequence=1

Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G. & Voll, L. O. (2016). *Teknologi og programmering for alle - En faggjennomgang med forslag til endringer i grunnopplæringen*.

Utdanningsdirektoratet. <https://www.udir.no/globalassets/filer/tall-og-forskning/forskningsrapporter/teknologi-og-programmering-for-alle.pdf>

Sevik, K. & m.fl. (2016). *Programmering i skolen*. Senter for IKT i utdanningen.

https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Statped. (2021a, 01.03.2021). *Analog programmering*. Statlig spesialpedagogisk tjeneste.

<https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/analog-programmering/>

Statped. (2021b, 01.03.2021). *Programmeringsspråk*. Statlig spesialpedagogisk tjeneste.

<https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/programmeringssprak/>

Stenseth, B., Kaufmann, O. T. & Forsström, S. (2019). Programmering og matematikk.

Tangenten–tidsskrift for matematikkundervisning, 30(2), 7-12.

Stylianides, A. J. (2007). Proof and Proving in School Mathematics. *Journal for Research in Mathematics Education*, 38(3), 289-321.

<https://www.jstor.org/stable/30034869?seq=1>

Thorland, E. M. (2018). *Hvilke ulike praksiser eksisterer for faget programmering i ungdomsskolen, og hvilke innholdskomponenter legger lærerne vekt på i utformingen av faget?* [Høgskulen på Vestlandet]. <https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/2586400>

Thunberg, S. & Arnell, L. (2021). Pioneering the use of technologies in qualitative research—A research review of the use of digital interviews. *International Journal of Social Research Methodology*, 1-12. <https://doi.org/10.1080/13645579.2021.1935565>

Utdanningsdirektoratet. (2019). *Algoritmisk tenkning*. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>

Utdanningsdirektoratet. (2020). *Kjerneelement*. <https://www.udir.no/lk20/mat01-05/om-faget/kjerneelementer>

Utdanningsdirektoratet. (2021). *Fagrelevans og sentrale verdier*.

<https://www.udir.no/lk20/mat01-05/om-faget/fagets-relevans-og-verdier>

Utdanningsdirektoratet. (2022). *Tilpasset opplæring*. <https://www.udir.no/laring-og-trivsel/tilpasset-opplaring/>

Uzunboyly, H., Kinik, E. & Kanbul, S. (2017). An Analysis of Countries which have Integrated Coding into their Curricula and the Content Analysis of Academic Studies on Coding Training in Turkey. *TEM Journal*, 6(Issue 4), 783-791.

<https://doi.org/10.18421/TEM64-18>, November 2017

Wengraf, T. (2001). *Qualitative research interviewing: Biographic narrative and semi-structured methods*. sage.

https://www.pm.lth.se/fileadmin/migrated/content/uploads/5_Conceptual_Frameworks_for_Studying_and_Inferring_from_Research_Interview_Interaction_Practice.pdf

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

<https://doi.org/10.1145/1118178.1118215>.

7.0 Vedlegg

7.1 Vedlegg 1: Intervjuguide

Holdninger	<p>Hvordan har fagfornyelsen påvirket din lærerhverdag og undervisning?</p> <p>Programmering har blitt en del av læreplanen - hva tenker du om dette?</p> <p>Hva legger du i begrepet programmering? Hva er programmering?</p> <p>Hva forbinder du med programmering i skolen?</p> <p>Hvordan har programmering påvirket undervisningen i ditt fag?</p> <p>Hvordan kan programmering føre til bedre forståelse i faget ditt?</p>
------------	--

	<p>Programmering kan ses på som enda et ekstra element som skal inkluderes inn i de ulike fagene, hva tenker du om dette? Hvorfor? I læreplanene er det allerede mange mål fra før og mye som elevene skal lære. Ser du på programmering som en ekstra byrde som skal inn i skolen? På hvilken måte?</p>	
Undervisnings praksis	<p>Har du startet med å inkludere programmering i din undervisning?</p>	
	<p>JA</p> <ul style="list-style-type: none"> -Hvilke konkrete eksempler har du bruk av programmering i undervisningen din? -Hvilke programmer bruker du, og hvorfor bruker du disse? -Underviser du annerledes i programmering sammenlignet med andre fag/tema i skolen? - Underviser du annerledes i programmering i matematikk sammenlignet med de andre fagene du har? - Ser du likheter/ulikheter ved programmering mellom de ulike fagene? -Hvilke arbeidsmetoder bruker du? -Hvordan kombinerer du programmering og fag? Har du så langt undervist i programmering som noe 	<p>NEI</p> <ul style="list-style-type: none"> -Hvorfor har du ikke startet med dette? -Har du blitt introdusert for programmer eller undervisningsopplegg for programmering? Hvilke programmer er da blitt introdusert? -I en tenkt situasjon hvor du skulle brukt et av de nevnte programmene. Hvordan ville du presentert dette for elevene? -Hvilke utfordringer/muligheter ser du ved å ha programmering i undervisningen din?

	<p>eget, eller har du knyttet det opp mot et tema i et annet fag?</p> <p>-Hvordan opplever du elevene dine når du underviser i programmering?</p> <p>-Hvordan hjelper du elevene videre i faget? Enten om de mestrer det eller trenger ekstra hjelp?</p> <p>Hvordan måler du elevenes læring når du underviser i programmering?</p>	
Opplæring	Har du fått noe opplæring knyttet til programmering?	
	<p>JA</p> <p>-Hvilket omfang og hvordan</p> <p>-Hvordan opplevde du opplæringen? Formell kompetanse innen programmering?</p> <p>-Hva var nyttig og relevant for skolehverdagen?</p> <p>-Hvilken opplæring ønsker du deg som du ikke allerede har fått?</p> <p>-Finnes det undervisningsmateriell om</p>	<p>NEI</p> <p>-Har du fått tilbud om dette?</p> <p>-Har du behov for dette?</p> <p>- Hva ønsker du deg av opplæring?</p> <p>-Finnes det undervisningsmateriell om opplæring innenfor programmering?</p>

	opplæring innenfor programmering?	
Individuell og kollektiv kompetanse	<p>Hvordan er din kompetanse innenfor programmering i forhold til dine kollegers?</p> <p><i>Tror du resten av ditt kollegium føler seg like forberedt/lite forberedt på programmering som deg? Hvorfor/hvorfor ikke?</i></p> <p>Hvordan føler du at dine kolleger stiller seg til programmering inn i skolen?</p> <p>Hvordan jobber dere med programmering på deres skole?</p> <ul style="list-style-type: none"> - På tvers av fag? - På tvers av trinn? 	
Avslutning	<p>Noe å tilføye?</p> <p>Oppfølgingsspørsmål?</p>	

7.2 Vedlegg 2: Samtykkeskjema

Vil du delta i forskningsprosjektet

Hvordan underviser lærere i programmering på 5-7 trinn?

Dette er en forespørsel til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke hva lærere legger i begrepet programmering og hvordan de underviser i programmering på 5-7 trinn. I dette skrevet gir jeg deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

I den nye læreplanen er programmering blitt en sentral del av matematikken, men også i andre fag. I tillegg er programmering en kompetanse som trengs for å leve og jobbe i dagens samfunn, men også i det samfunnet som er når barna vokser opp. Programmering er noe som skal jobbes med i flere fag, slik at elevene utvikler de ferdighetene og kunnskapene som trengs i programmering.

Målet med studien er å forske på hvordan lærere underviser i programmering på 5-7 trinn, og hvilke kompetanser som finnes ute i klasserommet, og da med utgangspunkt i lærere som jobber på 5-7 trinn nå, i fagene som har programmering på læreplanen. Jeg vil spørre om hvordan du underviser, hvilke verktøy du bruker, hvordan du legger opp undervisningen og om hvilken kompetanse du har. Jeg vil også se på hva du legger i begrepet programmering og hva du forbinder med det begrepet.

Foreløpig problemstilling for masteroppgaven er: Hva legger et utvalg lærere i begrepet programmering og hvordan underviser de i programmering på 5-7 trinn?

Hvem er ansvarlig for forskningsprosjektet?

Universitetet i Sørøst-Norge er ansvarlig for prosjektet.

Prosjektansvarlig er Nikolai Ellingsen(student) og Reiar Kravik(veileder)

Hvorfor får du spørsmål om å delta?

Denne henvendelsen går til lærere som underviser i fagene: matematikk, naturfag, musikk, kunst og håndverk på 5-7 trinn. Jeg ønsker et tilfeldig utvalg av lærere som vil gjennomføre et intervju. Dette vil danne grunnlaget for arbeidet i denne masteroppgaven

Hva innebærer det for deg å delta?

Ved å delta vil du samtykke til et intervju, hvor det blir tatt lydopptak, som blir transkribert i etterkant av intervjuet. Intervjuet vil vare i maksimum 40 minutter, og gjennomføres mellom uke 41 og 45. Intervjuet inneholder spørsmål knyttet til programmering på 5-7 trinn. I intervjuet vil det være spørsmål om hvordan du som lærer underviser i programmering, hvilken kompetanse du har i programmering og hva du legger i begrepet programmering. Fordeler, ulemper, forventninger og kunnskap til programmering i skolen er noen av temaene vi kan komme innom i intervjuet.

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 jeg oppbevarer og bruker dine opplysninger

Jeg vil bare bruke opplysningene om deg til formålene jeg har fortalt om i dette skrivet. Jeg behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Det er kun jeg (Nikolai Ellingsen) som har tilgang på innhentede opplysninger, men veileder vil også kunne se opplysningene i veiledningstimer.
- Opplysningene skal anonymiseres.
- Opplysningene vil bli lagret lokalt på en personlig datamaskin.
- Svarene som gis vil anonymiseres. Dette gjelder også for analysene som foretas i etterkant. Ingen informasjon vil derfor kunne tilbakeføres til enkeltelever, skoler, klasser eller lærere.

Hva skjer med opplysningene dine når jeg avslutter forskningsprosjektet?

Alle opplysningene transkriberes og anonymiseres fortløpende i forskningsprosjektet. Alle lydopptak og notater slettes ved slutten av forskningsprosjektet, 30. juni 2022.

Dine rettigheter

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

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hva gir meg rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Sørøst-Norge har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan du finne ut mer?

Hvis du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:

- Nikolai Ellingsen, nikolai.ellingsen@gmail.com, tlfnr: 41304483
- Reiar Kravik, reiar.kravik@usn.no, tlfnr: 47660044

Vårt personvernombud: Paal Are Solberg, Paal.A.Solberg@usn.no eller personvernombud@usn.no

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

- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Med vennlig hilsen

Nikolai Ellingsen
(Student)

Reiar Kravik
(Veileder)

Samtykkeerklæring

Jeg har mottatt og forstått informasjonen om prosjektet «Hvordan underviser lærere i programmering på 5-7 trinn» 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)

7.3 Vedlegg 3: Godkjenning fra NSD

05.10.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 05.10.2021, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 30.06.2022.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER

NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
- lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER

NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20).

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER

NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

Universitetet i Sørøst-Norge er felles behandlingsansvarlig institusjon. NSD legger til grunn at behandlingen oppfyller kravene til felles behandlingsansvar, jf. personvernforordningen art. 26.

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og eventuelt rådføre dere med behandlingsansvarlig institusjon.

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde: <https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema> Du må vente på svar fra NSD før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!