

MASTEROPPGAVE

Ungdomsskolelæreres refleksjoner om hvordan
programmering kan bli brukt til å lære elever matematikk

Simen Holt Didriksen

16.05.2022

Grunnskolelærerutdanning for trinn 5-10
Fakultet for lærerutdanninger og språk

Sammendrag

Med en økende digitalisering av verden, og dermed skolen, har vi et økende behov for opplæring i digital kompetanse. Det fins en rekke studier som undersøker programmering i sammenheng med matematikk og elevers læring, motivasjon, ferdigheter og innstilling. Derimot er det manglende innsikt i læreres syn på programmering og hva deres erfaringer har vært. Studien skal undersøke hvilke tanker og refleksjoner lærere på ungdomsskolenivå har om deres erfaringer innen programmering, særlig i sammenheng med matematikk.

Forskningsspørsmålene er: (1) Hvordan forstår lærere viktige begreper i programmering? (2) Hva er læreres erfaringer med programmering før og etter den nye læreplanen? (3) Hvordan kan programmering bidra til å få elevene interessert i matematikk? (4) Hvilke fordeler og ulemper har programmering i skolen?

Det har i denne oppgaven blitt gjennomført en kvalitativ forskningsstrategi i form av individuelle intervjuer med tre ungdomsskolelærere. Det har blitt tatt i bruk tematisk analyse av intervjuenes transkripsjoner for å finne gjennomgående temaer.

Resultatene viser informantenes forståelse for viktige begreper i programmering sammenlignet med teorien, informantenes erfaringer med programmering både privat og i en skolesammenheng. Det ble også avdekket ulike implikasjoner for tilpasninger i arbeid med programmering og matematikk.

Abstract

With an increasing digitalization of the world, and thus the school, we have an increasing need of training in digital competence. There are several studies that examine programming in the context of mathematics and students' learning, motivation, skills, and attitude. On the other hand, there is a lack of insight into teachers' attitudes towards programming and what their experiences have been. This study will examine what thoughts and reflections teachers at junior high school level have about their experiences in programming, especially in relation to mathematics.

The research questions are: (1) How do teachers understand important terms in programming? (2) What are teachers' experiences with programming before and after the new curriculum? (3) How can programming contribute to get students interested in mathematics? (4) What advantages and challenges have programming in school?

In this thesis, a qualitative research strategy has been used in the form of individual interviews with three junior high school teachers. Thematic analysis of the interviews' transcripts has been used to find common themes.

The results show the informants' understanding of important concepts in programming compared to the theory, the informants' experiences with programming both privately and in a school education context. Various implications for adaptations in work with programming and mathematics were also revealed.

Forord

Denne masteroppgaven er siste innspurt i min tid på lærerskolen. Den største takken går til min veileder Khaled Ben Latief Jemai, som tålmodig har veiledet arbeidet mitt gjennom hele perioden. Tusen takk for gode tilbakemeldinger, konstruktiv kritikk, tips til litteratur og hyggelige prater.

En stor takk til venner og familie som har vist interesse og støtte for arbeidet mitt. Muligheten til å fortelle hva jeg driver med og dele mine frustrasjoner om alt mulig har hjulpet meg stort i å komme gjennom semesteret.

En siste takk går til informantene som tok tid til å stille til intervjuer, til tross for at de har svært travle hverdager som lærere. Jeg setter stor pris på at dere ville dele deres tanker og erfaringer med meg og andre.

Takk!

Innhold

Figur-oversikt	viii
Tabell-oversikt	viii
Forkortelser – Norsk.....	ix
Forkortelser – Engelsk	ix
1. Innledning.....	1
1.1 Bakgrunn	1
1.2 Hvorfor arbeide med programmering i skolen?	1
1.3 Problemstilling og avgrensninger.....	2
1.4 Tidligere forskning	3
1.5 Begrepsforklaringer.....	4
1.5.1 Programmering.....	4
1.5.2 Læringsmetode og problemløsningsstrategi.....	4
1.5.3 Algoritmisk tenkning vs. algoritmisk resonnering	5
1.5.4 Dybdelæring	6
1.6 Oppgavens forskningsdesign – Metode	7
1.7 Faglig sammenheng.....	8
1.8 Oppgavens struktur	8
2. Teori	9
2.1 Hva er egentlig matematikk?.....	9
2.2 Grunnleggende ferdigheter i matematikk.....	11
2.3 Algoritmisk tenkning.....	12
2.4 Programmering og algoritmisk tenkning i skolen	13
2.4.1 Fordeler	13
2.4.2 Utfordringer.....	14
2.4.3 Læreres kunnskap og holdninger	16
2.5 Kilpatrick, Nosrati og Wæge, matematisk kompetanse og dybdelæring	17

2.6 «Learning by doing» og Dewey	19
2.7 Proksimal utviklingssone og Vygotsky	20
2.8 Det helhetlige teoretiske rammeverket	22
3. Metode	24
3.1 Kort oversikt over studien	24
3.2 Vitenskapelig forankring	25
3.2.1 Hermeneutikk	26
3.2.2 Fenomenologi	27
3.3 Det kvalitative forskningsintervju	28
3.4 Datainnhenting og oppbevaring og utvalg	28
3.4.1 Forarbeid og intervjuguide	28
3.4.2 Utvalg av informanter	29
3.4.3 Gjennomførelsen av intervjuene	30
3.5 Metode for analysen av datamaterialet	31
3.5.1 Tematisk analyse	31
3.5.2 Fra rådata til koder	33
3.5.3 Induktiv tilnærming	33
3.6 Etiske betraktninger og studiens kvalitet	34
3.6.1 Reliabilitet og validitet	34
3.6.2 Informantens beste	35
3.6.3 Metodekritikk	36
3.7 Refleksivitet	37
4. Presentasjon og analyse av funn	39
4.1 Analyseprosessen	39
4.2 Lærernes begrepsforståelse	43
4.2.1 Programmering	43
4.2.2 Læringsmetode	44

4.2.3 Problemløsningsstrategi	45
4.2.4 Algoritmisk tenkning.....	45
4.3 Læreres erfaringer med programmering.....	46
4.4 «Du skjønner det ikke før du har prøvd»	48
4.5 Aspekter ved programmering i skolen	49
4.5.1 Fordeler	49
4.5.2 Utfordringer.....	50
4.5.3 Hvordan undervise med programmering?.....	51
5. Diskusjon av funn.....	53
5.1 Lærernes syn på de ulike begrepene i forhold til teorien	53
5.2 Implikasjoner.....	54
5.2.1 Læreres erfaringer og programmering i skolen.....	54
5.2.2 Metodiske implikasjoner	55
5.2.3 Didaktiske implikasjoner.....	57
5.2.4 Implikasjoner for lærerutdanningen og forskeren	59
5.3 Refleksjoner av egne funn.....	60
6. Konklusjon og oppsummering	62
6.1 Funn og implikasjoner for praksisen	62
6.2 Studiens begrensninger	63
6.3 Videre forskning.....	64
6.4 Avsluttende sammendrag	64
Referanser.....	65
Vedlegg	74
Vedlegg 1 - Samtykkeskjema.....	74
Vedlegg 2 - Intervjuguide.....	77
Vedlegg 3 – Godkjennelse fra NSD	78

Figur-oversikt

Figur 1 – Dybdelæring satt mot overflatelæring, Ludvigsen-utvalgets oversettelse.....	7
Figur 2 - Durrheim (2006, s. 34) sin oversikt over forskningsprosessen	7
Figur 3 - Kilpatrick et al. (2001, s. 5), tråder om matematisk kompetanse.....	17
Figur 4 – Illustrasjon av den proksimale utviklingssone, https://www.simplypsychology.org/vygotsky.html	21
Figur 5 - Grov oversikt over arbeidsprosessen i studien	25
Figur 6 - Eksempel fra transkripsjon	33
Figur 7 - Utdrag av tankekart fra første analyse (Miro)	39
Figur 8 - Utdrag fra tankekart fra endelig analyse	41

Tabell-oversikt

Tabell 1 - Læringsmiljøer, Skovsmose (2003), side 149	10
Tabell 2 - Demografisk oversikt av informanter	30
Tabell 3 - Oversikt over temaer og beskrivelser	42

Forkortelser – Norsk

AR – Algoritmisk resonnering

AT – Algoritmisk tenknings

Udir – Utdanningsdirektoratet

Forkortelser – Engelsk

AR – Algorithmic Reasoning

AT – Algorithmic Thinking

CS – Computer Science

CT – Computational Thinking

IPA – Interpretative Phenomenological Analysis

PR – Positive Reinforcement

ZPD – Zone of Proximal Development

1. Innledning

I den nylig utgitte læreplanen i matematikk for 1.-10. trinn fra Utdanningsdirektoratet (2020c) har programmering fått en sentral rolle som en del av digitale ferdigheter. Programmering skal brukes til å behandle informasjon, utforske og løse matematiske problemer (Utdanningsdirektoratet, 2020a). Elevene skal hensiktsmessig bruke digitale verktøy til å utvikle sin digitale kompetanse, hvor de skal progressivt ta i bruk hjelpemidler i sammenheng med utforskning, finne løsninger og presentere matematiske problemer (Utdanningsdirektoratet, 2020a). Dette fokuset på programmering i den nye læreplanen kommer av den voksende internasjonale interessen for å få programmering i skolen (Sevik, 2016, s. 6).

Jeg skal i denne innledende delen ta for meg bakgrunnen for masteroppgaven. Det vil også bli presentert forskerens tanker og erfaringer med programmering, oppgavens problemstilling, hvilke avgrensninger som er blitt gjort, tidligere forskning, begrepsavklaringer, forskningsdesign, programmeringens faglige sammenheng med matematikk og oppgavens struktur.

1.1 Bakgrunn

Jeg har selv fått mye ut av å arbeide med programmering, både på videregående, to år på dataingeniørskolen og lærerskolen. Jeg har opplevd hvor gøy og lærerikt det er å arbeide med noe jeg trives med og blir utfordret av. I tillegg har jeg fått troen på å arbeide med færre, større, sammenhengende oppgaver i stedet for mange, små, like oppgaver for å tilegne seg kunnskap om et tema.

1.2 Hvorfor arbeide med programmering i skolen?

I perioden fra 2016-2017 deltok 153 norske skoler på et kurs i programmering etter at Kunnskapsdepartementet åpnet for muligheten til å la ungdomsskoleelever delta (Lie et al., 2017, s. 28). Fokuset var å utvikle digitale ferdigheter, som er en av de grunnleggende ferdighetene i både den gamle (Utdanningsdirektoratet, 2006) og nye læreplanen (Utdanningsdirektoratet, 2020a). Et eksempel på et prosjekt for å utvikle digitale ferdigheter er Lego League. Også kalt FIRST LEGO League Challenge, som er mer bestemt for barn og unge mellom 10 og 16 år. Det er et 8 ukers prosjekt med mål om å inspirere barn og unge til å

arbeide med teknologi og programmering (Hjernekraft, 2021b). På Hjernekraft (2021a) sine hjemmesider er det blitt publisert en artikkel som gjennomgår alle kompetansemålene som er relevante i gjennomføringen av Lego League. Her er matematikk, naturfag, samfunnsfag, norsk, KRLE, kunst og håndverk, engelsk, programmering og teknologi og design inkludert i arbeidsmålene (Hjernekraft, 2021b).

1.3 Problemstilling og avgrensninger

Som forsker ønsker jeg å få innsikt i læreres erfaringer med programmering. Jeg har tidligere lest forskning om hvordan programmering er nyttig fra et pedagogisk perspektiv, men jeg har funnet lite som går i dybden av læreres tanker, erfaringer og følelser om programmering. Derfor vil jeg ha fokus på lærere, hvor de jeg har planlagt å intervju har undervisningserfaring innen programmering. Derfor er min problemstilling i denne oppgaven følgende:

Hva tenker ungdomsskolelærere om programmering i skolen og potensial som læringsmetode og problemløsningsstrategi?

Jeg vil avgrense oppgaven ved å ta i bruk fire forskningsspørsmål utarbeidet fra problemstillingen. Det er mer bestemt disse spørsmålene som er utgangspunktet for oppgaven. Forskningsspørsmålene er:

1. Hvordan forstår lærere viktige begreper i programmering?
2. Hva er læreres erfaringer med programmering før og etter den nye læreplanen?
3. Hvordan kan programmering bidra til å få elevene interessert i matematikk?
4. Hvilke fordeler og utfordringer har programmering i skolen?

Første forskningsspørsmål innebærer lærernes forståelse av et utvalg begreper. Andre forskningsspørsmål undersøker i hvilken grad lærerne har jobbet med programmering før og etter den nye læreplanen fra 2020, og hva slags programmeringsspråk eller programmer de har brukt. Denne informasjonen kan brukes til å sammenligne foreliggende kunnskaper med den kunnskapen læreren har måttet tilegne seg etter 2020. Her har informantene mulighet til å fortelle om både personlige erfaringer med programmering, og deres observasjoner av elevenes erfaringer og utbytte. I dette forskningsspørsmålet går vi også dypere i hvordan

programmering har påvirket informantens kunnskaper innen matematikk, siden jeg også er interessert i lærernes personlige erfaringer med programmering, og ikke bare elevhistorier. Tredje forskningsspørsmål går på lærernes refleksjoner rundt programmering og elevene. Spørsmålet gir informantene mulighet til å fortelle om situasjoner hvor elever har jobbet med programmering i skolen og andre steder. Med informant menes de lærerne som har deltatt i de individuelle forskningsintervjuene. Det fjerde forskningsspørsmålet går ut på lærernes refleksjoner om innføringen av programmering i skolen. Her er tanker om utfordringer, fordeler og annet velkomne.

1.4 Tidligere forskning

Vi er ifølge Klaus Schwab (2017) midt i den fjerde industrielle revolusjon. Revolusjonen går ikke bare på utvikling av nye maskiner og systemer, men også det økende behovet for sammensettingen av forskjellige teknologier (Schwab, 2017). Denne digitale revolusjonen spiller på felt som gen-sekvensering, nanoteknologi og kvanteberegning, og hva som oppstår av sammensetninger av disse (Schwab, 2017). Vi ser at revolusjonen også er i gang i Norge, med det økende behovet for digital kompetanse blant både voksne og barn. Jeg ønsker i denne oppgaven å trekke frem Vygotsky og den proksimale utviklingssone (Allazzam, 2015; Britton, 1987; Hall & Rieber, 2012), Dewey og å lære ved å gjøre (Allazzam, 2015; Jarning, 2006; Williams, 2017) og Nosrati og Wæge (2015) sin forståelse av dybdelæring i matematikk i sammenheng med Kilpatrick sine tråder for matematisk kompetanse (2001). Disse læringsteoriene vil bli et grunnlag for hvordan programmering kan bli tatt i bruk for å lære matematikk. Teoriene skal videre settes opp mot hverandre og bli brukt som et helhetlig teoretisk rammeverk.

Det har blitt publisert flere studier om motivasjon til å lære matematikk (Lambić, 2011), introdusering av programmering i skolen (Moreno-León et al., 2016) og utvikling av ferdigheter i forhold til programmering (Falloon, 2016; Husain et al., 2017), hvor det diskuteres noe om de erfaringer elever og studenter har hatt. Det er også omfattende litteratur om undervisning, programmering og informatikk (Computer Science) på et høyere nivå (Grover & Pea, 2013, s. 40). Det er derimot manglende forskning om lærere som underviser i programmering til ungdom og yngre barn (Rich et al., 2019, s. 312).

1.5 Begrepsforklaringer

I denne oppgaven vil det bli tatt i bruk noen faglige begreper, og i dette underkapittelet skal disse begrepene bli diskutert og klargjort. Oppgaven har grunnlag i matematikken og didaktikk, og derfor vil alle begreper bli sett i en matematisk-didaktisk sammenheng.

1.5.1 Programmering

Programmering er nok et begrep de fleste har hørt om, men ulike personer kan ha ulike definisjoner av begrepet. Programmering er et begrep som har ulike forklaringer etter hvem som blir spurt (Rich et al., 2019, s. 314). Forsström og Kaufmann (2018, s. 19) beskriver programmering som en prosess, hvor en gir instruksjoner til en datamaskin som skal gjennomføre ulike oppgaver, som problemløsning. I denne oppgaven blir programmering sett på som et synonym med begrepet koding, hvor de engelske begrepene «programming», «coding» og «computing» kan bli sett på som oversettelser av programmering og koding. Gjøvik og Torkildsen (2019, s. 34) skiller mellom disse ved å se på programmering som å løse et problem med en algoritme eller et program. Programmering innebærer alt av forarbeid, gjennomføring og testing av en algoritme for å løse et problem (Gjøvik & Torkildsen, 2019, s. 34). Koding kan bli sett på som aktiviteten med å ta i bruk programmering med et bestemt programmeringsspråk (Gjøvik & Torkildsen, 2019, s. 34). Programmering og koding kan bli sett på som synonymer, men programmering er begrepet som vil bli tatt i bruk i denne oppgaven.

1.5.2 Læringsmetode og problemløsningsstrategi

Programmering vil i denne oppgaven bli brukt i sammenheng med begrepet «læringsmetode» eller «læringsstil». Læring er et komplekst begrep, og det har ingen fast definisjon (Lave, 2009, s. 1). Læring kan bli sett på som en prosess for å lage en ny forståelse av en erfaring, som gir bedre forståelse og verdsettelse for erfaringen (Lave, 2009, s. 53). En læringsstil er foretrukne måter å lære og studere på, som å jobbe sammen med andre eller bruke bilder i stedet for tekst (Pritchard, 2017, s. xii, 5). En kan derfor se på læring som en form for utvikling av kompetanse. En læringsmetode kan dermed bli forstått som å bruke et system som utgangspunkt for å utvikle kompetanse. Programmering kan derfor være en metode for å lære algoritmisk tenkning. Et annet begrep som kan kobles til programmering er «problemløsningsstrategi». Først må begrepet «problem» bli undersøkt. Nickles (1981, s. 111)

definerer et problem ut ifra de ulike mulige løsningene tilgjengelig. Et problem kan bli forstått som noe som oppstår når en har et mål, men uten en klar framgangsmåte for å nå dette målet.

Videre har begrepet «problemløsning» ingen fast definisjon, men kan ses på som en måte å få elever til å tenke kreativt (Schoenfeld, 2016, s. 4). Holyoak (1990) ser på problemløsning som et av de viktigste delene av menneskelig tankemåte, hvor en møter problemer hele tiden i dagliglivet. Problemløsning kan også oppfattes som utforskende, hvor utforskning i matematikk går på å «undersøke en matematisk problemstilling» (Stedøy, 2018, s. 3).

Problemløsning skal ikke bli sett på som et innat mål, men som et middel for å gjøre det lettere å nå et mål (Schoenfeld, 2016, s. 5). Polya (2004, s. 5-6) beskriver problemløsning ved hjelp av fire steg: (1) forstå, (2) planlegge, (3) utføre og (4) reflektere. For å løse et problem må problemet først (1) forstås, slik at en kan fortsette til (2) planleggingsfasen.

Planleggingsfasen (2) innebærer å tenke tilbake på lignende problemer som er kjent fra før, og om den kunnskapen kan anvendes i problemet. Etter planleggingen er det på tide med (3) utførelsen, som går på å teste ut kunnskapen og se om problemet kan bli løst (Simpol et al., 2017, s. 1). (4) Reflektere innebærer å se tilbake på hva en har gjort og undersøke om noe kan bli gjort på en annen måte eller endres. Å lage en vane for å kritisere eget arbeid kan gi flere løsninger, og evnen til problemløsning vil utvikles (Polya, 2004, s. 6-7). Dermed kan begrepet «problemløsningsstrategi» bli forstått som prosessen for å løse en oppgave hvor fremgangsmåten er ukjent. Dette kommer av manglende erfaring med denne oppgavetypen, og ulike strategier for å løse problemet må bli utforsket.

1.5.3 Algoritmisk tenkning vs. algoritmisk resonnering

Siden studien tar for seg matematikk og programmering, må forskjellene mellom begrepene algoritmisk resonnering og algoritmisk tenkning eller tankegang, som lett kan forveksles med hverandre, bli klargjort. Det engelske begrepet «Computational Thinking» (CT), oversatt til norsk som algoritmisk tenkning, ble først tatt i bruk av Seymour Papert (2020). Det er også noe forskjell mellom uttrykkene på norsk og engelsk (Gjøvik & Torkildsen, 2019, s. 32).

Bocconi et al. (2018, s. 8) beskriver algoritmisk tankegang som et paraply-begrep, som inkluderer enkelte CT-trekk. Algoritmisk tankegang blir dermed sett på som en del av CT, hvor det norske og engelske begrepet blir forstått som synonymer. Dette stemmer derimot ikke overens med bruken av begrepet i annen litteratur, hvor forskjellige definisjoner blir tatt i bruk i forskjellig litteratur (Tedre & Denning, 2016, s. 1). Definisjonen av CT har blitt

presentert av Wing (2011, s. 1), kalt for «Cuny-Snyder-Wing-definisjonen» (Tedre & Denning, 2016, s. 124-125):

«Computational thinking is the thought process involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.»

Algoritmisk tenkning innebærer å løse et problem ved å formulere problemet og løsningen på en måte som kan effektivt bli løst av en datamaskin, et menneske eller en kombinasjon av disse (Wing, 2011, s. 1). Algoritmisk resonnering (Algorithmic Reasoning, AR) går på å ta i bruk en algoritme for å nå et svar (Lithner, 2008, s. 259). Gjøvik og Torkildsen (2019, s. 31) beskriver bruken av begrepet som mer problematisk når det brukes i en skolesammenheng. Algoritmisk tenkning vil bli diskutert nærmere i kapittel 2.3.

1.5.4 Dybdelæring

Ludvigsen-utvalget fikk i 2013 en oppgave om å vurdere skolens faglige innhold i lys av kravene til kompetanse. Det ble avdekket at innholdet i alle fag var lite sammenhengende og altfor omfattende. Det ble dermed anbefalt av Ludvigsen-utvalget å fokusere på dybdelæring for å bedre læringsutbytte. Dybdelæring har ulike definisjoner etter hvor en ser, men Sawyer (2006, s. 4) setter dybdelæring opp mot overflatelæring for å bedre beskrive begrepet. I figur 1 (neste side) blir ulike aspekter ved dybdelæring satt opp mot overflatelæring for å bedre forklare hva begge begrepene går ut på.

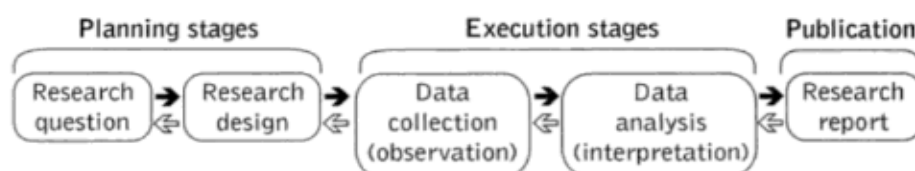
Dybdelæring	Overflatelæring
Elever relaterer nye ideer og begreper til tidligere kunnskap og erfaringer.	Elever jobber med nytt lærestoff uten å relatere det til hva de kan fra før.
Elever organiserer egen kunnskap i begreps-systemer som henger sammen.	Elever behandler lærestoff som atskilte kunnskaps-elementer.
Elever ser etter mønstre og underliggende prinsipper.	Elever memorerer fakta og utfører prosedyrer uten å forstå hvordan eller hvorfor.
Elever vurderer nye ideer og knytter dem til konklusjoner.	Elever har vanskelig for å forstå nye ideer som er forskjellige fra dem de har møtt i læreboka.
Elever forstår hvordan kunnskap blir til gjennom dialog og vurderer logikken i et argument kritisk.	Elever behandler fakta og prosedyrer som statisk kunnskap, overført fra en allvitende autoritet.
Elever reflekterer over sin egen forståelse og sin egen læringsprosess.	Elever memorerer uten å reflektere over formålet eller over egne læringsstrategier.

Figur 1 – Dybdelæring satt mot overflatelæring, Ludvigsen-utvalgets oversettelse.

I denne oppgaven vil dybdelæring bli sett på som å se mønstre og sammenhenger i ny kunnskap, og relatere dette til kunnskap en allerede besitter. Dybdelæring vil bli nærmere gjort rede for i kapittel 2.5.

1.6 Oppgavens forskningsdesign – Metode

Durrheim (2006, s. 34) beskriver forskningsdesign som en strategi for å binde forskningsspørsmål med utførelsen av forskning. Som vist i Figur 2 (under) er et forskningsdesign en del av planleggingsfasen, sammen med utviklingen av forskningsspørsmål. Videre skal data bli samlet og analysert, for så å publisere en forskningsrapport. Denne forskningsprosessen er i grove trekk hva oppgaven tar utgangspunkt i.



Figur 2 - Durrheim (2006, s. 34) sin oversikt over forskningsprosessen

Som planlagt ble tre lærere på ungdomsskolenivå intervjuet, og dataene hentet fra intervjuene er utgangspunkt for oppgaven. Det er ikke meningen at de tre informantenes refleksjoner og tanker skal representere de refleksjoner og tanker resten av Norges lærere har. Det er hverken utgangspunktet eller målet for oppgaven, siden antall informanter er såpass begrenset.

Oppgaven tar derfor utgangspunkt i bare de utvalgte læreres tanker og refleksjoner, uten et mål om generalisering. Mer detaljer om metode og design kommer i kapittel 3.

1.7 Faglig sammenheng

Å trekke linjer mellom programmering og matematikk er ikke noe nytt, siden algoritmisk tenkning har blitt sterkt knyttet til matematikk tidligere (Feurzeig et al., 1970). De første datamaskinene ble bygget av matematikere, og siden den gang har programmering og matematikk blitt sett på som sterkt tilknyttede aktiviteter (Misfeldt & Ejsing-Duun, 2015, s. 2524). Det er flere land som har inkludert programmering i de nye læreplanene sine. For eksempel har Frankrike lagt inn programmering sammen med matematikk, og Sverige og England har koblet programmering med design og ingeniørarbeid (Misfeldt & Ejsing-Duun, 2015, s. 2524). Det har de siste årene blitt flere land i Europa og Asia som har satt et fokus på programmering i opplæringen, og derfor er det flere lærere som blir bedt om å lære bort programmering til yngre barn (Rich et al., 2019, s. 311-312).

1.8 Oppgavens struktur

For å best mulig svare på min problemstilling har jeg valgt å dele oppgaven i følgende kapitler: Først blir relevant teori innen matematikk og programmering presentert i kapittel 2. Teori-delen tar for seg hvordan programmeringsbegrepet kan bli forstått, ulike begreper knyttet til programmering og studier som er relevant for oppgaven. Kapittel 3 tar for seg den kvalitative metode, som har blitt tatt i bruk i denne oppgaven. Jeg skal også ta for meg hvordan studiens kvalitet kan bli sikret, og viktigheten av etikk i forskning. Tre lærere har blitt individuelt intervjuet, og intervjuene har blitt transkribert. Kapittel 4 tar for seg analysen av disse intervjuene og presenterer hva som ble funnet, hvor det er transkripsjonene som skal bli nøye analysert. Funns blir diskutert i kapittel 5, og hele oppgaven blir avslutningsvis oppsummert i kapittel 6.

2. Teori

For å svare på oppgavens problemstilling må det bli satt søkelys på teorier og forskning som er relevant til feltet. Målet med oppgaven er å se på læreres tanker om hvordan programmering kan bli brukt til å få elever til å lære og bli interessert i matematikk. Dermed vil dette kapittelet ta for seg hva matematikk og læring er, og hvordan disse temaene kan fungere i samsvar med programmering. Dette kapittelet inneholder en oversikt over hva begrepet matematikk egentlig inneholder, Skovsmoses undersøkelseslandskap, de grunnleggende ferdighetene i matematikk, en dypere gjennomgang av algoritmisk tenkning, programmerings rolle i skolen, læringsteoriene til Vygotsky, Nosrati og Wæge og Dewey, og en oppsummering av oppgavens sammenlagte teoretiske rammeverk.

For å finne forskning relevant for denne oppgaven ble det tatt i bruk søkemotoren Google Scholar og biblioteksdatabasen Oria. Ved søk etter ord og intervaller for utgivelsesår fant disse verktøyene relevant litteratur som er dagsaktuell.

2.1 Ha er egentlig matematikk?

Hans Freudenthal (1968, s. 7) beskriver matematikk som en menneskelig aktivitet, og gir mulighet til å «matematisere» virkeligheten. Matematikk er ulik andre undervisningstemaer siden mengden kunnskap er sammenlignet liten, samtidig som kunnskapen kan bli brukt i flere ulike situasjoner (Freudenthal, 1968, s. 5). Freudenthal (1968, s. 5) tar for seg to ekstreme måter matematikkundervisningen kan utfolde seg på: (1) elevene får opplæring i håp om at de selv forstår hvordan matematikk kan bli brukt i hverdagen, og (2) elevene får opplæring i «brukbar» matematikk, som innebærer at elever kan bruke hva de har lært i hverdagslige situasjoner. En utfordring med brukbar matematikk (2) er risikoen for å bli ubrukelig hvis situasjonen eller konteksten endrer seg (Freudenthal, 1968, s. 5). Hvis en elev møter en utfordring som ikke er tilstrekkelig lik noe kjent fra før, vil ikke eleven klare å bruke kunnskapen de innebærer. Dermed er det vanskelig for mange elever å anvende kunnskapen sin, ikke bare i fysikk- eller kjemi-laboratoriene på skolen, men også i hverdagen (Freudenthal, 1968, s. 5). Freudenthal (1968, s. 5) beskriver ikke-anvendbar matematikk til hverdagen som ren matematikk. Ren matematikk beskrives av Skovsmose (2003) som regneoppgaver uten kontekst til virkeligheten. Ren matematikk er inkludert i Skovsmoses (2003) modell for undersøkelseslandskaper, som tar for seg hvordan matematikk kan bli tatt i bruk i forskjellige situasjoner.

Ole Skovsmose (2003) beskriver undersøkelseslandskapet som elevers mulighet for utforskning og undring. Elever trenger muligheten til å bruke kreativiteten sin og finne ut av ting selv, hvor elevene selv også må gripe denne muligheten. Dette må komme gjennom lærerens spørsmål og hvordan disse spørsmålene blir formulert. Det er læreren sin oppgave å spørre elevene om hvordan ting fungerer, og gi spørsmål som skaper undring og lyst til å utforske. Skovsmose (2003) understreker at ikke alle elevgrupper er villige til å arbeide på denne måten, etter hva de kan og hva de er vant til. Dermed er det også læreren sin oppgave å vurdere hva som fungerer for sine egne elever. På denne måten er ikke oppgaver med undersøkelseslandskapet som motivasjon ideelt for alle elevgrupper.

Skovsmose (2003) skiller videre mellom undersøkelseslandskapet og mer tradisjonell matematikk, som også blir omtalt som oppgaveparadigmet. Oppgaveparadigmet går på å arbeide med oppgaver etter at læreren har vist hvordan lignende oppgaver skal arbeides med og løses. Dette står i kontrast med undersøkelseslandskapet, siden elevene frarøves muligheten til å arbeide med utforskning. Ifølge Skovsmose (2003) kan oppgaver være i stil med ren matematikk, altså rene utregning-oppgaver uten noen form for virkelighetsnær kontekst. En mellomting av ren matematikk og virkelighetsnærhet i oppgaver er hva Skovsmose (2003) refererer til som en semivirkelighet. Dette går på oppgaver som har forsøkt å ha virkelighetsnær kontekst, men som ikke helt når målet om den reelle konteksten. Undersøkelseslandskapet og oppgaveparadigmet, sammen med oppgaver av ren matematikk, semivirkelighet og virkelighetsnære oppgaver kan bli funnet i tabell 1 (under) laget av Skovsmose (2003).

	Oppgaveparadigmet	Undersøkelseslandskaper
Referanser til «ren» matematikk	(1)	(2)
Referanser til en «semi-virkelighet»	(3)	(4)
Reelle referanser	(5)	(6)

Tabell 1 - Læringsmiljøer, Skovsmose (2003), side 149

2.2 Grunnleggende ferdigheter i matematikk

Det er skolens oppgave å legge til rette for en tilfredsstillende opplæring i de grunnleggende ferdighetene (Utdanningsdirektoratet, 2020d). De grunnleggende ferdighetene i matematikk innebærer lesing, skriving, regning, muntlige ferdigheter og digitale ferdigheter (Utdanningsdirektoratet, 2020a). Hver grunnleggende ferdighet har et hensiktsmessig mål i opplæringen, og læreplanen beskriver hvordan utviklingen til hver ferdighet kan utfolde seg.

Å kunne lese går på å kunne forstå og danne mening av både hverdagslige og matematikkfaglige tekster. Elevene skal møte progressivt mer kompliserte tekster og utvikle leseferdighetene ved å finne og bruke informasjon. Skriving utgjør hensiktsmessige representasjoner av sammenhenger, oppdagelser og ideer. Elevenes utvikling går på å gå fra bruk av hverdagsspråket til et mer komplisert matematisk språk. Å kunne regne går på vurdering av løsninger og kunne bruke ulike framgangsmåter til å gjøre utregninger. Elevene må gjenkjenne i hvilke problemer det er hensiktsmessig å ta i bruk regning, og formulere spørsmål om disse problemene. Utvikling innen regning går på å løse stadig mer komplekse problemer med bruk av hensiktsmessige begreper og strategier. Muntlige ferdigheter i matematikk innebærer å bli forstått og skape mening i samtalen om matematikk. Elevene skal klare å kommunisere ideer, tanker og strategier med andre. Utviklingen av muntlige ferdigheter bygger på overgangen fra det hverdagslige språket til et mer spesialisert matematisk språk. Til slutt er de digitale ferdighetene, hvor elevene skal ta i bruk en rekke ulike verktøy for å løse matematiske problemer. Elevene skal lære å bruke verktøy på en hensiktsmessig måte til å løse, presentere og utforske ulike problemer. Eksempler på slike verktøy er regneark, CAS, ulike geometriske programmer og programmering. Det er ikke programmering i seg selv som blir argumentert for å bli iverksatt i skolen, men heller den algoritmiske tenkningen som kommer av programmering. Det er flere kilder som trekker frem overførbarheten av ferdigheter fra algoritmisk tenkning i programmering til matematikk (Forsström & Kaufmann, 2018; Grover & Pea, 2013; Wing, 2006, 2011).

2.3 Algoritmisk tenkning

Fra tidligere har begrepet algoritmisk tenkning blitt sammenlignet med det engelske begrepet «Computational Thinking», noe Utdanningsdirektoratet også tar i bruk (Utdanningsdirektoratet, 2019). Dette kapittelet vil ta for seg innholdet i begrepet algoritmisk tenkning og hva de ulike aspektene innebærer.

Grover og Pea (2013, s. 39) presenterer syv aspekter som hjelper med å forklare hva algoritmisk tenkning innebærer:

1. En menneskelig aktivitet
2. Abstraksjon
3. Tilrettelegging av læring
4. Verktøy
5. Kreativ prosess
6. Tilnærminger for problemløsning
7. Innovasjon til andre felt

(1) Algoritmisk tenkning skjer gjennom programmering, og kan beskrives som en kreativ menneskelig aktivitet. (2) Abstraksjon går på å generalisere og se på mønstre for å løse vanskelige oppgaver. (3) Tilrettelegging av læring kommer av data og informasjon. (4) Algoritmer fungerer som verktøy for å utvikle og gi uttrykk for løsninger av algoritmiske problemer. (5) Programmering er en kreativ prosess som danner et produkt. (6) Digitalt utstyr og systemer vil bidra til å se sammenhenger mellom algoritmiske problemer og algoritmiske løsninger. (7) Algoritmer påvirker innovasjon til felt som medisin, ingeniørfag og andre virksomheter. Algoritmer blir sett på som gjennomsyrende i den globale økonomien (Grover & Pea, 2013, s. 40).

Knuth (1985, s. 170) tar i bruk begrepet «Algorithmic Thinking» (AT) for å beskrive det overordnede innholdet i informatikk (data science), til forskjell fra andre som tar i bruk CT (Bocconi et al., 2018; Grover & Pea, 2013; Leonard et al., 2016; Nouri et al., 2020; Tedre & Denning, 2016; Wing, 2006). Knuth (1985, s. 170) forteller om noen av hans kolleger, som er uenige i hans bruk av begrepet. Dette kommer av Knuth sin bredere forståelse av begrepet algoritme. Algoritmisk tenkning kan bli forstått som alle konseptene innenfor godt definerte prosesser, i tillegg til hvordan strukturen og sekvensen av data blir forstått (Knuth, 1985, s.

170). Også begrepet tenkning må bli klargjort, siden det er noe forskjell mellom tenkning, tenking, tankegang og tenkemåte.

Gjøvik og Thorkildsen (2019, s. 31) refererer til Språkrådet (2007) om disse forskjellene. Tenkning er mer omfattende enn tenking, som betyr «det å tenke» (Gjøvik & Torkildsen, 2019, s. 31). Tenkning er også et eget felt innen kognitiv psykologi (snl.no, 2018). Tenking går på å bearbeide noe en husker av følelser eller opplevelser (snl.no, 2018). Tankegang kobles direkte til begrepet «tankerekke» (Gjøvik & Torkildsen, 2019, s. 31). Tenkning er et begrep som kobles mer til mennesker (snl.no, 2018). Tenkning-begrepet blir satt opp mot biologi, som hjernen og nervesystemet, og sosiologi, med språk og kultur. Siden forskningen i denne oppgaven ser på mennesket og deres opplevelse av verden, vil begrepet tenkning med definisjon underforstått bli tatt i bruk, med forbehold om bruk av synonymer som tankegang og tankemåte for å bedre tekstens flyt.

2.4 Programmering og algoritmisk tenkning i skolen

Ifølge Wing (2006, s. 35) er det et sterkt behov for å tenke algoritmisk for å løse matematiske problemer ved hjelp av programmering på datamaskiner. Algoritmisk tenkning bli også sett på som en viktig ferdighet for alle, ikke bare for de som driver med data (Wing, 2006, s. 33). Dette fordi algoritmisk tenkning innebærer problemløsning, å lage systemer og forstå menneskelig oppførsel. Misfeldt og Ejsing-Duun (2015, s. 2526) presenterer en metafor for å beskrive en algoritme, som går på å følge en oppskrift: (1) Først skal alle tørre ingredienser sammen. (2) Så skal alt røres sammen. (3) Etter legges til vann og rør igjen. (4) Hvis deigen er god, rør i to minutter. Hvis ikke, gå tilbake til (3) og legg til mer vann. Algoritmisk tenkning handler om å utvikle, utføre og lage maskiner som skal håndtere slike algoritmer (Misfeldt & Ejsing-Duun, 2015, s. 2526).

2.4.1 Fordeler

Å kunne tenke med algoritmer og prosedyrer blir sett på som viktige mål i matematikken (Misfeldt & Ejsing-Duun, 2015, s. 2524). Humble et al. (2019a, 2019b) beskriver analog programmering som å drive med programmering uten digitale hjelpemidler.

Dette kan være å få medelever til å utføre handlinger i en bestemt rekkefølge eller løse et puslespill uten å selv se puslespillet (Aranda & Ferguson, 2018). Humble et al. (2019a, 2019b) har funnet at læreres tiltro til analog programmering er høy, når det gjelder å trene

elever til å vise sine tankemåter og prosedyrer. Elevene får trening i å bryte ned problemer i mindre deler og strukturere matematiske konsepter (Humble et al., 2019a, 2019b). Rich et al. (2019, s. 328) konkluderer i sin artikkel med en rekke positive sider ved å drive med programmering for 5-14-åringer. Det ble oppdaget (1) overveldende entusiasme fra elever, lærere og foreldre, (2) lærere som spør sine egne elever om hjelp og (3) elever som jobber uten å gi opp til tross for utfordringer. Det må også nevnes at denne entusiasmen kan komme av programmeringens fremmedhet og spenninger rundt temaet (Rich et al., 2019, s. 328). Det er med andre ord mulig at programmering ikke lenger er like spennende etter at elevene har blitt vant med det. Etter som elevene programmerer kan de også oppleve ulik grad av positiv forsterkning (Positive Reinforcement), som går på gjentatte beskjeder om at det de gjør er riktig (de ser at programmet fungerer, mindre feilmeldinger enn tidligere) (Pritchard, 2017, s. 11).

2.4.2 Utfordringer

I skolen kan bruken av teknologi for å løse matematiske oppgaver bli sett på som instrumentell (Guin et al., 2006). Dette kan bli beskrevet som en pragmatisk mekling, som går på å bare bruke teknologi for å løse en oppgave (Guin et al., 2006; Rabardel & Bourmaud, 2003). Hvis det derimot er et fokus på å lære noe med hjelp av teknologi, skjer en epistemisk mekling (Guin et al., 2006; Rabardel & Bourmaud, 2003). En utfordring kan dermed være å opprettholde et epistemisk fokus, slik at programmering og teknologi ikke blir redusert til et verktøy for læring. Misfeldt og Ejsing-Duun (2015, s. 2529) tar i bruk observasjoner gjort i klasserommet fra *Children as Learning Designers in a Digital School*. Prosjektet tar for seg viktigheten av hvordan elevers digitale produksjon påvirker læring og hvordan informasjonsteknologi bidrar til aktiv læring (Levinsen et al., 2014). Fra prosjektet vet vi at elever trenger hjelp med matematiske konsepter når de skal lære seg et programmeringsspråk for å utvikle spill. Misfeldt og Ejsing-Duun (2015, s. 2527-2528) trekker frem to eksempler, der elevene må kommunisere med lærer og hverandre, og teste ut spill utviklet av andre.

En annen rekke utfordringer presentert av Rich et al. (2019, s. 325) er angående tilgang på ressurser. Disse utfordringene ble delt inn i fire kategorier: kvantitet, forbindelse, funksjonalitet og undervisning. Kvantitet omfatter lærernes tilgang til nødvendig utstyr, som datamaskiner eller nettbrett. Forbindelse var gjennomgående og går på nettverkskapasitet og netthastighet. Siden programmer i stor grad er nettbaserte, er en trygg og kjapp

nettforbindelse nødvendig for å få en god læringssituasjon (Rich et al., 2019, s. 325). Funksjonalitet spiller på utfordringer med utstyret, som at datamaskinene ikke fungerte eller vanskeligheter med å få lastet ned programvare (Rich et al., 2019, s. 325). Det kom også fram et manglende behov om lisenser for programvare eller andre ressurser. Til slutt var å planlegge, tilrettelegge og hvordan integrere programmering i andre fag i undervisningen en utfordring (Rich et al., 2019, s. 325). Programmering ses på som vanskelig å integrere i skolen, som kommer av eksempelvis svak lærerkompetanse og mangel på spesifikke timeantall (Sanne et al., 2016, s. 73).

Skolene har i mange år hatt ulike tradisjoner for de ulike fagene, og integrering av programmering ses på som vanskelig. I tillegg er det uklart hva selve faget programmering skal inneholde (Sanne et al., 2016, s. 73). Stigberg og Stigberg (2020, s. 493) presenterer forskning som fremhever utfordringer fra svenske læreres ståsted i forhold til programmering i den nye læreplanen i matematikk: Matematiske ferdigheter er ikke oppdaterte, lite informasjon om hvorfor programmering er inkludert i algebra-delen av læreverket, og innholdet i programmering beskrives som ambisiøst. Først er det uklart om programmering er nødvendig i matematikkundervisningen, hvordan kunnskaper innen programmering burde bli testet og om programmering har en plass i de nasjonale testene. Det er i tillegg manglende forskning som tilsier at programmering har en direkte positiv virkning på matematikkferdigheter. Forsström og Kaufmann (2018, s. 28) har i sin forskning funnet lite som støtter påstander om at programmeringens potensial i utdanningen til å lære matematikk. Derimot fant de delvis gjennomgående resultater om at programmering øker interesse og prestasjoner i matematikk til en viss grad, men at disse resultatene ikke kan generaliseres (Forsström & Kaufmann, 2018, s. 28). Videre kommer utfordringen om hvorfor programmering har fått en særegen plass i algebra-temaet, når forskning har funnet at programmering i stor grad blir arbeidet med i sammenheng med geometri (Forsström & Kaufmann, 2018, s. 30). Dette kommer av sirkelgeometriens tilknytning til robotikkaktiviteter, siden omkretsen av hjulene til roboten må bli funnet for å kunne bevege seg en viss distanse (Forsström & Kaufmann, 2018, s. 27). Til slutt er det uklareheter i læreplanen om programmeringens mål.

2.4.3 Læreres kunnskap og holdninger

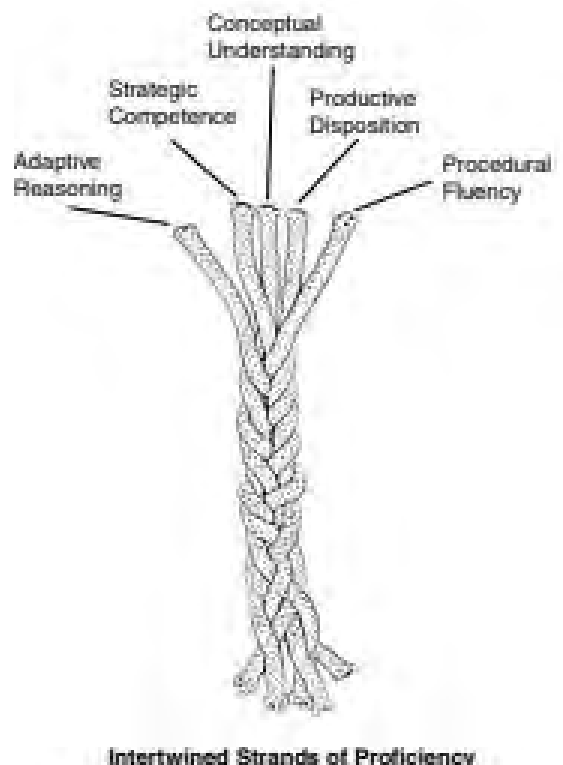
En holdning kan bli sett på som en tendens mot noe (Zeigler-Hill & Shackelford, 2020). Dette kan være en situasjon, en ny læreplan eller en del av den nye læreplanen, som programmering. En holdning kan også bli definert som en samlet vurdering av noe, for eksempel et tankeobjekt (Bohner & Wänke, 2002, s. 5). Med tankeobjekt menes noe konkret eller abstrakt. Videre inkluderes hva som assosieres med holdningen, om følelser, tanker og erfaringer (Zeigler-Hill & Shackelford, 2020). Mine forskningsspørsmål tar for seg de holdninger informantene har til programmering, og hvilke erfaringer de har til den nye delen av læreplanen. Summen av de erfaringer og tanker fra informantene kan gi et godt bilde av den samlede holdningen til programmering i skolen de har. Det er derimot ikke sikkert at informantenes holdninger kommer fram på riktig måte, eller at holdningene blir tolket på en annen måte av forskeren.

Forskning viser at fem kategorier for engstelse ble identifisert fra programmeringsopplæring med lærere: utvendig, innvendig, personlig, elev og ingen (Rich et al., 2019, s. 325). Blant disse var det tydelig personlige grunner som var mest omfattende blant lærere, som går på egne usikkerheter innen kodekunnskaper, potensial for å lære eller å mindre kunnskap enn sine egne elever.

2.5 Kilpatrick, Nosrati og Wæge, matematisk kompetanse og dybdelæring

For en vellykket opplæring i matematisk kompetanse har Kilpatrick med kolleger (2001, s. 5) presentert fem komponenter i en tråd-metamorfe.

Metaforen bygger på at hver for seg er en tråd lite sterk, men sammen kan de bygge et tau som er slitesterkt og solid. Dette konseptet er illustrert i Figur 3 (til høyre). Trådene er koblet til de engelske begrepene (fra venstre til høyre i Figur 3) «Adaptive Reasoning», «Strategic Competence», «Conceptual Understanding», «Productive Disposition» og «Procedural Fluency». Disse uttrykkene har i nevnt rekkefølge blitt oversatt av Ludvigsen-utredningen (2015, s. 57) til resonnering, beregning, forståelse, anvendelse og engasjement. Begrepene oversettelser og innhold har blitt tatt fra samme utredning.



Figur 3 - Kilpatrick et al. (2001, s. 5), tråder om matematisk kompetanse

Resonnering går på muntlige ferdigheter, evnen til å kommunisere hva en tenker, følge en logisk resonnering og være kritisk til resonneringen. En må ta utgangspunkt i kunnskap for å bygge en god resonnering, finne hva som er ukjent og videre undersøke dette.

Beregning innebærer å hensiktsmessig utføre matematiske prosedyrer på en fleksibel og nøyaktig måte. Elever som forstår hvilken prosedyre som er mest nyttig i en gitt situasjon har mulighet til å bytte mellom forskjellige prosedyrer etter hva som er hensiktsmessig.

Forståelse går på kunnskap om begreper og sammenhengen mellom disse begrepene. I tillegg trenger elevene å se sammenhenger mellom ideer og prosedyrer dannet av disse begrepene. Elevene må også forstå hvilke representasjoner som er mest relevant, og variere mellom disse etter måte.

Anvendelse går på å gjenkjenne problemer og klare å representere problemene på ulike måter. Elevene må selv utvikle strategier for problemer og selv vurdere om løsningene er gyldige og relevante. Det nevnes også at med problemer menes både teoretiske problemer og problemer fra samfunnet og hverdagen.

Engasjement tar utgangspunkt i å se på skolearbeidet som noe relevant til sitt eget liv. Også å ha troen på at matematikk er noe en kan få til, er viktig for elevenes engasjement.

Oppsummert må skolearbeidet være verdifull og nyttig, og gi elevene tro på at det er mulig å utvikle gode ferdigheter som er langvarige. Kilpatrick et al. (2001) beskriver disse trådene som gjensidig avhengige av hverandre (codependency), og må fungere og utvikles parallelt. Hvis en tråd er ekskludert, vil ikke de andre trådenes funksjon ta i kraft. Faglig utvikling med utgangspunkt i disse trådene kan bidra til en langvarig forståelse av faget som er både nyttig, fleksibel og relevant (Ludvigsen et al., 2015).

Nosrati og Wæge (2015, s. 4-6) presenterer fem komponenter i et forsøk på å beskrive hva begrepet dybdelæring i sammenheng med matematikk innebærer. Komponentene er satt sammen av modeller fra ulike forskere (Flavell, 1979; Hiebert & Lefevre, 1986; Kilpatrick et al., 2001; Schneider & Artelt, 2010; Skemp, 1976). Komponentene er (1) begrepsmessig forståelse, (2) prosedyrekunnskap, (3) anvendelse, (4) resonnering og (5) metakognisjon og selvregulering. (1) Det er viktig å kunne mer enn bare regler og formler. Begreper og strukturer bygd opp av begreper bidrar til forståelse rundt ideer og konsepter. (2) Prosedyrekunnskap innebærer en automatisering og fleksibilitet i forhold til hvilke prosedyrer som skal bli tatt i bruk i en viss situasjon. Elevene må selv velge hvilke strategier som skal bli tatt i bruk i arbeidet sitt, og det er deres valg om en prosedyre skal forkastes eller ikke. (3) Anvendelse går på å kunne gjenkjenne matematiske problemer, og finne en strategi for å løse problemene. Elevene må også kunne stille seg kritiske til de resultatene de får, og se på resultatene i kontekst av problemet. (4) Elevene trenger ferdigheter innen kommunikasjon, hvor de må ha ferdigheter til å forklare hvordan de selv tenker, og klare å følge resonneringen til en annen. (5) Igjen må elevene stille seg kritisk til eget arbeid, hvor de må stoppe arbeidet sitt og vurdere om egen framgangsmåte er hensiktsfull. De må reflektere over hvordan og hva de lærer, og vurdere om en annen strategi er hensiktsmessig å ta i bruk. Denne type bevisstgjøring av egne læringsprosesser og strategier gir elevene mulighet til å ta i bruk selvregulering i arbeidet sitt (2015, s. 6). Avslutningsvis skriver Nosrati og Wæge (2015, s. 6) at de fem komponentene skal både fungere og utvikles sammen, i likhet med Kilpatrick sine tråder for dybdelæring.

2.6 «Learning by doing» og Dewey

John Dewey var en amerikansk filosof og psykolog som blir sett på som den første forkjemperen for at barn skal lære med aktivitet (Imsen, 2005). Dette står i motsetning til den tradisjonelle undervisningen, hvor det er vanlig at elevene skal sitte i ro og høre passivt på lærer eller foreleser (Dale, 2001). Denne passiviteten kan føre til et overskudd av energi for elevene, som igjen vil føre til at undervisningen ikke blir så optimal som den kunne ha vært. Elevene må sitte i ro i lengre perioder og passivt følge med på hva som skjer på tavla, hvor de bare har mulighet for utløp av energi når det er pause. Dewey mente at kroppen var et verktøy, som skulle brukes til å både kvitte seg med den ekstra energien og lære av det som skjer i en undervisningssammenheng (Dale, 2001; Imsen, 2020).

Det berømte uttrykket «learning by doing» stammer fra J. A. McLellan sin utgivelse i 1889, *Applied Psychology*, som brukte uttrykket: «Learn to do by knowing and to know by doing» (McLellan & Dewey, 2008, s. 182). Dewey skrev ikke utgivelsen med McLellan, men på grunn av Dewey sin sterke innflytelse ble navnet hans nevnt ved siden av McLellan sitt (Reitan, 2015). Dette berømte uttrykket ble fulgt opp med kritikk fra Dewey om mangelfull aktivitet i den tradisjonelle skolen.

Det var ikke bare den tradisjonelle undervisningen Dewey kritiserte, men oppfatningen av å lære ved å gjøre. Han bemerket at det ikke bare er å erstatte tekstbøker med fysiske oppgaver eller lignende, men bruke hele kroppen for å lære (Dewey, 1899). På denne måten mener ikke Dewey at «Learning by doing» skal være et synonym for erfaring, men heller at hender, øyne, ører og kroppen som helhet skal bidra til å skaffe informasjon (Dewey, 1899). McLellan (2008), som er sterkt inspirert av Dewey, gir sterk kritikk til de som reduserer «Learning by doing» til enkel aktivitet. Noe som skal gjøres kan tross alt ikke bli gjort før kunnskapen er til stede (Dewey, 1899). Rolandsson (2020) skriver om å programmere med hensikt, og problematiseringen å programmere i par. Ifølge Dewey lever mennesker etter vaner, uten noen form for refleksjon (Thayer, 1988, s. 270). Denne teorien er viktig å kjenne til når det skal bli satt søkelys på hvordan nye konsepter blir forstått og lært. Rolandsson (2020, s. 2) trekker linjer mellom Dewey (Thayer, 1988) sin forståelse av konsepter og koding. Artikkelen tar for seg de ulike måtene en person kan lære konsepter i forhold til koding, og de utfordringer som kommer av å lære i en sosial setting. Det som er mer relevant i forhold til denne oppgaven er Rolandsson (2020, s. 2) sin beskrivelse av arbeidsprosessen: I arbeid med

koding får elever oppleve de utfordringene, hindringene og forstyrrelsene en kan møte når en jobber med et problem. Dette arbeidet gir i tur elevene erfaring i hvordan en potensiell løsning kan se ut, etter hvert som de får bedre forståelse for problemet. Kort fortalt må elevene oppleve og erfare problemet, slik at de kan forstå problemet og etter hvert jobbe mot en løsning selv. De må lære ved å gjøre.

2.7 Proksimal utviklingssone og Vygotsky

Lev Semyonovich Vygotsky, en psykolog fra Sovjetunionen, var opptatt av hvordan verden og mennesker som lever i den kan bli forstått. Verden er både fysisk, kulturell og sosial, og Vygotsky var kanskje mest kjent for sitt arbeid med loven for kulturell utvikling. Denne loven kan defineres som at enhver funksjon i den kulturelle utviklingen av et barn forekommer to ganger: en gang på et sosialt nivå og en gang på et psykologisk nivå (Vygotsky, 1997). En kan se på dette som at barn lærer først i en sosial setting, og danner så forståelse i sitt eget hode ut ifra de erfaringene en har hatt. Denne forståelsen av læring tar oss videre til en annen teori Vygotsky er kjent for, den proksimale utviklingssone, eller «Zone of Proximal Development» (ZPD) på engelsk.

Vygotsky definerte den proksimale utviklingssone som:

«the distance between the actual development level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers»
(Vygotsky & Cole, 1978, s. 86)

ZPD ser på en persons ferdighetsnivå akkurat nå, og personens mulighet til å lære nye ferdigheter og informasjon med hjelp av en annen med forkunnskaper om temaet. Det må bemerkes at ZPD tar utgangspunkt i en sosial situasjon med en lærer, foreleser eller medelev som kan mer enn seg selv, ut ifra definisjonen. Det er elevenes samarbeid med lærer som vil bli satt fokus på i denne oppgaven.



Figur 4 – Illustrasjon av den proksimale utviklingssone,
<https://www.simplyphsychology.org/vygotsky.html>

Distansen mellom de ulike nivåene er en metafor som viser hvordan læring utvikles progressivt i forhold til egne forkunnskaper. En person har ingen til liten mulighet for å lære uten at grunnkunnskapene er til stede. Et eksempel på dette er elever som arbeider med et problem i matematikktimen: Eleven kan bruke lang tid på å løse oppgaven alene, eller redusere tiden brukt med hjelp fra lærer eller en medelev som enten er kunnskapsrik eller har sett et lignende problem før. På denne måten kan eleven både løse oppgaven raskere, og få kunnskap om lignende problemer til senere arbeid. Læringsprosessen går først fra sosial til individuell. Vygotsky har tilsynelatende ikke presentert de ulike typer hjelp en voksenperson kan gi et barn, men dette har blitt undersøkt ytterligere av andre (Chi et al., 2001, s. 490).

Det er også relevant å undersøke hvordan en lærer kan sørge for at elevene arbeider innenfor sin egen proksimale utviklingssone. Det kan derfor være hensiktsmessig å gjennomføre periodevise spørsmål- og svar-evalueringer for å sjekke om elevene henger med (Edwards, 2005, s. 832). En utviklingssone kan derfor bare bli undersøkt og forklart med en sosial kontekst, siden arbeidsmiljøet inkluderer andre. Arbeidsmiljøet kan inkludere en lærer, foreleser, medelev eller en kombinasjon av alle, som kan mer om temaet enn seg selv. Nyere forskning som baserer seg på Vygotsky sine teorier vil dermed ha et pedagogisk grunnlag i samsvar med et sosiokulturelt perspektiv. Kort fortalt virker det som å lære er best med andre på samme eller høyere nivå.

Edwards (2005, s. 833) forklarer at det pedagogiske grunnlaget for å ta i bruk sosiale arbeidsmetoder kommer fra Vygotsky sine teorier, eller mer nøyaktig tilhengere av Vygotsky. Den proksimale utviklingssone er sterkt knyttet til samarbeid, hvor alle inkludert jobber med å løse samme oppgave eller problem. Selv om denne type læringsmetode baserer seg på læring i et sosialt miljø, er det fortsatt den individuelle læring som er i fokus. Den individuelle vil tilegne seg kunnskap samtidig som den bidrar med gruppens aktiviteter, hvor kunnskapen er både faglig og sosial. Edwards (2005, s. 833) beskriver dette som et bytte fra Vygotsky sitt syn på selvidentitet til et nyere syn som baserer seg på intersubjektivitet, altså enighet mellom personer som deler en erfaring eller opplevelse. Denne endringen krever derfor en ny metodologi i skolen som endrer fokuset fra beskrivelse til forklaring. Elevene skal ikke bare beskrive en hendelse eller noe de har lært eller opplevd, men de skal forklare i stedet. Forklare hva som har skjedd, eller forklare en idé eller et konsept.

2.8 Det helhetlige teoretiske rammeverket

Det har hittil blitt diskutert en rekke teorier som dekker noen aspekter av det store, vanskelige begrepet som kalles læring, i en matematisk sammenheng. Teoriene har sine ulikheter, men det er også en del som er felles. Det skal i dette underkapittelet bli gjort et forsøk på å oppsummere viktige aspekter ved læring i lys av disse teoriene. Disse aspektene går på:

- (1) Elevene må arbeide i sin egen sone av forståelse, og progressivt utvide denne sonen. Å la elever arbeide med oppgaver eller problemer som er for vanskelig er lite hensiktsmessig for deres utvikling av forståelse.
- (2) Læring skjer best i en sosial kontekst, hvor elevene arbeider i samarbeid med en veileder. Elever bruker andre for å legge til ny kunnskap til den gamle, og utvide kunnskapslandskapet sitt på denne måten.
- (3) Selv om læring skjer i en sosial setting, er det den individuelle faglige og sosiale utvikling som er viktig. Eleven må vise sin kompetanse på forskjellige måter og klare å formidle kunnskap på en god måte i ulike situasjoner. Disse situasjonene kan være individuelle eller i grupper, skriftlige eller muntlige.
- (4) Elever lærer på forskjellige måter, ikke bare enten auditivt, visuelt eller fysisk. For mange er praktisk arbeid i kombinasjon med undervisning nyttig. Elevene må få hjelp med arbeidet sitt, men også prøve ut ting selv og lære av egne feil. Variasjon i skolehverdagen er viktig.

(5) Elevene må ikke bare lære, men oppleve. De må bli utfordret i virkelighetsnære kontekster og bli gitt muligheten til å utforske på eget nivå. Elevene må være kritisk til eget arbeid og tankegang. Samtidig skal de klare å overføre tilegnet kunnskap til andre felt, både i skolen og hverdagslivet.

Nå som relevant teori har blitt diskutert og oppsummert, vil metoden for studien og en analyse av innsamlet data med utgangspunkt i teorien bli gjennomgått. I metodekapittelet vil hele studiens prosess bli lagt fram, med innsamling og oppbevaring av data, og hvordan datasettet har blitt analysert.

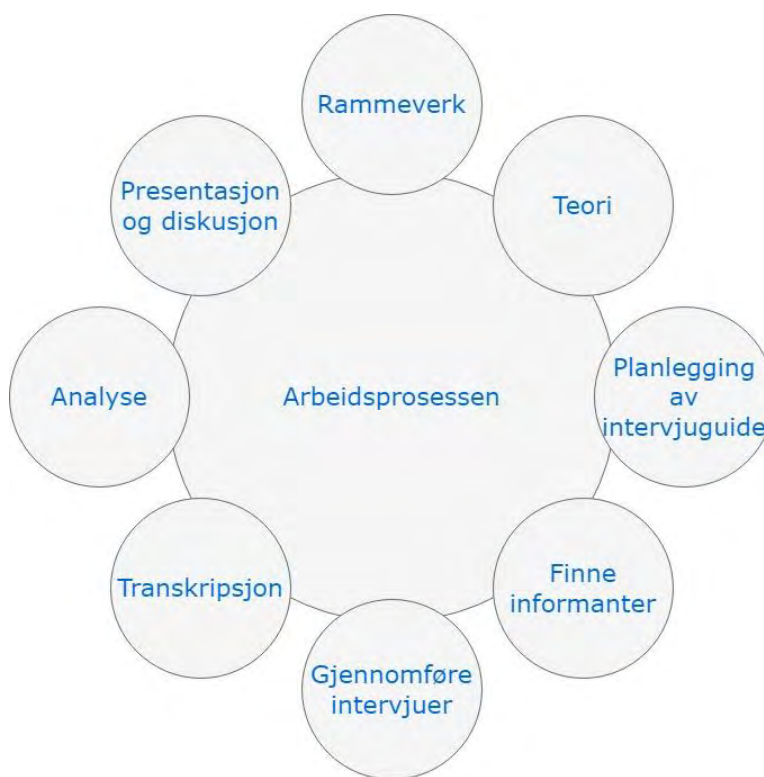
3. Metode

3.1 Kort oversikt over studien

Formålet med denne oppgaven er å finne de gjennomgående temaene i læreres refleksjoner og tanker om innføringen av programmering i skolen, samt de erfaringene lærerne selv har gjort. Derfor er det hensiktsmessig å ta i bruk den kvalitative metoden intervju for å finne ut av dette. Jeg har valgt å ta i bruk individuelle intervjuer med lydopptak. Det er ifølge Christoffersen og Johannessen (2012, s. 84) umulig å huske alt som blir fortalt i et intervju. Det er derfor naturlig å ta lydopptak av intervjuene for å bedre arbeidet med dataene i ettertid. En annen mulighet er å gjennomføre gruppeintervjuer, eller kalt fokusgruppeintervjuer, hvor vanligvis fire til åtte personer har en samtale, ledet av en moderator (Davidsson Bremborg, 2021, s. 313). Selv om det kan være nyttig å føre en samtale mellom flere personer, er det noen utfordringer som kommer med denne intervjumetoden. I motsetning til intervjuer som er en til en, hvor informanten svarer direkte på hva som blir spurt, kan informantene i et gruppeintervju bli utfordret av de andre deltakende og miste fokus (Davidsson Bremborg, 2021, s. 313). Dessuten er transkripsjonsprosessen mye mer utfordrende i en gruppesituasjon enn i en individuell, siden det er flere personer å ta hensyn til samtidig.

Prosjektet er meldt til Norsk Samfunnsvitenskapelig Datatjeneste (NSD), hvor jeg har fått tillatelse til å bruke lydopptak under intervjuene mine. Jeg har benyttet meg av UiO sine nettsider og appen «Diktafon» for å trygt ta opptak og lagre opptakene av intervjuene. Informantene har gitt skriftlig samtykke til intervjuene, og intervjuene er anonymisert. Det er ikke skrevet noe sted hvilke skoler jeg utførte intervjuene på, eller navnene til informantene. Som informantene har blitt opplyst om, vil opptak og transkripsjoner bli slettet når prosjektet er slutt.

Figur 5 (under til høyre) viser en oversikt over viktige elementer i studien. Først ble det utarbeidet en midlertidig problemstilling og forskningsspørsmål ut ifra personlige interesser og mål for studien. Etter ble det gjort et valg av rammeverk med teori om læring, matematikk og programmering. Så ble det planlagt en intervjuguide etter problemstilling og forskningsspørsmålene. Videre ble ulike matematikklærere som passer rammeverket kontaktet med informasjon om intervjuprosessen. Omsider ble intervjuene gjennomført og fulgt opp av transkripsjonsarbeid og analyse kort tid etter. Til slutt ble de felles temaene presentert og diskutert. Det må nevnes at det ikke er en klar kronologisk rekkefølge i arbeidsprosessen, siden studien har blitt revidert som en helhet fra starten. Sirkel-formasjonen i figur 5 skal derfor ikke representere en lineær, kronologisk prosess.



Figur 5 - Grov oversikt over arbeidsprosessen i studien

3.2 Vitenskapelig forankring

Brottveit (2018, s. 25-31) tar for seg de tre hovedretningene for vitenskapsteori: positivisme, hermeneutikk og poststrukturalisme. Av disse tre er det hermeneutikk som skal bli diskutert og tatt i bruk i denne oppgaven. Alle forskningsmetoder, ikke bare tekststudier, tar utgangspunkt i en hermeneutisk refleksjon (Gilhus, 2021, s. 314). Dette går på hva slags virkelighet- og vitenskapssyn en har, og hvordan en velger å se på virkeligheten som blir opplevd. Vitenskapssyn kan defineres som «hvordan vi forstår, forklarer eller nærmer oss den sosiale virkeligheten (Brottveit, 2018, s. 17). Hele poenget med forskning er å hjelpe med å forstå virkeligheten. Vitenskap kan i tur beskrives som den kunnskapen som er utviklet gjennom forskningsbaserte metoder (Brottveit, 2018, s. 17). Denne studien tar utgangspunkt i en hermeneutisk vitenskapstradisjon, der beskrivelser av informantenes livsverden har en

fenomenologisk metode. I denne delen av oppgaven vil begrepene hermeneutikk og fenomenologi bli gjort rede for.

3.2.1 Hermeneutikk

Forskningsprosessen i denne oppgaven handler i bunn og grunn om hvordan en fortolker informantenes tanker og uttrykk, og derfor er et hermeneutisk utgangspunkt i høyest grad hensiktsmessig. Hermeneutikk var egentlig en vitenskapsfilosofisk tradisjon som var knyttet til å analysere og tolke tekster (Befring, 2015, s. 20; Gilhus, 2021, s. 275-276; Thagaard, 2009, s. 39). I moderne tid er denne tradisjonen brukt til å fortolke alle menneskelige uttrykk (Brottveit, 2018, s. 29). Videre beskriver Befring (2015, s. 20) hermeneutikk som en «subjektivt fortolkende prosess», som vil gi en økende forståelse av en tekst. Forskeren tar med seg alle erfaringer og kunnskap i en situasjon, og ser på en situasjon bundet av disse erfaringene. I en hermeneutisk tilnærming vil forskerens tolkninger påvirke sin forståelse, slik at tolkningene i stor grad er subjektive. En kan si at forskeren blir begrenset av egen horisont for forståelse (Brottveit, 2018, s. 34). En forståelseshorisont kan beskrives som de erfaringer og forestillinger en forsker har, vil i tur danne en subjektiv forståelse av den nye informasjonen (Brottveit, 2018, s. 35). Dette er en annen grunn til at oppgavens funn ikke kan generaliseres, siden forskerens subjektive utgangspunkt kan være annerledes enn noen andres. Som forsker vil ny informasjon være relevant eller interessant i situasjonens kontekst, og det er opp til forskeren om informasjonen er relevant. Både hvordan forskeren fortolker de enkelte deler, men også helheten, vil dermed bli subjektivt (Brottveit, 2018, s. 35-36). Oppgavens teoretiske rammeverk kan knyttes sterkt til fortolkningen av det kvalitative datamaterialet, samtidig som datamaterialet kan påvirke forståelsen av teorien (Thagaard, 2009, s. 219). På denne måten vil oppfattelsen av oppgavens problemstilling, teori og innsamlet datamateriale endre seg gjennom hele forskningsprosessen.

Denne oppgaven tar som nevnt tidligere utgangspunkt i de ulike informantenes uttrykk og meninger innenfor et gitt tema. Studien ønsker å gå dypere i de utsagnene og tankene lærerne har enn det som hadde vært vanlig i en hverdagslig kontekst. For å få noe ut av de utsagnene informantene har, må de i tur tolkes. Hvordan en tolker beskrives av Thagaard (2009, s. 39) som å tilegne et utsagn større mening. Transkripsjonene vil bli lest igjennom flere ganger, tematisert og sortert, og de ulike temaene vil bli satt i kontekst med hele datamaterialet. Denne gjentakende prosessen med å gå igjennom det samme datamaterialet flere ganger kan

bli representert ved en sirkel, mer bestemt den hermeneutiske sirkel. Denne hermeneutiske sirkelen har som formål å forstå noe som har mening, ved å forstå enkeltdelene i kontekst av helheten og motsatt (Gilje, 2019, s. 74-75). Forståelsen er fortsatt subjektiv, og blir sett i lys av alle de erfaringene forskeren innehar. I denne oppgavens tilfelle innebærer disse erfaringene de dokumenter som har blitt lest, forskerens opplevelser i skolen og egne erfaringer med programmering som både student og lærervikar.

3.2.2 Fenomenologi

Fenomenologi er en vitenskapelig tilnærming og går på menneskers erfaringer. Tilnærmingen har som mål å finne ut av essensen av et fenomen ved å utforske perspektivet til de som har opplevd det (Neubauer et al., 2019, s. 91). I fenomenologisk forskning er det ikke bare hva som har skjedd som er viktig, men hvordan det skjedde (Teherani et al., 2015, s. 669).

Fenomenologi innebærer å forske på aktørenes egne perspektiver med mål om å forstå sosiale fenomener (Kvale & Brinkmann, 2015, s. 45). Det er ulike fenomenologiske variasjoner etter hvilket utgangspunkt i filosofien en tar utgangspunkt i, og etter hva en skal finne (Neubauer et al., 2019, s. 93). Siden det er en rekke ulike filosofier en forsker kan ta utgangspunkt i, er det en rekke ulike fenomenologiske tradisjoner forskeren kan velge fra (Neubauer et al., 2019, s. 93). Eksempler på disse er ontologien, som ser på en persons erfaringer gjennom personens livsverden, i tillegg til epistemologien, hvor den observerende er en del av verdenen og dermed ikke nøytral (Embree et al., 1997, s. 259). Begrepet livsverden blir brukt i den hermeneutiske fenomenologien, som stammer fra arbeidet til Martin Heidegger (Neubauer et al., 2019, s. 94). Med livsverden menes den individuelle syn på virkeligheten, og at denne virkeligheten blir påvirket av selve verdenen den lever i (Lopez & Willis, 2004, s. 729). Fenomenologi setter søkelys på de beskrivelsene personer har om et fenomen (Thagaard, 2009, s. 38). Dette vitenskapsteoretiske synet ble videreutviklet til å ta for seg hvordan mennesker opplever verdenen, kalt ontologi (Brottveit, 2018, s. 23). Brottveit (2018, s. 23) beskriver dette som menneskers evne til å kommunisere om opplevelsen av verden, i kontekst av menneskets egen livsverden.

Det er ikke fenomenologisk analyse som vil bli tatt i bruk i denne oppgaven, selv om oppgavens problemstilling tar utgangspunkt i andres tanker. Derimot vil den tematiske analysen bli tatt i bruk, hvor enkelte verdier fra fenomenologien blir sett på som viktige. Disse

verdiene går på hvordan informantenes subjektive forståelse av enkelte handlinger blir oppfattet (Brottveit, 2018, s. 147), samtidig som forståelsen blir satt i datasettets kontekst.

3.3 Det kvalitative forskningsintervju

For å drive med forskning må det bestemmes hvilken tilnærming av datainnsamling som skal bli tatt i bruk. Det er i hovedsak to tilnærminger for metode; kvantitativ og kvalitativ. Ved kvantitativ metode skal det gjøres et forsøk på å måle hyppigheten av et fenomen for å kunne tallfeste den (Postholm & Jacobsen, 2011, s. 41-42). Slike undersøkelser er gode på å hvordan ting er og utvikler seg, men mindre gode på å si hvorfor eller hvordan (Postholm & Jacobsen, 2018, s. 121-122). Et eksempel på denne type metode er spørreskjema, hvor store mengder data av et sett med spørsmål blir samlet. Siden denne metoden for forskning ikke er relevant for denne oppgaven, vil den ikke bli tatt i bruk. Derimot er kvalitativ metode en god tilnærming for innsamling av data ut ifra oppgavens problemstilling. Intervju som metode har sine begrensninger, men fungerer godt til å undersøke erfaringer og tanker som en person har om en hendelse eller situasjon (Brinkmann & Tanggaard, 2012, s. 20-23).

3.4 Datainnhenting og oppbevaring og utvalg

Det har blitt tatt opptak av tre individuelle intervjuer, hvor opptakene har blitt manuelt transkribert ord for ord. Grunnen til en så nøye transkripsjon er å få med alle lyder som kan beskrive intervjusituasjonen, og dermed gi forskeren bedre innsikt i arbeidet med data (Tjora, 2021, s. 103). Intervjuene ble gjennomført i perioden januar og februar i 2022 på to forskjellige skoler i Viken-området. Lydopptak av intervjuene ble midlertidig lagret lokalt på forskerens datamaskin, og transkripsjonene ble oppbevart på privat OneDrive, hvor det bare er forskeren og veileder som har tilgang til både lydopptakene og transkripsjonene. Både OneDrive og enheten med tilgang til forskerens OneDrive var gjennom hele arbeidsprosessen beskyttet med passord.

3.4.1 Forarbeid og intervjuguide

Informantene for prosjektet har blitt valgt ut ifra oppgavens problemstilling. Siden problemstillingen har et fokus på lærere som går på ungdomsskolen, er det lærere på det nivået som er relevant. Lærere som har utdanning i matematikk og undervisningserfaring i programmering har vært krav for deltakelse, hvor også lærere med utdanning i programmering er ønskelig, men ikke nødvendig. Informantene har blitt valgt ut ifra disse

kravene, og er fra Viken-området. På grunn av pandemien var det hensiktsmessig for forskeren å ikke dra langt unna eget bosted.

Christoffersen og Johannessen (2012) skriver om homogene målgrupper, som går på at individene i gruppen er relativt like hverandre etter visse kriterier. Hvis en gruppe er homogen, trenger forskere færre informanter enn når målgruppen er heterogen, altså når informantene er forskjellige fra hverandre etter visse kriterier (Christoffersen & Johannessen, 2012). Utvalget av informanter er derfor homogen, siden alle informantene er ungdomsskolelærere med omtrent like kriterier for å delta på intervjuene. Ut ifra problemstillingen er det derfor ønskelig å forske på en homogen målgruppe, siden alle informantene skal intervjues med de samme spørsmålene.

For å få mest ut tiden på intervjuene, ble det utarbeidet en intervjuguide (vedlegg 2) før intervjuene ble gjennomført. Intervjuguiden hadde semistruktur, som betyr at intervjuet ikke må følge spørsmålene i en bestemt rekkefølge. Formålet med semistrukturerte intervjuer er å gi informantene mulighet til å bytte fritt mellom spørsmålenes temaer og rekkefølge (Christoffersen & Johannessen, 2012). Grunnen til dette er å åpne mulighetene for at informantene kan snakke om ting som interesserer dem når de vil, uten at en fast struktur på intervjuguiden skal begrense intervjuet. Dette åpner også muligheten for at informantene kan fortelle om egne erfaringer og tanker når som helst i intervjuet, som er ønskelig med tanke på oppgavens problemstilling. I tillegg er det en prioritet at informantene har det behagelig i en intervjusituasjon, og ikke føler seg presset til å snakke om noen undertemaer framfor andre. Det er tross alt informantenes ærlige tanker og refleksjoner som er relevant for denne oppgaven.

3.4.2 Utvalg av informanter

For å sikre kvaliteten i det empiriske arbeidet ble informanter valgt ut ifra hva som passer problemstillingen, slik at innsamlet data er reliabel (Thagaard, 2009, s. 190, 200). Hvis oppgaven omtaler alle informanter, betyr det at alle de tre informantene som bidro til denne oppgaven er inkludert. Hvis det står «de fleste informantene», betyr det at to av informantene er involvert i hva enn som blir diskutert. Alle informanter i denne studien er lærere fra Viken-områder, og er representert fra to ulike skoler. Kriteriene for å bli inkludert i prosjektet var at læreren har hatt eller har matematikk som fag og undervisningserfaring i programmering.

Igjen må det gjøres klart at informantenes tanker og refleksjoner ikke skal fungere representativt på noen måte. Informantene er valgt ut etter visse kriterier, og dataene vil dermed ikke kunne generaliseres. Alle informanter er mannlige lærere forskeren kjenner til fra før, og det var de tre forskeren ville intervju fra studiens start. I stedet for å omtale informantene som Informant 1, 2 og 3, vil det bli tatt i bruk fiktive navn for å få bedre flyt i oppgaveteksten. I tabell 2 (under) vises en oversikt over de ulike informantenes fiktive navn, eller aliaser, i tillegg til deres erfaring i skolen og erfaring med programmering.

Informant	Alias	Erfaring i skolen	Erfaring med programmering i skolen
1	Jon	7 år, diverse fag i barneskolen, naturfag og matematikk på ungdomsskolen	5 år, Lego League, MicroBit, blokk-programmering
2	Mads	21 år, fag fra universitetet, PPU	7 år, Har vært lærer i programmering valgfag, Scratch, Python, C++, Blockly, Basic, Simula
3	Karl	8 år, 60 stp. i matte, engelsk og musikk + blanding av fag, GLSM	3 år, Scratch, Python

Tabell 2 - Demografisk oversikt av informanter

3.4.3 Gjennomførelsen av intervjuene

Christoffersen og Johannessen (2012) legger også frem noen retningslinjer for hvordan et intervju skal planlegges og gjennomføres. Studiens problemstilling omfatter personers tanker og opplevelser, og derfor er det viktig å følge enkelte retningslinjer for behandling av informasjon og data. En forsker må ha god innsikt og ha hensyn til de regler for atferd som er relevant for forskningen (Christoffersen & Johannessen, 2012). Med disse retningslinjene ble intervjuene gjennomført på omtrent samme måte: intervjuene startet med at informanten fikk takk for at de stilte opp, for så å bli forklart formålet med studien og deres rettigheter sammen med underskrift av samtykkeskjema (vedlegg 1). Det ble ekstra tydelig klargjort at informantene deltar som anonyme, hvor de har mulighet til å trekke seg når som helst. Intervjuene ble vel gjennomført, uten at jeg merket noe nervøsitet eller ubehag fra noen av informantene. Det var tydelig at alle informantene var interesserte i programmering og hadde

mye å komme med, samtidig som de ikke ville vike for langt ifra temaet for oppgaven min. Alle informantene ga gode og innsiktsfulle svar til de spørsmål jeg hadde å komme med, og det gav meg inntrykk at vi snakket om noe de syns er interessant. Etter hvert som intervjuene utfoldet seg var det enkelt å komme med oppfølgingsspørsmål hvis noe var uklart eller usikkert fra min side, hvor informantene med glede svarte på de spørsmål jeg hadde. Jeg avsluttet alle intervjuene med å spørre om de hadde noe å tilføye, noe ingen benyttet seg av.

3.5 Metode for analysen av datamaterialet

Etter at de tre individuelle intervjuene ble gjennomført og transkribert var det mye data som skulle kategoriseres. Siden dataene har blitt samlet spesifikt for denne studien, og temaene i intervjuene samsvarer med tema i denne oppgaven, er det hensiktsmessig å ta i bruk en induktiv analyse (Braun & Clarke, 2006). På denne måten kan problemstillingen utvikle seg, og endres etter behov for hva som kommer fram i intervjuene.

3.5.1 Tematisk analyse

Virginia Braun og Victoria Clarke (2006, s. 78) beskriver den tematiske analyse som den første metode for analyse en kvalitativ forsker burde lære seg. Dette fordi metoden er i seg selv ganske fleksibel, siden en forsker kan finne ulike temaer på flere forskjellige måter (Braun & Clarke, 2006, s. 83). Tematisk analyse er i seg selv ikke en metodologi, men er en metode for å forstå kvalitative data. Den tematiske analysen har som formål å identifisere og gjenkjenne mønstre, også kalt temaer, fra et datasett (Braun & Clarke, 2006, s. 79). Et tema er noe viktig i forhold til et forskningsspørsmål, og representerer et mønster som går igjen i et datasett (Braun & Clarke, 2006, s. 82). Den tematiske analysen skiller seg fra andre typer analyser (for eksempel fortolkende fenomenologisk analyse (IPA) og databasert teoriutvikling) med at tematisk analyse ikke er direkte koblet til et teoretisk rammeverk (Braun & Clarke, 2006, s. 81). De andre typene analyse ser på datasettene i nøye detalj, mens tematisk analyse går mer i dybden på de individuelle dataene og finner temaer mellom hele datasett (Braun & Clarke, 2006, s. 82). Tematisk analyse kan bli brukt til å undersøke deltakeres erfaringer, eller se på hvordan hendelser og meninger kan påvirke samfunnet (Braun & Clarke, 2006, s. 81). Siden undersøkelsen er induktiv, vil oppgavens problemstilling tilpasse seg etter hvilke temaer som blir identifisert i oppgaven. Temaene i denne oppgaven er valgt uten å se på temaer fra tidligere forskning, noe som beskrives som i tråd med den induktive tilnærming (Braun & Clarke, 2006, s. 84). På grunn av fleksibiliteten og

mulighetene som tematisk analyse gir, vil denne analysemetoden bli tatt i bruk, selv om enkelte deler av fenomenologien er relevante.

Den tematiske analysen blir beskrevet som fleksibel og «rett frem», men det er noen fallgruver forskeren må unngå (Braun & Clarke, 2006, s. 94):

Først er det å mislykkes med å i det hele tatt analysere. Med dette menes at tematisk analyse ikke er et sammendrag av ulike temaer med lite til ikke eksisterende analytisk narrativ. Det er heller ikke en samling av utdrag som bare blir parafrasert med analytiske kommentarer.

For det andre må forskeren unngå å bruke spørsmålenes tema som endelige presenterbare temaer. I dette tilfellet er det analytiske arbeidet minimalt eller ikke-eksisterende, og det kan dermed ikke bli sett på som en god tematisk analyse.

Den tredje fallgraven er å danne en analyse som ses på som «svak» eller «ikke overbevisende» (Braun & Clarke, 2006, s. 94-95). Her vil temaene ikke fungere sammen som representerende for helheten, det vil være mye overlapp og lite forståelige. For eksempel kan et tema bli presentert, men ha manglende eksempler fra datamaterialet. Selv om et tema virker interessant eller relevant, kan det ikke bli presentert som gjennomgående i hele datasettet.

En fjerde fallgruve vil gå på en manglende sammenheng mellom datasettet og de analytiske utsagnene som blir presentert (Braun & Clarke, 2006, s. 95). Hvis det som blir presentert ikke stemmer med hva som ble funnet, kan ikke analysen bli sett på som god. Det er forskerens oppgave å sørge for at datasettet og hva som blir presentert stemmer og gir mening. Det er også viktig å legge merke til en analyse som ikke tar for seg mangler, for det er ikke noe mønster i et datasett som er absolutt gjennomgående eller har en total mangel på motstridende utsagn. Hvis en analyse blir presentert som feilfri, er det grunn for mistanke. Den femte fallgraven beskrevet går på en manglende sammenheng mellom teori og de analytiske utsagn som blir presentert. Også en god sammenheng mellom forskningsspørsmålene og formen av tematisk analyse brukt er viktig.

Avslutningsvis understreker Braun og Clarke (2006, s. 95) viktigheten av å klargjøre de teoretiske antagelsene og hvordan og hvorfor disse ble gjort. Det spiller ingen rolle om analysen er god eller interessant, så lenge forskeren ikke beskriver hvilke teoretiske antagelser som har blitt gjort.

3.5.2 Fra rådata til koder

Det har blitt tatt i bruk fargekoder i transkripsjonene for å skille mellom ulike temaer som kom fram i intervjuene. Innsamlet data trenger å bli tolket i en kontekst for å gi mening, og dette vil bli gjort ved å inndeles datamaterialet for å bli analysert (Johannessen et al., 2016).

Det ble gjennomført til sammen 117.5 minutter intervjuetid, og produsert 44 sider transkripsjoner ut ifra intervjuene. Figur 6 (under) er et eksempel fra en transkripsjon hvor dette har blitt gjort. Spørsmål fra forskeren (S) er markert i rødt, og svar fra informanten (I) er markert i ulike farger etter tema. Figur 6 viser hvordan flere temaer kan forekomme i ett og samme svar.

S: Og hva legger du da i læringsmetodebegrepet?

I: Det er jo å... bli trygg på å... Det er jo nok en gang oppskriften, det å tørre å feile. Ja, det kan godt hende at det går skeis, men allikevel, prøv en gang til, a. Prøv å bytte ut noen variabler, bytt ut noen steder. Og så se. Det er egentlig hva hele livet er bygd opp på. Det er jo prøving og feiling. Sånn det er der sånn

Figur 6 - Eksempel fra transkripsjon

3.5.3 Induktiv tilnærming

Den induktive tilnærmingen for analyse blir beskrevet av Braun og Clarke (2006, s. 83) som «fra bunn og opp» (min oversettelse). Dette spiller på tilnærmingens evne til å sterkt koble sammen hvilke temaer som er identifisert med selve dataene. Den induktive tilnærming gir også mulighet til å forme problemstillingen og forskningsspørsmål i løpet av kodingsprosessen (Braun & Clarke, 2006, s. 84). Siden oppgavens problemstilling og forskningsspørsmål vil endre seg gjennom hele arbeidsprosessen, vil den induktive tilnærmingen passe best. Det har vært et fokus på å la informantene komme med egne refleksjoner og tanker, selv om det går utenfor den opprinnelige problemstillingen. Semi-struktur i intervjuguiden er et eksempel på dette. Den induktive tilnærmingen er sterkt knyttet til kvalitativ forskning, siden forskere ofte vil undersøke temaet uten å teste ut for mange ideer på en gang (Postholm & Jacobsen, 2018, s. 224). Induktiv analyse er da et forsøk på å organisere data uten å prøve å få dataene til å passe i et allerede dannet teoretisk rammeverk (Braun & Clarke, 2006, s. 83).

3.6 Etske betraktninger og studiens kvalitet

I arbeid med forskning er det en del etske betraktninger forskeren må ta hensyn til. To viktige begreper som trenger søkelys er reliabilitet og validitet, også kalt pålitelighet og gyldighet. Pålitelighet og gyldighet er begreper som rommer en rekke kriterier for å sikre studiens troverdighet (Postholm & Jacobsen, 2018, s. 211-213). Jeg vil ta i bruk begrepsparene reliabilitet og pålitelighet, og validitet og gyldighet som synonymmer i denne oppgaven.

3.6.1 Reliabilitet og validitet

Begrepene reliabilitet og validitet kommer opprinnelig fra naturvitenskapene, og må derfor beskrives i den kvalitative forskningens kontekst for å gi mening (Brottveit, 2018, s. 144). Disse begrepene beskriver en rekke punkter som hele tiden endrer seg etter undersøkelsen og undersøkelsens mål. Derfor beskrives det som umulig å ha en sjekkliste for vurdering av etske betraktninger i forskning. Betraktningene endrer seg etter hva som skal undersøkes, hvordan forskeren kommer til funnene og den vitenskapsteoretiske forankringen.

Begrepet reliabilitet brukt av Thagaard (2009, s. 33, 190), eller pålitelighet brukt av Postholm og Jacobsen (2018, s. 275), beskriver om forskning er utført på en god og pålitelig måte. Med dette menes om forskningen med samme metode kan bli utført av noen andre, og gi omtrent samme resultat. Det å identisk etterligne et studie er vanskelig (Postholm & Jacobsen, 2018, s. 129), noe som støttes av LeCompte og Goetz (1982, s. 55) sine tidligere funn. Konklusjonen gikk ut på å oppnå absolutt validitet og reliabilitet i en hvilken som helst forskningsmodell blir sett på som umulig (LeCompte & Goetz, 1982, s. 55). Dette kommer av at det er umulig å etterprøve en studie og få helt identiske resultater. Samtidig må forskningens prosess bli tydelig lagt synlig, slik at andre personer med interesse har mulighet til å reflektere over den (Postholm & Jacobsen, 2018, s. 129). På denne måten har jeg forsøkt å legge fram prosessen i min egen forskning, på en så oversiktlig måte som mulig. Jeg har også forsøkt å definere ulike begreper ut ifra relevant teori, samt gi min egen vinkling på disse viktige begrepene utover oppgaven. Bruken av lydopptak bidrar til å styrke påliteligheten av studien, hvor jeg har mulighet til å høre på opptakene flere ganger (Kvale & Brinkmann, 2015, s. 205).

Begrepet validitet brukt av Thagaard (2009, s. 190), eller gyldighet brukt av Postholm og Jacobsen (2018, s. 126), går på om forskeren måler det som faktisk skal bli målt, og om det som har blitt funnet er relevant for selve studien.

Bruken av lydopptak sikrer at analysen skjer ut ifra akkurat det som ble sagt i intervjuene (Kvale & Brinkmann, 2015, s. 205). Dataene er dermed ikke analysert ut ifra forskerens hukommelse eller notater fra intervjuene. Samtidig er det mulighet for misforståelser og feiltolkninger fra forskerens side, både fra informantens og andre personers perspektiv. For å videre sikre validitet for oppgaven er intervjuene gjennomført med et mål om ikke bare å få innsikt i informantenes tanker, men også vise andre lærere hva de kan forvente av å jobbe med programmering i skolen. Derfor er undersøkelsen både gyldig og relevant for andre lærere.

3.6.2 Informantens beste

Etiske utfordringer er til stede i alle forskningsgrener, og god bruk av etikk gjelder når forskeren skal unngå å påføre skade på mennesker (Orb et al., 2001, s. 93). Denne form for beskyttelse er avgjørende i kvalitative studier. Dalen (2011, s. 100-104) presenterer fem viktige krav for å opprettholde god og profesjonell forskning: (1) samtykke, (2) informere, (3) konfidensialitet, (4) beskytte barn og (5) hensyn til svakstilte grupper. Oppgavens problemstilling har ikke fokus på barn eller svakstilte, og det er dermed ikke hensiktsmessig å forholde meg til krav 4 og 5. Derfor er det kravene 1, 2 og 3 som er viktige, og i denne delen av oppgaven skal disse kravene bli presentert.

(1) Før intervjuene kunne bli gjennomført, ble det utarbeidet et informasjonsskriv ut ifra en gitt mal fra NSD. Informasjonsskrivet fungerer også som et skriv om samtykke, hvor informantene må skrive under for å delta på intervju. Informasjonsskrivet og intervjuguide inkludert i prosjektet ble godkjent av NSD i slutten av 2021.

Informasjonsskrivet ble utarbeidet med formål om å gi informasjon (2) om prosjektets tema, studiens mål, brukte metoder og informantens rettigheter. Dette ble gjort for å fjerne all usikkerhet ved studiens formål og hva informantene kan forvente av intervjuene.

Konfidensialitet (3) handler om at informasjonen til informantene blir behandlet på riktig måte. Informasjonen må ikke bli brukt på en måte som setter informantene i et negativt lys. Som nevnt tidligere er all informasjon om informantene anonymisert, og studien krever ikke informanter å oppgi sensitiv informasjon. Dette er gjort for at informantene skal føle trygghet i både intervjusituasjonen, og i ettertid med at hva de sier ikke kan lede tilbake til dem.

Informantene vil bli skrevet om med andre navn i presentasjonsdelen av oppgaven (kapittel 4), for å ytterligere sikre deres anonymitet. Som beskrevet i informasjonsskrivet (vedlegg 1)

vil taleopptak bli slettet så fort transkripsjonene er ferdige. Transkripsjonene vil også bli slettet når prosjektet er gjennomført og levert.

3.6.3 Metodekritikk

Dette underkapittelet tar for seg de svakheter, men også styrker, som er viktig å være bevisst på i dette forskningsprosjektet. Siden problemstillingens mål er å oppnå innsikt og dybde i informantenes tanker, var det naturlig å ta i bruk den kvalitative metode. Av samme grunn var det dermed ikke hensiktsmessig å ta i bruk kvantitativ metode, i form av spørreskjema eller lignende. Intervju ga også på muligheten til oppfølgingsspørsmål der og da hvis det var behov for begrunnelser eller forklaringer. Denne fleksibiliteten er unik i forhold til den kvantitative metode, som er mer fastlåst (Anderson, 2010, s. 2).

Selv om kvalitativ metode er et godt utgangspunkt for å utforske tankene til informantene, er det også enkelte svakheter ved kvalitativ forskning. Intervjuenes kvalitet avhenger i stor grad på intervjuerens erfaring (Anderson, 2010, s. 2). Forskere som mangler erfaring, vil stille manglende i forhold til en mer erfaren forsker og mulig få mindre ut av intervjuene. Det er dermed intervjuerens ansvar å lese om hvordan et intervju burde gjennomføres i forkant. Uerfaren som jeg er, med bare en håndfull kvalitative intervjuer under beltet, ble det derfor min oppgave å lese meg opp på intervjumetode. Selv om forberedelser ble gjort, og intervjuene ble tilsynelatende gjennomført på en god måte, kan det diskuteres at mer erfaring kunne forbedre intervjuenes kvalitet. I tillegg er oppgavens tema noe jeg er veldig interessert i. På grunn av dette kan kvaliteten også påvirkes av forskerens fordommer eller ensidighet (Anderson, 2010, s. 3).

Patton (2015, s. 685) trekker frem fire kriterier for å sikre et studies kvalitet, som innebærer troverdighet, overførbarhet, pålitelighet og bekreftbarhet.

Troverdighet innebærer at informantenes tanker og meninger kommer fram på en måte som stemmer med intervjuene. Dataen innhentet skal altså ikke bli transformert og omgjort etter som det passer forskeren.

Overførbarhet handler om studien kan generaliseres. Michael Curtin og Ellie Fossey (2007, s. 92) beskriver at det er forskerens ansvar å gi leseren nok informasjon til å vurdere en sammenligning av andre studier, som også støttes av Patton (2015). I utgangspunktet hevder

ikke kvalitative studier å være overførbare, men for å oppnå troverdighet og pålitelighet burde kvalitativ forskning være overførbart (Curtin & Fossey, 2007).

Det tredje kriteriet, pålitelighet (reliabilitet), har allerede blitt klargjort i kapittel 3.6.1 etter Thagaard (2009) og Postholm og Jacobsen (2018) sine definisjoner. Gibbs (2018, s. xiii) legger fram en rekke punkter for å sikre reliabiliteten av kvalitative prosedyrer: sjekke transkripsjoner for åpenbare feil, sørge for at kodingen av data har definisjoner som ikke overlapper og en kryss-sjekk av koder. Det er nyttig hvis kryss-sjekken skjer i forhold til andre forskeres koder, slik at en overenstemmelse kan skje. Hvis ulike personer får lite overlapp i koder, må dermed forskeren gå tilbake til arbeidet og sjekke kodene sine. Bekreftbarhet spiller på å vise at forskerens oppfattelse av data ikke bare er et produkt av forskerens fantasi (Patton, 2015, s. 685). Det ferdige produktet av data skal være en sammenkobling av funn og oppdagelser til dataene selv på en merkbar måte. Dette kan være å sjekke om funnene gir mening i forhold til datamaterialet, sørge for at den som blir intervjuet er kjent med materialet eller få en annen til å gjennomgå arbeidsprosessen med datamaterialet (Lincoln & Guba, 1985, s. 321-323).

En utfordring forskeren kan møte er avveiningen om hvor mye informasjon informantene skal få. Ønsket i dette studien var å informere deltakerne på en god og oversiktlig måte, så samtykkeskjemaet (vedlegg 1) ble sendt til informantene i forkant av intervjuenes start. I hvilken grad informantene leste igjennom samtykkeskjemaet på forhånd er uvisst. Rett før intervjuene fikk informantene en kort oppsummering av deres rettigheter som anonymitet, retten til å trekke seg fra prosjektet og oppbevaring og sletting av personopplysninger. I intervjusituasjonen ble informantene innledningsvis spurt om en rekke begreper relevant for prosjektet, men det ble ikke gjennomgått alle de begreper som er sentrale for oppgaven. Dette var for å unngå at informantene følte seg i en vurderingssituasjon. Det var viktig å ha faglighet og enighet om visse begreper i intervjuene, men en full faglig gjennomgang av alle viktige begreper ble utelatt.

3.7 Refleksivitet

Som nevnt tidligere er all kvalitativ forskning satt i en viss kontekst (Brottveit, 2018; Thagaard, 2009). Forskingen skjer på et spesifikt tidspunkt mellom to eller flere mennesker, og det er forskerens oppgave å fremstille forskningssituasjonen i en passende kontekst (Dodgson, 2019, s. 220). Denne måten for forskeren å formidle sin posisjon i forhold til

forskningssituasjonen i sin helhet kalles refleksivitet. Refleksivitet er et forsøk på å gi klarhet til leseren om forskerens egen bevisstgjøring. Brottveit (2018, s. 125) beskriver begrepet som en bevisstgjøringsprosess, hvor forskeren må vurdere sin egen posisjon i forhold til forskningsprosessen. Teh og Lek (2018, s. 520) beskriver refleksivitet som en prosess som skal skape gjentagende bevissthet og refleksjon knyttet til mennesker likheter og ulikheter. Hvis forskeren beskriver de kontekstuelle, kryssende forholdene mellom informantene og seg selv, som rase, sosioøkonomisk status, alder, kulturell bakgrunn, vil det ikke bare øke troverdigheten til funnene, men gi leseren en bedre forståelse for arbeidet (Berger, 2015; Dodgson, 2019, s. 220). Curtin og Fossey (2007, s. 92-93) oppsummerer refleksivitet som forskerens anerkjennelse for at forskeren er en aktiv deltager gjennom hele forskningsprosessen, og på denne måten har gir en påvirkning. Derfor vil hele oppgaven være preget av begrunnelser og forklaringer for hvorfor enkelte ting ble gjennomført på måten de ble.

Det har blitt nevnt tidligere at jeg kjente informantene fra før. Dette er både fordi jeg ville intervju personer som hadde arbeidsplass i Viken-området og egnet seg til oppgavens problemstilling. Selv om dette er risiko for å påvirke dataene innsamlet, er det hensiktsmessig å intervju personer som passer til problemstillingen. En utfordring med dette er at forskeren kan bli sett på som noe annet enn en forsker, som kollega eller bekjent.

4. Presentasjon og analyse av funn

Oppgavens problemstilling og forskningsspørsmål ble utarbeidet i starten av arbeidsprosessen, og disse spørsmålene ble utgangspunktet for oppgaven. Det var klart at spørsmålene kunne endre seg etter hvordan oppgaven utfoldet seg; både i intervjuene, gjennomgang av teori og skriveprosessen. Forskerens rolle i denne delen av oppgaven er å gjennomgå analyseprosessen og presentere resultater av analysen.

For å analysere innsamlet data vil den tematiske analysen av Braun og Clarke (2006) bli tatt i bruk. Hele den analytiske prosessen har blitt oppsummert av Eggebø (2019), som vil bli presentert og tatt i bruk i denne oppgaven.

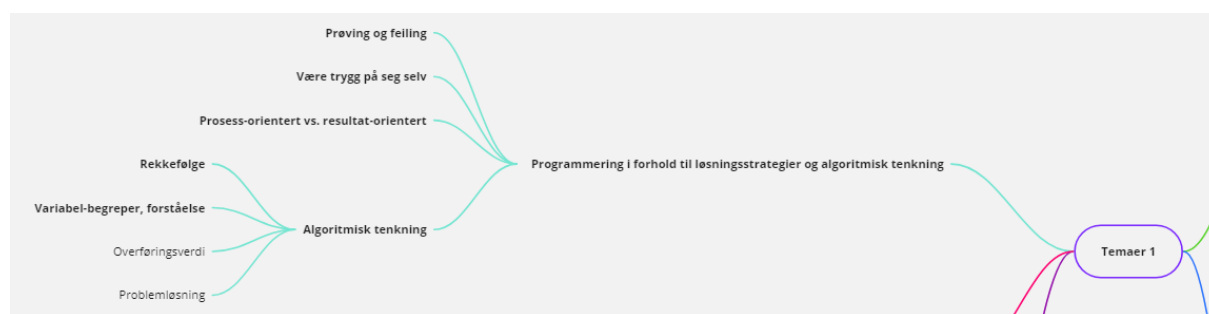
4.1 Analyseprosessen

Hvordan analyseprosessen utfolder seg blir beskrevet gjennom seks steg av Eggebø (2019). I denne delen av oppgaven vil disse stegene først bli presentert (skrevet i kursiv). Deretter vil arbeidsprosessen med stegene i oppgaven bli fortløpende gjennomgått.

1. Bli kjent med datamaterialet – alt av datamateriale skal bli transkribert ordrett. Dette gir mulighet til å lese igjennom hele datasettet flere ganger og bli kjent godt med det.

Datamaterialet skal bli undersøkt grundig, hvor det skal bli identifisert mønstre på tvers av transkripsjonene.

Først og fremst ble alle transkripsjonene lest igjennom flere ganger, hvor alle spørsmål fra forskeren som virket relevant, ble markert ved første gjennomlesning. Grunnen til å gå gjennom alt flere ganger var for å se på alle intervjuene som en helhet, og få en god oversikt over datamaterialet. Overordnede temaer ble markert med en annen farge enn spørsmålene fra forskeren, og de store temaene var nå funnet. Naturlig samsvarte temaene i stor grad med forskningsspørsmålenes temaer. Nedenfor (Figur 7) er et utdrag fra det første tankekartet laget i Miro.



Figur 7 - Utdrag av tankekart fra første analyse (Miro)

I første steg av analysen ble en rekke temaer identifisert: Tolkninger av begreper, tanker om innføring av programmering i skolen, elevers oppfatning av programmering, programmering i forhold til løsningsstrategier og algoritmisk tenkning og egen kompetanse etter ny læreplan.

2. Lag de første kodene – det skal først bli laget en liste over de temaene som kan være relevant for datamaterialet. Temaene skal bli kodet ved å identifisere ulike aspekter som virker interessante å analysere, som kan gjøres ved å notere i margen eller fargekode. Det samme utdraget fra intervjuet kan inneholde flere temaer, og dermed tilhøre flere farger. Til slutt vil det være en rekke farger som representerer en liste med de ulike temaene.

Etter første oversikt av temaer ble det gjort en ny runde med analyse, for å finne de temaene som var felles på tvers av oppgavens forskningsspørsmål. Hvilke temaer som er gjennomgående i de tre intervjuene er hva som er interessant og ønskes fremhevet. Dette endte med en ny rekke temaer, i tillegg til lærernes tolkninger av begreper. Begrepstolkning er ikke et tema i seg selv som et produkt av analyseprosessen, men dette er viktig å nevne for å forstå informantenes forståelse av begrepene gjennom hele oppgaven.

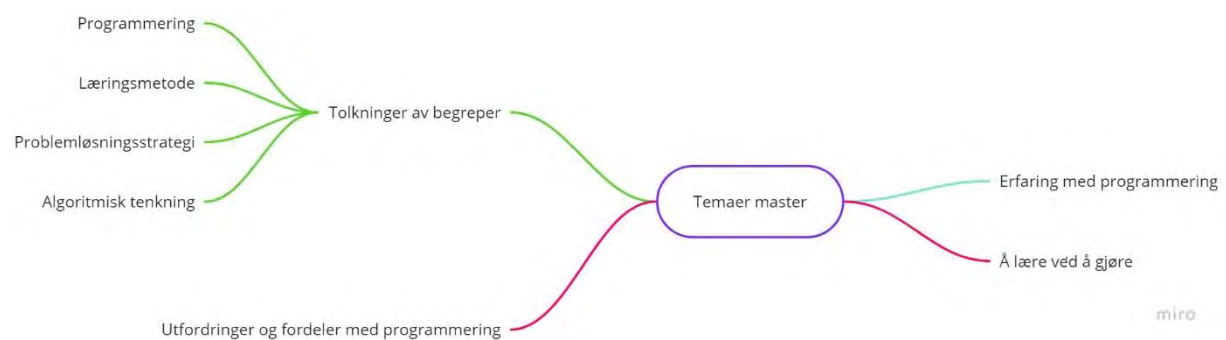
Disse temaene, inkludert begrepsforståelse, ble identifisert som å lære ved å gjøre, prøve og feile og utfordringer med programmering. Temaene vil bli næyere gjennomgått og diskutert i kapittel 5.

3. Let etter tema – finne overordnede temaer og samle kodene under disse temaene. Videre finne viktige temaer og undertemaer, og alle utdrag som passer til disse. Forskeren skal nå ha en god oversikt over hele datamaterialet med en liste av temaer og utdrag.

Etter å ha fargekodet alle de tre transkripsjonene på nytt, ble det tatt ut en rekke utdrag etter tema som beskrevet i forrige punkt. Dette ga en god oversikt over de gjennomgående temaene som var felles i flere av spørsmålene fra intervjuene. Utdragene ble sortert etter hvilken informant som sa hva etter tema, som gir en bedre oversikt over helheten av intervjuene.

4. Gå kritisk gjennom tema – sjekke om enkelte temaer kan sammenslås eller om temaer må brytes i flere undertemaer. Sjekke om de ulike sitatene passer til det valgte temaet, eller om de passer andre steder. Gjør en kritisk vurdering om sluttproduktet gir en god oversikt over hele datamaterialet. Dette punktet repeteres til forskeren er fornøyd med temalisten som er dannet.

Kodene ble gjennomgått og revidert flere ganger etter som overlapp og mangler i temaene ble funnet. Etter flere gjennomganger ble følgende temaer identifisert: Begrepsforståelse, erfaringer med programmering, å lære ved å gjøre og aspekter ved programmering i skolen. Disse temaene vil bli næyere gjennomgått i kommende delkapitler. Figur 8 (under) viser en oversikt over de ulike temaene.



Figur 8 - Utdrag fra tankekart fra endelig analyse

5. Definer og gi temanavn – hvert tema trenger et navn som godt beskriver hva temaet handler om. Temaene må kryssjekkes med den overordnede historien til datamaterialet, og sjekke om det gir mening i kontekst. Ved slutten av dette punktet skal en god skildring av de ulike temaene være klart, samt presise navn som gir leseren et innblikk i hva de handler om. Under er en oversikt (tabell 3) som viser de endelige temaene, hva temaene vil bli kalt i oppgaven og en kort beskrivelse til hvert tema.

Tema	Temanavn	Skildring
Tolkninger av begreper	Lærernes begrepsforståelse	Her blir de ulike begrepene som er viktige for oppgaven gjennomgått og oppsummert fra lærernes perspektiv.
Erfaring med programmering	Læreres erfaringer med programmering	Dette temaet inneholder læreres personlige erfaringer med programmering både på skolen og privat, i tillegg til erfaringer lærernes elever har gjort.
Å lære ved å gjøre	«Du skjønner det ikke før du har prøvd»	Det var gjennomgående i alle intervjuene at å tørre å prøve og ikke være redd for feil er viktig i prosessen med å lære seg programmering.
Utfordringer og fordeler med programmering	Aspekter ved programmering i skolen	Det kom fram flere aspekter (positive og negative) ved programmering i skolen, både opplæring av lærere og de fordeler og utfordringer lærere og elever kan møte.

Tabell 3 - Oversikt over temaer og beskrivelser

6. Skriv rapporten – nå som temaene er utarbeidet og navngitt, skal rapporten bli skrevet. Treffende sitater som gir en god representasjon over det tilhørende tema skal bli tatt ut og brukt i rapporten. Rapporten skal ha en analytisk narrativ, og ha en historie om datamaterialet som gir god oversikt. Til slutt kommer et argument som passer til forskningsspørsmålet.

Det virker som om det er en felles forståelse om at programmering i skolen er utfordrende for mange, både lærere og elever. Noen lærere har et syn på programmering som lite treffende for mange elever. Programmering oppleves som både lite interessant i skolen, og lite nyttig i hverdagen, for enkelte elever. Det må også nevnes at temanavn og innhold til temaene ble gjennomgått og forbedret etter hvert som kapittel 4 og 5 ble utarbeidet. Funnene vil nærmere bli presentert og analysert i dette kapittelet, og så diskutert i kapittel 5.

4.2 Lærernes begrepsforståelse

Det kan argumenteres som hensiktsmessig å gi en oversikt over informantenes forståelse av de ulike begrepene som er relevant for studien. I denne delen av oppgaven vil dermed informantenes syn på fire begreper; programmering, læringsmetode, problemløsningsstrategi og algoritmisk tenkning, bli kort presentert. Definisjonene presentert i denne delen av oppgaven er hva informantene viste av forståelse i intervjusituasjonene, og er ikke nødvendigvis offisielle eller tatt direkte fra teorien. For å tydelig vise hva som er direkte sitater, vil sitatene fra transkripsjonene oftest bli presentert i kursiv, med innrykk og innenfor anførselstegn i denne delen av oppgaven. Kortere sitater kan også forekomme som en del av avsnittet.

4.2.1 Programmering

Alle informantene beskrev programmering som en fremgangsmåte for hvordan en oppgave kan bli løst. I spørsmål om hva Jon legger i begrepet programmering, svarte han:

«Det kan være så enkelt som at når du skal knyte skolissa, så må du tenke hele tiden: første steget er alltid å ta tak i lissene. Og så er det bare å tre den ene lissa rundt den andre, altså du tenker jo programmering hele tida. Det er bare det atter, du er ikke obs på det, kanskje.»

På denne måten forklarer Jon at alt en gjør kan bli sett på som programmering, i hvert fall logikken som ligger bak programmering. For å utføre noe på en god måte, trenger prosessen å bli gjennomført med visse trinn i en bestemt rekkefølge. Denne måten å utføre en prosess på, kommer av å ha utført prosessen flere ganger. En person som aldri har knyttet en skolisse før vil ikke gjenkjenne trinnene, og enda viktigere: rekkefølgen av trinnene, for å få gjennomført knyttingen på en god måte.

Mads forsto begrepet på forskjellige plan:

«På et nivå er det at man gir instruksjoner til en datamaskin, som kan kjøre et program. Litt videre betydning så er det jo alle handlinger som har med en slags algoritme, en steg for steg-løsning, dekomponere et problem, og forklare litt hvordan systemet henger sammen.»

Denne forklaringen er veldig lik Jon sin, som knytter logikken bak programmering med hva enn som skjer i hverdagslivet. Karl beskriver programmering som alt fra å lage et script, et enkelt program, en løkke eller en animasjon. Karl ser ut til å forstå programmering som en sekvens av kode for å få noe til å skje. Begrepet programmering har oppsummert blitt assosiert med språk, rekkefølge og å gi instruksjoner til en datamaskin.

4.2.2 Læringsmetode

Mads beskriver læringsmetode på denne måten:

«... læringsmetoder for programmering vil være prøve og feile... og problemsøke, følge oppskrifter, modifisere oppskrifter...»

Mads nevnte i tillegg algoritmisk tenkning for å lære å programmere i oppstartsfasen. Mads forteller videre om prøving og feiling, problemsøking, og følge og modifisere en oppskrift som læringsmetoder for programmering. Karl nevner også metode og innfallsvinkler, for eksempel å ta i bruk tverrfaglighet for å lære elevene programmering. Det å prøve og feile virker sentralt for alle informantene når de beskriver læringsmetode i forhold til programmering. Jon forklarer:

«Det er jo nok en gang oppskriften, det å tørre å feile. Ja, det kan godt hende at det går skeis, men allikevel, prøv en gang til, a. Prøv å bytte ut noen variabler, bytt ut noen steder. Og så se. Det er egentlig hva hele livet er bygd opp på. Det er jo prøving og feiling.»

Jon beskriver å følge en oppskrift, samtidig som den programmerende må utforske, prøve og feile for å lære. Han kobler det direkte opp mot livet selv, at en må prøve og feile for å

komme gjennom livet. Karl beskriver videre læringsmetode som «ulike innfallsvinkler på det du skal lære». Han sammenligner det med språkopplæring, hvor en kan bruke læringsmetoder i matematikk som om det var et eget språk, på samme måte som norsk, engelsk eller programmering er et eget språk. Det å lære å prøve og feile fremstår som et viktig og gjennomgående tema i forhold til programmering, og vil bli nærmere gjennomgått i kapittel 4.4.

4.2.3 Problemløsningsstrategi

I spørsmål om hva informantene legger i begrepet problemløsningsstrategi, beskriver Karl:

«Å kunne ha såpass oversikt over de metodene du har, at du klarer å velge de metodene som er best tilpassa målet ditt...»

Mads forklarer det som en grunnopplæring, om hvordan ting i programmering fungerer, både i blokk og tekst. Blokkprogrammering er da programmering hvor en flytter og bytter på plassering og rekkefølge av ulike blokker med ulike funksjoner, som programmet Scratch tar i bruk. Mads beskriver:

«Så vil det ofte være at vi får et problem, eller en bestilling. Gjør noe som... lag et program som får til dette. Og da vil det i programmering være 20 måter å løse det på.»

Altså oppgaver en person møter i programmering kan være ukjente problemer, og det er mange ulike måter for hvordan en løser problemet. Selv om en elev får en utfordring de har sett før, er det flere ulike måter å løse det på. Også her knytter inn Jon det å prøve og feile, og ser på dette som noe felles for alle de viktige begrepene tatt i bruk i denne oppgaven.

4.2.4 Algoritmisk tenkning

Ved spørsmål om hva algoritmisk tenkning går ut på, beskriver Mads det som en verktøykasse. Verktøykassen inneholder summen av ulike ferdigheter som problemløsning, prosessforståelse og bruken av variabler, verdier og størrelser. Med prosessforståelse mener Mads det å ta i bruk, følge, justere og være kritisk til egen prosess. Både Jon og Karl nevnte rekkefølge i sammenheng med algoritmisk tenkning. Karl forklarer:

«... det er en slags, en regel eller rekkefølge du skal følge. For at du skal skjønne det. Og dem kan gjerne bruke en eller annen formel, da, som utgangspunkt for det.»

Jon forklarer noe lignende:

«... hva må inn først for at det skal få et sluttprodukt ... Det er jo rett og slett en rekkefølge. Det første jeg tenker er regnerekkefølge, da, at vi har en fast måte å løse dette her sånn på.»

Jon refererte også til en video på YouTube (Gaines, 2017). Videoen viser en far som tester sine barns ferdigheter innen algoritmisk tenkning. Barna får i oppdrag å skrive en instruksjonsmanual for hvordan en skal lage et smørbrød med peanøttsmør og syltetøy. Gaines tar instruksjonene veldig bokstavelig og utfyller manglende informasjon, som gjør hele situasjonen absurd og komisk. Dette gir et godt bilde på hvordan algoritmisk tenkning og programmering henger sammen. Logikken kan være riktig, men så lenge riktig rekkefølge og klare instruksjoner ikke er på plass, vil ikke systemet fungere som det skal. På samme måte trenger et program riktig syntaks for å forstå programmeringen som brukeren legger inn.

4.3 Læreres erfaringer med programmering

De ulike lærerne har i en viss grad forskjellige erfaringer med programmering, både privat og i skolesammenheng. Samtidig er det en del felles mellom disse lærerne når det gjelder opplevelsen av programmering. Informantene ble som nevnt tidligere valgt på bakgrunn av deres interesse og erfaring med programmering. Dermed ble det intervjuet lærere som har alt fra tre til 30 års erfaring med programmering i en privat sammenheng (se tabell 2 for demografisk oversikt). I tillegg har de en god del erfaring med programmering i skolesammenheng (3 til 7 år), til tross for at det er nytt i skolen. De ulike læreres erfaringer med programmering vil nå bli presentert hver for seg.

For noen år siden ble Jon introdusert til First Lego League, som blir beskrevet som en praktisk måte å introdusere teknologi og matematikk til barn mellom 4 og 16 år (First Lego League, 2022; Hjernekraft, 2021b). Jon forteller om sitt eget arbeid med programmering:

«Etter å ha sittet, og prøvd meg litt hjemme, så er det fine fremgangsmåter for hvordan du skal klare det. Også kommer det noen utfordringsoppgaver, for hvordan du skal klare det. Også kommer det noen utfordringsoppgaver mot slutten, for hvordan du kan videreutvikle det.»

Jon, som de andre informantene, har en over gjennomsnittlig interesse for programmering, og har derfor jobbet med dette på egen tid. Jon har jobbet med Python og blokkprogrammering, hovedsakelig sistnevnte, som er «hva de fleste benytter seg av». Han forteller at lærere står fritt i å velge hvilket språk dem vil, etter sitt eget kunnskapsnivå og interesse. Jon la fram programmering på en positiv måte i intervjuet, og han beskrev mange situasjoner der elever har mestret og hatt det gøy med programmering. Jon beskriver også Python som noe for de mer spesielt interesserte.

Karl forteller at han har «nesten utelukkende» brukt Scratch i vanlige fag. Han kjenner også til bruken av Python for de som har programmering som valgfag, selv om han ikke underviser i det selv. Karl fremstår som veldig positiv til innføringen av programmering i skolen, og beskriver det selv som «supert». Han forteller også at Python er tatt i bruk i det nye læreverket som er kjøpt inn på skolen.

Mads har 30 års erfaring med programmering og har tidligere vært lærer i programmering som valgfag. Han har erfaring med en rekke ulike programmeringsspråk, som C++, Blockly, Microbit, Simula og Basic. Mads er i første runde med programmering i skolen dette året, og beskriver en uttesting av hva som fungerer og ikke i klasserommet. Mads har gjort et forsøk på å hente opp tidligere temaer elevene har jobbet med og bruke dette i kontekst med programmering for å gi det mer relevans til elevene. Han ser også en sterk korrelasjon mellom ferdigheter i matematikk og programmering, selv om en ikke nødvendigvis trenger å være høyt presterende i matematikk for å mestre og like programmering. Elever som presterer lavt i matematikk kan være lei av den mer teoriorienterte regnedelen, og ser fram til mer praktiske metoder i programmering. Når det gjelder kollegers innstilling til programmering, opplever Mads at det er noe blandet. Noen lærere tar det godt i bruk i timen og får elevene engasjerte, mens andre vil helst ikke ha så mye med det å gjøre.

Informantene er som beskrevet erfarne i programmering, og noe erfarne med programmering i en skolesammenheng, siden programmering er såpass nytt i læreplanen. Som en av informantene ga uttrykk for, er ferdigheter og kunnskap innen programmering noe som gradvis utvikler seg etter hvert som det arbeides med. Informantene ga uttrykk for liten forskjell av kompetanse for programmering før og etter innføringen av det nye læreplanverket.

4.4 «Du skjønner det ikke før du har prøvd»

Dette underkapittelets tittel er et direkte sitat fra Jon sitt intervju. Sitatet representerer en forståelse for å lære ved å gjøre, som tidligere beskrevet i sammenheng med Dewey (1899, 1986, 2005). Informantene nevnte gjennomgående at det er å prøve og feile som er viktig som en strategi for å arbeide med programmering. Dette temaet og dets undertemaer svarer godt på oppgavens forskningsspørsmål 3: hvordan få elevene til å bli interessert i matematikk ved hjelp av programmering. Etter å ha fått spørsmålet om hva som legges i begrepet problemløsning, kom Jon med følgende tanker:

«Prøv. Går det gæli, så går det gæli, så start på nytt igjen! Altså vi skal ikke stresse med å komme i mål med det greiene her sånn, nå. Prøv og feil. Trenger du det, ja så skaff det, også ser vi hvordan det funker. Funker ikke det, så da prøver du en annen innfartsvinkel, for å se om det kan funke ... Det er prøving og feiling. Learning by doing.»

Her nevner Jon eksplisitt læringsteorien til Dewey, og kobler det direkte til prøving og feiling i programmering. Dette kan i tur kobles til algoritmisk tenkning. Jon fremstår tydelig om at prøving og feiling i klasserommet er lov, og det er forventet at elevene tester ut ulike metoder for å komme fram til en god løsning. Også Karl snakker om programmering som 90 prosent feilsøking, og at en må teste ut ulike ting og sjekke hva som fungerer. Karl forteller:

«For dem har ikke lyst til å starte et program og få det veldig tydelig at, oj! Det her funker ikke, det jeg har gjort. Men da blir dem enda grundigere i arbeidet sitt.»

Karl beskriver videre hvordan elever trenger å komme over en «kneik» før de føler mestring med programmering. Elevene må teste ut selv hva som fungerer, og virkelig prøve å forstå

hva de faktisk driver med. Det er ikke hva elevene får som resultat som er viktig, men prosessen for å komme til resultatet. Mads nevnte prøving og feiling tidlig i intervjuet, og koblet det med læringsmetode og programmering. Hvis elevene jobber med en oppgave, kan det være titalls ulike måter å løse denne oppgaven på. Elevene må da selv prøve ut ulike metoder og velge den metoden de forstår best.

Alle informantene forteller om elevenes behov for å lage et produkt og vise det fram til andre. Dette kan være en blokk med kode som de har jobbet med i en time, eller et helt spill som har vært et arbeidsprosjekt over en lengre periode. Karl forteller med engasjement om en elev som har gjennomført et prosjekt i Minecraft, et utforskende spill hvor en kan bygge nesten hva en vil med ulike blokker av byggematerialer. Karl beskriver eleven som stille og den type elev som aldri sier noe i timen. Så fort eleven fikk lov til å drive med noe som var interessant og gøy, ble det dannet et sluttprodukt som Karl selv beskriver som «kanskje det råeste opplegget jeg har sett».

4.5 Aspekter ved programmering i skolen

Dette kapittelet tar for seg de utfordringer og fordeler som innføringen av programmering har i skolen. Lærerne nevnte positive og negative sider både for lærere og elever, og i dette kapittelet skal de ulike aspektene bli presentert. Det må tydeliggjøres at lærernes tanker ikke er absolutte fakta, men deres egen opplevelse av programmering og sine elevers opplevelse av programmering.

4.5.1 Fordeler

I intervjuene var innstillingen til informantene i forhold til programmering i skolen generelt positiv. Det var noen tanker om hvordan opplæringen av lærere kan bli bedre, men det fremlegges ytterligere senere. Det kom fram at programmering er enkelt å iverksette i fag som matematikk og naturfag. Dette kan komme av den algoritmiske tenkningen som er gjennomgående i matematikk (Knuth, 1985), eller mulighetene for tverrfaglighet med naturfag.

Det kom frem en del gode aspekter med programmering i forhold til elevene og deres observerte opplevelser: De elevene som er interesserte i programmering er i stor grad selvdrevne i arbeidet sitt. Elevene vil gjerne utføre et godt arbeid og vise frem produktet sitt

til så mange som mulig. Dette driver elevene til å videreutvikle arbeidet sitt og jobbe utenfor oppgavens rammer på en kreativ måte. Denne muligheten for videreutvikling utover oppgaven er også en drivkraft for videre kompetanse innen programmering. Det er på denne måten lettere å lage gode relasjoner til disse elevene. Dette kommer av at elever og lærer kan jobbe og lære sammen, og arbeide i et miljø som er lærerikt og morsomt for alle. Elevene fremstår som positive til å vise fram hva de har gjort, samtidig som de vil forbedre arbeidet sitt etter medelevers tanker og lærerens veiledning. Det oppfattes at de fleste elever opplever programmering og logikken som interessant, gøy og virkelighetsnært.

4.5.2 Utfordringer

Det å få inn programmering i de fleste fagene i skolen, spesielt språkfag (norsk, engelsk, fremmedspråk) og praktiske fag (mat og helse, kunst og håndverk), blir sett på som en stor utfordring av alle informantene. Karl forklarer:

«... det er et problem for de lærerne som skal jobbe med programmering i kunst og håndverk, for eksempel. Eller musikk da. Det er de lærerne som ikke har den bakgrunnen. De som har musikk hos oss er ikke matematikere, for eksempel.»

Mangel på interesse og forkunnskaper innen programmering kan ses på som en bekymring i forhold til den nye læreplanen. Oppfattelsen er at de lærerne som ikke har erfaring med programmering, ikke er interesserte i å arbeide med det videre. Det blir sett på som noe påtvunget og unødvendig.

Programmering oppfattes å ikke fange interessen til alle elever. Jon forklarer:

«Nå skal det sies da, det er vanskelig å treffe samtlige på dette her sånn. Noen syns det er bønn i bøtta bare når du sier «programmering» og det er det døllest dem vet.»

Noen elever ser på programmering som unyttig og forstår ikke hvorfor det er hensiktsmessig å lære programmering i skolen. De ser ikke noen sammenheng mellom programmering, logikken og det virkelige liv som kommer etter skolen. Det er dermed ekstremt vanskelig å motivere og hjelpe elever som opplever programmering på denne måten.

En annen utfordring er læreres kompetanse innen programmering og muligheter til opplæring. Som nevnt er de fleste lærere mindre erfarne innen programmering, og mangler en grunnleggende opplæring. Selv de som har tidligere erfaring med programmering beskriver opplæringen som et dagskurs, hvor de bare har fått ressurser å utforske på egenhånd. Det virker som om mange lærere føler at programmering er et påtvunget ansvar de ikke forstår. Mads har også vært med på arbeidsgrupper som har kommet med utspill til læreplanen. Om dette beskriver han en blandet innstilling til innføring av programmering, også blant mattelærere. Mads forteller om en skepsis blant noen lærere om hvorfor programmering skal bli inkludert i læreplanen, og noen «reservasjoner» som arbeidsgruppen kom med. Ressurser blir gitt til ulike skoler som tar det i bruk på forskjellige måter, som i tur fører til et spredt ferdighetsgrunnlag blant både elever og lærere.

4.5.3 Hvordan undervise med programmering?

Gjennom samtalene kom det en del gode råd til hvordan undervisning med programmering kan gjennomføres. I opplæringen får lærerne et kort kurs i programmering, og en rekke ulike ressurser de kan ta i bruk. Lærerne kan ta i bruk ressursene enten for å lære seg programmering selv eller i en undervisningssammenheng.

Mads forteller at å starte programmering med blokker først er det beste. På denne måten får elevene inn logikken og den algoritmiske tenkningen, og de slipper å stoppe opp på grunn av syntaks i arbeidsprosessen. Arbeidet med blokkprogrammering er mer visuelt og håndterlig enn å arbeide med programmeringsspråk. Han forteller videre at det mest sannsynlig vil komme et «stort sprik» mellom ulike elever, siden noen mestrer programmering kjapt, og andre vil aldri få det helt til. Derfor er det lurt «å ha en plan for å hjelpe de som trenger hjelp videre, men også utfordringer for de som tar disse konseptene veldig fort». Samtidig tipser Mads om at lærere burde bruke god tid på å lære seg programmering, slik at de føler de har bedre kompetanse enn elevene. De kan ikke gjøre kurs på internett og regne med at det er nok.

Karl snakker lite om introduserende undervisning med programmering, men mer om hvordan elevene jobber med større prosjekter. Etter opplæringsfasen kan elevene stå fritt til å velge metode for hvordan de vil lage et produkt innen programmering, om det er en kode som skal utføre en oppgave, eller lage et spill som viser et konsept. På denne måten må læreren være åpen for ulike vurderingsformer for elevene, og la de bestemme hva de skal vise for vurdering. Jon forteller også om elever som arbeider omfattende med programmering, både på skolen, etter skoletid og på egen fritid. Han beskriver elever som møtes i grupper i helgene for å sammen arbeide med oppgaver de har fått på skolen. En viktig ting som alle informantene gjorde tydelig, var elevenes forståelse for at det er lov å ikke lykkes i arbeidet sitt med en gang. Dette temaet er direkte knyttet med temaet i kapittel 4.4, men trenger å bli tydeliggjort i en undervisningssammenheng. Elevene må vite at arbeid med programmering er en prøve- og feile-prosess, og at det er lov å gjøre feil, så lenge arbeidet fortsetter.

5. Diskusjon av funn

Studien gjør et forsøk på å forstå læreres tanker om hvordan arbeid med programmering kan gi elever interesse for og innsikt i matematikk. For å finne ut av dette ble tre lærere på ungdomsskolen intervjuet, og intervjuene ble kort tid etter transkribert. Transkripsjonene har blitt tematisk analysert og sortert med tilhørende utdrag. Siden det bare er tre lærere som har blitt intervjuet, er det å trekke generelle konklusjoner utfordrende. Likevel er det forskning som underbygger studiens funn. Denne delen av oppgaven skal ta for seg og diskutere disse funnene i lys av forskning.

5.1 Lærernes syn på de ulike begrepene i forhold til teorien

Det er viktig å se på lærernes forklaringer og definisjoner til de ulike begrepene, både i forhold til hverandre og i forhold til hva litteraturen sier. Hvordan lærerne ser på de ulike begrepene er utgangspunkt for de kommende temaene. I denne delen vil lærernes forståelse av disse begrepene bli diskutert nærmere.

Rich et al. (2019) beskriver definisjonen av programmering som varierende etter hvem en spør. Dette kom også frem i intervjuene, hvor en informant sammenlignet det med prosesser fra det virkelige liv, som å knyte en skolisette eller lage mat fra Toro. En annen informant koblet begrepet direkte til hvordan en gir instruksjoner til en datamaskin, og videre til alle handlinger som kan utføres steg for steg. Dette stemmer godt med hva Forsström og Kaufmann (2018) beskriver, som en prosess for problemløsning.

Med læringsmetode knyttet Mads og Karl direkte inn det å prøve og feile, sett i sammenheng med programmering. Det å følge en oppskrift, men også gå utenfor oppskriften og teste ut nye ting en ikke har sett før blir assosiert sterkt med læringsmetodebegrepet. Karl hadde et mer generelt syn på læringsmetode, og så på det i likhet med Pritchard (2017). I den sammenhengen går læringsmetode på å ha en oversikt over de metodene en besitter, og velge hvilken metode som passer best for situasjonen. Som Mads nevnte, kan dette ses på som en verktøykasse. Dette kobles da direkte til problemløsning, hvor Karl beskrev problemløsningsstrategi på akkurat denne måten. Det er tross alt ikke bare en måte å løse et problem på, og det er løseren sin evne til å velge riktig framgangsmåte som blir testet. I likhet med læringsmetode ble det å prøve og feile også nevnt i sammenheng med problemløsningsstrategier. Alle informantene ga uttrykk for at det å prøve og feile er essensen for læring i programmering, og at det ikke er noe galt i å gjøre feil. Å utforske og utprøve blir

også sett på som en læringsmetode i seg selv, som kan bli tatt i bruk av elevene for å lære hvordan programmering fungerer.

Informantenes forståelse av begrepet algoritmisk tenkning er nok det som viker mest fra hva litteraturen sier, men fortsatt er det ganske likt. Dette er forståelig, siden algoritmisk tenkning mulig er et mindre brukt begrep i forhold til de andre i en skolesammenheng. Wing (2011, s. 1-2) beskriver algoritmisk tenkning i sammenheng med problemløsning, der målet er å løse et problem ved å formulere det slik at løsningen kan bli utført av en datamaskin. Mads nevnte spesifikt problemløsning og beskrev godt dets aspekter, som passer godt med Wing (2011) sin definisjon. Jon og Karl blander inn rekkefølge og prosess, noe som kan kobles opp mot problemløsning og i tur algoritmisk tenkning. Oppfattelsen av informantene er at de forstår begrepene hver for seg, men at de fortsatt ser sammenhengen mellom begrepene og temaet for oppgaven, og mellom begrepene selv.

5.2 Implikasjoner

På hvilken måte er denne studien nyttig for lærere på ungdomstrinnet? Er studien også nyttig for andre personer i skolen? Hva var egentlig informantenes erfaringer med programmering i skolen, og hvilke utfordringer må lærere være bevisste på når programmering skal bli tatt i bruk i skolen? Denne delen av oppgaven skal diskutere og gi et forsøk på å besvare disse spørsmålene.

5.2.1 Læreres erfaringer og programmering i skolen

Informantene valgt ut i dette studien er lærere som allerede har en interesse for programmering, og har jobbet med programmering i mange år. På grunn av dette er det vanskelig å si nøyaktig hva lærere som en helhet mener om programmering i skolen, noe som heller ikke er hensikten med denne studien. Derimot kan det diskuteres hva som ble sagt av de tre informantene, og ta med denne innsikten i videre diskusjoner.

Programmering oppfattes som vanskelig å få benyttet i de fleste fag, spesielt språkfag og mer praktiske fag som musikk og kunst og håndverk. Særdeles er matematikk og naturfag hver for seg og tverrfaglig oppfattet som godt egnet til programmering, noe som kommer av at alle informantene har matematikk som fag. Bekymringene om at lærere innen språk og praktiske fag, i tillegg til de som ikke har erfaring med programmering, er i stor grad reelle.

Informantene ga noe uttrykk til at opplæringen med programmering var utfordrende. Både egen opplæring i programmering og opplæring av elevene. Utfordringene kom i stor grad av manglende læreropplæring fra institusjonen, hvor lærerne fikk et kort kurs i bruk av ressurser, og ble i stor grad overlatt til seg selv i å faktisk lære seg å programmere. Siden programmering har fått en viktig rolle i den nye læreplanen, burde opplæring og oppfølging av kompetanse innen programmering bli forbedret. Bare ett eller et par dagskurs med en enkel gjennomgang av tilgjengelige ressurser kan ikke beskrives som en god opplæring for noe som blir sett på som sentralt i matematikk i fagfornyelsen. Det er en rekke ressurser tilgjengelig på internett for å lære programmering, men ressurser med en pedagogisk grunnmur virker mangelfull og ettertraktet.

I denne studien kom det fram at for å lære og mestre programmering er det hensiktsmessig å ta i bruk en prøve- og feile-metodikk, slik at en kan utforske og teste ut hva som fungerer og ikke. Dette gjelder ikke bare for elevene, men også lærere og andre voksne som vil lære seg programmering. På denne måten kan det bli sett på som hensiktsmessig å gi lærere de ressursene som er tilgjengelig, og la de utforske selv. Hva som er pedagogisk riktig; å la lærere stå for egen opplæring i programmering selv eller gi de nøye opplæring over en lengre periode, er usikkert, og kan være verdt å undersøke ytterligere. Det som står klart, er i hvert fall informantenes misnøye og uttrykk av mangel på læreropplæring i programmering i en undervisningssammenheng. Selv om lærere får opplæring i å programmere, er det ikke sikkert at lærerne føler seg gode nok til å ta det med inn i undervisningen. En av grunnene til at programmering kan bli sett på noe som utfordrende, er tross alt lærernes tanker om egen kompetanse i programmering (Sanne et al., 2016, s. 73). Hvis en lærer opplever manglende kompetanse i programmeringsferdigheter og programmering i en pedagogisk sammenheng, er det forståelig at programmering virker lite hensiktsmessig å ta i bruk i undervisningen.

5.2.2 Metodiske implikasjoner

Som nevnt tidligere kan både matematikk og algoritmisk tenkning med programmering bli sett på som en menneskelig aktivitet, og det er derfor hensiktsmessig å se det i sterk sammenheng med programmeringsprosessen, som også oppfattes som en aktivitet. Etter intervjuene kom det frem en rekke utfordringer som kan komme av å undervise i programmering, både som en del av fagene i skolen som vi allerede kjenner til, og som et nytt valgfag.

Oppfattelsen er at programmering ikke er hverken interessant eller blir sett på som relevant for noen elever og lærere. Programmering er som nevnt noe som de fleste elever får til, men det er ikke mange som blir så interesserte at de opplever mestring og glede av arbeidet. Inntrykket fra informantene var at visse elevgrupper ble spesielt interessert i programmering i skolen, som også tok det som valgfag. På denne måten ble kompetansenivået deres mye høyere enn de som ikke tok programmering som valgfag. På denne måten vil skillet mellom kompetanse innen programmering variere stort i et gjennomsnittlig klasserom. Enkelte elever kan være så interesserte og selvdrevne at de arbeider med programmering i helgene og på fritiden. Ellers blir programmering sett på som noe kjedelig og lite relevant for dagliglivet, både av elever og lærere. Programmering oppfattes som noe fremmed og lite nyttig, til den grad at det ikke trenger å være en del av læreplanen. Enkelte elever (og lærere for øvrig) ser ingen sammenheng mellom den algoritmiske tenkningen og utfordringer som oppstår i hverdagslivet, og da blir programmering lite interessant. Hvis programmering blir sett på som noe en lærer i skolen og vil bli brukt bare i skolen, vil ikke elever ha interesse til å arbeide med programmering. Hvis elever i tillegg får et inntrykk om at lærere har et negativt syn på programmering, kan dette også påvirke elevenes eget syn på programmering. Derfor er det vanskelig å si om programmering har en direkte positiv innvirkning på alle elever, selv om det oppleves som interessant og nyttig for mange. Dette er ikke en kvantitativ studie, så hva «mange» betyr er opp til leserens oppfatning av informantenes meninger.

Det er kjent at arbeid med programmering har ofte blitt knyttet til geometri (Forsström & Kaufmann, 2018). En informant nevnte at elevenes forståelse for det ukjente og ligninger ble forbedret etter arbeidet med programmering. Dette kan komme av at elevene ser hvordan en ukjent variabel fungerer i praktiske situasjoner, for eksempel når de skal lage et spill eller en blokk med kode. Når elever jobber med det ukjente, kan det være i sammenheng med teoretiske oppgaver i oppgaveparadigmet (Skovsmose, 2003), slik at de ikke får utforsket hvordan en variabel fungerer i praksis. Læreren som en veileder i undervisningen ses på som viktig i programmering, siden elevene må selv prøve ut hva som fungerer og ikke. Elevene må starte utforskningen selv, og bygge seg opp til en forståelse for hva programmering egentlig går ut på og kan brukes til. Elevene starter å programmere noe fra starten av, og utvikler programmet sitt til et ferdig produkt til slutt. I hele arbeidsprosessen må elevene finne ut av hva som fungerer og ikke, og tilpasse seg de utfordringer de møter. Dette kan bli sett i

sammenheng med både Vygotsky (1997; 1978) sin teori om den proksimale utviklingssone og veiledning fra lærer, og Dewey (1899, 1986, 2005) sin læringsteori om å lære ved å gjøre.

5.2.3 Didaktiske implikasjoner

For å hensiktsmessig ta i bruk programmering i skolen, er det viktig å tydeliggjøre både de utfordringer som programmering har, men også hva slags utbytte som kan oppnås. Det har tidligere blitt forsket på hvordan programmering påvirker elevers kompetanse innen matematikk (Forsström & Kaufmann, 2018). Det ble funnet at oppgaver innen programmering hadde et fokus på geometri framfor andre temaer i matematikken. Dette er noe som ikke var planlagt i intervjuguiden, og informantene ga ingen indikasjon på at dette stemmer fra deres erfaring. Derimot ble det nevnt i intervjuene at det er en sammenheng mellom elevenes kompetanse i matematikk og deres arbeid med programmering. De elevene som har gode kunnskaper innen matematikk og forstår konsepter innenfor algoritmisk tenkning har det lettere for å prestere godt i programmering. Det er derimot ikke alltid at dette stemmer, siden noen av informantene nevnte enkelte elevers interesse og dyktighet i programmering tross lavt prestasjonsnivå i matematikk. Det kom også fram at flere informanter så en sammenheng mellom programmering og elevenes forståelse for ukjente variabler. Elevene får selv testet ut hvordan programmeringsblokken fungerer, ved å bytte ut variabler og se hva som skjer, som også ble nevnt av informantene.

Nosrati og Wæge (2015) sine komponenter for en opplæring i matematikk preget av dybdelæring kan også kobles til arbeidet med programmering. Eleven må først forstå hva problemet er og prøve å sammenligne det med tidligere problemer. For å utføre en større oppgave må eleven makte å utføre en prosedyre som datamaskinen forstår, og ikke gi opp i arbeidet sitt. Samtidig må eleven ha gode muntlige ferdigheter til å ytre hvordan de står fast og hva slags hjelp som kreves i en veiledende situasjon. Alle disse komponentene må samtidig fungere i samhold med hverandre, og må ha blitt utviklet i en innledende periode med tett veiledning, der eleven lærer om grunnleggende programmering, og det å ikke gi opp med arbeidet sitt.

For å drive god undervisning med programmering, er det funnet enkelte tilpasninger lærere kan ha i bakhodet. Først må oppgavene som gis være passende til elevenes kompetansenivå. Siden det mest sannsynlig er et stort sprik i kompetansenivå mellom elevene i et

gjennomsnittlig klasserom, er den tilpassede opplæringen som beskrevet tidligere fortsatt reell. Elever som har programmering som valgfag og elever som helst ikke vil arbeide med programmering vil nok ha et naturlig sprik i programmeringsferdigheter. Elevene har dermed behov for oppgaver som er på deres nivå, som gir plass til selvstendighet. Derfor er det viktig at læreren har en veilederrolle framfor en lederrolle. Selvfølgelig skal læreren være en klar og ansvarlig leder i klasserommet. Samtidig må elevene få tid og mulighet til å utforske og prøve seg ut med dette som kalles programmering. For det andre kan det være hensiktsmessig å starte opplæringen i programmering med blokkprogrammering. Dette er fordi blokkprogrammering beskrives som mer visuelt og håndterlig, og passer derfor bedre til nybegynnere i programmering.

En andre tilpasning er lærerens bevisstgjøring om at det er mange ulike måter å løse en oppgave på. Dette gjelder også i programmering. En lærer vil nok i planleggingsarbeidet sitt oppdage flere løsninger på en enkelt oppgave. Samtidig vil mulig flere løsninger oppstå i elevenes eget arbeid. Dermed er det viktig at læreren har kompetanse til å se og forstå de ulike elevløsningene som mest sannsynlig vil komme. De ulike elevene vil ha ulik erfaring både med matematikk og programmering, som igjen vil føre til en rekke ulike løsningsstrategier til en og samme oppgave. Læreren trenger kompetanse for å forstå hvorfor en blokk kode fungerer, hvorfor koden ikke fungerer og eventuelt hva som kan forbedres. Denne måten å se mangler og forbedringer i et arbeid er likt med mange andre fag i skolen, for eksempel matematikk. En lærer må forstå hva en elev har regnet ut, og se hva som kan forbedres eller hva som er feil.

En tredje tilpasning går på lærerens oppgave til å holde arbeidet interessant og relevant. Denne tilpasningen kan knyttes sterkt til den første tilpasningen angående den tilpassede opplæringen alle elever har krav på. Uansett trenger denne tilpasningen et eget avsnitt, på grunn av viktigheten for å gjøre en arbeidsprosess virkelighetsnært og relevant. Alle lærere uttalte seg om elever som så mangelfull relevans til programmeringsarbeidet, og ga uttrykk til en nedsatt lyst blant elever til å arbeide med programmering på grunn av dette. Derfor er lærerens ansvar og arbeid med å gjøre programmeringsarbeidet relevant til dagliglivet i høyest grad viktig. Elevene må føle at arbeidet de driver med er nyttig, og at de kan få brukt det i livet etter skolen. Dette skal for øvrig være en tilpasning for alle fag i skolen.

En fjerde tilpasning, og som kanskje er den viktigste av alle, er læringsmiljøet i klasserommet. Elevene må vite og skjønne at det er lov å gjøre feil. For er det noe som vil komme i arbeidet med programmering, så er det feilmeldinger. Elevene må lære å holde ut, selv om programmet de arbeider med forteller dem at alt de gjør er feil. Elevene må finne ut av en enkelt ting som ikke stemmer, rette opp i dette og tålmodig arbeide med neste feil. Et arbeidsmiljø hvor elevene føler at det ikke er rom til feil vil ikke fungere generelt i skolen, og det kan argumenteres at dette gjelder særdeles i arbeidet med programmering.

En femte tilpasning, som bygger på funnene om å lære ved å gjøre, er hva som allerede er diskutert av Rich et al. (2019, s. 313). Læreren starter opplæringen med en veiledende funksjon. Elevene trenger å arbeide med oppgaver som forteller dem trinn for trinn hva de trenger å gjøre, slik at de ser hvordan de må jobbe for å utvikle et program. De trenger kort fortalt en oppgave med en oppskrift, og med lærer tilgjengelig for hjelp. Etter hvert som elevene øker sin kompetanse og forståelse for programmering, er det hensiktsmessig å gi elevene større, mer åpne prosjekter. Her kan de arbeide over en lengre periode med noe som interesserer dem, og som gir dem et sluttprodukt de kan utprøve.

5.2.4 Implikasjoner for lærerutdanningen og forskeren

Allerede etter 2. trinn skal norske elever ha arbeidet med algoritmisk tenkning i matematikk, og derfor trenger matematikklærere i alle trinn å vite hva dette går ut på. Elevene skal «kjenne igjen og beskrive repeterende enheter i mønstre og lage egne mønstre» og «lage og følge regler og trinnvise instruksjoner i lek og spill» (Utdanningsdirektoratet, 2020b). Disse kompetansemålene handler om mønstergjenkjenning og å følge trinnvise instruksjoner, som kan bli sett i sammenheng med definisjonen til algoritmisk tenkning. Studien har funnet en erfaring av mangel på opplæring i programmering for lærere, hvor de i stor grad blir overlatt til seg selv med noen få ressurser. Senter for IKT i utdanningen har utviklet et nettbasert kurs i programmering for lærere (MOOC), men nettsiden for påmelding av kurset er ikke lenger fungerende. Dette var et kurs dannet hovedsakelig for lærere som har programmering som valgfag, men det var også laget kurs for lærere generelt. Det fins noen kurs om informasjonsteknologi, men disse tilhører enkelte institusjoner i landet.

Utdanningsdirektoratet (2022) har gitt ut kompetansepakker for ulike temaer, om alt fra sosiale medier og omvendt undervisning til digital kompetanse og programmering. Det har

nylig kommet en kompetansepakke, «Støtte til arbeid med læreplanverket», med formål om å gi en innføring av det nye læreplanverket. Kompetansepakken inneholder tre hovedmoduler, som inneholder en rekke informasjon og aktiviteter som skal hjelpe lærere til å bli kjent med det nye læreplanverket. I en rapport av Rogde et al. (2020) kom det fram statistikk som viser hvor stor andel av respondenter som sier de tar i bruk kompetansepakken «innføring av nytt læreplanverk», som er ment til å støtte skoleledere og lærere til å forstå det nye læreplanverket. Oppsummert viste rapporten at en stor andel av respondentene oppgir kompetansepakken som nyttig, og varierende nytthet etter hvilke trinn respondentene jobber på. En større andel av skoleledere på barneskolen oppgir at kompetansepakken ga god støtte i forberedelsen av fagfornyelsen, enn hva skoleledere på ungdomsskolen oppga. Det kan være interessant å se på i hvilken grad de nyeste kompetansepakkene blir tatt i bruk, og hvor relevant det er for lærere.

Fra egen utdanning i matematikk har programmerings viktighet i skolen vært minimal. Gjennom fem år har det vært ett arbeidskrav i ett emne hvor programmering har blitt nevnt. Det må nevnes at dette arbeidskravet var ett av fire for emnet, og i grunn valgfritt, siden bare to av arbeidskravene skulle leveres inn til vurdering. Selv om den nye lærerutdanningen i stor grad komplementerer kravene som settes av den nye læreplanen, er det mye å innhente når det gjelder programmering, spesielt i matematikk. Nå som begrepet «programmering» blir spesifikt nevnt i læreplanen i matematikk, virker det merkelig å ikke gi lærerstudentene i matematikk en solid, obligatorisk grunnopplæring i programmering.

5.3 Refleksjoner av egne funn

For å besvare problemstillingen ble det intervjuet tre lærere på ungdomstrinnet som har erfaring med programmering i skolen. Funnene ble utarbeidet etter en tematisk analyse av transkripsjonene fra intervjuene. Samtidig som det ikke er meningen at funnene skal gjøres generaliserbare, vil funnene gi noe innsikt av programmerings bruk i skolen. I tillegg ble intervjuene gjennomført med en semistrukturert intervjuguide som hjelp, for å få informantene til å snakke om temaer de selv følte var interessant. Dette gjorde at informantene snakket om forskjellige temaer etter å ha blitt spurt det samme spørsmålet. Dette var ønskelig, samtidig som det gjorde analyseprosessen ganske utfordrende. Det var utfordrende å finne gjennomgående temaer i alle intervjuene som ikke sto i direkte sammenheng med spørsmålene i intervjuguiden. Tematisk analyse skal tross alt ikke bare

være en oppsummering av intervjuguiden, som blir sett på som en av fallgruvene for tematisk analyse (Braun & Clarke, 2006, s. 94). Heldigvis hadde informantene en god grunnforståelse av de ulike begrepene diskutert i denne oppgaven, og det var ikke behov for å klargjøre for informantene hva begrepene gikk ut på. Derfor kunne intervjuene bli gjennomført uten noen form for nøye gjennomgang av begreper. Informantene, som var spesifikt valgt etter hvilke fag de har i skolen, viste stor interesse for temaet og kom med gode utspill og historier. Alle informantene fortalte om egne opplevelser og hendelser som har skjedd på skolen med stort engasjement. Inntrykket var at informantene var både interessert i å fortelle om deres egne opplevelser med programmering, og hvordan de følte elevene opplevde programmering. Som nevnt tidligere ser informantene på opplæring i programmering for lærere som mangelfull. Hvis andre typer lærere hadde blitt valgt som informanter, kunne det ha kommet fram andre interessante aspekter ved programmering i skolen. Dette er en studie som tar utgangspunkt i informantenes uttalelser, og forskerens oppfatning av uttalelsene vil variere etter hvem forskeren er.

6. Konklusjon og oppsummering

Dette kapittelet tar for seg de viktigste funnene for studien, og hva disse funnene har å si for praksisen. Det vil bli presentert en kort gjennomgang av studiens begrensninger, og hva forskeren etterspør av videre forskning. Helt til slutt kommer et sammendrag som avslutning på oppgaven.

6.1 Funn og implikasjoner for praksisen

Det ble gjennomført tre individuelle intervjuer med ulike ungdomsskolelærere, hver av intervjuene med en varighet mellom 20 og 30 minutter. Dette datamaterialet ble nøye transkribert og analysert for å få en dypere forståelse om læreres opplevelser og erfaringer med programmering. Det er eksklusivt ungdomsskolelærere som var informanter i studien, selv om det kunne ha vært ønskelig å undersøke et bredere spekter av informanter. På bakgrunn av dette kan ikke studiens funn generaliseres. Dette var heller ikke formålet med studien. Formålet var derimot å bidra med innsikt i hva en kan gjøre for å ta i bruk programmering i skolen på en hensiktsmessig måte, mest i sammenheng med matematikk. Funnene viser at opplæringsfasen i programmering for lærere er noe mangelfullt, hvor lærere får korte kurs og en rekke ressurser til å ta med seg tilbake. Det kan også argumenteres at dette er den rette måten å sette lærerne i gang med programmering: gi de ressurser til å starte med, og la de lære seg programmering ved å prøve ut selv. Lærerne må også prøve og feile i arbeidet med programmering. Dette kan kobles til et av de overordnede temaene i funnene: Dewey sin læringsteori om å lære ved å gjøre. For å lære programmering må en prøve og feile, og progressivt bygge kunnskaper om algoritmisk tenkning og programmeringsspråket. I tillegg ble læreres forståelse av ulike begreper undersøkt og satt opp mot litteraturen. Dette viste at læreres forståelse av begrepene var god, og var i stor grad relevant med hva litteraturen sier.

Studien viser ytterligere skepsis til hvordan programmering fungerer i skolen, selv om informantene ble sett på som over gjennomsnittlig interessert i programmering. Lærernes erfaringer var at det er vanskelig å iverksette programmering på en god måte i de fleste fag, spesielt språkfag og praktiske fag. Dessuten er oppfatningen at mange lærere føler de blir påtvunget et ansvar de ikke forstår. De skjønner ikke hvorfor programmering og algoritmisk tenkning skal være sentralt i alle fag, også i matematikk. I tillegg er programmering noe som ikke kan bli sett i sammenheng med det virkelige liv for enkelte. Dette fører til et stort sprik i

kompetanse og interesse både på de ulike skolene, mellom lærere og mellom elever på samme skole. En grunn til sprik av kompetanse blant elevene er at de som er interesserte vil ta ekstra opplæring i programmering som valgfag, og få ekstra timer i uka som skal benyttes til å lære programmering. Samtidig vil de som ikke er interessert i programmering ikke få denne ekstra opplæringen, samtidig som de virker uinteresserte i de vanlige fagene i tillegg. Motsigende, som en av informantene nevnte, er den algoritmiske tenkningen noe en tar i bruk hele tiden i hverdagen. Ved opplæring av programmering blir elevene mer teknologiorienterte, og selve logikken kan brukes til alt en gjør (også lage Toro (metafor fra informanten Jon om logikken bak programmering og algoritmisk tenkning)).

6.2 Studiens begrensninger

En av begrensningene til denne studien er antallet og omfanget av informanter. Det ble håndplukket informanter som ble stilt spørsmål direkte koblet til problemstillingen. Alle informantene hadde mye felles, og det kan argumenteres for at variasjonen av informanter var minimal. Det kunne ha vært nyttig å ha forsket på et bredere spekter av matematikklærere, ikke bare de som har arbeidet med programmering i flere år. Det kunne også ha blitt tatt i bruk gruppeintervjuer, som kunne gitt informantene mulighet til å diskutere og spille på hverandres utsagn. Denne type intervjusituasjon kunne ha gitt en helt annen dynamikk i intervjusituasjonen, og andre temaer kunne ha oppstått. På denne måten kunne det ha blitt presentert flere varierte utsagn som ville gitt oppgaven mer bredde. Det var også en del utsagn fra informantene som kunne ha blitt presentert og diskutert, men på grunn av oppgavens tid- og plassbegrensninger lar ikke dette seg gjøre.

Oppgaven er begrenset til læreres meninger om sine egne erfaringer og elevers erfaringer. Det teoretiske rammeverket ser på hvordan en kan oppfatte andres ytringer og hvordan matematikk og programmering kan fungere sammen. Derimot tar ikke det teoretiske rammeverket for seg elevers læringsutbytte og interesse for programmering, og hvordan de selv vil arbeide med programmering i en skolesammenheng. Programmering i grunnskolen i Norge er nytt, og dermed er det mangelfull forskning på akkurat dette.

For å sikre studiens kvalitet og pålitelighet ble det utarbeidet en intervjuguide (vedlegg 2) med spørsmål, som allerede var plassert i sine egne kategorier. Disse kategoriene hadde utgangspunkt i et teoretisk rammeverk, som gir mulighet for å sammenligne studien med

foregående og kommende studier. På en annen side vil oppfattelsen av de ulike begrepene og informantenes utsagn påvirke hvordan funnene fremstår, som i tur vil påvirke problemstilling og oppgaven som en helhet. Hvis tolkningene av begreper og informantenes utsagn hadde vært annerledes, kan i tur denne studiens funn endres.

6.3 Videre forskning

Nå som programmering og algoritmisk tenkning står ordrett og har fått en sentral rolle i læreplanen for matematikk, er de ikke noe som kan unngås. Lærere i Norge må i fremtiden utvikle gode undervisningsopplegg som tar i bruk programmering på en hensiktsmessig måte. Det kan videre være interessant å undersøke:

- hvordan programmering oppleves fra elevers perspektiv (også en sammenligning av ulike elevgrupper)
- elevenes læringsutbytte av programmering, både innledende opplæring og arbeid med oppgaver alene, i par eller grupper med andre elever (med og uten lærerveiledning).
- på hvilke måter programmering og algoritmisk tenkning kan bli tatt i bruk på en hensiktsmessig måte i matematikk.
- i hvilken grad lærere og skoleledere tar i bruk de nye kompetansepakkene fra Utdanningsdirektoratet, og opplevelsen av hvor nyttig disse kompetansepakkene er for både elever, lærere og skoleledere.

6.4 Avsluttende sammendrag

I arbeid med denne oppgaven har jeg fått god innsikt i enkelte læreres erfaringer innen programmering, både som spesielt interessert privatperson og fremtidig profesjonell lærer. Studien har gitt innsikt i informantenes begrepsforståelse, utfordringer og fordeler i arbeid med programmering og matematikk, både innledningsvis opplæring og hva som kommer videre. Programmering blir sett på som noe fremmed og unødvendig for enkelte, samtidig som det omtales som noe interessant og relevant i skolen av andre. Programmering fremstår som hensiktsmessig å bli tatt i bruk hovedsakelig i skolefag som matematikk og naturfag. Samtidig oppleves det utfordrende å benytte programmering i språkfag og mer praktisk rettede fag, som musikk og kunst og håndverk. Håpet er at hensiktsmessig bruk av programmering og algoritmisk tenkning i skolen vil gi elever mulighet til å få interesse for matematikk.

Referanser

- Allazzam, M. (2015). Learning from Dewey and Vygotsky Perspective. 14.
- Anderson, C. (2010). Presenting and evaluating qualitative research. *American journal of pharmaceutical education*, 74(8). <https://www.ajpe.org/content/ajpe/74/8/141.full.pdf>
- Aranda, G. & Ferguson, J. P. (2018). Unplugged Programming: The future of teaching computational thinking? *Pedagogika*, 68(3).
<https://doi.org/10.14712/23362189.2018.859>
- Befring, E. (2015). *Forskningsmetoder i utdanningsvitenskap*. Cappelen Damm akademisk.
- Berger, R. (2015). Now I see it, now I don't: Researcher's position and reflexivity in qualitative research. *Qualitative research*, 15(2), 219-234.
<https://doi.org/10.1177/1468794112468475>
- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. *Report prepared for the Nordic@ BETT2018 Steering Group*, 397-400.
- Bohner, G. & Wänke, M. (2002). *Attitudes and Attitude Change* (1. utg.). East Sussex, UK: Psychology Press.
- Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 77-101. <https://doi.org/10.1191/1478088706qp063oa>
- Brinkmann, S. & Tanggaard, L. (2012). Introduksjon. I S. Brinkmann & L. Tanggaard (Red.), *Kvalitative metoder. Emipiri og teoriutvikling*, 2, 11-16.
- Britton, J. (1987). Vygotsky's contribution to pedagogical theory. *English in Education*, 21(3), 22-26.
- Brottveit, G. (2018). *Vitenskapsteori og kvalitative forskningsmetoder: om å arbeide forskningsrelatert*. Gyldendal akademisk.
- Chi, M. T., Siler, S. A., Jeong, H., Yamauchi, T. & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive science*, 25(4), 471-533.
- Christoffersen, L. & Johannessen, A. (2012). *Forskningsmetode for lærerutdanningene*. Abstrakt.
- Curtin, M. & Fossey, E. (2007). Appraising the trustworthiness of qualitative studies: Guidelines for occupational therapists. *Australian occupational therapy journal*, 54(2), 88-94. <https://doi.org/10.1111/j.1440-1630.2007.00661.x>
- Dale, E. L. (2001). *Om utdanning: klassiske tekster*. Gyldendal akademisk.

- Dalen, M. (2011). Intervju som forskningsmetode-en kvalitativ tilnærming (2 utg.): Universitetsforlaget.
- Davidsson Bremborg, A. (2021). Interviewing. I S. Engler & M. Stausberg (Red.), *The Routledge handbook of research methods in the study of religion*. Routledge.
- Dewey, J. (1899). The intellectualist criterion for truth. *The Collected Works of John Dewey, The Middle Works*, 50-75.
- Dewey, J. (1986). Experience and education. *The educational forum*, 50(3), 241-252.
<https://doi.org/10.1080/00131728609335764>
- Dewey, J. (2005). *Art as experience*. Penguin.
- Dodgson, J. E. (2019). Reflexivity in qualitative research. *Journal of Human Lactation*, 35(2), 220-222. <https://doi.org/10.1177/0890334419830990>
- Durrheim, K. (2006). Research design. *Research in practice: Applied methods for the social sciences*, 2, 33-59.
https://books.google.no/books?hl=en&lr=&id=7eKA1SMlOzAC&oi=fnd&pg=PA33&dq=what+is+research+design&ots=qKj2ToeH_V&sig=066xNZ8uJxW2ejUt8ExA_UtKtEI&redir_esc=y#v=onepage&q&f=false
- Edwards, J.-A. (2005). Exploratory talk in peer groups: Exploring the zone of proximal development. 831-840.
https://iris.uniupo.it/retrieve/handle/11579/122068/97258/CERME4_WG8.pdf#page=45
- Eggebo, H. (2019). Tematisk analyse – metodeartikkelen som løyser alt.
<http://helgaeggebo.no/tematisk-analyse-metodeartikkelen-som-loyser-alt>
- Embree, L. E., Behnke, E. A., Carr, D., Evans, J. C., Huertas-Jourda, J., Kockelmans, J. J., McKenna, W. R., Mickunas, A., Mohanty, J. N. & Seebom, T. M. (1997). *Encyclopedia of phenomenology*. Springer. <https://doi.org/10.1007/978-94-015-8881-2>
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning*, 32(6), 576-593. <https://doi.org/10.1111/jcal.12155>
- Feurzeig, W., Papert, S., Bloom, M., Grant, R. & Solomon, C. (1970). Programming-languages as a conceptual framework for teaching mathematics. *ACM SIGCUE Outlook*, 4(2), 13-17.
- First Lego League. (2022). *What is FIRST® LEGO® League?*
<https://www.firstlegoleague.org/about>

- Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10), 906.
- Forsström, S. E. & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. <https://doi.org/10.26803/ijlter.17.12.2>
- Freudenthal, H. (1968). Why to teach mathematics so as to be useful. *Educational studies in mathematics*, 3-8.
- Gaines, J. (2017). *Exact Instructions Challenge - THIS is why my kids hate me.* | Josh Darnit [[Video]]. Josh Darnit. https://www.youtube.com/watch?v=cDA3_5982h8&ab_channel=JoshDarnit
- Gibbs, G. R. (2018). *Analyzing Qualitative Data: Thematic coding and categorizing* (Bd. 6). Sage. <https://doi.org/10.4135/9781849208574>
- Gilhus, I. S. (2021). Hermeneutics. I *The Routledge handbook of research methods in the study of religion* (s. 314-323). Routledge.
- Gilje, N. (2019). Hermeneutikk som metode: Ein historisk introduksjon. *Oslo: Samlaget*.
- Gjøvik, Ø. & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten–tidsskrift for matematikkundervisning*, 30(3), 31-37.
- Grover, S. & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Guin, D., Ruthven, K. & Trouche, L. (2006). *The didactical challenge of symbolic calculators: turning a computational device into a mathematical instrument* (Bd. 36). Springer Science & Business Media.
- Hall, M. J. & Rieber, R. W. (2012). *The collected works of LS Vygotsky: Child psychology*. Springer Science & Business Media. <https://doi.org/10.1007/978-1-4615-5401-1>
- Hiebert, J. & Lefevre, P. (1986). Conceptual and procedural knowledge in mathematics: An introductory analysis. *Conceptual and procedural knowledge: The case of mathematics*, 2, 1-27.
- Hjernekraft. (2021a). *Hjernekraft sin hjemmeside*. <https://hjernekraft.org/>
- Hjernekraft. (2021b). *Hva er FIRST® LEGO® League Challenge?* <https://hjernekraft.org/challenge/om-first-lego-league>
- Holyoak, K. J. (1990). Problem solving. *Thinking: An invitation to cognitive science*, 3, 117-146.

- Humble, N., Mozelius, P. & Sällvin, L. (2019a). On the role of unplugged programming in K-12 education. *Proceedings of the 18th European Conference on E-Learning, ECEL*, 224-230. <http://miun.diva-portal.org/smash/record.jsf?dswid=8280&pid=diva2%3A1369126>
- Humble, N., Mozelius, P. & Sällvin, L. (2019b). Programmering i matte och teknik. <https://www.diva-portal.org/smash/get/diva2:1372308/FULLTEXT01.pdf>
- Husain, H., Kamal, N., Ibrahim, M. F., Huddin, A. B. & Alim, A. A. (2017). Engendering problem solving skills and mathematical knowledge via programming. *Journal of Engineering Science and Technology*, 12(12), 1-11. http://jestec.taylors.edu.my/Special%20Issue_PEKA_2017/PEKA%202017_01.pdf
- Imsen, G. (2005). *Ellevens verden: innføring i pedagogisk pedagogikk*. Universitetsforlaget.
- Imsen, G. (2020). *Lærereens verden: innføring i generell didaktikk*. Universitetsforlaget.
- Jarning, H. (2006). Dewey square: Lærerarbeid, didaktikk og improvisation. I: K. Steinsholt & H. Sommerro (red.). *Improvisasjon. Kunsten å sette seg selv på spill*, 215-236.
- Johannessen, A., Tufte, P. A. & Christoffersen, L. (2016). Introduksjon til samfunnsvitenskapelig metode (5. utg.) Abstrakt forlag.
- Kilpatrick, J., Swafford, J., Findell, B. & council, N. r. (2001). *Adding it up: Helping children learn mathematics* (Bd. 2101). Citeseer. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.487.9364&rep=rep1&type=pdf>
- Knuth, D. E. (1985). Algorithmic thinking and mathematical thinking. *The American Mathematical Monthly*, 92(3), 170-181.
- Kvale, S. & Brinkmann, S. (2015). Det kvalitative forskningsintervju. *Oslo: Gyldendal akademisk*, (3).
- Lambić, D. (2011). Presenting practical application of mathematics by the use of programming software with easily available visual components. *Teaching Mathematics and Its Applications: International Journal of the IMA*, 30(1), 10-18. <https://doi.org/10.1093/teamat/hrq014>
- Lave, J. (2009). The practice of learning. I K. Illeris (Red.), *Contemporary theories of learning* (s. 208-216). Routledge.
- LeCompte, M. D. & Goetz, J. P. (1982). Problems of reliability and validity in ethnographic research. *Review of educational research*, 52(1), 31-60.

- Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T. & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 25(6), 860-876. <https://doi.org/10.1007/s10956-016-9628-2>
- Levensen, K., Sørensen, B. H., Tosca, S., Ejsing-Duun, S. & Karoff, H. S. (2014). Research and Development Projects with ICT and students as learning designers in Primary Schools: A methodological challenge. *Designs for Learning: Expanding the Field*.
- Lie, J., Hauge, I. & Meaney, T. (2017). Computer programming in the lower secondary classroom: learning mathematics. *Italian Journal of Educational Technology*, 25(2), 27-35. <https://doi.org/10.17471/2499-4324/911>
- Lincoln, Y. S. & Guba, E. G. (1985). *Naturalistic inquiry*. Sage.
- Lithner, J. (2008). A research framework for creative and imitative reasoning. *Educational studies in mathematics*, 67(3), 255-276. <https://doi.org/10.1007/s10649-007-9104-2>
- Lopez, K. A. & Willis, D. G. (2004). Descriptive versus interpretive phenomenology: Their contributions to nursing knowledge. *Qualitative health research*, 14(5), 726-735. <https://doi.org/10.1177/1049732304263638>
- Ludvigsen, S., Elverhøi, P., Gundersen, E., Indregard, S., Ishaq, B., Kleven, K. & Øye, H. (2015). NOU 2015: 8 Fremtidens skole-Fornyelse av fag og kompetanser. <https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001>
- McLelland, J. A. & Dewey, J. (2008). *Applied Psychology: An Introduction to the Principles and Practice of Education*. Kessinger Legacy Reprints.
- Misfeldt, M. & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. *CERME 9-Ninth Congress of the European Society for Research in Mathematics Education*, 2524-2530. https://hal.archives-ouvertes.fr/CERME9/public/CERME9_NEW.pdf
- Moreno-León, J., Robles, G. & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum. *Journal of Information Technology Education: Research*, 15(1), 283-303. <https://www.informingscience.org/Publications/3521>
- Neubauer, B. E., Witkop, C. T. & Varpio, L. (2019). How phenomenology can help us learn from the experiences of others. *Perspectives on medical education*, 8(2), 90-97. <https://doi.org/10.1007/s40037-019-0509-2>
- Nickles, T. (1981). What is a problem that we may solve it? *Synthese*, 85-118.

- Nosrati, M. & Wæge, K. (2015). Dybdelæring i matematikk. *realfagsloyper.no*.
https://realfagsloyper.no/sites/default/files/2021-03/T3.P1.M1A-Dybdel%C3%A6ring%20i%20matematikk_2.pdf
- Nouri, J., Zhang, L., Mannila, L. & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17. <https://doi.org/10.1080/20004508.2019.1627844>
- Orb, A., Eisenhauer, L. & Wynaden, D. (2001). Ethics in qualitative research. *Journal of nursing scholarship*, 33(1), 93-96. <https://doi.org/10.1111/j.1547-5069.2001.00093.x>
- Papert, S. A. (2020). *Mindstorms: Children, computers, and powerful ideas*. Basic books.
- Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice*.
- Polya, G. (2004). *How to solve it: A new aspect of mathematical method*. Princeton university press.
- Postholm, M. B. & Jacobsen, D. I. (2011). *Læreren med forskerblikk: innføring i vitenskapelig metode for lærerstudenter*. Høyskoleforlaget.
- Postholm, M. B. & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm akademisk.
- Pritchard, A. (2017). *Ways of learning: Learning theories for the classroom*. Routledge.
<https://doi.org/10.4324/9781315460611>
- Rabardel, P. & Bourmaud, G. (2003). From computer to instrument system: a developmental perspective. *Interacting with computers*, 15(5), 665-691.
[https://doi.org/10.1016/S0953-5438\(03\)00058-4](https://doi.org/10.1016/S0953-5438(03)00058-4)
- Reitan, J. B. (2015). Learning-by-watching as concept and as a reason to choose professional higher design education. *DS 82: Proceedings of the 17th International Conference on Engineering and Product Design Education (E&PDE15), Great Expectations: Design Teaching, Research & Enterprise, Loughborough, UK, 03-04.09. 2015*, 512-517.
- Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E. & Belikov, O. M. (2019). Coding in K-8: International trends in teaching elementary/primary computing. *TechTrends*, 63(3), 311-329. <https://doi.org/10.1007/s11528-018-0295-4>
- Rogde, K., Federici, R. A., Vika, K. S., Bergene, A. C., Pedersen, C., Denisova, E. & Wollscheid, S. (2020). Spørsmål til Skole-Norge: Analyser og resultater fra Utdanningsdirektoratets spørreundersøkelse til skoler og skoleeiere våren 2020.

- Rolandsson, L. (2020). Programming with a purpose. *2020 IEEE Frontiers in Education Conference (FIE)*, 1-7. <https://doi.org/10.1109/FIE44824.2020.9273936>
- 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-august 2016.
- Sawyer, R. K. (2006). The new science of learning. *The Cambridge handbook of the learning sciences*, 1, 18.
- Schneider, W. & Artelt, C. (2010). Metacognition and mathematics education. *Zdm*, 42(2), 149-161. <https://doi.org/10.1007/s11858-010-0240-2>
- Schoenfeld, A. H. (2016). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics (Reprint). *Journal of Education*, 196(2), 1-38. <https://doi.org/10.1177/002205741619600202>
- Schwab, K. (2017). *The fourth industrial revolution*. Currency.
- Sevik, K. (2016). *Programmering i skolen*. Utdanningsdirektoratet. https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Simpol, N. S. H., Shahrill, M., Li, H. C. & Prahmana, R. C. I. (2017). Implementing thinking aloud pair and Pólya problem solving strategies in fractions. *Journal of Physics: Conference Series*, 943(1), 012013. <https://doi.org/10.1088/1742-6596/943/1/012013>
- Skemp, R. R. (1976). Relational understanding and instrumental understanding. *Mathematics teaching*, 77(1), 20-26.
- Skovsmose, O. (2003). Undersøgelseslandskaber. I *Kan det virkelig passe?* L&R Uddannelse.
- snl.no. (2018). tenkning. *Store Norske Leksikon*. <https://snl.no/tenkning>
- Språkrådet. (2007). *Spørsmål og svar*. <https://www.sprakradet.no/Vi-og-vart/Publikasjoner/Spraaknytt/Arkivet/Spraknytt-2007/Spraknytt-22007/Sporsmal-og-svar/>
- Stedøy, I. M. (2018). Utforskende matematikkundervisning. https://realfagsloyper.no/sites/default/files/2018-04/T2.P1.M3A%20Artikkel%20Utforskende%20undervisning_0.pdf
- Stigberg, H. & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures in Education*, 18(4), 483-496. <https://doi.org/10.1177/1478210319894785>

- Tedre, M. & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling international conference on computing education research*, 120-129. <https://doi.org/10.1145/2999541.2999542>
- Teh, Y. Y. & Lek, E. (2018). Culture and reflexivity: Systemic journeys with a British Chinese family. *Journal of Family Therapy*, 40(4), 520-536. <https://doi.org/10.1111/1467-6427.12205>
- Teherani, A., Martimianakis, T., Stenfors-Hayes, T., Wadhwa, A. & Varpio, L. (2015). Choosing a qualitative research approach. *Journal of graduate medical education*, 7(4), 669-670. <https://doi.org/10.4300/JGME-D-15-00414.1>
- Thagaard, T. (2009). *Systematikk og innlevelse: en innføring i kvalitativ metode* (Bd. 3). Fagbokforlaget Bergen.
- Thayer, H. (1988). Logic: The Theory of Inquiry John Dewey, the Later Works, 1925-1953, Vol. 12. <https://www.jstor.org/stable/40320234>
- Tjora, A. (2021). *Kvalitative forskningsmetoder i praksis* (Bd. 4). Gyldendal akademisk Oslo.
- Utdanningsdirektoratet. (2006). *Læreplan i matematikk (MAT1-01)*. https://www.udir.no/kl06/MAT1-01/Hele/Grunnleggende_ferdigheter
- Utdanningsdirektoratet. (2019). *Algoritmisk tenkning*. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2020a). *Grunnleggende ferdigheter: Matematikk 1-10 (MAT01-05)* <https://www.udir.no/lk20/mat01-05/om-faget/grunnleggende-ferdigheter?lang=nob>
- Utdanningsdirektoratet. (2020b). Kompetansemål og vurdering: Kompetansemål etter 2. trinn (MAT01-05). <https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv20?lang=nob>
- Utdanningsdirektoratet. (2020c). *Læreplan i matematikk 1.-10. trinn (MAT01-05)*. <https://www.udir.no/lk20/mat01-05>
- Utdanningsdirektoratet. (2020d). *Overordnet del: Grunnleggende ferdigheter*. <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/grunnleggende-ferdigheter/>
- Utdanningsdirektoratet. (2022). *Velkommen til Utdanningsdirektoratets kompetanseportal*. https://bibsyst.instructure.com/search/all_courses?design=udir
- Vygotsky, L. S. (1997). The historical meaning of the crisis in psychology: A methodological investigation. *The collected works of LS Vygotsky*, 3, 233-343.

- Vygotsky, L. S. & Cole, M. (1978). *Mind in society: Development of higher psychological processes*. Harvard university press. <http://ouleft.org/wp-content/uploads/Vygotsky-Mind-in-Society.pdf>
- Williams, M. K. (2017). John Dewey in the 21st century. *Journal of Inquiry and Action in Education*, 9(1), 7. <https://digitalcommons.buffalostate.edu/cgi/viewcontent.cgi?article=1147&context=jae>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145%2F1118178.1118215>
- Wing, J. M. (2011). Research notebook: Computational thinking—What and why. *The link magazine*, 6, 20-23. <https://people.cs.vt.edu/~kafura/CS6604/Papers/CT-What-And-Why.pdf>
- Zeigler-Hill, V. & Shackelford, T. K. (2020). *Encyclopedia of personality and individual differences*. Springer. <https://doi.org/10.1007/978-3-319-24612-3>

Vedlegg

Vedlegg 1 - Samtykkeskjema

Vil du delta i forskningsprosjektet

Undersøkelse om lærers erfaringer med programmering og matematikk

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er undersøke læreres erfaringer med bruk av data og programmering i matematikkundervisningen. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Prosjektet er på masternivå og skal undersøke læreres erfaringer bruk av programmering for å lære elevene matematikk. Det skal bli intervjuet tre til fem lærere, og problemstillingen omhandler hvordan man kan bruke programmering til å gi elever mer forståelse og interesse i matematikk.

Hvem er ansvarlig for forskningsprosjektet?

Høgskolen i Østfold er ansvarlig for prosjektet, hvor Simen Holt Didriksen skal gjennomføre intervjuer, transkripsjon og arbeid med oppgaven.

Hvorfor får du spørsmål om å delta?

Prosjektet retter seg mot lærere som har noe eller mye erfaring med å bruke programmering i matematikkundervisningen, og lærere som planlegger å bruke programmering.

Hva innebærer det for deg å delta?

Hvis du velger å delta i prosjektet, vil det bli gjennomført et anonymt intervju med studenten, Simen, som vil ta taleopptak av intervjuet, og lagre det elektronisk. Taleopptaket og transkripsjoner vil bli slettet etter levert oppgave.

Det er frivillig å delta

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

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. Navnet ditt vil bli erstattet med et annet navn i prosjektarbeidet, slik at ingen vet at det er du som gir informasjon. Du vil bli spurt om å oppgi kjønn og alder, som vil bli brukt i oppgaven. Appen Diktafon vil bli tatt i bruk, som anonymt lagrer intervjuene elektronisk.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er 15. mai 2022. Etter dette vil personopplysninger og taleopptak bli slettet.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

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

Dine rettigheter

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

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

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

Høgskolen i Østfold ved Khaled Ben Latief Jemai, khaled.jemai@hiof.no, +47 696 08 132

Vårt personvernombud: Line Mostad Samuelsen, line.m.samuelsen@hiof.no, +47 696 08 234

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

Khaled Ben Latief Jemai
(Forsker/veileder)

Simen Holt Didriksen
(Student)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet Undersøkelse om læreres erfaringer med programmering og matematikk, og har fått anledning til å stille spørsmål. Jeg samtykker til:

- ☐ å delta i intervju
- ☐ at Simen Holt Didriksen kan gi opplysninger om meg til prosjektet – hvis aktuelt
- ☐ at opplysninger om meg publiseres slik at jeg kan gjenkjennes (alder, kjønn) – hvis aktuelt

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

(Signert av prosjektdeltaker, dato)

Vedlegg 2 - Intervjuguide

Intervjuguide til masterprosjekt – læreres refleksjoner om innføringen av programmering i læreverket og bruken av programmering som læringsmetode.

1. Tidligere erfaringer

- Hvor mange år har du jobbet som lærer?
- Hva er din utdanningsbakgrunn?
- Hvor lenge har du drevet med programmering i undervisningen?
- Gir skolen tilbud om kurs/videreutdanning innen programmering?
- Hvis ja, hvilket programmeringsspråk, hvor ofte, over hvor lang tid?

2. Begreper

- Hva legger du i begrepet «programmering»?
- Hva legger du i begrepet «læringsmetode»?
- Hva legger du i begrepet «problemløsningsstrategi»?
- Hva legger du i begrepet «algoritmisk tenkning»?
- Hvilke(t) programmeringsspråk er du kjent med?

3. Elever og programmering

- Hva er dine tanker rundt innføring av programmering som en del av matematikkfaget?
- Hva tenker du elevene får ut av å holde på med programmering som læringsmetode?
- Hvordan tenker du programmering er nyttig i forhold til problemløsningsstrategier og algoritmisk tenkning?

4. Etter nye læreplanen

- Har din egen kompetanse innen programmering blitt bedre etter den nye læreplanen?
- Hvis ja, hvordan og i hvor stor grad?
- På hvilken måte har din egen undervisning blitt endret etter den nye læreplanen?

5. Annet

- Noe annet som kan nevnes i sammenheng med temaet?
- Eksempler fra undervisning om programmering som er merkverdig?
- Andre erfaringer fra eget arbeid med programmering?

26.04.2022, 09:53

Meldeskjema for behandling av personopplysninger

NSD NORSK SENTER FOR FORSKNINGSDATA

Vurdering

Referansenummer

622848

Prosjekttittel

Masteroppgave

Behandlingsansvarlig institusjon

Høgskolen i Østfold / Fakultet for lærerutdanninger og språk / Institutt for real-, praktisk-etiske-, samfunns- og religionsfag

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Khaled Ben Latief Jemai, khaled.jemai@hiof.no, tlf: +4798859675

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Simen Holt Didriksen, simenhd96@gmail.com, tlf: 93635491

Prosjektperiode

15.10.2021 - 15.05.2022

Vurdering (1)

15.10.2021 - 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 15.10.2021. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 15.05.2022.

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).

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 meldeskjemat. 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.

Kontaktperson hos NSD: Henning Levold
Lykke til med prosjektet!