

Signe Sandengen Kveseth

Programmeringsverktøy i matematikkundervisning

En kvalitativ intervjustudie av et utvalg
grunnskolelæreres refleksjoner

Masteroppgave i matematikdidaktikk 5.-10.

Veileder: Melih Turgut

Mai 2022

Signe Sandengen Kveseth

Programmeringsverktøy i matematikkundervisning

En kvalitativ intervjustudie av et utvalg
grunnskolelæreres refleksjoner

Masteroppgave i matematikdidaktikk 5.-10.
Veileder: Melih Turgut
Mai 2022

Norges teknisk-naturvitenskapelige universitet
Fakultet for samfunns- og utdanningsvitenskap
Institutt for lærerutdanning



Kunnskap for en bedre verden

Sammendrag

Bakgrunn for temaet som denne masteroppgaven tar opp er innføringen av programmering i den norske grunnskolen gjennom Fagfornyelsen, der hovedansvaret for opplæringen ble lagt til matematikkfaget. Det blir dermed opp til matematikklærere å sørge for å gjennomføre opplæringen, til tross for at ikke alle nødvendigvis har forkunnskaper om programmering (Forsström & Kaufmann, 2018). Det bringer frem en rekke spørsmål om hva slags kunnskap matematikklærere trenger for å legge til rette for at elever utvikler ferdigheter og kompetansemål definert i læreplanen. Siden programmering i matematikkfaget fortsatt er relativt nytt i Norge vil det være behov for forskning på fagfeltet. Det vil være spesielt relevant å undersøke temaet fra et lærerperspektiv, da det finnes lite kunnskap om læreres syn på programmering og undervisning av programmering i matematikk, og desto mindre om hvordan de håndterer de siste reformene som har funnet sted i flere nordiske læreplaner (Pörn et al., 2021).

Hensikten med denne studien har derfor vært å bidra med mer kunnskap om hvordan matematikklærere underviser med programmeringsverktøy og hva som påvirker valgene de tar gjennom denne prosessen. Studiens problemstilling var: Hva kjennetegner matematikklæreres pedagogiske teknologiske kunnskap (PTK) i en programmeringskontekst i grunnskolen?

Studien har benyttet kvalitative metoder med intervju av fem matematikklærere fra forskjellige barne- og ungdomsskoler. Datamaterialet ble analysert gjennom en teordrevet tematisk analyse, med forhåndsbestemte kategorier. Både hoved- og underkategoriene var basert på begreper fra det teoretiske rammeverket PTK og dets tre teoretiske komponenter. Rammeverket er designet for å forstå læreres praksis og deres faglige utvikling ved bruk av digitale hjelpemidler i matematikkundervisning.

Studien viser at samtlige lærere beskriver programmeringsbegrepet, med noen variasjoner, som å få noe til å skje ved hjelp av instruksjoner i en bestemt rekkefølge. Alle gir uttrykk for positive orienteringer mot bruk av programmeringsverktøy i matematikkundervisning, og alle virker å benytte flere ulike undervisningstilnæringer. Studien tyder likevel på at til tross for programmerings plass i matematikkfaget, er det utfordrende for lærerne å knytte programmering og matematikk sammen. Lærerne fra ungdomsskoler forteller om tekniske utfordringer som legger begrensninger for at programmering kan bli et verktøy for å lære matematikk bedre. Lærerne fra barneskoler forteller om utfordringer med å knytte programmeringsaktiviteter direkte til temaer innenfor matematikken.

Nøkkelord: matematikkundervisning, programmeringsverktøy, lærerens pedagogiske teknologiske kunnskap

Abstract

The background for the topic that this master's thesis addresses is the introduction of programming in the Norwegian primary and lower secondary school through Fagfornyelsen, where the main responsibility for the training of programming was added to the subject of mathematics. Thus, it is up to mathematics teachers to complete the training, despite the fact that not everyone necessarily has prior knowledge of programming (Forsström & Kaufmann, 2018). That raises number of questions about what kind of knowledge mathematics teachers need to help students develop skills and competencies defined in the curriculum. Since programming in the mathematics subject is still relatively new in Norway, there will be a need for research in the field. It will be particularly relevant to examine the topic from a teachers' perspective, as there is little knowledge about teachers' views on programming and teaching of programming in mathematics, and nevertheless about how they handle the latest reforms that have taken place in several Nordic curricula (Pörn et al., 2021).

The purpose of this study has therefore been to contribute with more knowledge about how mathematics teachers teach with programming tools and what influences the choices they make during this process. The study's research question was: What characterizes mathematics teachers' pedagogical technological knowledge (PTK) in a programming context in primary and lower secondary school?

The study has used qualitative methods with interviews of five mathematics teachers from different primary and lower secondary schools. The data material was analyzed through a theory-driven thematic analysis, with predetermined categories. Both the main and subcategories were based on concepts from the theoretical framework PTK and its three theoretical components. The framework is designed to understand teachers' practice and their professional development through the use of digital tools in mathematics teaching.

The study shows that all teachers describe the concept of programming, with some variations, as making something happen using instructions in a specific order. All express positive orientations towards the use of programming tools in mathematics teaching, and all seem to use a number of different teaching approaches. Nevertheless, the study suggests that despite the place of programming in the subject of mathematics, it is challenging for teachers to link programming and mathematics together. The teachers from upper secondary schools talk about how technical challenges place restrictions on the fact that programming can become a tool for learning mathematics better. Teachers from primary schools talk about challenges of linking programming activities directly to topics within mathematics.

Keywords: mathematics teaching, programming tools, teachers' pedagogical technological knowledge

Forord

Fem fantastiske år i Trondheim og på lærerstudiet ved NTNU, nærmer seg slutten. Arbeidet med denne masteroppgaven har til tider virket uoverkommelig, men jeg kom meg i mål til slutt jeg også. Innlevering av oppgaven markerer at studenttilværelsen er over, noe som føles rart og vemodig. Heldigvis venter det en lærerjobb på meg i Oslo, en by flere gode studievenner også har valgt seg ut som neste stopp.

Det er mange som fortjener en stor takk, først og fremst lærerne som stilte opp på intervju, uten dere ville det ikke blitt noe av denne masteroppgaven. På tross av travle lærerhverdager tok dere dere tid til å svare på spørsmålene med stor interesse og engasjement. Det er inspirerende.

Tusen takk til veilederen min, Melih Turgut, for at du har vært tilgjengelig og behjelpelig gjennom hele prosessen. Det har vært lærerikt å diskutere med deg, og jeg er veldig takknemlig for all tiden du har brukt på å lese gjennom det jeg har skrevet og på de konstruktive tilbakemeldingene du har gitt meg underveis.

Til slutt vil jeg takke mine nærmeste som har oppmuntret og støttet meg gjennom dette halvåret, jeg setter uendelig stor pris på dere.

Signe Sandengen Kveseth,

Trondheim 25. mai 2022.

Innholdsfortegnelse

1 Innledning	1
1.1 Problemområdet og forskningsspørsmål	3
1.2 Oppgavens oppbygging	4
2. Teoretisk perspektiv og bakgrunn.....	6
2.1 Valg av teoretisk rammeverk	6
2.2 Pedagogisk teknologisk kunnskap (PTK)	7
2.2.1 Instrumentell genesis	8
2.2.2 Matematisk kunnskap for undervisning (MKT)	11
2.2.3 Personlige orienteringer.....	12
2.3 Algoritmisk tenkning	13
2.4 Programmering	15
2.5 Tidligere forskning.....	15
2.5.1 Programmering og matematikk	15
2.5.2 Programmering i matematikkfaget fra et lærerperspektiv	16
3 Metode	18
3.1 Metodisk tilnærming	18
3.2 Vitenskapsteoretisk forankring	18
3.3 Datainnsamling	19
3.3.1 Utvalg og rekruttering	19
3.3.2 Intervju.....	20
3.4 Analysearbeid	21
3.4.1 Metode for analyse av datamaterialet.....	22
3.5 Forskningens kvalitet	24
3.5.1 Pålitelighet	24
3.5.2 Gyldighet	25
3.5.3 Generaliserbarhet	26
3.6 Etske betraktninger	26
4 Presentasjon av funn.....	28
4.1 Personlige lærerorienteringer	28
4.1.1 Beskrivelse av programmeringsbegrepet	28
4.1.2 Verdien av å bruke programmering i matematikk	29
4.1.3 Selvtillit til å bruke programmering	30
4.2 Matematisk digital kunnskap om undervisning	31
4.2.1 Kunnskap om læreplaninnhold.....	31
4.2.2 Undervisningstilnærming	32

4.2.3 Kunnskap om elevene og deres tenkning	33
4.3 Dobbel instrumentell genesis	34
4.3.1 Utfordringer knyttet til blokkbasert programmering.....	34
4.3.2 Lærerutfordringer og bekymringer om hvordan knytte programmering til matematikk.....	35
5 Drøfting.....	38
6 Avslutning	46
6.1 Videre forskning.....	47
Referanser.....	48
Vedlegg.....	53

Figurer

Figur 1. Modell av den nyeste versjonen av PTK (Thomas and Palmer, 2014, s. 76).	7
Figur 2. Oversatt modell av prosessen instrumentell genesis (Guin & Trouche 1998, s. 202).	9
Figur 3. Oversatt modell av Tabach og Trglovás' (2019, s. 190) Matematisk Digital Kunnskap for Undervisning (MDKT).	11
Figur 4. Oversatt modell av underdimensjoner i sammenheng med undervisning med teknologi (Thurm & Barzel, 2022, s. 48).....	13
Figur 5. Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019c).....	14
Figur 6. Grafisk fremstilling av hvordan forskningsspørsmålene henger sammen med problemstillingen.....	24

Tabeller

Tabell 1. Introduksjon av deltakerne i studien.....	20
Tabell 2. Eksempler på datakoding og temaer.....	23

1 Innledning

Tradisjonelt sett har informatikk, og programmering spesielt, vært ansett som relevant først og fremst for fagfolk innenfor IT-bransjen (Nouri et al., 2019). I løpet av de senere årene har imidlertid dette synet endret seg. Som følge av den økende digitaliseringen i verden spiller programvare og teknologi nå en stor rolle på nesten alle områder i samfunnet, og i livene til folk flest. Dette reiser et behov for å forstå hvordan den digitale verden fungerer, samt hvilke muligheter og risikoer den gir. Haraldsrud et al. (2020) forbinder denne forståelsen med begrepet *digital dannelse*. Ifølge Balanskat og Engelhardt (2015) vil mange av dagens elever være med på å utvikle teknologi i fremtiden som er viktig for samfunnet. Følgelig har algoritmisk tenkning blitt en stadig viktigere kjernekompetanse for ferdigheter i det 21. århundre, og viktig for utdanningspolitikken som ønsker å tilpasse utdanningssektoren med hensyn til fremtidige samfunnskrav (Forsström & Kaufmann, 2018).

En årsak til begrepets oppmerksomhet i nyere har tid har med det allment aksepterte synet om at algoritmisk tenkning er anvendelig i en rekke forskjellige disipliner, som for eksempel matematikk. Programmering er ofte uløselig knyttet til utvikling av algoritmisk tenkning, noe som handler om at programmering forutsetter at eleven demonstrerer en viss mestring av algoritmisk tenkning (Lye & Koh, 2014). Algoritmisk tenkning utvikler seg imidlertid ikke utelukkende gjennom programmeringsaktiviteter, men strekker seg langt utover denne disiplinen (Lye & Koh, 2014). Programmering er grunnlaget for alle digitale løsninger, systemer og programvarer vi bruker (Nouri et al., 2019). For å forstå den digitale verden innebærer det altså en viss forståelse for hva programmering er. Sevik (2016) definerer programmering først og fremst som en problemløsningsprosess med å skrive koder som kan forstås av datamaskiner, men trekker også frem programmering som en kreativ prosess som handler om å skape noe nytt eller forbedre noe annet.

Interessen for algoritmisk tenkning og programmering har ført til at begrepene har fått tildelt en betydelig rolle i flere lands nasjonale læreplaner (Bråting & Kilhamn, 2021; Pörn et al., 2021). Noen land har valgt å introdusere informatikk som et eget fag, for eksempel England, mens Finland og Sverige har hatt en tverrfaglig tilnærming som bygger på konseptet digital kompetanse. I Norge ble hovedansvaret for opplæring av programmering lagt til matematikkfaget. Nedenfor er kompetansemålene i matematikk som kan knyttes til algoritmisk tenkning og programmering listet opp (Utdanningsdirektoratet, 2019a):

2. trinn: Lage og følge regler og trinnvise instruksjoner i lek og spill.

3. trinn: Lage og følge regler og trinnvise instruksjoner i lek og spill knyttet til koordinatsystemet.

4. trinn: Lage algoritmer med bruk av variabler, vilkår og løkker.

5. trinn: Lage og programmere algoritmer med bruk av variabler, vilkår og løkker.

6. trinn: bruke variabler, løkker, vilkår og funksjoner i programmering til å utforske geometriske figurer og mønstre.

7. trinn: Bruke programmering til å utforske data i tabeller og datasett.

8. trinn: Utforske hvordan algoritmer kan skapes, testes og forbedres ved hjelp av programmering.

9. trinn: Simulere utfall i tilfeldige forsøk og beregne sannsynligheten for at noe skal inntreffe, ved å bruke programmering.

10. trinn: Utforske matematiske egenskaper og sammenhenger ved å bruke programmering.

I debatten om hvorfor programmering skal inn i skolen, har det vært løftet frem som en nødvendig kunnskap for alle medborgere (Nouri et al., 2020). Andre argumenter har handlet om programmering som et pedagogisk verktøy for å utvikle elevens algoritmiske tenkning (Bråting & Kilhamn, 2021). Siden den første interessen for programmering i skolen vokste frem, har det blitt utviklet en rekke nye programmeringsmiljøer. Noen av miljøene er spesielt utviklet for nybegynnere og unge elever, hvorav flere er blokkbaserte (Duncan et al., 2014). En av de mest populære er Scratch (<https://scratch.mit.edu/>), utviklet ved Massachusetts Institute of Technology. Programmet er verdens største kodefelleskap for barn og fremmer blant annet algoritmisk tenkning og problemløsningsferdigheter. I tillegg til Scratch finnes det flere tilgjengelige ressurser spesielt designet for at barn skal kunne lære seg konsepter ved programmering. Micro:bit (<https://microbit.org/>) lar barn utforske hvordan programvare og maskinvare fungerer sammen. Dette lar seg gjøre ved å skrive et program på datamaskinen (programvare) som deretter overføres til den lille datamaskinen i lommeformat som kalles Micro:bit (maskinvare), via en USB-ledning. Selv om den vanligste strategien for å lære bort algoritmisk tenkning er gjennom datastyrt aktiviteter (plugged activities), er også analoge aktiviteter (unplugged activities) en vanlig tilnærming i skolen, spesielt på lavere skoletrinn (Brackmann et al., 2017). CS unplugged (<https://www.csunplugged.org/en/>) er en samling gratis undervisningsmaterieell som lærer algoritmisk tenkning gjennom aktiviteter som ikke krever digitale enheter.

Rollen programmering fikk i den norske læreplanen gjennom Fagfornyelsen høsten 2020, forutsetter at matematikklærere er nødt til å integrere det i sin praksis. Forsström og Kaufmann (2018) hevder at lærerrollen kan bli utfordrende i forbindelse med integrering av programmering i et matematikkpensum, fordi matematikklæreren ikke nødvendigvis har forkunnskaper om programmering. Ifølge Stenseth et al. (2019) vil det være størst behov for lærernes kunnskaper i planleggingsfasen og for å få elevene i gang. De trekker frem at noe av det mest krevende i denne fasen handler om å finne riktig vanskelighetsgrad på oppgavene som skal tilfredstille matematiske og programmeringsmessige aspekter (Stenseth et al., 2019). I utarbeidelsen av oppgavedesign er genuine og engasjerende kontekster for aktiviteter pekt på som utgangspunkt for å skape overbevisende utfordringer som forutsetter matematikk for å løse, og der teknologien spiller en viktig rolle (Bray & Tangney, 2017). Lærere må altså mestre kunsten med å integrere teknologi på en slik måte at den formidler læring av matematikk (NCTM, 2000, s. 24):

Technology cannot replace the mathematics teacher, nor can it be used as a replacement for basic understandings and intuitions. The teacher must make prudent decisions about when and how to use technology and should ensure that the technology is enhancing students' mathematical thinking.

Ifølge Stephens og Kadijcevic (2020) bør det i dagens kontekst være et toveis forhold der matematikk bidrar til digital kompetanse og algoritmisk tenkning. Ideelt skal dette forholdet bidra til utvikling av dypere matematisk forståelse på alle trinn i løpet av skolegangen. Dette er i tråd med intensjonen om at elever i den norske skolen skal oppnå

dybdelæring i alle fag, deriblant matematikk, slik det er beskrevet i overordnet del av læreplanen (Utdanningsdirektoratet, 2019b).

1.1 Problemområdet og forskningsspørsmål

Mens programmering tidligere ikke var en obligatorisk del av norsk grunnopplæring, med unntak av enkelte studieretninger i videregående skole, betyr tillegget til læreplanene i dag at nytt innhold vil bli undervist av de aller fleste lærere. Dette til tross for at realiteten var at mange lærere ikke hadde programmeringskunnskaper da Fagfornyelsen ble innført (Johansen, 2020). Det bringer frem en rekke spørsmål knyttet til hva lærere skal undervise og hvordan de skal undervise for å legge til rette for at elever utvikler ferdighetene og kompetansemålene som er definert i læreplanen. Selv om det finnes noe litteratur om undervisning og læring av programmering i skolesammenheng, er den meste forskningen gjennomført på universitetsnivå (Heintz et al., 2016). Som følge av den korte tidsperioden programmering har spilt en rolle i matematikkundervisning, er forskning på temaet fra norsk skolekontekst begrenset. Herheim (2019) trekker frem at på tidspunktet før programmering ble implementert i matematikkfaget, mente fagmiljø og lærere at skoler ikke var godt nok forberedt til å lage god undervisning med programmering. Det er desto større grunn til å belyse temaet, spesielt hvilke erfaringer lærere har gjort seg med å bruke programmering i matematikkundervisning i løpet av de siste to årene.

I forbindelse med forskning på lærerkunnskap og pedagogiske teknologier har Technological Pedagogical Content Knowledge (TPACK) vært et populært teoretisk rammeverk i flere forskningsartikler (f.eks. Koehler, Mishra & Yahya, 2007; Sundberg, 2015). I denne masteroppgaven vil jeg ta utgangspunkt i rammeverket Pedagogical Technological Knowledge (PTK) (Thomas & Palmer, 2014) som er spesifikt for matematikkundervisning. Som det vil bli beskrevet i neste kapittel, består PTK av tre komponenter: matematisk kunnskap for undervisning (Ball), instrumentell genesis (f.eks. Haspekian, 2005; Trouche, 2020) og personlige orienteringer.

Med denne masteroppgaven ønsker jeg derfor å bidra til å fylle forskningsgapet beskrevet ovenfor ved å undersøke læreres erfaringer med programmering og programmering i matematikkundervisning, i lys av det teoretiske rammeverket PTK. Det leder meg til følgende problemstilling:

«Hva kjennetegner matematikklæreres pedagogiske teknologiske kunnskap (PTK) i en programmeringskontekst i grunnskolen?»

Denne problemstillingen skal jeg undersøke ved hjelp av tre forskningsspørsmål, der det første undersøker ...

- 1. Hva er aspekter ved læreres personlige orienteringer mot programmering i matematikkundervisning?*
- 2. Hvilke aspekter ved læreres matematiske kunnskap for undervisning kommer til uttrykk i deres beskrivelser av deres undervisningstilnærming til matematikk med programmeringsverktøy?*
- 3. På hvilke måter kan læreres instrumentelle genesis påvirke koblingen mellom programmering og matematikk i undervisning?*

For å oppnå dette har jeg intervjuet til sammen fem matematikklærere fra barneskolen og ungdomsskolen som har brukt programmering i matematikkundervisning siden

Fagfornyelsen tredde i kraft høsten 2020. Videre har jeg gjennomført en tematisk analyse av det innsamlede datamaterialet, før jeg til slutt diskuterte funnene i lys av relevant teori og tidligere forskning om temaet lærere og programmering i matematikkundervisning.

1.2 Oppgavens oppbygging

Resten av oppgaven er organisert som følger: først presenterer jeg en mer detaljert bakgrunn for masteroppgaven i kapittel 2, som inneholder en beskrivelse av PTK rammeverket, andre relevante teoretiske begrep og tidligere forskning på programmering i skolen og læreres syn på programmering i matematikkfaget. Deretter vil jeg presentere og utdype de metodiske valgene for oppgaven i kapittel 3. Videre presenterer jeg resultatene fra fem analyserte lærerintervjuer i kapittel 5, som drøftes i lys av tidligere forskning og relevant teori i kapittel 6. Til slutt vil det være en oppsummering av funn og en kort presentasjon av idéer for fremtidig arbeid i kapittel 7.

2. Teoretisk perspektiv og bakgrunn

Dette kapittelet beskriver grunnlaget for masteroppgavens analytiske linse, samt tidligere forskning som er relevant for å forstå mer om programmering og tilhørende lærerpraksis i Norge. Det er særlig forskning på pedagogisk teknologisk kunnskap for å integrere verktøy i matematikk og programmering i grunnskolen som er viktig å se på. I tillegg vil det være relevant for oppgaven å inkludere tidligere forskning på læreres syn på programmering og dets rolle i matematikkfaget. Sentrale begreper er pedagogisk teknologisk kunnskap, instrumentell genesis, matematisk kunnskap for undervisning, personlige orienteringer, programmering og algoritmisk tenkning.

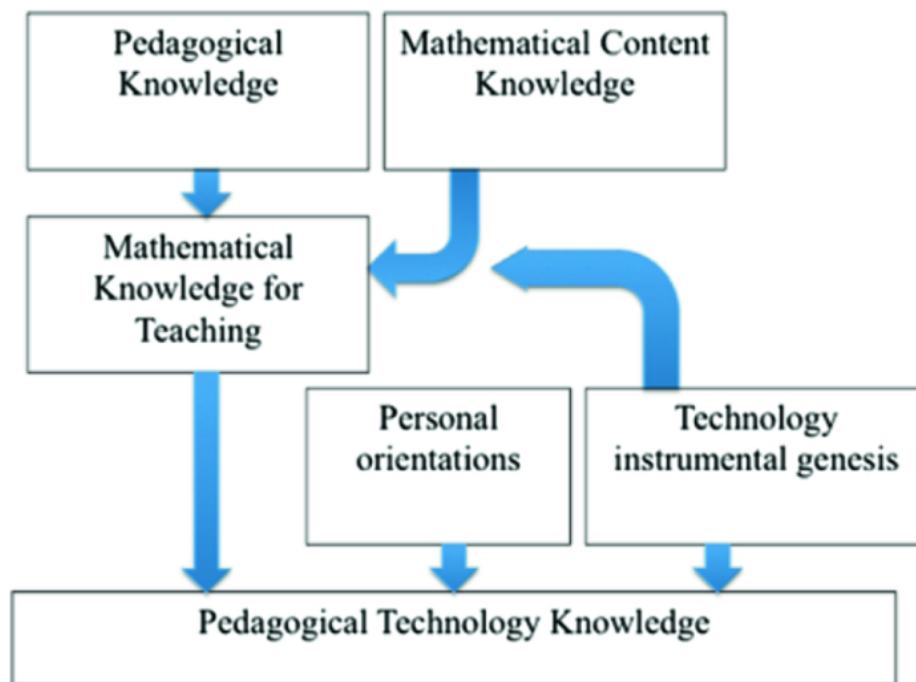
2.1 Valg av teoretisk rammeverk

For å beskrive og studere læreres kunnskap og læring ved bruk av digitale hjelpemidler, finnes det en rekke teoretiske rammeverk. PTK og TPACK er to teoretiske rammeverk som er designet spesifikt for å forstå læreres praksis og deres faglige utvikling. Begge rammeverkene bygger på en idé om at kunnskapen lærere trenger i en slik kontekst består av to hovedkomponenter, en innholdskomponent og en pedagogisk komponent (Shulman, 1986). Komponentene er relatert til henholdsvis matematikkdisiplinen og til den faglige kunnskapen for undervisning.

Mishra og Koehler (2006) introduserte TPACK for å kunne teoretisere den unike kunnskapen som er nødvendig for å undervise med digital teknologi på generell basis. Blant rammeverkene som tar for seg læreres digitale kunnskap og ferdigheter er TPACK det som brukes oftest med flere aksepterte tolkninger (Tabach & Trglóvá, 2019). TPACK er ikke spesifikt for et emne eller fag, det kan betraktes for et hvilket som helst. Omtrent på samme tid ble PTK (Hong & Thomas, 2006; Thomas & Hong, 2005) introdusert, og til forskjellig fra TPACK har det alltid vært fokusert på matematikk som har sine egne nyanser av innholdskunnskap. Det er også et par andre faktorer som skiller PTK fra TPACK. For det første bruker PTK det teoretiske grunnlaget for *instrumentell genesis* (Rabardel, 1995) med sin forklaring på hvordan et verktøy forvandles til et didaktisk instrument. For det andre inkluderer PTK det avgjørende elementet av læreres *personlige orienteringer* til bruk av teknologi, og deres rolle i å påvirke målsetting og beslutningstaking, noe som virker fraværende i TPACK (Thomas & Palmer, 2014). Basert på de fremlagte forskjellene mellom PTK og TPACK, vil jeg argumentere for at PTK vil være det mest hensiktsmessige å bruke i min studie. PTK vil gjøre det mulig å vurdere læreres orienteringer, noe TPACK ikke vil fange opp. Lærers orienteringer som inkluderer troen på teknologiens verdi og deres selvtillit til å bruke den, vil være relevant å undersøke fordi det ifølge Thomas og Palmer (2014) er avgjørende for implementering av teknologi i undervisning. Teknologi vil i denne studien gjelde programmering og programmeringsverktøy. Videre vil jeg argumentere for at siden PTK er fokusert mot matematikkdisiplinen, vil det være spesielt relevant for denne studien som tar utgangspunkt i programmering i matematikkfaget. Jeg ser altså størst potensial i å bruke PTK som teoretisk rammeverk for min forskning, og i det følgende vil rammeverket og dets komponenter beskrives nærmere.

2.2 Pedagogisk teknologisk kunnskap (PTK)

Figur 1 nedenfor viser den nyeste versjonen av rammeverket PTK som beskrives i det følgende. I matematikklasserommet spiller læreren en avgjørende rolle for hvor ofte og hvordan teknologi brukes (Thomas & Palmer, 2014). En rekke ytre og iboende faktorer viser seg å kunne påvirke disse avgjørelsene (Thomas, 2006). Ytre faktorer kan være mangel på ressurser eller en skolekultur som ikke har for å vane å bruke teknologi i undervisning. Iboende faktorer kan handle om læreres orienteringer til teknologi og hvilke oppfatninger de har om matematikk og hvordan det burde læres. En vurdering av faktorer som påvirker læreres bruk av teknologi i undervisning, ledet Thomas og Hong (2005) til å foreslå rammeverket PTK som de definerte som «knowing how to teach mathematics with the technology» (s. 256). Thomas og Hong (2005) foreslo at rammeverket kunne fungere som en indikator på læreres fremgang i implementering av teknologibruk. I matematikkfaget vil en lærers PTK inneholde prinsippene, konvensjonene og teknikkene som kreves for å undervise i matematikk gjennom teknologi (Thomas & Palmer, 2014). Lærere må altså ved bruk av teknologi også legge til rette for læring i matematikk: "Thus PTK includes the need to be a proficient user of the technology, but more importantly, to understand the principles and techniques required to build didactic situations incorporating it, to enable mathematical learning through the technology" (Thomas & Palmer, 2014, s. 75).



Figur 1. Modell av den nyeste versjonen av PTK (Thomas and Palmer, 2014, s. 76).

Etter at rammeverket først ble presentert av Thomas og Hong (2005), har Thomas og Palmer (2014) senere videreutviklet konseptet ved å sammenveve en rekke iboende lærerfaktorer for å produsere PTK. Blant disse faktorene er: læreres instrumentelle genesis med hensyn til teknologi, matematisk kunnskap for undervisning (Ball et al., 2008) og lærerorienteringer og mål, spesielt oppfatninger om verdien av teknologi og naturen for å lære matematisk kunnskap. Videre vil jeg utdype hver av de tre hovedpilarene.

2.2.1 Instrumentell genesis

I denne studien tar jeg utgangspunkt i den doble instrumentelle genesis foreslått av Haspekian (2011). Den teoretiske konstruksjonen ble utviklet i samsvar med den instrumentelle tilnærmingen (Rabardel, 2002), og omfatter både den *personlige* og den *profesjonelle instrumentelle genesis* hos lærere som bruker teknologi i matematikkundervisning. For å forstå konstruksjonen er det først nødvendig å forklare noen sentrale begreper fra den instrumentelle tilnærmingen (Rabardel, 2002).

Den instrumentelle tilnærmingen

Ifølge Rabardel (2002) oppstår det et forhold mellom mennesker og de tekniske objektene og systemene de befinner seg rundt. Begrepet *artefakt* brukes om slike objekter og systemer som ikke eksisterer av naturlige årsaker, men som følge av målbevisste menneskelige handlinger. Datamaskin og kalkulator er eksempler på artefakter. I den instrumentelle tilnærmingen kan digitale verktøy betraktes som artefakter. Monaghan og Trouche (2016) forklarer forskjellen mellom et artefakt og et verktøy gjennom et originalt eksempel med en passer: «The compass becomes a tool when it is used to draw a circle (which is its intended purpose); the same artifact becomes a different tool when it is used to stab someone» (s. 6). Det samme artefaktet kan altså bli forskjellige verktøy avhengig av hvilket formål det brukes til.

Verillon og Rabardel (1995) søkte å forstå hvordan tanker forholder seg til bruk av artefakter. Deres bidrag fikk tittelen instrumenterte aktivitetssituasjoner, og var bygget for å studere situasjoner der mennesker deltar i aktiviteter med bestemte formål i tankene, og hvor de deretter bevisst bruker et artefakt for å løse oppgaven. Verillon og Rabardel (1995) påpeker at det er viktig å understreke forskjellen mellom artefakt som det menneskeskapte objektet og *instrumentet* som en psykologisk konstruksjon: «The instrument does not exist in itself, it becomes an instrument when the subject has been able to appropriate it for himself and has integrated it with his activity» (Verillon og Rabardel, 1995, s. 86). Det samme artefaktet kan bli ulike instrumenter basert på personen som bruker det sine intensjoner, men også artefaktets spesifikke egenskaper. I oppdagelsen av et artefakts ulike bruksområder, vil en person kunne utvikle såkalte *bruksskjemaer*. Det er først når personen utvikler ett eller flere bruksskjemaer gjennom problemløsning med artefaktet at det kan forvandles til et instrument. Dette kan forenkles med formelen: $\text{Instrument} = \text{Artefakt} + \text{Skjema}$. Relatert til et eksempel med programmeringsverktøyet Scratch, så vil en person med intensjon om å simulere terningkast kunne utføre de nødvendige stegene i programmet dersom vedkommende har utviklet passende bruksskjemaer for denne spesifikke aktiviteten.

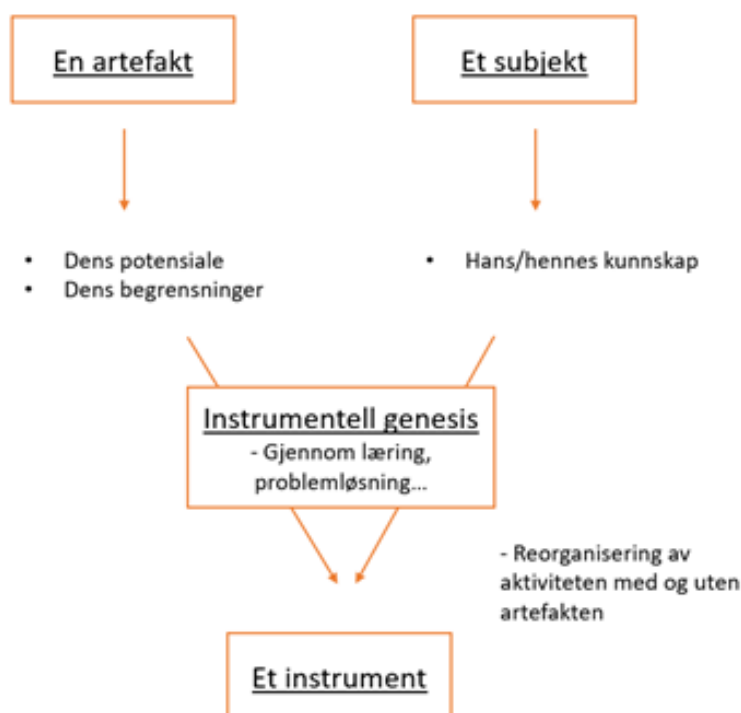
Instrumentell genesis

Proessen der et artefakt forvandles til et instrument har av Rabardel (2002) fått navnet *instrumentell genesis*. Prosessen søker å forstå utviklingen av artefaktene i henhold til brukerens aktivitet og fremveksten av de nye bruksområdene. Prosessen av instrumentell genesis har to dimensjoner. De kommer fra både artefakten og fra bruksskjemaene. Dimensjonene kan skilles fra hverandre, *instrumentalisering* rettet mot artefakten og *instrumentering* knyttet til personen. Gjennom den instrumentelle genesis skjer det dermed en dobbel bevegelse mellom artefaktet og personen:

Instrumentalisering (bevegelse fra personen mot artefaktet): Brukeren tilpasser verktøyet etter sitt behov. Kunnskap den personen besitter vil gjøre det mulig å velge og bruke den nødvendige funksjonen for en pågående handling. Denne prosessen er basert på

artefaktets egenskaper og de iboende egenskapene hos en person. Det kan forklares som en prosess der personen forbedrer artefaktets egenskaper. Nye funksjoner kan oppstå som ikke nødvendigvis var intensjonen til artefaktets designere. Med programmeringsverktøyet Scratch kan en slik prosess innebære at en person gjør seg kjent med de ulike funksjonene i programmet.

Instrumentering (bevegelse fra artefaktet mot personen): Artefaktets begrensninger og potensial påvirker handlingen til brukeren. Brukeren vil derfor endre aktivitetene, handlingene og bruksskjemaene sine for å bruke verktøyets funksjoner. I denne prosessen vil brukeren utvikle instrumenterte bruksskjemaer som tillater vedkommende å løse spesifikke oppgaver. Relatert til eksempelet ovenfor med Scratch, vil brukeren for eksempel mestre å simulere terningkast.



Figur 2. Oversatt modell av prosessen instrumentell genesis (Guin & Trouche 1998, s. 202).

Som figur 2 oppsummerer, innebærer instrumentell genesis en kobling mellom brukeres mentale skjemaer og teknikker for bruk av et artefakt, som hentes fram når det er nødvendig. Et artefakt vil ikke uten videre oppleves meningsfullt for en bruker, og en prosess av instrumentell genesis er derfor nødvendig: "An artefact is initially not meaningful to the user until he or she develops associated schemes of instrumented action to use the artefact for achieving a task, and effectively turning the artefact into a useful mathematical instrument" (Bozkurt et al., 2018, s. 44).

I matematikkundervisning oppstod begrepet instrumentell genesis for første gang i en artikkel av Guin og Trouche (1998) med tittelen «The complex process of converting tools into mathematical instruments: The case of calculators». Der hevder de at elever involverer seg i en kompleks instrumenteringsprosess som ikke nødvendigvis leder til bedre matematisk forståelse. Videre argumenterer de for at lærere derfor må bli oppmerksomme på det mulige gapet mellom et matematisk objekt og dens representasjon av spesifikt verktøy eller artefakt. For å hjelpe elevene i denne prosessen må lærere spille

en sentral rolle ved å guide, monitorere og orkestrere deres instrumentelle genesis Drijvers et al. (2009). Drijvers et al. (2010) beskriver instrumentell orkestrering som den bevisste og systematiske organiseringen, og bruken av tilgjengelige artefakter i et læringsmiljø i en matematisk kontekst.

Den doble instrumentelle genesis

Med de siste avsnittenes beskrivelser og definisjoner kan vi returnere til den teoretiske konstruksjonen av den doble instrumentelle genesis (Haspekian, 2011). Som nevnt innledningsvis omfatter den både den personlige og profesjonelle genesis hos lærere som bruker teknologi i undervisningen. Mens den personlige instrumentelle genesis er relatert til utviklingen av læreres personlige instrument, gir den profesjonelle instrumentelle genesis et profesjonelt instrument for en læreres didaktiske aktivitet. Ifølge Haspekian (2011) må lærere først tilegne seg grunnleggende ferdigheter for å mestre den spesifikke teknologien de har tenkt til å bruke, og dermed utvikle bruksskjemaer knyttet til denne teknologien (personlig instrumentell genesis). I tillegg må de utvikle en forståelse for hvordan de kan støtte elevenes matematikk læring i et digitalt miljø (profesjonell instrumentell genesis). Der den personlige instrumentelle genesis er felles for alle lærere, er den profesjonelle genesis unik for alle matematikklærere. Det samme artefaktet, Haspekian bruker et eksempel med regneark, kan altså bli et instrument for elevenes matematiske aktivitet og et annet instrument for lærerens didaktiske aktivitet. Lærere må derfor forholde seg til flere genesis: «Thus, for a given artefact, the teacher must consider various geneses enrolled in complex systems of various interacting instruments” (Haspekian, 2005, s. 118). Med en instrumentell tilnærming til et programmeringsverktøy som undervisningsinstrument, skapt av lærerens profesjonelle genesis, fremheves to prosesser av Haspekian (2011):

- En instrumenteringsprosess: verktøyet instrumenteres av læreren for å tjene didaktiske mål. Ved å gå vekk fra noen av verktøyet opprinnelige funksjoner kan det gradvis skapes et didaktisk potensial, eller oppdages eller tilegnes i tilfellet med et pedagogisk verktøy.
- En instrumenteringsprosess: lærere innlemmer nye opplegg i sine allerede etablerte undervisningsopplegg. I de nye oppleggene blir bruken av et verktøy integrert. Gradvis vil læreren spesifisere bruken av verktøyet for en bestemt type situasjoner (for eksempel gjennom å dra fordel av Scratch for læring i geometri) og organisere aktiviteten på en måte som gradvis er stabil for denne type situasjoner.

Instrumentet som er et resultat av lærerens profesjonelle genesis (Scratch for å lære bort geometri) er forskjellig fra instrumentet skap gjennom en personlig genesis (Scratch for å gjøre personlig arbeid med beregninger, feilsøking eller skrive koder). Lærere er derfor involvert i den doble instrumentell genesis. En lærers personlige genesis er noe liknende det man ønsker elevene skal bygge, ifølge Haspekian (2005). Læreres profesjonelle genesis derimot er mer komplisert ettersom den inkluderer elevenes instrumentelle genesis.

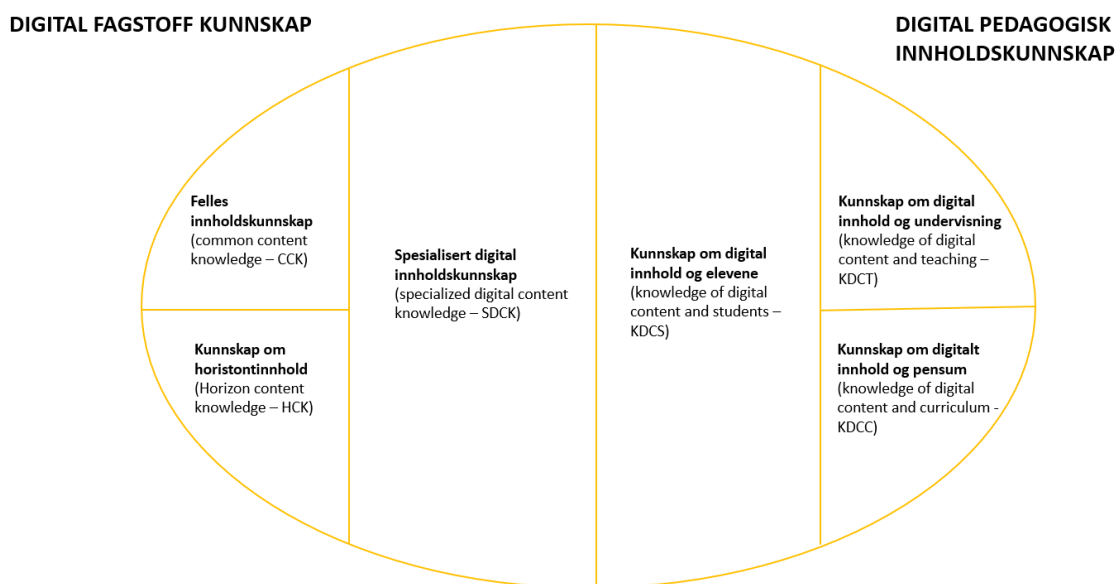
Videre i oppgaven vil jeg ta utgangspunkt i Haspekians (2011) teori om den doble instrumentelle genesis i forbindelse med faktoren instrumentell genesis i rammeverket PTK. Grunnen til dette er fordi den doble instrumentelle genesis betrakter instrumentell genesis fra et lærerperspektiv, deriblant profesjonell genesis som er unik for matematikklærere. Det samsvarer med problemstillingen i denne studien.

2.2.2 Matematisk kunnskap for undervisning (MKT)

“The mathematical knowledge needed to carry out the work of teaching mathematics” (Ball et al., 2008, s. 395).

En av PTKs tre komponenter, MKT, er en praksisbasert teori om innholdskunnskap for undervisning bygget på Shulmans (1986) forestilling om pedagogisk innholdskunnskap (PCK). Ifølge Shulman (1986) oppstår det et forhold mellom lærerens pedagogiske kunnskap og innholdskunnskap når læreren tolker innholdet og finner de mest hensiktsmessige metodene for å gjøre innholdet forståelig for elevene. Teorien MTK ble foreslått i en studie av Ball et al. (2008) der hensikten var å undersøke karakteren av profesjonsrettet fagstoffkunnskap i matematikk. Det ble forsøkt gjort ved å studere matematikkundervisning og identifisere matematisk kunnskap for undervisning basert på analyser av de matematiske problemene som oppstår i undervisningen. Deres tilnærming i studien fokuserte på hvordan lærere trenger å kjenne til temaene og prosedyrene de underviser i. I tillegg ønsket de å finne ut hva lærere ellers trenger å vite om matematikk, og hvordan de kan bruke slik matematisk kunnskap i praksis. Fra deres analyser kommer det frem at matematikklærere behøver kunnskap som er detaljert på måter som er unødvendig for hverdagen. I den forbindelse trekker de frem et eksempel basert på en enkel subtraksjonsberegning. Lærere som underviser subtraksjon må selv kunne utføre beregningen, det er nødvendig, men ikke tilstrekkelig. Ifølge Ball et al. (2008) krever dyktig undervisning evnen til å identifisere kilden til en matematisk feil. Denne feilen må i tillegg identifiseres raskt siden lærere forholder seg til flere elever samtidig. Studien konkluderer med at lærere må kunne matematikk på måter som er nyttig for blant annet å gi mening til elevsvar, og for å velge hensiktsmessige måter å representere det på slik at det gir mening for elevene.

I PTK er matematisk kunnskap for undervisning betraktet med fokus på læreres bruk av teknologi. For å gi mening til Ball et al.'s (2008) 6 identifiserte kunnskapsområder i konteksten for denne masteroppgaven, ønsker jeg videre å benytte Tabach og Trglovás (2019) modifiserte versjon.



Figur 3. Oversatt modell av Tabach og Trglovás' (2019, s. 190) Matematisk Digital Kunnskap for Undervisning (MDKT).

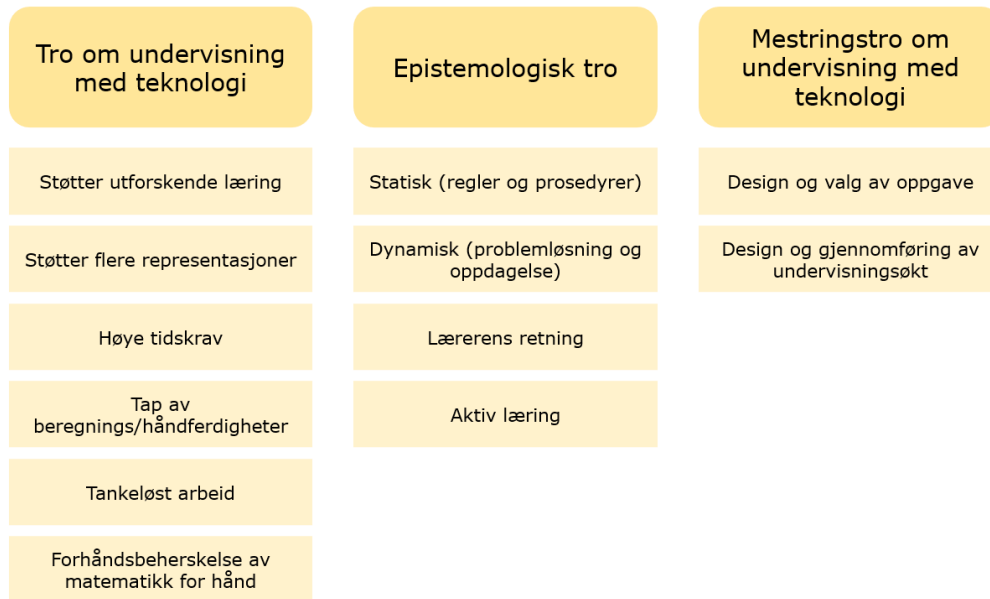
I denne versjonen, som har fått tittelen Matematisk Digital Kunnskap for Undervisning (MDKT), har Tabach og Trglová (2019) tilpasset Ball et al.'s (2008) MKT i et teknologisk miljø. Begrunnelsen for å benytte denne versjonen ligger i tilpasningen til det teknologiske miljøet, og i koblingene Tabach og Trglovás (2019) gjør i forbindelse med kunnskapsområdene og instrumentell genesis, som samsvarer med PTK rammeverket.

2.2.3 Personlige orienteringer

I PTK er læreres personlige orienteringer knyttet til deres tro, motivasjon og holdninger relatert til bruken av teknologi i matematikkundervisning (Clark-Wilson et al., 2020). Det handler spesielt om deres tro på verdien av teknologi for læring av matematikk og deres selvtillit til å bruke det (Thomas & Palmer, 2014). Tabach og Trglovás (2019) som også benytter PTK rammeverket i sin studie, betrakter lærerorienteringer som relatert til affektive aspekter og deres oppfatninger om matematikk, undervisning i matematikk og teknologi. I den forbindelse trekker de frem en rekke spørsmål som kan knyttes til dette aspektet:

What kind of discipline is mathematics and what is its norms and values? What is the teacher's role in teaching mathematics, and how can this role be accomplished? What added value will technology bring to learning mathematics, what affordances and constraints does it involve, and how confident do teachers feel with respect to its use? (Tabach & Trglovás, 2019, s. 190).

Siden Thomas og Palmer (2014) påpeker at læreres orienteringer spesielt handler om deres tro på verdien av teknologi for læring av matematikk og deres selvtillit til å bruke det, vil jeg av hensyn til oppgavens omfang ta utgangspunkt i begrepet tro og ikke motivasjon og holdninger som kan ha andre definisjoner. Ifølge Thurm og Barzel (2022) er nemlig læreres tro om undervisning med teknologi, epistemologiske tro og mestringstro om undervisning med teknologi ansett som avgjørende faktorer for undervisning i matematikk med teknologi (se figur 4). De trekker imidlertid frem at det er mangel på forskning som undersøker de nevnte faktorene på et mer detaljert nivå. I den forbindelse har de samlet underdimensjoner av de tre faktorene i sammenheng med undervisning med teknologi, i en modell (se figur 4). I modellen har de samlet underdimensjoner av læreres tro om undervisning med teknologi som har vært identifisert i forskning. Videre har de samlet tro om matematikkens natur i to hovedretninger, og oppfatninger om læring i matematikk i to perspektiver. Til slutt er mestringstro om undervisning med teknologi delt inn i to basert på forskning som har fremhevet to funksjonsdomener.



Figur 4. Oversatt modell av underdimensjoner i sammenheng med undervisning med teknologi (Thurm & Barzel, 2022, s. 48).

Videre i oppgaven vil jeg knytte lærers orienteringer til læreres tro og mestringstro i henhold til Thurm og Barzel (2022) dimensjoner.

2.3 Algoritmisk tenkning

Begrepet computational thinking oversettes som regel til algoritmisk tankegang på norsk (Bocconi & Cioccarillo, 2018; Sevik, 2016). Begrepet har også vært oversatt til algoritmisk tenkning (Gjøvik & Torkildsen, 2019) og algoritmisk tenking (Utdanningsdirektoratet, 2019c). Det kan derfor oppstå forvirring rundt begrepenes betydning når de oversettes til norsk, noe Gjøvik og Torkildsen (2019) sier kan skyldes at algoritmisk tenkning på norsk ikke brukes på samme måte som algorithmic thinking på engelsk. For å rydde opp i eventuell forvirring, vil jeg derfor presisere at jeg i denne masteroppgaven legger det samme i algoritmisk tenkning som i det engelske begrepet computational thinking. Der det i forskning er brukt algoritmisk tenkning med engelsk betydning, vil jeg som Gjøvik og Torkildsen (2019), oversette det til algoritmebehandling. Begrepet vil da brukes om det å følge og forklare algoritmer.

Algoritmisk tenkning ble for første gang introdusert i Seymour Paperts (1980) banebrytende arbeid *Mindstorms: Children, Computers and Powerful ideas*. Et drøyt tiår tidligere, i 1967, utviklet han sammen med kollegaer det første programmeringsspråket for barn, Logo. Ambisjonene bak Papert's idéer forsvant derimot gradvis etter at Logo ble introdusert. Omtrent 30 år senere returnerte man til begrepet, da Wing (2006) argumenterte for at det burde undervises i skolen på lik linje som lesing, skriving og regning. Det å kunne tenke som en informatiker mente hun ville være en fordel for alle, uavhengig av hvilket yrke som er involvert. Videre gir Wing (2006) en samlet beskrivelse av algoritmisk tenkning, og sier det involverer: «(...) solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science» (s. 33). På tross av begrepets utbredte bruk i dag, mangler det en allment akseptert definisjon (Mouza et al., 2017). I mangfoldet av definisjoner er det likevel noen elementer som går igjen. Shute et al. (2017) trekker frem dekomponering, abstraksjon og

algoritmer, det samme gjør Hoyles og Noss (2015) som antar at algoritmisk tenkning er basert på dekomponering, abstraksjon, algoritmebehandling, og i tillegg mønstergjenkjenning. Når det gjelder karakterisering av algoritmisk tenkning i matematikkundervisning, har Kallia et al. (2021) i en nylig studie utarbeidet tre hovedaspekter (ss. 179-180, mine oversettelser):

- *Problemløsning* (som i å forstå problemet, finne en strategisk tilnærming, utføre strategien),
- *Kognitive prosesser* (som abstraksjon, dekomposisjon, mønstergjenkjenning, algoritmisk tenkning, modellering, logisk og analytisk tenkning, generalisering og vurdering av løsninger og strategier),
- *Transposisjon* (formulere løsningen på et matematisk problem på en slik måte at det kan overføres til en annen person eller maskin).

I læreplanen for matematikk i Norge er algoritmisk tenkning ansett som en del av kjerneelementet *utforsking og problemløsning*. Plassen algoritmisk tenkning har fått i kjerneelementer for faget, som angir det viktigste og mest sentrale elevene skal lære, forteller hvilken viktig rolle begrepet har fått i LK20. Av Utdanningsdirektoratet (2019c) defineres algoritmisk tenkning som:

Å tenke algoritmisk er å vurdere hvilke steg som skal til for å løse et problem, og å kunne bruke sin teknologiske kompetanse for å få en datamaskin til å løse (deler av) problemet. I dette ligger også en forståelse av hva slags problemer/oppgaver som kan løses med teknologi og hva som bør overlates til mennesker.

Det er denne definisjonen som legges til grunn i den nye læreplanen LK20. Utdanningsdirektoratet presenterer også en modell for algoritmisk tenkning som har fått navnet *den algoritmiske tenkeren* (se figur 4). Modellen illustrerer viktige nøkkelbegreper som inngår i algoritmisk tenkning og arbeidsmåter den algoritmiske tenkeren benytter i problemløsning. Sentrale begreper er logikk, algoritme, dekomposisjon, mønstre, abstraksjon og evaluering. Arbeidsmåtene modellen fremhever er fikle, skape, feilsøke, holde ut og samarbeide.



Figur 5. Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019c).

2.4 Programmering

Programmering er ifølge Gjøvik og Torkildsen (2019) et naturlig miljø å implementere algoritmisk tenkning gjennom. Den samme oppfatningen deler Kilhamn et al. (2020) som løfter frem programmering som et pedagogisk verktøy for å utvikle elevers algoritmiske tenkning. Tradisjonelt sett er programmering et begrep for aktiviteten å skrive en programkode. Det vil si et sett instruksjoner datamaskiner og andre digitale hjelpemidler blir bedt om å utføre (Haraldsrud et al., 2020). I de senere årene har begrepet "koding" fått en større plass i dagligtalen (Sevik, 2016). Det er likevel viktig å skille mellom begrepene. Å programmere innebærer hele prosessen "fra å identifisere et problem og tenke ut mulige løsninger på problemet, til å skrive kode som kan forstås av en datamaskin, og å feilsøke og kontinuerlig forbedre denne koden" (Sevik, 2016, s. 9). Koding derimot, er selve aktiviteten å skrive instruksjoner ved hjelp av et programmeringsspråk som datamaskinen skal utføre. Koding er med andre ord en del av å programmere og av snevrere betydning. Programmering og algoritmisk tenkning er generelt ansett som separate ferdigheter, men programmering krever algoritmisk tenkning (Hickmott et al., 2018; Lye & Koh, 2014). Programmering er aktiviteten å skrive en kode som instruerer en datamaskin til å utføre en rekke handlinger, mens algoritmisk tenkning er en problemløsningsmetode. Undervisning av algoritmisk tenkning krever nødvendigvis ikke at elever lager program. Introduksjonen av programmering og algoritmisk tenkning i læreplanen har ført til en rekke utfordringer for lærere, siden læring av disse ferdighetene ikke vanligvis har vært en del av lærerutdanningen (Hickmott et al., 2018).

2.5 Tidligere forskning

2.5.1 Programmering og matematikk

Programmering i skolen er ikke et nytt fenomen. Allerede på 1960-tallet vokste det frem en interesse for å lære matematikk gjennom å skrive algoritmer. Den amerikanske matematikeren Seymour Papert utviklet sammen med kollegaer programmeringsspråket Logo i 1967, som ble satt inn i en brukervennlig programmeringsomgivelse. I Logo miljøet kunne elevene programmere en datamaskin til å styre en skilpadde på en dataskjerm ved hjelp av enkle kommandoer. Papert (1980) hevdet at datamaskiner i moderne utdanningssituasjoner som regel ble brukt til å gi elever tilpassede oppgaver, eller for å overlevere informasjon. Med andre ord programmerte datamaskinen barna. Tanken bak utviklingen av LOGO var derfor å snu på dette forholdet, slik at barnet kunne programmere datamaskinen. En annen grunnleggende filosofi i Logo var at elevene skulle få oppleve matematikken innenfor rammene av programmet, og ved å reflektere over det som ble skapt kunne de lære seg matematikk (Kilhamn et al., 2020). Læringsprosessene elevene gikk gjennom i løpet av slikt arbeid hevdet Papert (1980) var et ideal for kognitiv utvikling hos elever: "Powerful intellectual skills are developed in the process" (s. 60).

Utover 1980- og 1990-tallet ble det gjennomført flere undersøkelser internasjonalt, for å studere om programmering hadde overføringsverdi til andre områder. Papert (1980) hadde som nevnt tidligere, beskrevet hvordan datamaskiner kunne ha en grunnleggende effekt på barns intellektuelle og kognitive utvikling. Det viste seg at det ikke var så enkelt å finne støtte for denne idéen (Noss & Hoyles, 1996). Hvorvidt programmeringskompetanse bidro til bedret problemløsningskompetanse eller fordeler i andre emner, som var en ambisjon i utgangspunktet, lot seg heller ikke påvise (Koschmann, 1997). Yelland (1995) undersøkte Logo-miljøets potensial til å fungere som et matematisk miljø. Forskingen bestod av en litteraturgjennomgang av en mengde kvalitative og kvantitative studier som hadde

undersøkt forholdet mellom kognitive gevinster, problemløsning og elevers sosiale interaksjonsevne i matematikk. Resultatene var motstridende, for der noen studier kunne bevise den positive effekten av å inkludere Logo i matematikkundervisningen, feilet andre i forsøk på å oppdage forskjeller i elevenes problemløsningsferdigheter og matematikkprestasjoner etter at de hadde gjennomført Logos programmeringsprosjekt. Utover det nevnte, gikk kritikken også på at programmering ble ansett som tidkrevende og vanskelig, og at det var et behov for at lærere trengte mer didaktisk kunnskap om programmering for at undervisningen skulle utvikle elevene matematiske evner (Kilhamn et al., 2020).

Siden da har det skjedd en betydelig utvikling av programmeringsspråk, spesielt i forbindelse med et skifte fra tekstbasert til grafisk programmering. I tillegg har teknologitilgangen gjort ressurser lettere tilgjengelig gjennom internett. Ifølge Benton et al. (2017) kan mye nå gjøres for å rette opp mangelen på klarhet i tidligere arbeid om algoritmisk tenkning, rollen programmering spiller i utviklingen av den, og spesielt hvilken annen kunnskap som kan læres gjennom programmering. Papert så tidlig at elever kunne utvikle mikroverdener ved hjelp av grafiske figurer. Denne idéen ble plukket opp og videreutviklet i Scratch, et programmeringsspråk med grafiske blokker med en rekke funksjoner (Resnick et al., 2009). I løpet av den seneste tiden har Scratch blitt utbredt internasjonalt, men også i den norske skolen (se f.eks. Herheim & Severina, 2020). Benton et al. (2017) gjennomførte et forskningsprosjekt i England for aldergruppen 9-11, der det viser seg at det er mulig å undervise i programmering i Scratch og samtidig lære seg matematikk, men at det krever mye arbeid for å designe aktiviteter som inneholder viktige matematiske idéer og som bygger bro mellom matematikk og programmering.

2.5.2 Programmering i matematikkfaget fra et lærerperspektiv

Mozelius et al. (2019) har undersøkt svenske læreres holdninger til integreringen av programmering på mellomtrinnet. I deres studie kommer det frem at flertallet av lærerne diskuterer vanskelighetene med å undervise i programmering på grunn av manglende kompetanse og fagkunnskap. Andre utfordringer lærerne trekker frem er mangelen på tid, mangelen på materiell og verktøy og vanskeligheter med å tolke og forstå nasjonale styringsdokumenter. Til tross for den manglende kunnskapen i programmering, var lærerne positive til introduksjonen av programmering. Videre kommer det frem at elevtilfredshet og økt motivasjon og selvtillit er sett på som positive bivirkninger av å bruke programmering. Moeliuz et al. (2019) anbefaler til slutt å tilby lærere kurs i grunnleggende programmering så fort som mulig.

Når det gjelder forskning på læreres syn på programmering, har Pörn et al. (2021) gjennomført en spørreundersøkelse av finske barneskolelærere, relatert til landets nye matematikkpensum. Resultatet viser at programmering først og fremst er knyttet til å skrive, gi og følge instruksjoner. Den nest vanligste kategorien var knyttet til et syn på programmering som logisk tenkning og identifisering av mønstre. Videre kommer det frem at de færreste knytter programmering til spesifikt matematisk innhold. Fraværet av denne forbindelsen tolker Pörn et al. (2021) til å handle om at barneskolelærere ikke helt forstår samspillet mellom matematisk og algoritmisk innhold og læring. Denne tolkningen kan ses i sammenheng med Misfeldt et al.'s (2019) funn fra innsamlet data fra 133 svenske lærere, som viser at selv om lærerne var positive til å jobbe med programmering i matematikkundervisning, var det ikke alle som kunne se forholdet mellom de to eller relevansen av å gjøre det.

Det mest fremtredende synet på programmering som kommer frem i Pörn et al.'s studie (2021), stemmer overens med funn i Kilhamn et al.'s (2021) studie. 20 svenske matematikklærere deltok i studien, og selv om de ikke eksplisitt ble bedt om å definere programmering, impliserte flere av argumentene deres et syn på programmering som først å fremst for å lære om å gi og ta instruksjoner i små sekvensielle trinn. Videre poengterer Kilhamn et al. (2021) at det er en generisk ferdighet, ikke bare matematisk, men grunnleggende i matematikk. Totalt ble det identifisert fire argumenter for å undervise programmering i matematikk: for å utvikle algoritmisk tenkning, for å øke engasjement, for å lære matematikk og fordi det er et kraftig verktøy. Til slutt konkluderer de med at programmering både kan og kanskje ikke kan løfte matematikken. De skriver at det handler om hvorvidt lærere er villige til å se etter muligheter der programmering kan gjøre dette, og er åpne for endringer i matematisk tenkningspraksis.

Læreres undervisningspraksis har vært undersøkt av blant annet Kilhamn et al. (2020). Fra deres undersøkelser av 32 svenske lærergrupper som samarbeider om å planlegge og gjennomføre timer i programmering i matematikk i grunnskolen, kommer det frem at en tredjedel av programmeringsaktivitetene ikke var knyttet til noe tradisjonelt matematisk innhold. Videre i materialet er det identifisert fire sammenhenger mellom matematikk og programmering: 1) kun programmering, 2) matematikk som kontekst for programmering, 3) programmering som et verktøy for å effektivisere beregninger og 4) programmering som et verktøy for å utforske matematikk. I den forbindelse diskuterer de om programmering bør legges til bare matematikk når undervisning i programmering mangler kobling til tradisjonell matematikk. De peker også på at i deres materiale er det å lære seg hva programmering er og hvordan det fungerer, mer fremtredende enn spørsmålet om hva programmering kan brukes til og hvorfor.

I tidligere forskning på programmering i matematikkundervisning har fokuset ofte vært rettet mot elevenes læring heller enn læreres undervisning (f.eks. Clark-Wilson et al., 2014; Kilhamn et al., 2020). Ifølge Clark-Wilson et al. (2014) finnes det underutviklet kunnskap om hvordan læreres praksis påvirker bruken av teknologi, og hvordan lærere integrerer det i sitt profesjonelle yrke med formål om å lære elevene matematikk. De trekker frem PTK som en teori for å beskrive bruken og kunnskapen om teknologier hos lærere. Det finnes i tillegg lite kunnskap om læreres syn på programmering og undervisning av programmering i matematikk, og desto mindre om hvordan de håndterer de siste reformene i ulike land (Pörn et al., 2021). I Norden virker det å være Sverige og Finland som har forsket mest på programmering i matematikkundervisning fra et lærerperspektiv (f.eks. Kilhamn et al., 2020; Mozelius et al., 2019; Pörn et al., 2021). På bakgrunn av det tidligere forskning viser, ønsker jeg dermed å undersøke dette nærmere fra en norsk kontekst, gjennom beskrivelsene til denne studiens deltakere. Ved å benytte PTK rammeverket vil det være mulig å belyse flere aspekter som i forskning har fått mangelfull oppmerksomhet. Eksisterende forskning fra en norsk kontekst har ikke fokusert på læreres instrumentelle genesis, MTK og orienteringer, og sånn sett kan det forsterke denne studiens originalitet.

3 Metode

Dette kapitlet vil beskrive metodiske og praktiske sider ved gjennomføringen av forskningsprosjektet, deriblant metodiske valg som har vært ansett som hensiktsmessige. Metodologi handler om "alle vurderinger og tiltak som gjøres for å kunne svare på forskningsspørsmål i lys av forskningens hensikt" (Johnsen-Høines & Alrø, 2019, s. 36). Med utgangspunkt i definisjonen, vil det først redegjøres for kvalitativ metode og masteroppgavens vitenskapsteoretiske forankring. Videre gis en beskrivelse av innsamlingsmetode og utvalg. Deretter presenteres analyseprosessen og tilhørende valg av metoder. Avslutningsvis diskuteres oppgavens kvalitet opp mot begrepene pålitelighet, gyldighet og generaliserbarhet, før etiske betraktninger til slutt belyses.

3.1 Metodisk tilnærming

I samfunnsvitenskapelig forskning er det vanlig å skille mellom to ulike tradisjoner eller tilnærminger, en kvalitativ og en kvantitativ. Blant samfunnsforskere er det kjent at begge tilnærmingene er nødvendig for sammensatt forskning, og gunstig såfremt man har ressurser til det (Tjora, 2021). Dersom forskningen av ulike årsaker må begrenses til én tilnærming, vil for eksempel en kvalitativ tilnærming kunne forklare andre aspekter ved det samme fenomenet sammenliknet med en kvantitativ, og motsatt. Ifølge Tjora (2021) handler det om at man stiller andre type spørsmål eller stiller spørsmålene annerledes i de ulike tilnærmingene. Med bakgrunn i denne masteroppgavens problemstilling og forskningsspørsmål, ble kvalitativ metode en naturlig fremgangsmåte. Gjennom en kvalitativ metode søker man å oppnå en forståelse av sosiale fenomener (Thagaard, 2018). Det innebærer å undersøke hvordan personer opplever og reflekterer over sin situasjon. Ifølge Thagaard (2018) egner kvalitativ metode seg godt for å undersøke temaer det finnes lite forskning på fra før. Temaet for denne oppgaven, programmering i matematikkfaget, er et fenomen som har oppstått i Norge i løpet av de siste par årene, og som det finnes begrenset forskning på fra et lærerperspektiv. For å undersøke dette nærmere vil fokuset være på lærernes opplevelser og refleksjoner. Videre kan kvalitative metoder deles inn i fem kategorier, der intervju og deltakende observasjon er de mest brukte (Thagaard, 2018). I denne oppgaven er intervju anvendt som innsamlingsmetode. Valget mellom kvalitativ og kvantitativ metode er gjerne forbundet med hva man ønsker å undersøke og hvordan man posisjonerer seg vitenskapsteoretisk (Ringdal, 2013). Neste avsnitt vil derfor ta for seg denne studiens vitenskapsteoretiske forankring.

3.2 Vitenskapsteoretisk forankring

Den samme virkeligheten kan alltid beskrives ut fra ulike perspektiver, og derfor må jeg som forsker beskrive min posisjon og mitt perspektiv. Denne studien kan plasseres innenfor et fenomenologisk vitenskapsperspektiv. I kvalitativ forskning peker det fenomenologiske perspektivet på en interesse for å få "deltakere til å sette ord på hvordan de forstår sin verden, hvordan "ting" er og hvorfor de er slik de er – gjerne i forbindelse med et avgrenset fenomen" (Tjora, 2021, s. 31). I fenomenologien er altså menneskets måte å være på sentral. Det kan føle, det kan oppleve og erfare, og det kan utføre handlinger og ha meninger. Gjennom intervjuer har jeg fått innblikk i læreres beskrivelser av egne valg og handlinger, og deres opplevelser og erfaringer ved å undervise programmering i matematikkfaget. Med andre ord, jeg har fått tilgang på data som er

lærernes tolkning av situasjonen. Det har vært viktig for å senere kunne belyse aspekter ved deres PTK. Thagaard (2018) poengterer at det i fenomenologien er sentralt å forstå og beskrive hvordan deltakere opplever det aktuelle fenomenet. Mennesker har ulik bakgrunn, ulike interesser og kunnskap, og selv om de er med på det samme kan oppfattelsen være svært ulik. I det fenomenologiske intervjuet mener Kvale og Brinkmann (2015) at det asymmetriske maktforholdet mellom forsker og den intervjuede er for sjeldent adressert. Videre poengterer de at intervjuere derfor bør reflektere over hvordan ulike maktforhold kan påvirke kunnskapen som kommer frem fra intervjuene.

3.3 Datainnsamling

Datamaterialet i denne studien ble samlet inn gjennom lydopptak av fem intervjuer med matematikklærere i grunnskolen. Fire av lydopptakene ble gjort av digitale intervjuer på Teams, mens ett ble gjort av et fysisk intervju på deltakerens arbeidsplass. Lydopptakene har blitt transkribert og brukt i analyse. I det følgende vil jeg beskrive utvalget og rekrutteringen av deltakere til studien, og innsamlingsmetoden intervju.

3.3.1 Utvalg og rekruttering

Hovedregelen for utvalg i kvalitative intervjustudier er ifølge Tjora (2021) «at man velger deltakere som av ulike grunner vil kunne uttale seg på en reflektert måte (s. 145). Slike utvalg kalles strategiske. I mitt tilfelle foretok jeg en slik utvelgelse, basert på noen kriterier. Et kriterium var at lærerne underviste i matematikk. I utgangspunktet var det matematikk på barneskolen, men på grunn av vanskeligheter med å få tak i et tilstrekkelig antall deltakere til studie, endret jeg det til hele grunnskolen. Det kan være positivt i den forstand at fenomenet programmering i matematikkfaget kan belyses fra et større perspektiv, men samtidig forhindre en dypere forståelse på et mer innsnevret område. Et annet kriterium var at lærerne brukte eller hadde brukt programmeringsverktøy i matematikkundervisning. Bakgrunnen for det kriteriet var rammeverket PTK, som studien tar utgangspunkt i, og som fokuserer på blant annet verktøybruk i matematikk. Det kan derfor sies at deltakerne ble valgt utfra hensiktsmessighet heller enn representativitet.

Deretter begynte rekrutteringsprosessen. Det å få tak i deltakere som oppfylte kriteriene, viste seg å være ganske problematisk. Det var vanskelig å komme i direkte kontakt med lærere, da jeg i første omgang ringte administrasjonen til diverse skoler. To lærere jeg til slutt kom i kontakt med takket nei til å delta. Etter hvert så jeg meg derfor nødt til å spørre nettverket mitt om forslag til lærere og skoler jeg kunne kontakte. På denne måten skaffet jeg meg fem deltakere fra tre forskjellige barneskoler og to forskjellige ungdomsskoler, spredt utover Midt-Norge og Vestlandet. På grunn av vanskelighetene med å skaffe deltakere vil jeg i stor grad kategorisere utvalget som et tilgjengelighetsutvalg. I slike tilfeller peker Tjora (2021) på et vesentlig problem der man ikke har kontroll på hva personer som ikke deltok i studien ville sagt. Det kan føre til at man «mister» aspekter ved det som undersøkes. De fem deltakerne i studien hadde alle en tilsynelatende positiv holdning til programmering og var tilfeldigvis rundt samme alder. Kanskje kunne eldre lærere eller lærere med en mer kritisk holdning til programmering tilført studien annen innsikt og erfaring.

Nedenfor (tabell 1) følger en kort introduksjon av deltakerne i studien, detaljer om deres utdanning og undervisningserfaring. Av hensyn til deltakernes anonymitet har de blitt tildelt fiktive navn: Mats, Bjørn, Kristoffer, Anders og Silje.

	År med undervisningserfaring i matematikk	Deltatt på lærerkurs i programmering	Annen erfaring
Mats, 27 år	5	Nei	Erfaring med blokkprogrammering og diverse læringsplattformer med mulighet for programmering
Bjørn, 29 år	5	Ja	Erfaring med blokkprogrammering og flere fysiske programmeringsverktøy
Kristoffer, 30 år	8	Ja	Erfaring med blokk- og tekstprogrammering. Har også brukt regneark for simuleringer.
Anders, 27 år	5	Ja	Erfaring med blokk- og tekstprogrammering
Silje, 26 år	5	Nei	Erfaring med en programvare som er designet for undervisning og læring av databehandling på grunnskolenivå

Tabell 1. Introduksjon av deltakerne i studien.

3.3.2 Intervju

Intervju er den mest utbredte datagenereringsmetoden innenfor kvalitativ forskning, og særlig semistrukturerte- eller dybdeintervjuer er populære (Tjora, 2021). Problemstillingen i denne oppgaven legger til orde for å studere matematikklæreres meninger, holdninger og erfaringer, noe som stemmer godt overens med Tjoras (2021) hovedregel for å bruke semistrukturerte intervjuer. Utgangspunktet for slike intervjuer er en overordnet intervjuguide som er utarbeidet av forskeren på forhånd. Intervjuguiden inneholder forslag til spørsmål knyttet til bestemte temaer og springer ut av problemstillingen undersøkelsen skal belyse (Johannessen et al. (2016). Videre er semistrukturerte intervju basert på et fenomenologisk perspektiv, «hvor forskeren ønsker å forstå informantenes opplevelser samt hvordan informanten reflekterer over dette» (Spradley, 1979, sitert i Tjora, 2021, s. 128). Formålet med intervjuene i denne studien var å få utfyllende informasjon om lærernes forståelse av programmeringsbegrepet og dets rolle i matematikkfaget, og hvordan de underviser med programmering. Ved å benytte en semistrukturert intervjumetode vil det være mulig å oppnå dette, og dersom deltakerne skulle bringe nye temaer på banen tillater metodens struktur å endre på rekkefølgen av spørsmål (Johannesen et al., 2016). Den samme intervjuguiden ble brukt i samtlige intervju i denne studien.

Når man skal bruke intervju som metode er viktig å være forberedt på utfordringer og uforutsette situasjoner. Johannessen et al. (2016) peker på en utfordring knyttet til relasjonen mellom forskeren og den som blir intervjuet. De understreker viktigheten av å

være oppmerksom på at intervjueren kan påvirke deltakerens svar. En annen utfordring trekkes frem av Tjora (2021) og handler om at deltakeren kan forsøke å svare det de mener er «riktig» eller det de tror intervjueren vil høre. I tillegg er det viktig å være bevisst antakelsespotensialet som kan ligge i spørsmål utformet i intervjuguiden. Det kan ifølge Olsson (2003) være knyttet til deltakerens kunnskaper, innstilling og atferd, og kan gi deltakeren en følelse av å ikke forstå enkelte spørsmål eller ha kompetanse til å svare på dem.

Gjennomføring av intervju

Stedet for gjennomføring av intervju ble bestemt på deltakernes premisser. 4 av 5 intervjuer ble gjennomført digitalt på Teams, på bakgrunn av store avstander og koronasituasjonen på daværende tidspunkt. Det siste intervjuet ble gjennomført på deltakerens arbeidsplass. For å skape trygge rammer i intervjuene begynte jeg med å fortelle hva de skulle omhandle, før jeg deretter stilte enkle bakgrunnsspørsmål som: «hvor lenge har du jobbet som lærer» og «hvor lenge har du undervist i matematikk». Videre benyttet jeg den semistrukturerte intervjuguiden jeg utarbeidet i forkant av intervjuene (se vedlegg 3). En semistrukturert intervjuguide lar deltakerne få anledning til å uttale seg fritt om de forhåndsbestemte temaene, samtidig som den tillater å endre på rekkefølge og spørsmålsform. Etter oppvarmingsfasen med studiens formål og biografiske spørsmål, ble fokuset rettet mot studiens problemstilling og kjerne. Her ble hoveddelen delt inn i tre hovedtemaer: teknisk og pedagogisk kunnskap, refleksjon over egne erfaringer fra undervisning og til slutt personlige oppfatninger og holdninger knyttet til programmering. Hvert hovedtema inneholdt forslag til flere spørsmål jeg kunne stille. Til slutt ble det gitt for rom for at deltakerne kunne tilføye annet de måtte ha på hjertet. I etterkant av intervjuene har jeg innsett at jeg burde skrevet ned forslag til oppfølgingsspørsmål som kunne fått deltakerne til å utdype svar i større grad. Av transkripsjonene ser jeg at de spontane oppfølgingsspørsmålene jeg stilte ikke nødvendigvis hadde denne effekten. Derfor kunne samtale om enkelte temaer foregå mer i overflaten enn i dybden. Dette er likevel en erfaring jeg tar med meg videre fra prosessen. Som intervjuer forsøkte jeg å vise en interessert holdning ved å gi små nikk og bekræftende «ja» og «mhm».

3.4 Analysearbeid

I første fase av analysearbeidet transkriberte jeg lydopptakene av intervjuene. Det innebærer å strukturere intervjusamtaler fra muntlig til skriftlig form, slik at de er bedre egnet for analyse (Kvale & Brinkmann, 2015). Bryman (2016) løfter frem at man ved å transkribere sine egne intervjuer kommer tettere på dataene. Det kan for eksempel handle om at man begynner å identifisere nøkkeltemaer eller blir klar over forskjeller og likheter mellom intervjuene. Selv om oversettelse fra det talte til det skriftlige ord virker tilsynelatende uproblematisk, understreker Bryman (2016) at det ikke er en problemstilling som skal tas lett på. Jeg vil derfor beskrive hvordan transkripsjonene i denne studien er utført. Kvale og Brinkmann (2015) omtaler det som den ene grunnregelen for transkripsjon.

Lydopptakene ble transkribert så fort det lot seg gjøre etter hvert enkelt intervju. Fordi transkripsjonene ikke skulle brukes til noe som fokuserte på språk eller samtale, valgte jeg å utelate detaljerte transkripsjoner av for eksempel pauser, «eh»-er og intonasjoner. Likevel forsøkte jeg å ivareta meningsinnholdet til deltakeren etter beste evne ved å transkribere så nøyaktig som mulig. Transkripsjonene ble oversatt til bokmål for lesbarhetens skyld, men først og fremst av hensyn til deltakernes anonymitet.

3.4.1 Metode for analyse av datamaterialet

I analysen av de nå transkriberte intervjuene valgte jeg å benytte tematisk analyse, lansert av Braun og Clarke (2006). Dette er en grunnleggende form for analyse som ikke er bundet til et bestemt teoretisk rammeverk, men som innebærer at man ser etter *temaer* i dataene sine. Målet med en tematiske analyse er å gruppere utsagn fra gjennomførte intervju i mer generelle kategorier, eller temaer, som til sammen skal svare på forskningsspørsmålet for studien (Johannessen et al., 2018). Braun og Clarke (2006) definerer 6 ulike faser for tematisk analyse, som Johannessen et al. (2018) senere har forenklet og justert med hensyn til studenter. Jeg vil derfor benytte meg av deres versjon som består av fire ulike faser. Selv om Braun og Clarke (2018) hevder at fasene i tematisk analyse ikke må ses på som en lineær prosess, er jeg likevel enig med Johannessen et al. (2018) om at fasene kan fungere som nyttige holdepunkter for uerfarne forskere. Det kan gjøre det enklere å ha en systematisk tilnærming til analysen av datamaterialet. Videre vil jeg redegjøre for disse og vise hvilke beslutninger jeg har tatt underveis i denne studiens analyseprosess.

Fase 1: Forberedelse

Det første steget i tematisk analyse handler om å bli kjent med datamaterialet som er samlet inn. For min del handlet det om at jeg leste grundig gjennom de fem transkriberte intervjuene, for å få en viss oversikt over ulike idéer eller mønstre som så ut til å fornye eller gjenta seg. Før intervjuene hadde jeg allerede lest en del teori jeg ønsket å benytte i oppgaven, og dette påvirket hvordan jeg så for meg å senere kategorisere datamaterialet. På dette tidspunktet i prosessen vurderte jeg også at noen av intervjuene nok kunne legge føringer for hva jeg anså som interessante funn, og at det var viktig å være bevisst på dette i det videre arbeidet.

Fase 2: Koding

Etter å ha blitt kjent med datamaterialet beveger man seg over i fase 2, som innebærer koding av informasjonen fra materialet. Johannessen et al. (2018) definerer det til å handle om «å fremheve og sette ord på viktige poenger i dataene våre» (s. 284). I denne prosessen benyttet jeg de tre teknikkene Johannesen et al. (2018) anbefaler: markering, stikkordoppsummering og notere refleksjoner som dukker opp underveis. Deretter ordnet jeg en tabell for hvert av intervjuene hvor jeg plasserte markerte utdrag fra transkripsjonene inn i de forhåndsbestemte hovedkategoriene. Noen eksempler på temaene som kommer fra dataanalysen er gitt i figur 4.

Kode	Tema	Eksempel fra datamaterialet
Programmering blir kunstig på barne- og ungdomsskolen	Kunnskap om læreplaninnhold (MDKT)	Bjørn: (...) jeg tror ofte det blir veldig kunstig på barne- og ungdomsskolen fordi når jeg leser læreplanmålene, spesielt de på åttende trinn, så leser jeg det ikke som et matematikkmål. Jeg leser det som et mål der det står «lær deg programmering».
Vanskelig å bruke programmering til noe skolefaglig	Lærerbekymring om hvordan knytte programmering til matematikk (profesjonell instrumentell genesis)	Anders: Det vanskeligste har kanskje vært å få knytta det mot undervisning da, synes jeg. Det er liksom artig å sitte med programmering og forskjellige animasjoner og spill og sånn, men å få kobla det opp mot kompetansemålene og få brukt det til noe skolefaglig, for å si det sånn, det har kanskje vært det vanskeligste.
Hive seg ut i det [programmering]	Lærerselvtilitt i å bruke programmering i undervisning (orientering)	Kristoffer: Det som jeg har gjort, som jeg nevnte tidligere, det er at det er ikke alt jeg kan, men jeg hiver meg ut i det.

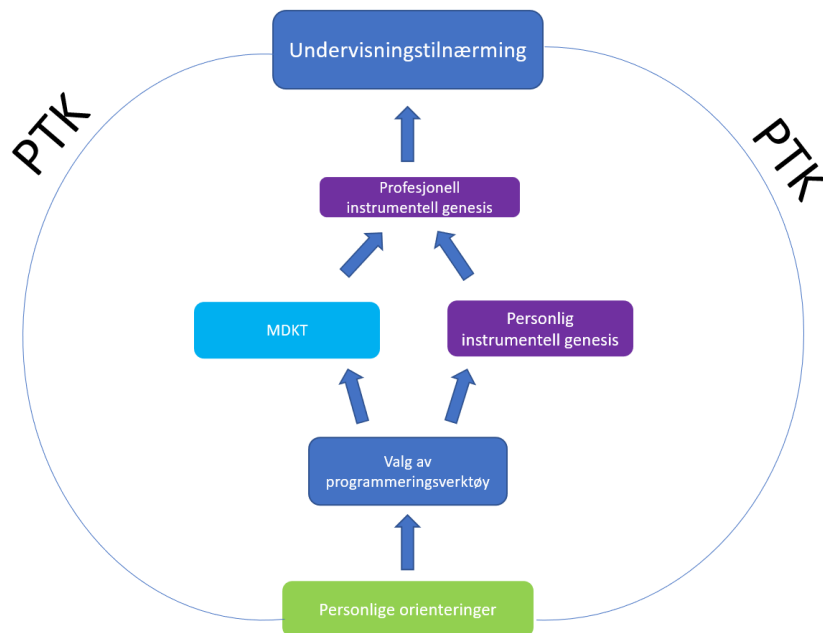
Tabell 2. Eksempler på datakoding og temaer.

3. Kategorisering

Den tredje fasen i tematisk analyse er kategorisering. Etter at materialet er ferdig kodet, skal det settes sammen til en større helhet. Det innebærer å sortere dataene i overordnede kategorier eller temaer, som er funnene som senere skal presenteres i resultatdelen av oppgaven (Johannessen et al., 2018). Etter kodingen av datamaterialet i fase 2 satt jeg igjen 275 empirinære koder. Gjennom denne prosessen fikk jeg en klarere forståelse for innholdet i empirien. Neste steg var deretter å systematisere kodene i ulike kategorier for å samle koder som kunne virke å handle om det samme. Følgende tema fremstod som gjennomgående i datamaterialet: holdninger og oppfatninger blant deltakerne, undervisningsaspekter og bruken av programmeringsspråk og verktøy. Med utgangspunkt i rammeverket PTK, ble kodene systematisert i kategorier som belyste aspekter ved PTK. Det resulterte i tre hovedkategorier: (1) Personlige lærerorienteringer, (2) Matematisk digital kunnskap om undervisning og (3) Dobbelt instrumentell genesis. Hver av hovedkategoriene fikk tilhørende underkategorier som også samsvarte med teori.

For å belyse problemstillingen har jeg altså sett etter elementer i PTK rammeverket og dets tilhørende teoretiske komponenter og elementer i utsagn fra deltakerne som kan

knyttes sammen. Dette danner grunnlag for videre resultater og diskusjon i oppgaven. En visuell fremstilling av denne sammenhengen er illustrert nedenfor:



Figur 6. Grafisk fremstilling av hvordan forskningsspørsmålene henger sammen med problemstillingen.

4. Rapportering

Til slutt skal temaene, og altså funnene, rapporteres. Det innebærer å skrive frem resultatet på forskningsspørsmålet/ene for studien. Dette blir gjort i kapittel 4.

3.5 Forskningens kvalitet

I kvantitativ forskning er det vanlig å bruke begrepene pålitelighet (reliabilitet), gyldighet (validitet) og generaliserbarhet for å etablere og vurdere kvaliteten på forskningen. De senere årene har det derimot vært diskutert om disse begrepene har den samme relevansen for kvalitativ forskning (Bryman, 2016). Det er spørsmålet om måling som kan være problematisk dersom begrepene skal være av samme betydning i de to forskningsdisiplinene. Likevel er begrepene pålitelighet, gyldighet og generaliserbarhet godt innarbeidede kriterier på forskning generelt, og innføring av nye begreper er sett på som lite hensiktsmessig ifølge Tjora (2021). For at begrepene skal gi mening for å vurdere kvalitativ forskning, må de imidlertid tillegges en annen betydning enn de har i kvantitativ forskning. Videre vil jeg vurdere kvaliteten på min egen kvalitative forskning i lys av Tjoras (2021) forståelse av begrepene.

3.5.1 Pålitelighet

Forskningens pålitelighet handler om at det er en intern sammenheng gjennom hele forskningsprosessen (Tjora, 2021). Ved å redegjøre for forhold som gjelder internt i et enkelt forskningsprosjekt, og for alle valg som er gjort underveis, vil påliteligheten til en studie kunne styrkes. Et annet sentralt forhold ved en studies pålitelighet er forskerens posisjon og forforståelse. I kvalitativ forskning vil det være vanskelig for forskeren å forholde seg nøytralt gjennom hele forskningsprosessen. Det er heller ikke et mål, men ifølge Tjora (2021) er det viktig å være bevisst sin egen forforståelse og være åpen for at

den kan justeres underveis. I mitt tilfelle er forforståelsen min om temaet programmering i skolen begrenset. Jeg har lite erfaring med programmering generelt, både fra fritiden og fra undervisningssammenheng. I tillegg er det først i løpet av det siste året jeg har utviklet en interesse for temaet. Likevel kan informasjon jeg har lest eller hørt påvirke hvordan jeg ordlegger meg om temaet, og hvilke spørsmål jeg stiller eller eventuelt ikke stiller informantene i denne studien.

For å styrke påliteligheten i denne masteroppgaven, har jeg forsøkt å beskrive hvordan jeg har kommet frem til resultatene i den ferdigstilte rapporten. En måte jeg har sikret at dette kommer frem på en ærlig og redelig måte, er ved å bruke sitater fra intervjuene for å bekrefte funn. Deltakerne ble rekruttert gjennom et strategisk utvalg for å sikre at de har erfaring med å bruke programmering i matematikkundervisning. I forkant av intervjuene gjennomførte jeg et pilotintervju for å få en bekreftelse på at jeg kunne få svar på det jeg ville undersøke. Videre ble intervjuguiden gjennomgått med masterveileder for å sørge for at spørsmålene var i samsvar med det teoretiske rammeverket jeg ønsket å ta utgangspunkt i. Den samme intervjuguiden ble brukt i alle intervjuene, for å ha muligheten til å sammenligne de med hverandre. Av de fem deltakerne jeg intervjuet, hadde jeg kjennskap til en av de fra før. Det trenger nødvendigvis ikke være negativt, men det er viktig å være klar over at en relasjon kan påvirke intervjusituasjonen. Dette var for så vidt det eneste intervjuet som ble gjennomført ansikt til ansikt, noe som gjorde det enklere å få med seg det som ble sagt i samspill med ansiktsuttrykk, kroppsspråk og gestikuleringer. De fire andre intervjuene ble gjennomført på teams av praktiske årsaker. Når det gjelder universitets retningslinjer angående masterforskning, så har jeg brukt godkjente lagringsmetoder og universitets egen diktafon. Ved å formidle valgene jeg har tatt i forskningsprosessen ønsker jeg å invitere lesere til å selv ta stilling til forskningens kvalitet. Dette er forbundet med begrepene *transparens* eller *gjennomsiktighet*, og danner grunnlaget for en studies pålitelighet (Tjora, 2021).

3.5.2 Gyldighet

Begrepet gyldighet kan ifølge Tjora (2021) forklares som den logiske sammenhengen mellom et prosjekts utforming og dets funn. Det handler om å reise spørsmål om hvorvidt det er en sammenheng mellom prosjektets design og resultater, og de forskningsspørsmålene man forsøker å finne svar på. Valg av metode i en studie bør altså samsvare med problemstillinger og forskningsspørsmål dersom gyldigheten skal styrkes. Andre faktorer som er med på å styrke gyldigheten av en studie er dens faglige forankring, og at den bygger på relevant forskning og litteratur.

Jeg har forsøkt å styrke gyldigheten til denne forskningen ved å forholde meg til relevante teorier og perspektiver, og til tidligere forskning om samme tema. Oppgaven tar utgangspunkt i et teoretisk rammeverk som er spesielt designet for å undersøke læreres kunnskap om å integrere digitale hjelpemidler i matematikkundervisning, noe som samsvarer med valg av problemstilling. Videre er intervjuguiden utarbeidet med utgangspunkt i samme rammeverk, og med inspirasjon fra lignende undersøkelser om samme tema der innsamlingsmetoden også var kvalitative intervju. Jeg intervjuet fem matematikklærere for å forsøke å finne svar på forskningsspørsmålene mine, men av praktiske årsaker måtte jeg vike fra den opprinnelige planen om å inkludere observasjon. Fraværet av direkte data av hvordan læreren underviser svekker forskningens triangulering. Datamaterialet gir kun tilgang til det lærerne sier at de gjør, men ikke nødvendigvis hva de faktisk gjør. Forskningsprosessen går ikke alltid slik man ønsker, men det er viktig å være bevisst hvilke implikasjoner valg kan ha for resultatene. En annen svakhet ved studiens gyldighet vil være at jeg har utført transkribering og tolkning av

datamaterialet alene. Jeg har derfor, etter beste evne, forsøkt å være kritisk til mine egne tolkninger og stille kontrollspørsmål underveis om for eksempel datamaterialet er tilstrekkelig til å trekke de slutningene jeg har gjort. Arbeidet med denne masteroppgaven er et av mine første møter med forskning på egenhånd, og min manglende erfaring har påvirket valg gjennom hele prosessen. Jeg har likevel forsøkt å veie opp for dette ved å lese meg opp på faglitteratur knyttet til forskningsprosessen, og ved å lese annen forskning for inspirasjon.

3.5.3 Generaliserbarhet

I forskning generelt etterstreber man som regel en eller annen form for generalisering (Tjora, 2021). Tjora (2021) forklarer at det handler om hvorvidt forskningen har relevans utover de enhetene som faktisk er undersøkt. Hvordan man vurderer generaliserbarheten til forskning varierer ut ifra om designet er kvalitativt eller kvantitativt. Generaliserbarhet i kvalitativ forskning har lenge vært et omdiskutert tema. Payne og Williams (2005) har i en kartleggingsstudie av hvordan kvalitative forskere generaliserer resultatene sine, funnet at de færreste forskere diskuterer grunnlaget for sine generaliseringer og at de nærmest produseres ubevisst. Videre problematiserer de fraværet av en slik diskusjon, og mener det er en fare for at leseren selv kan anta et generaliseringsbehov som ikke er der. Tjora (2021) foreslår hvordan dette kan unngås: «En mer eksplisitt redegjørelse for generaliseringens grenser i presentasjonen av kvalitativ forskning kan begrense overdreven, men ikke eksplisitt uttalt, tro på generaliserbarhet» (s. 270). Jeg vil derfor i det følgende redegjøre for generaliseringens grenser i min forskning.

I utvalgsprosessen benyttet jeg et strategisk- og tilgjengelighetsutvalg. Dette gjør det vanskelig å argumentere for moderat generalisering, fordi alle enhetene i en populasjon ikke hadde samme sannsynlighet for å bli med i forskningen. Dersom alle matematikklærere i landet ville hatt en mulig sannsynlighet for å delta, kunne resultatet vært generalisert til å gjelde hele populasjonen matematikklærere i Norge. Utvalget i denne forskningen bestod av fem matematikklærere. Det lave antallet lærere som er representert gjør det vanskelig å generalisere funn til å gjelde andre matematikklærere. Det er kun de aktuelle lærerens individuelle opplevelser og beskrivelser av disse denne studien gir innblikk i. På en annen side kan man argumentere for at tolkningen som er utviklet i denne konteksten kan være nyttig og relevant i andre kontekster. Det handler om at kunnskapsoverføring kan være et sentralt aspekt ved generalisering (Thagaard 2013; Kvale & Brinkmann, 2015). For matematikklærerne som deltok kan studien ha en verdi eksempelvis som et bidrag til refleksjon og bevissthet over egen undervisning. Videre kan studien bidra til kunnskap for lesere, som ved å se funnene i lys av relevant teori og forskning kan vurdere om de er nyttige for deres egen situasjon.

3.6 Etske betraktninger

På lik linje som med annen forskning, må man også i masterforskning forholde seg til etiske retningslinjer og prinsipper (Johannessen et al., 2006). Den nasjonale forskningsetiske komité for samfunnsvitenskap og humaniora (NESH) viser til retningslinjer som skal fungere rådgivende for å sikre at forskning organiseres og utøves forsvarlig (se NESH, 2021). Videre påpekes det at retningslinjene bør ligge til grunn gjennom hele forskningsprosessen. Jeg har derfor tatt flere etiske hensyn underveis i arbeidet med denne masteroppgaven, både ved planlegging, gjennomføring, behandling og gjengivelse av datamaterialet.

Før prosessen kunne begynne søkte jeg Norsk Senter for Forskningsdata (NSD) om godkjenning for det tiltenkte forskningsprosjektet. Dette var nødvendig ettersom

forskningsdesignet mitt gikk ut på å intervju deltakere, og lydopptak av disse kunne inneholde personidentifiserbart materiale. Den godkjente søknaden finnes som vedlegg 1. I forkant av intervjuene ble det utformet og utstedt et informasjonsskriv med samtykkeskjema til deltakerne (se vedlegg 2). Dette ble gjort for å forsikre at deltakerne var klar over hva forskningsprosjektet gikk ut på og hva en deltakelse ville innebære (se vedlegg). Skrivet poengterte i tillegg at deltakeren når som helst kunne trekke seg uten behov for begrunnelse, og det ikke ville medføre negative konsekvenser. Etter gjennomlesing kunne de signere på om de ga et frivillig samtykke til å delta dersom de ønsket det. Lydopptakene av de fem deltakerne som samtykket til å delta, ble umiddelbart kryptert på telefonen. Dette skjedde gjennom appen nettskjema-diktafon som er et anbefalt innsamlingsverktøy av NTNU. På bakgrunn av medstudenters opplevelser med at appen ikke alltid lagret intervjuene deres, besluttet jeg å bruke en ekstern diktafon i tillegg. Etter transkriberingen ble opptakene slettet fra denne. I gjengivelsen av datamaterialet ble deltakernes navn byttet ut med fiktive navn for å ivareta deres anonymitet. Hvordan lærerne ble gjengitt og fremstilt er også tatt i betraktning. Det var kun jeg som hadde tilgang til de anonymiserte transkripsjonene.

4 Presentasjon av funn

I dette kapitlet presenteres funn fra datamaterialet i henhold til tre av faktorene i Thomas og Palmers (2014) teoretiske rammeverk PTK. Gjennom analysearbeid med koding og kategorisering, har utsagn fra deltakerne ført til et relevant materiale som kan svare på problemstillingen: *hva kjennetegner læreres PTK ved bruk av programmeringsverktøy i matematikkfaget i grunnskolen?* Den induktive og teoretiske tilnærmingen til kodingen, resulterte i tre brede hovedkategorier med tilhørende underkategorier som til sammen skal svare på de tre forskningsspørsmålene. 4.1 tar for seg forskningsspørsmål 1, mens 4.2 og 4.3 tar for seg henholdsvis forskningsspørsmål 2 og 3. Til sammen danner kategoriene grunnlaget for oppbyggingen av kapitlet.

4.1 Personlige lærerorienteringer

Jeg har valgt å inkludere deltakernes beskrivelser av programmeringsbegrepet under denne kategorien. Den teoretiske komponenten personlige lærerorienteringer sier ingenting om hva lærere legger i teknologiske begrep, men heller hvilken verdi teknologi kan ha i matematikkfaget. Jeg mener likevel det er relevant å inkludere for å gi en beskrivelse av lærernes utgangspunkt.

4.1.1 Beskrivelse av programmeringsbegrepet

Alle lærerne virket å dele den samme grunnoppfatningen av programmeringsbegrepet. De beskrev begrepet, med noen variasjoner, som å få noe til å gjøre noe ved hjelp av ulike steg. Noen lærere bruker begrepene instruksjoner og algoritmer i sine beskrivelser. Det ble også trukket frem at de stegvise instruksjonene kan resultere i et ønsket mål. Dette kom blant annet til uttrykk i intervju med Silje [merk: navnene er pseudonymer], hvor hun beskrev programmering som å følge en oppskrift.

Intervjuer: Kan du beskrive hva du legger i begrepet programmering?

Silje: Ja, for meg er det på en måte det å skulle følge en oppskrift på et vis. Følge noen steg frem til et ønsket mål. Det er vel egentlig det jeg ser på som programmering.

Siljes beskrivelse knytter programmering til det hverdagslige begrepet oppskrift. Hun utdyper videre at å følge oppskrifter er viktig i vårt daglige liv, for å få de resultatene vi ønsker. Hun trekker i den forbindelse frem et eksempel om å lage middag ved hjelp av oppskrifter. Silje forteller at dette er noe hun pleier å fortelle elevene sine på 2.trinn, og hennes beskrivelse av programmeringsbegrepet kan være preget av alderstrinnet hun underviser på. En annen lærer, Kristoffer, trekker også frem elevene sine når han forklarer hva han legger i begrepet:

Det er rett og slett å få en gjenstand, en ting, et eller annet til å gjøre det du ønsker at den skal gjøre. Enkelt forklart. Det er hvert fall sånn jeg prøver å forklare det til elevene mine. Og det virker som de forstår det da.

Programmering er å få noe til å gjøre noe, ut ifra ønskene man har for hva som skal skje, ifølge Kristoffer. Han underviser også, i likhet med Silje, på barneskolen og opplever at elevene virker å henge med på denne forklaringen. Bjørn på sin side, som til vanlig underviser på ungdomsskolen, bruker en del andre begreper i sine beskrivelser. Likevel forbinder han også programmering til å få noe til å gjøre noe. På spørsmål om hva han

legger i begrepet, trekker han først frem at koding og programmering brukes om hverandre, men at han forstår de som to ulike begreper:

Når jeg tenker koding så blander jeg det kanskje inn med at du koder en ting til at det her betyr ... Den her tingen tilsvarer koden. Mens programmering blir en samling av alle algoritmene og instruksjonene som gjør at datamaskinen gjør det du ønsker den skal gjøre.

Bjørn forteller at koding kanskje handler om å tillegge noe betydning ved hjelp av en kode, mens programmering er samlingen av det som får datamaskinen til å utføre noe. Han knytter programmering med engang til datamaskiner, og bruker algoritmebegrepet og instruksjoner for å forklare hva denne samlingen inneholder. Den umiddelbare tilknytningen til datamaskiner gjør også Anders:

Jeg tenker jo det handler om å på en måte få datamaskinen til å gjøre noen instruksjoner i den rekkefølgen du bestemmer, også blir det gjort akkurat sånn. Ja, så det er liksom tanken jeg har, det å utføre, datamaskinen skal utføre det du tenker da.

Han forteller at rekkefølgen på instruksjonene har noe å si for hvordan resultatet av programmeringen blir. I tillegg kommer det frem at programmering handler om å overføre det man tenker en datamaskin skal gjøre, til datamaskinen gjennom instruksjoner i en bestemt rekkefølge. Videre påpeker Anders at vi ved hjelp av datamaskiner gjenta instruksjoner uendelig mange ganger, noe som gjør de svært effektive om de er skrevet inn på riktig måte. Mats forteller at befinner seg i læringsfasen av hva det egentlig vil si å programmere. Derfor hevder han at han på nåværende tidspunkt har en snever forståelse av begrepet. Han kommer deretter inn på lignende beskrivelser som de andre lærerne, og nevner i tillegg assosiasjoner han får til begrepet:

Enn så lenge vil det [programmering] være det å lage noe som får noe til å skje egentlig. Og når jeg hører programmering så er det første jeg tenker på micro:bit. (...) og de blokkene, og litt scripting, koding generelt.

Selv om Mats mener han har en snever forståelse av programmeringsbegrepet, virker det likevel som han kjennskap til en del aspekter ved det. Både blokker, scripting og koding er elementer som kan inngå i aktiviteten ved å programmere. I sum ser vi altså at programmeringsbegrepet forstås som varianter av å få noe til å gjøre noe. Det nevnes ting, gjenstander og datamaskiner. Noen beskrivelser er mer detaljerte enn andre og inneholder ord som algoritme, instruksjoner og rekkefølge. To av lærerne forklarer sin oppfattelse av begrepet på måter de ville brukt for elevene sine.

4.1.2 Verdien av å bruke programmering i matematikk

I forsøket på å undersøke hvilken verdi programmering ser ut til å ha i matematikkfaget for lærerne, kommer flere perspektiver til syne. Verdien av programmering i matematikkfaget ses både fra et teknologisk fremtidsperspektiv, et pedagogisk perspektiv og et matematisk perspektiv. Flertallet av lærerne forteller at programmering er viktig med tanke på å holde tritt med samfunnsutviklingen og for å møte etterspørselen i det fremtidige arbeidsmarkedet. Her er noen eksempler på dette:

Intervjuer: Hvorfor bør elever lære programmering i matematikk?

«Jeg tenker jo at det er fordi samfunnet blir mer og mer teknologisk sånn i utgangspunktet. (...) Mye foregår på skjerm og det finnes jo uendelig med ting som er programmert på en eller annen måte i samfunnet» (Mats).

«Det er jo det at det er stor etterspørsel etter det, eller hvert fall økende etterspørsel i veldig mange forskjellige yrker» (Kristoffer).

«(...) det at mange mest sannsynlig vil få bruk for det senere i arbeidslivet» (Anders).

Verdien av å bruke programmering i matematikk fra et pedagogisk perspektiv, knyttes av lærerne først og fremst til effekten det har på elevenes mestringsfølelse. Dette kommer blant annet til uttrykk i intervju med Bjørn, som forteller hvordan programmering kan treffe flere elevtyper:

Jeg vet ikke om det har noe med verktøyene eller undervisninga, eller om det handler om essensen programmering og at det er på datamaskin og er noe annerledes enn vanlig undervisning. Som jeg sier, at det er mer ... De som vanligvis er engasjert er ikke mer engasjert, men du treffer andre elevtyper rett og slett.

Bjørn vet ikke akkurat hva det er som gjør at programmering treffer andre elevtyper, men at variasjonen fra vanlig undervisning kan fange interessen hos flere elever. Anders forteller at digital teknologi, som programmering, kan gjøre at elever får vist andre typer kompetanser i matematikkfaget:

(...) hvert fall det jeg har lagt merke til, er at det trenger ikke være de som er faglig sterkest som gjør det best der [i programmering], kanskje ofte litt motsatt. Det er jo artig at de som ikke gjør det best i matematikk til vanlig, kanskje er veldig gode til digitale ting.

Fra et matematisk perspektiv, er utforskning og problemløsning to områder flere av lærerne mener programmering kan være et positivt supplement. Det kommer eksempelvis frem i det Silje og Kristoffer forteller om fordeler ved å bruke programmering i matematikkundervisningen sin:

Jeg ønsker jo egentlig at elevene skal se mer av det utforskende i den vanlige matematikken og. Så det skal bli mer spennende og de skal prøve seg litt mer frem da. Så det syns jeg er veldig gode ting med programmering som jeg tar med meg inn i matematikk da (Silje).

Det som er gøy med dette er vel den gleden av å få til ting, utforske, undringa, hvorfor skjer det (...). Det klarer å skape en del spørsmål rundt dette her da som jeg syns er bra. Den undringsdelen, det tror jeg er det beste med programmering (Kristoffer).

Mine analyser avdekker at lærerne ser verdien av å bruke programmering i matematikkundervisning, men at det nødvendigvis ikke bare knyttes til utvikling av matematisk kunnskap. Flere av lærerne ser også verdien av å bruke programmering for å gi elevene mestringsfølelse og nødvendig digital kunnskap for fremtidige yrker.

4.1.3 Selvtillit til å bruke programmering

Læreres selvtillit til å bruke teknologi i matematikkundervisning er av Thomas og Palmer (2014) løftet frem som en vesentlig del av deres orienteringer. Lærerne ble ikke stilt direkte spørsmål om deres selvtillit, men av datamaterialet kommer den likevel til syne i flere av deres refleksjoner. Hos Silje kommer det til syne på denne måten:

Jeg tenker spesielt for de som er av den litt mer eldre garden, som synes programmering høres kjempeskummelt og vanskelig ut. Også er det jo egentlig ikke det. Det er jo mange forskjellige måter å gjøre det på.

Silje virker å ha gode opplevelser med å bruke programmering, og ser flere muligheter for hvordan hun kan bruke det. Mats forteller at det er enklere å bruke programmering for de som har kjennskap til det fra før, slik som han selv, mens det for Kristoffer handler om å tørre og hive seg ut i ukjent farvann sammen med elevene:

Det jeg har gjort, som jeg nevnte tidligere, det er at det er ikke alt jeg kan, men jeg hiver meg ut i det. Jeg har lært masse av elevene for eksempel. Å våge og tørre og spille på lag med elevene og utforske sammen med dem. Det tror jeg man vinner på på mange forskjellige måter og kompetansemål egentlig i alle fag.

Kristoffer innrømmer at det er ting ved programmering han ikke kan, men å tørre og utforske sammen med elevene kan ha flere gevinster. Av analysen virker det som flere av

lærerne har god selvtillit til å benytte programmering i undervisningen sin, på tross av at de selv befinner seg i læringsfasen.

4.2 Matematisk digital kunnskap om undervisning

Å undersøke hvilken kunnskap lærere har er nødvendigvis ikke enkelt. Jeg har derfor forsøkt å tilnærme meg utfordringen gjennom indirekte spørsmål, og deretter hente ut informasjon om hvilke kunnskaper som kommer til syne. Min analyse avdekker følgende aspekter ved deres kunnskap: kunnskap om læreplaninnhold, undervisningsstrategier og bevissthet om elevene og deres tenkning.

4.2.1 Kunnskap om læreplaninnhold

Programmering ble en del av læreplanen i matematikkfaget høsten 2020. I tre av intervjuene kommer det frem at lærerne har kunnskap om læreplaninnhold for trinnene de underviser på, selv om noen uttrykker at enkelte kompetansemål kan være vanskelig å tolke. Dette kommer blant annet til uttrykk i intervju med Bjørn, hvor han oppga at kompetansemål med programmering i matematikkfaget kan oppleves kunstig på barne- og ungdomsskolen:

(...) jeg tror ofte det blir veldig kunstig på barne- og ungdomsskolen fordi når jeg leser læreplanmålene, spesielt de på åttende trinn, så leser jeg det ikke som et matematikkmål. Jeg leser det som et mål der det står «lær deg programmering».

Læreplanmålene for programmering i matematikk tolker Bjørn i retning av å lære programmering og ikke lære matematikk. Hans beskrivelse gir uttrykk for at intensjonen med å bruke programmering i matematikkfaget ikke kommer tydelig frem i kompetansemålformuleringene. Videre forteller han at dersom intensjonen er å forstå matematikken bedre gjennom programmering, må man ha inngående kunnskap om programmering i utgangspunktet. En realitet langt unna utgangspunktet i dag, ifølge Bjørn. Anders har utarbeidet undervisningsopplegg i forbindelse med masteroppgaven sin, der han tok utgangspunkt i kompetansemålet for sjette trinn:

I oppgaven min så hadde jeg jo om geometriske figurer på sjette trinn, siden det var det første kompetansemålet der. Og da fikk jeg egentlig masse gode tanker rundt ulike figurer, vinkelsum og hva en sirkel er. Om de [elevene] klarte å lage det i Scratch for eksempel. Så jeg føler jo at akkurat i geometri hvert fall, så har jeg fått brukt det til noe faglig.

Anders viser at han vet hva kompetansemålet sier, og forteller om en følelse av å ha klart å knytte programmering direkte til matematikken gjennom geometri. Videre kommer det til uttrykk at andre områder i matematikken har vært vanskeligere å nærme seg med programmering. Silje forteller fra ståsted på andre trinn:

Nei, det som er kompetansemålet for andre trinn er vel det å følge og lage regler og instruksjoner eller noe, i spill og aktiviteter og sånne ting. Så det er jo på en måte litt det vi gjør, men det jo kanskje ikke akkurat det man gjør i matematikk da egentlig. Da får man jo reglene og instruksjonene levert, så gjør du oppgaven på en måte.

Også Silje viser at hun vet hva kompetansemålet for trinnet hun underviser på sier, og at de gjør noe lignende i hennes undervisning. Hun gir i tillegg uttrykk for at det de gjør ikke nødvendigvis er det man gjør i matematikk. Kompetansemålet handler om det å følge og lage regler og instruksjoner, mens i matematikk får elevene det levert, hevder Silje. I sum ser vi altså at lærerne har innsikt i læreplaninnholdet i matematikk, men at det åpner for ulike tolkningsmuligheter.

4.2.2 Undervisningstilnærming

Lærerne forteller om flere forskjellige undervisningstilnærminger til bruk av programmeringsverktøy i matematikk. I introduksjonsfasen forteller 4 av 5 lærere at de bruker seg selv, elevene eller aktiviteter på ark for å demonstrere i korte trekk hva programmering handler om. Såkalt analog programmering uten datamaskin. Bjørn forklarer en aktivitet han pleier å bruke og hvilke fordeler den kan ha:

(...) Du har et par der den ene ser hva jeg skriver på tavla og den andre som ikke ser tavla. Og så skal den som ser tavla prøve og forklare hva jeg tegner på tavla til den som har ryggen til. Og så skal den personen tegne da. Det går heller aldri, for de ender opp med arket feil vei eller at de er upresise. (...) Så det skaper noen gode refleksjonssamtaler med hvorfor presise beskjeder og instruksjoner er viktig da. Og at det du tenker er logisk ikke nødvendigvis er logisk for den du jobber med.

Analog programmering kan illustrere nødvendigheten av presise instruksjoner i programmering, ifølge Bjørn. Kristoffer er også opptatt av å få frem det samme poenget for elevene sine, og forteller at han bruker seg selv for å illustrere det:

Hvis jeg skal begynne med programmering med en helt ny klasse som aldri har brukt det før, så bruker jeg meg selv. Der jeg lager en liten løype i klasserommet, så skal de gi meg kommandoer på hvor jeg skal for at jeg skal sette meg i den stolen. Og da forstår de ganske kjapt at «oi, jeg må akkurat ...» eller «jeg må si presist akkurat det jeg vil at han skal gjøre, hvis jeg sier en liten feil så blir det helt galt».

Det Kristoffer forteller viser at også hans erfaring er at analog programmering kan få elevene til å se nødvendigheten av presise instruksjoner. Når elevene etter hvert har fått en innføring i hva programmering er, forteller flere av lærerne at de gir elevene verktøy de kan utforske på egenhånd eller i par. Det kommer eksempelvis til uttrykk i det Mats forteller:

(...) Hvis du gir de verktøyene til det så er de veldig selvdrevne. De er flinke til å utforske selv. Spesielt når de sitter i par, da prøver de ut masse forskjellig selv også. Så oppdager de at «oi, jeg kan programmere lysene på bilen så de blinker i forskjellige farger, eller i regnbuemønstre, eller jeg kan ta annen hver sånn og sånn».

Mats opplever at elevene gjennom utforsking oppdager ulike funksjoner ved programmeringsverktøy. Det forutsetter imidlertid at disse er tilgjengelig for elevene, påpeker han. For å guide elevene i utforskningsfasen forteller Mats at han gir elevene instruksjonsark han har utarbeidet på forhånd. På disse arkene har han beskrevet ulike løyper elevene kan følge, med forslag til blokker de eventuelt kan bruke for å løse en oppgave. Dette har han brukt i forbindelse med roboten Bit:Bot, som kontrolleres med micro:bit. Anders forteller at han også har utarbeidet materiale i forkant av undervisningen:

Jeg laga jo et sånt hefte i undersøkelsen jeg hadde på masteren min, som var på en måte todelt da. En om hvordan man kom i gang og alt fra å lage seg bruker til hva de ulike delene på Scratch var, også et geometrihefte med oppgaver. Så jeg har egentlig brukt det og bygd videre på oppgavene jeg brukte til masteroppgaven min. Så jeg brukte jo masse tid på å lage oppgavene da, jeg får på en måte brukt dem igjen nå.

Elevene får altså en innføring i hvordan de bruker Scratch programmet og oppgaver knyttet til geometri, i heftet Anders har utarbeidet. Videre kommer det frem at dette er noe han har brukt mye tid på, og som han i etterkant av masterarbeidet har videreutviklet. Av analysearbeidet kommer det frem at lærerne varierer mellom ulike undervisningsstrategier når de bruker programmering og programmeringsverktøy i matematikkundervisningen. Analog programmering brukes av flere i introduksjonsfasen. Videre går det igjen at lærerne legger opp til utforskende arbeid der elevene er aktive på egenhånd. Instruksjonsark og

oppgavehefter er også brukt for å hjelpe elevene med å bli kjent med ulike funksjoner ved programmeringsverktøy.

4.2.3 Kunnskap om elevene og deres tenkning

I et forsøk på å nærme meg videre innsikt i lærernes matematiske kunnskap for undervisning, spør jeg hvordan de vet om elevene forstår idéene eller konseptene de forsøker å lære bort gjennom programmering. Noen opplevde dette som vanskelig, mens andre fortalte at de ikke hadde vurdert det noe særlig enda. Likevel kommer flere med forslag til hvordan det kan gjøres og hva det eventuelt forutsetter. Et eksempel på en forutsetning for å kunne vite om elevene følger undervisningen, er noe Bjørn oppga:

Det er vanskelig. Spesielt når du ikke kjenner ungene. (...) når du kjenner ungene så vil du i større grad kunne kjenne, altså, bare for å relatere til matematikk, holder jeg på med matematikkundervisning så kan jeg bare se på eleven, på kroppsspråk og ansiktsuttrykk, om de skjønner ingenting, om de skjønner litt, eller om de egentlig har catcha sammenheng. Det er litt det samme i programmering og egentlig.

Forutsetningen Bjørn gir uttrykk for er en relasjon til elevene. Ved å kjenne elevene mener han at han kan vurdere hvor mye de forstår ved å lese kroppsspråk og ansiktsuttrykk. Dette har han erfart i matematikkundervisning uten programmering, men opplever at det har overføringsverdi til når programmering brukes også. Å få elevene til å sette ord på det de har gjort, er en metode flere forteller de benytter seg av for å undersøke hva elevene tenker og forstår. Det kan vi se av eksemplene nedenfor:

Jeg spør de masse. «Åja, du har gjort sånn. Hvorfor gjør du det?» Hvorfor, hvorfor, den bruker jeg hele veien. Jeg har ikke noen type test eller noe sånt, jeg spør hele veien. Prøver å få de i tale og forklare hva de gjør (...). For når de plutselig får et forklaringsproblem, da prøver vi å ta opp det (Kristoffer).

Vi har Smartboard og det er veldig enkelt å bare koble til Ipaden, og på en måte vise fram det programmet du har gjort da. Så det er vel kanskje, ja, det jeg har gjort for å på en måte se litt da, om de faktisk har forstått det de har gjort, om de får til å forklare det for resten (Anders).

(...) Da vil du kanskje selv måtte gå inn og stille noen kontrollspørsmål for å spørre «hvorfor har du valgt å gjøre sånn og sånn? Og hvis det skulle vært litt annerledes, hvordan ville du gjort det da?». Og hvis de da klarer å forklare det på en god måte så vet du at de virkelig har skjönt det de har holdt på med da (Bjørn).

Både Kristoffer, Anders og Bjørn er opptatt av å få elevene til å forklare hva de har gjort for å kunne identifisere hva de forstår. Som Kristoffer sier, dersom de får et forklaringsproblem så er det noe som må tas opp. "

Angående det samme spørsmålet som nevnt ovenfor, velger Mats å se det fra en annen vinkling enn de øvrige lærerne:

Det er spennende. Det kunne vært en praktisk prøve. Få dem til å programmere noe. Få en bil til å kjøre gjennom et eller annet. Jeg vet ikke. Kunne vært et eget prosjekt der du programmerte et eller annet. Jeg vet at det finnes sånne kodeprøver.

Mats kommer heller med forslag til hvordan man kan vurdere elevene og få innsikt i hva de kan. Det vises at han ser flere muligheter for hvordan det kan gjøres, og han synes det er et spennende tema. I sum ser vi altså at gode relasjoner er trukket frem som en forutsetning for å vurdere hva elevene forstår. Å stille spørsmål som får elevene til å forklare hva de har gjort, er også løftet frem for å få innsikt i deres tankeprosess i forbindelse med programmeringsaktiviteter eller oppgaver.

4.3 Dobbel instrumentell genesis

For å undersøke lærernes instrumentelle genesis har jeg forsøkt å stille spørsmål som kan gi innsikt i ulike faser av denne prosessen, i henhold til Haspekians (2011) doble instrumentelle genesis. Fra egen læringsfase av et programmeringsverktøy til det blir et profesjonelt instrument for deres didaktiske aktivitet. I utdrag fra intervjuene vil verktøy gjelde både fysiske programmeringsverktøy, men også blokkbaserte og tekstbaserte programmeringsspråk.

4.3.1 Utfordringer knyttet til blokkbasert programmering

Lærerne har ulik erfaring når det kommer til programmering og bruk av programmeringsverktøy. Alle har likevel noe erfaring og alle bruker minst ett programmeringsverktøy i matematikkundervisningen sin. Micro:bit og Scratch er verktøy som trekkes frem flest ganger, og som flere har gode erfaringer med å bruke i undervisning. Dette er visuelle programmeringsspråk der elevene kan sette sammen ferdigprogrammerte blokker. En viktig grunn til at dette var verktøy flere av lærerne valgte, var at de selv hadde fått opplæring i dem og de var visuelle. Under intervjuene fikk lærerne som brukte Micro:bit og Scratch spørsmål om hva som var mest utfordrende ved å bruke disse. I den forbindelse er det bare Bjørn som utdyper en teknisk utfordring som la begrensninger for han selv i læringsfasen.

Intervjuer: Hva syns du har vært mest utfordrende når du har holdt på å lære deg å bruke Scratch og Micro:bit?

Bjørn: Det er å skjønne hvordan datamaskinen tenker egentlig. For du har jo en viss idé om hva du vil at den skal gjøre, men så selv om du skjønner sånne enkle begreper som at du vil at hvis det her skal skje så skal det skje, og at kanskje det med variabler var det vanskeligste i starten. Når du kom deg over den humpen så var på en måte alt annet lettere. Men når du var i den fasen at du ikke helt skjønnte hva en variabel var, så ga liksom ingenting mening. Så når det liksom klikka så falt alt på plass etter hvert.

Bjørn forteller om hvordan variabelfunksjonen i Scratch og Micro:bit la begrensninger for hans egen læringsprosess. Han beskriver også hvordan denne læringsprosessen innebar å forstå hvordan datamaskinen «tenker», noe som var utfordrende i starten. Videre forteller han om et slags vendepunkt da han lærte seg å bruke variabelfunksjonen:

(...) men når du endelig skjønnte hva en variabel var og hva den kunne brukes til, og hvordan du kunne nyttiggjøre det i en kode, så ble det mye mer spillerom for min egen kreativitet. Det var ingen begrensninger på at, jeg skjønnte ikke at hvis jeg hadde en idé om at det her vil jeg skal skje på skjermen, så klarte jeg ikke å se koblinga til hvordan det skulle skje. Men med engang jeg klarte å skjønne hva en variabel gjorde så kunne man i mye større grad tenke seg til at jeg kan skrive en kode som skal gjøre det jeg vil at den skal gjøre.

Bjørn gir uttrykk for at kunnskap om variabelfunksjonen i blokkbaserte programmeringsspråk som Scratch og Micro:bit var avgjørende for å oppnå ønsket resultat. Her mente Bjørn Scratch hadde en fordel fordi resultatene ble synlige på skjerm med engang. Ved bruk av Micro:bit er man derimot nødt til å lage programmet ferdig i nettversjonen, før det deretter kan lastes over på den fysiske enheten for å se resultatet. Dette er en utfordring både Kristoffer og Bjørn må hjelpe elevene sine med å forstå:

Ja, det har for enkelt tatt to-tre økter før de har forstått at «oi, jeg må faktisk gjøre det hver gang det skal skje». Og det er litt utfordrende med tanke på, jeg føler liksom at jeg henger sånn etter hele veien. Det er ikke like lett å få med det store lasset hele veien (Kristoffer)

Du må overføre til Micro:biten hele tiden. Det er mange som ikke skjønner det fordi de endrer på chrome booken da, også skjønner de ikke at det de da har gjort ikke fører til en instant forandring på Micro:biten (Bjørn).

Både Kristoffer og Bjørns kommentarer viser at bruk av fysiske enheter, som Micro:bit, der utviklingsomgivelsen kjøres direkte i en nettleser før det overføres via USB, kan være utfordrende fordi resultatene ikke umiddelbart blir synlige. Det kommer altså til uttrykk at blokkbaserte programmeringsspråk brukes av både barneskolelærere og ungdomsskolelærere, og at det visuelle aspektet trekkes frem som en fordel med verktøyene som brukes mest, Scratch og Micro:bit. Videre reflekterer Bjørn og Kristoffer over tekniske utfordringer ved blokkbaserte programmeringsspråk og trekker frem variabelfunksjonen og problematikken rundt overføring av program til Micro:bit.

4.3.2 Lærerutfordringer og bekymringer om hvordan knytte programmering til matematikk

Det er stor forskjell på bekymringene lærerne som jobber på ungdomsskolen og lærerne som jobber på barneskolen uttrykker. Det handler om at de bruker forskjellige verktøy blant annet. Av dataene kommer det frem at tekstbaserte programmeringsspråk, som for eksempel Python, brukes på ungdomsskolen. På barneskolen er fysiske verktøy og blokkbaserte programmeringsspråk mer utbredt. Refleksjoner over bruk av tekstbaserte programmeringsspråk var mest fremtredende i intervju med Anders og Bjørn, da var de som jobbet på ungdomsskoler. Anders forteller at på det ene lærerкурset han deltok på i programmering, kom det tydelig frem at tekstbasert programmering er hovedmålet å jobbe med på ungdomsskolen. Han opplever derfor at dette er noe han må prøve å bruke, selv om terskelen kan være høy:

Så den der terskelen for å bruke programmeringsspråk og ikke blokker, den, ja, det å begynne og bruke det synes jeg kan være utfordrende selv og. Jeg har ikke brukt det så mye jeg heller, og ja, det å sitte og leite etter feil i koder og sånn det kan være vanskelig. Og det er det for elever, og lærere og for så vidt.

Det kommer altså frem at overgangen fra å bruke blokkbaserte til tekstbaserte programmeringsspråk er utfordrende for både Anders og elevene. Det er spesielt feilsøking i koder som oppleves vanskelig. Bjørn på sin side opplever at det er utfordrende å bruke tekstspråket Python i undervisning, for å hjelpe elever på niende trinn mot kompetansemålet som innebærer å simulere utfall ved hjelp av programmering:

Altså det at du skal over til en ny ting med tekstprogrammering, og ting de aldri har vært borti før og sånne ting. Du må lære dem et helt tekstspråk og masse styr med lister og sånn i Python. Det blir så mye bare for at nå skal du få opp ti terningkast, hundre terningkast og tusen terningkast også skal du se på de resultatene og kanskje klare å relatere det til store talls lov. For når du gjør det så er det jo matematikk inn i det. Da er det inne på statistikk og sannsynlighet. (...) Men gjør du det i, jeg får ikke selv til å gjøre det i Python. Så ikke sant, det er der det stopper da. At de tingene som vil gjøre at du kan bruke det som et verktøy til å skjønne matematikk bedre, de har ikke elevene eller vi for øyeblikket.

Bjørn hevder verken lærere eller elever har kunnskapen som skal til for at programmering blir et verktøy for å forstå matematikken bedre. Han utdyper videre at dersom intensjonen med å integrere programmering i matematikkfaget er å bruke det på en måte som gjør elevene i stand til å se sammenhenger og relasjoner, er lærere nødt til å ha mer erfaring og flere studiepoeng, ikke nødvendigvis i matematikk, men i programmering. Dagens situasjon beskriver han slik:

Men nå så blir det heller sånn at vi skal lære oss programmering for at vi skal bli bedre på det. Og algoritmisk tenkning er jo en del av læreplanen, og det er sikkert masse i overordna del som passer bra med programmering. Men for matematikkfaget sin del så er programmeringskompetansen til elevene for lav til at du får noe mer matematikk ut av det.

Det kommer til uttrykk i det Bjørn forteller at fokuset i dag ligger på programmering for programmeringens skyld og ikke for matematikkens skyld. For at sistnevnte skal bli en realitet må altså læreres og elevers kunnskap om programmering blir bedre, mener Bjørn.

Silje, Mats og Kristoffer, som alle jobber på ulike barneskoler, knytter ikke utfordringen med å bygge bro mellom programmering og matematikk direkte til begrenset kunnskap om programmeringsverktøyene. Det kommer heller indirekte til syne eksempelvis i Anders og Mats' utsagn nedenfor:

Det som er, det har vi vært litt inno for, men det som er en stor utfordring det er, jeg syns det er litt vanskelig å knytte det opp direkte til tema. For eksempel, jeg har jo knyttet det opp til koordinatsystem, der jeg bruker en kule da som jeg snakket om, der de skal «okei, E6», også skal de kjøre til E6 eller noe sånt. Men så er det liksom sånn da har de gjort det, og etterpå hva skal vi videre? Det kjenner jeg er en utfordring med disse greiene her da (Anders).

Micro:bit har en egen side der det på en måte er lagt ut leksjoner for hva du skal gjennomføre. For eksempel å programmere en terning da på Micro:biten. Og veldig fine opplegg, veldig enkle opplegg, men det var veldig få av dem. Og det var kanskje 10-12 som var gjennomførbare. Det mest utfordrende var egentlig hvordan tar du det videre, hvordan kan du finne på mer (Mats).

Anders og Mats snakker her om to ulike programmeringsverktøy, henholdsvis den programmerbare robotballen Sphero BOLT og Micro:bit. Anders forteller at han har fått til å lage en aktivitet som lar elevene utforske et programmeringsverktøy innenfor et koordinatsystem, som samsvarer med kompetansemål i matematikk på barneskolen. Det er derimot en utfordring å vite hvordan slik opplegg kan utvides, ifølge Anders. En lignende opplevelse beskrives av Mats, som opplever at de eksisterende leksjonene på Micro:bits hjemmeside er enkle og gjennomførbare, men utfordrende å utvide.

På spørsmål om hva som er mest utfordrende med å bruke programmering i matematikkundervisning, forteller Silje at programmering oppleves som en ny måte å jobbe på for elevene hennes og at det derfor kan bli vanskeligere å se sammenhenger: Silje sier for eksempel:

Ja, det er vel det vel kanskje det at elevene ikke helt ser sammenheng bestandig da. Når det gjelder matematikk så er det jo ofte tall. Man jobber kanskje ofte i en bok. Eller liksom at det er veldig sånn "følg den her regelen" eller "nå forklarer jeg oppgaven også skal du gjøre sånn som jeg har sagt". Mens i programmering blir det jo en annen måte å jobbe på.

Silje forteller i forbindelse med denne kommentaren at hun ikke kaller det matematikk når hun har brukt programmeringsverktøy i undervisning med 2.trinn. Hun begrunner dette med at flere av elevene hennes er negative til matematikkfaget i utgangspunktet, og at hun dermed ikke ønsker å skape flere dårlige assosiasjoner. I sum ser vi altså en stor variasjon i hva lærerne opplever som utfordrende når de skal forsøke å bruke programmering for å lære elevene matematikk. Lærerne på ungdomsskolen reflekterer i større grad over tekniske utfordringer, spesielt ved tekstbaserte programmeringsspråk som Python, og hvordan de legger begrensninger for at programmering kan bli et verktøy for å forstå matematikken. Lærerne på barneskolen nevner ikke eksplisitt utfordringer tilknyttet bruk av programmeringsverktøy, men heller tilknyttet utvidelse av

undervisningsopplegg og til å venne elevene til at programmering er en ny måte å jobbe på.

5 Drøfting

I denne masteroppgaven har jeg valgt å studere læreres pedagogiske teknologiske kunnskap ved bruk av programmeringsverktøy i matematikkundervisning. Problemstillingsstillingen jeg søker å belyse er:

Hva kjennetegner læreres pedagogiske teknologiske kunnskap ved bruk av programmeringsverktøy i matematikkfaget i grunnskolen?

I kapittel 4.1, 4.2 og 4.3 presenterte jeg aspekter ved PTKs tre teoretiske komponenter som jeg identifiserte i datamaterialet. Dette jeg gjorde jeg ved hjelp av tre forskningsspørsmål realtert til de teoretiske komponentene personlige lærerorienteringer, MDKT og instrumentell genesis. I dette drøftingskapittelet vil jeg forsøke å nærme meg problemstillingen gjennom å diskutere funn i analysen i lys av det teoretiske perspektivet jeg har gjort rede for tidligere. Det vil være funnene som er ansett mest sentrale for å belyse problemstillingen som er løftet frem. Drøftingen vil i det videre følge samme rekkefølge som hovedkategoriene funnene ble presentert i ovenfor.

Først vil jeg oppsummere hovedfunnene fra kapittel 4. Samtlige lærere ga uttrykk for positive orienteringer mot bruk av programmeringsverktøy i matematikkundervisning. Flere viste tegn til god selvtillitt for å benytte slike verktøy, til tross for at de selv befant seg i læringsfasen av å bruke de. Alle reflekterte over verdien av å bruke programmeringsverktøy i matematikkundervisning, som i større grad var rettet mot et teknologisk fremtidsrettet perspektiv heller enn utvikling av matematisk forståelse. Videre viser analysearbeidet at flertallet av lærerne benytter analog programmering i introduksjonsfasen av programmering. Flere har kunnskap om læreplaninnhold, men noen av lærernes svar viser at intensjonen i kompetansemålene er vanskelig å tolke. Samtlige lærere beskriver programmeringsbegrepet, med noen variasjoner, som å få noe til å skje ved hjelp av instruksjoner i en bestemt rekkefølge. Studien tyder likevel på at til tross for programmerings plass i matematikkfaget, er det utfordrende for lærerne å knytte programmering og matematikk sammen. Lærerne fra ungdomsskolene forteller om tekniske utfordringer som legger begrensninger for at programmering kan bli et verktøy for å lære matematikk bedre. Lærerne fra barneskoler forteller om utfordringer med å knytte programmeringsaktiviteter direkte til temaer innenfor matematikken.

5.1 Lærernes personlige orienteringer

I dette delkapittelet vil jeg drøfte det første forskningsspørsmålet: *Hva er aspekter ved læreres personlige orienteringer mot programmering i matematikkundervisning?*

Av analysen fremkommer det at deltakerne beskriver programmeringsbegrepet på mer eller mindre lignende måter. Overordnet handler programmering for lærerne om å få noe, en datamaskin, en ting, et menneske, til å gjøre noe. Likvel er det noen av beskrivelsene som skiller seg ut ved at de fanger opp noen viktige konsepter i algoritmisk tenkning som for eksempel instruksjoner og algoritme. Spørsmålet om hva deltakerne legger i programmeringsbegrepet ble stilt som et innledende spørsmål til hoveddelen for intervjuet. Utover intervjuene ble det tydelig at selv om deltakerne i første omgang implisitt og eksplisitt knyttet programmering til å skrive, gi eller følge instruksjoner, i likhet med funn fra Pörn et al.'s (2021) studie, tolket de etter hvert begrepet fra ulike perspektiver. Det kan være flere årsaker til at deres syn på programmering kom til uttrykk på forskjellige

måter til forskjellige tider i intervjuet, og en mulig grunn kan være at det er forskjell på hvordan de beskriver begrepet fra et generelt ståsted, slik det første spørsmålet jeg stilte kan ha implisert, og fra et undervisningsståsted.

Uansett, som tidlig brukere (early adopters) er lærerne i denne studien positive til programmeringsverktøy og programmeringsaktiviteter, men i likhet med det lærere rapporterte i Misfeldt et al.'s (2019) studie, ser de ikke alltid den klare sammenhengen til matematikk. Dette kan potensielt være et resultat av visuelle miljøer, som Scratch, som er mer tilpasset historiefortellingsaktiviteter og animasjoner heller enn matematikk. Følgelig er det en utfordring å fremme visuelle miljøer og samtidig plassere programmering i matematikkfaget. Dette ble også vist av Benton et al. (2017) som understreket viktigheten av å designe Scratch aktiviteter med spesifikke matematiske mål, og lærerens rolle i å gjøre de matematiske sammenhengene eksplisitte. Akkurat i geometri, virker det av analysen som at det er enklere å balansere disse to forholdene. Hvorfor det er sånn kan kanskje forklares med at geometriske figurer ikke oppleves like abstrakt som for eksempel likninger, og at visuelle programmeringsmiljøer legger godt til rette for å kunne utforske geometriske egenskaper.

Det var få svar som eksplisitt knyttet programmering til annet matematisk innhold. Grunnen til dette kan være måten jeg formulerte spørsmålet på i intervjuene. I en av deltakernes spontane svar blir en programmeringsaktivitet knyttet til «De store talls lov» og sannsynlighet og statistikk, selv om deltakeren er usikker på hvordan aktiviteten skal la seg gjennomføre i praksis med tekstprogrammering. Matematisk innhold som for eksempel måling, aritmetiske uttrykk eller likningsløsning er det ingen som relaterer aktiviteter til. Fraværet av denne koblingen kan ha flere årsaker, det kan ha med spørsmålsformuleringen min å gjøre, eller, på linje med lærerbekymringene nevnt av Benton et al. (2017), handle om at grunnskolelærere ikke helt forstår samspillet mellom matematikk og programmeringsinnhold og læring. Som flere forskere påpeker, er det et behov for å tydeliggjøre koblingene mellom matematikk og programmering for lærere (Benton et al., 2017; Hickmott et al., 2018). En helt annen forklaring kan være det begrensede utvalget lærere jeg intervjuet. Et større volum av intervjuobjekter kunne potensielt belyst problemstillingen på en mer nyansert måte. Slik studien tyder på nå, er deltakerne utelukkende positive til programmering og programmeringsaktiviteter i skolen, selv om de opplever utfordringer i integreringen til matematikkfaget. Jeg mistenker derfor at bildet kan være mer nyansert, og at lærere som har mindre positive orienteringer kunne bidratt med andre synspunkter.

Ifølge Thurm og Barzel (2022) finnes det en generell oppfatning om at positiv tro på undervisning med teknologi er gunstig for undervisning i matematikk med teknologi. De omtaler denne mestringstroen, også kalt selvtillit, som hvordan en person oppfatter sin egen evne til å organisere og gjennomføre undervisning med teknologi som skal produsere gitte prestasjoner. Clark-Wilson og Hoyles (2019) har identifisert lav mestringstro som den sannsynlige årsaken til at lærere ikke gjennomfører planlagt undervisning med teknologi. Det kan derfor være rimelig å anta at mestringstroen til lærerne i denne studien var en sterk bidragsyter til at de tok i bruk programmeringsverktøy og gjennomførte programmeringsaktiviteter med en intensjon om å hjelpe elevene deres til å nå kompetansemål i matematikk. Selv om utfordringen med å knytte programmering og matematikk sammen legger begrensninger for å oppfylle visse kompetansemål, vil jeg likevel argumentere for at de har et godt utgangspunkt for å kunne få det til i fremtiden. Det er også verdt å nevne at Thurm og Barzel (2022) hevder at underdimensjoner ved læreres mestringstro har fått mangelfull oppmerksomhet i forskning. De trekker frem to

underdimensjoner som kan påvirke læreres mestringstro, nemlig design og valg av oppgaver og design og gjennomføring av undervisningsøkt (se figur 4). Basert på funn i denne studien kan det tyde på at det er nødvendig å vie oppmerksomhet til slike underdimensjoner, for å lykkes med å knytte programmering og matematikk sammen. Design og valg av oppgaver er avgjørende ettersom å realisere potensialene til teknologi for læring av matematikk krever passende oppgaver (Thurm & Barzel, 2022). Likledes må lærere ha mestringstro til å designe undervisning hvor teknologien implementeres på passende måter (Thomas & Palmer, 2014).

Resultater fra Thurm og Barzel's (2022) studie, indikerer at troen på potensielle fordeler ved bruk av teknologi og oppfatninger om tidskravene til implementering av teknologi kan veie tyngre enn viktigheten av lærernes oppfatninger om negative effekter av teknologibruk. Det kan derfor tyde på at positive orienteringer gir gode forutsetninger for å lykkes med integreringen av nye verktøy, som programmering. Det kan være flere grunner til at alle lærerne i denne studien uttrykte positive orienteringer. Som beskrevet i kapittel 3.3.1 befant lærervalget seg i aldersgruppen 27-30, alle hadde erfaring med minst ett programmeringsverktøy og majoriteten hadde deltatt på lærerkurs eller foredrag om programmering. Det kan derfor spekuleres i om deltakernes unge alder og lignende bakgrunner kan være mulige forklaringer på de nesten utelukkende positive orienteringene.

Det er ingen tvil om at undervisning med teknologi utfordrer måten lærere orkesterer læring på, og det krever nye måter å tenke på for å implementere det på hensiktsmessige måter i klasserommet. Det virker som at lærerne i denne studien ser seg nødt til å prøve seg en del frem egenhånd fordi det ikke finnes noen autoritet i hva som skal og burde gjøres i undervisning med programmeringsverktøy for å lære matematikk. Jeg vil derfor hevde, på bakgrunn av mangelen på forkunnskaper om programmering blant lærere, at det er et behov for at det blir utarbeidet undervisningsopplegg og undervisningsaktiviteter som kan sørge for at elevene lærer matematikk med programmeringsverktøy. Det virker hvert fall nødvendig så lenge både lærere og elever befinner seg i en læringsfase av hva det vil si å programmere. Det kan kanskje også føre til at lærere med lav mestringstro gir det et forsøk.

5.2 Lærernes matematiske digitale kunnskap om undervisning

I dette delkapittelet vil jeg drøfte det andre forskningsspørsmålet: *Hvilke aspekter ved læreres matematiske digitale kunnskap for undervisning kommer til uttrykk i deres beskrivelser av deres undervisningstilnærming til matematikk med programmeringsverktøy?*

Nå, så vel som tidligere, introduseres nye emner i den nasjonale læreplanen, og legger vekt på lærernes evne til å undervise i det nye innholdet. Det kan derfor forventes at i fremtidige klasserom vil andre utfordringer, som følge av teknologiske fremskritt og menneskelig tenking, stille store krav til lærere om å transponere innholdet slik at det blir tilgjengelig og aktuelt for alle elever. Følgelig understreker Shulman (1986) at det er en forskjell mellom å kunne et emne og å kunne undervise i det, noe som, når det gjelder programmeringskunnskap, reiser både praktiske og didaktiske bekymringer.

Funn i denne studien tyder på at kompetansemålene som gjelder programmering i matematikk kan være vanskelige å tolke. Det faktum at matematikkfaget har fått hovedansvaret for opplæringen av programmering gir uttrykk for at intensjonen må være at elevene skal bruke programmeringsverktøy for å lære matematikk. Likevel kan kompetansemålformuleringer gi uttrykk for at elevene skal lære programmering for

programmeringens skyld og ikke for matematikkfagets skyld. Spesielt åttende klasse kompetansemålet «utforske hvordan algoritmer kan skapes, testes og forbedres ved hjelp av programmering», er løftet frem av en av deltakerne i denne studien som et eksempel på et kompetansemål som kan føre til forvirring. Det kan muligens handle om at intensjonen ikke kommer eksplisitt frem, eller så kan det kanskje være at metodefriheten lærere har i den nye læreplanen oppleves for stor. Kompetansemålene slik de er utformet i dag, sier hva elevene skal kunne etter hvert endte skoleår, men de sier ingenting om hvordan elevene skal komme dit. Det blir derfor opp til hver enkelt lærer å tolke hva kompetansemålet faktisk vil at elevene skal kunne, og deretter velge den mest hensiktsmessige metoden for å hjelpe de dit. I denne prosessen oppstår det et forhold mellom lærerens pedagogiske kunnskap og innholdskunnskap, ifølge Shulman (1986).

Utdanningsdirektoratet (2021) argumenterer for at læreplanenes handlingsrom er der for å kunne tilpasse opplæringen til alle elever på alle trinn. Denne tanken har bakgrunn i at elever, lokalsamfunn og skoler er forskjellige, og at samfunnet stadig endrer seg. Som følge av handlingsrommet lærere er gitt, finnes det ingen lister over aktiviteter som skal gjennomføres eller detaljerte oversikter over kunnskapsinnhold. Dette kan oppleves befriende eller begrensende, alt ettersom hvilke muligheter man ser. Funn fra studien tyder på at handlingsrommet virker begrensende i form av at intensjonen ved å bruke programmeringsverktøy i matematikk ikke kommer tydelig frem. Lærerne virker derfor usikre på hvor fokuset skal være, skal det være på å lære å bruke verktøyene eller skal det være på hva de kan brukes til? Dette funnet kan ses i sammenheng med det Kilhamn et al. (2020) påpeker i deres studie av svenske læreres programmeringsaktiviteter i matematikk. Der er det å lære seg hva programmering er og hvordan det fungerer, mer fremtredende enn spørsmålet om hva programmering kan brukes til og hvorfor.

I lys av Tabach og Trglovás MDKT (se figur 3), er det aspekter ved enkelte kunnskapsområder som kommer tydeligere frem i datamaterialet enn andre. For eksempel KDCS, kunnskap om digitalt innhold og elever, kommer til uttrykk gjennom lærernes beskrivelser av hvordan de vet om elevene deres forstår det de prøver å lære bort. Kontrollspørsmål virker å være en gjentakende strategi for å få tilgang til elevenes resonneringer, og til å identifisere hva som kan være vanskelig. To av lærerne forteller at de har identifisert en utfordring som handler om at elevene deres ikke forstår hvorfor programmene de lager i nettversjonen ikke fører til umiddelbare resultateter på den fysiske enheten. Hvorfor denne forvirringen oppstår kan muligens ses i sammenheng med det Kallia (2021) kaller transposisjon. I tilfellet lærerne forteller om har elevene imidlertid ikke fått til å overføre løsningene eller programmene ders til Micro:bit. Det har altså ikke skjedd en transposisjon. Utfordringer som denne, hvor verktøyet i seg selv legger begrensninger for hva elevene får til, kan gjøre det vanskelig å balansere tiden som burde brukes på verktøyet og tiden som burde brukes på lære matematikk. Det er uunngåelig at lærere og elever ikke skal bruke tid på å lære seg nye verktøy, men hvor mye tid vil egentlig gå til tekniske utfordringer og vil det gå på bekostning av matematikken som skal læres?

Det ser derfor ut som det kan oppstå et spenningsforhold mellom den nødvendige kunnskapen lærere trenger om matematikk for undervisning og om digital teknologi. Digital teknologi vil i dette tilfellet gjelde programmeringsverktøyene lærerne velger å bruke. Ifølge Brennan (2015, sitert i Benton et al., 2017) må lærere oppnå en balanse «between knowledge about the tool and understanding of how to engage in creative design activities, using the computer¹ for personal expression and problem solving” for å håndtere

¹ For at sitatet skal gi mening i denne studiens kontekst, kan «using the computer» byttes ut med «using the programming tools».

denne spenningen. Fra et instrumentelt perspektiv kan dette forstås som en instrumenteringsprosess. Dette perspektivet vil jeg utdype videre i neste kapittel 5.3. I tillegg gjenstår den kritiske utfordringen med å utnytte kunnskap som er oppnådd innenfor programmeringssammenhenger for å fremme engasjement med matematiske idéer og resonnement (Hoyles & Noss, 1996). For å hjelpe lærere i møte med disse utfordringene, ville det vært hensiktsmessig om matematikklærere fikk tilbud om kurs for å lære hvordan programmeringsverktøyene fungerer, men ikke minst også hva de kan brukes til i matematikk.

5.3 Lærernes doble instrumentell genesis

I dette delkapitlet vil jeg drøfte det tredje og siste forskningsspørsmålet: *På hvilke måter kan læreres instrumentelle genesis påvirke koblingen mellom programmering og matematikk i undervisning?*

I henhold til Rabardel og Verillon (1995) befinner elever seg i instrumenterte aktivitetssituasjoner når lærere gjennomfører undervisning med formål om å lære de matematikk gjennom programmeringsverktøy. Funn fra denne studien tyder på at de fleste lærerne ser nytten av å bruke programmeringsverktøy i forbindelse med utforskning og problemløsning. I instrumenterte aktivitetssituasjoner kan derfor formålet for eksempel være å lære matematikk gjennom problemløsning og utforskning med det aktuelle programmeringsverktøyet. Men selv om det er formålet, betyr det nødvendigvis ikke at elevene faktisk lærer matematikk. Først må elevene ha utviklet passende bruksskjemaer for aktiviteten de ønsker å gjennomføre. Likevel kan man argumentere for at problemløsning og utforskning med programmeringsverktøy er aktiviteter som kan føre til utvikling av slike bruksskjemaer. Det at de fleste lærerne i studien forteller at de benytter programmeringsverktøy i forbindelse med nettopp problemløsning og utforskning, kan potensielt bidra til at elevene blir kjent med verktøyenes funksjoner og utvikler flere bruksskjemaer for hvordan de eventuelt kan brukes i andre type aktiviteter. Først da vil verktøyet kunne forvandles til et instrument. Denne prosessen har som nevnt i kapittel 2.2.1 fått navnet instrumentell genesis. Ifølge Bozkurt et al. (2018) er prosessen nødvendig for at artefaktet skal oppleves meningsfullt og forvandles til et nyttig matematisk instrument. Dette gjelder både for elever, men også for lærere. De må utvikle sine egne instrumenter, men også guide og veilede elevenes personlige instrumentelle genesis for at verktøyene skal kunne lede til matematisk forståelse. Basert på funn i studien, virker det som lærerne gjør nettopp dette. Flere forteller at de er opptatt av at elevene skal sett ord på det de gjør når de bruker programmeringsverktøy, de veileder underveis i undervisningsøkter og en av lærerne forteller at han lar elevene vise frem programmene sine på Smartboard. En slik bevisst organisering og bruk av tilgjengelige artefakter i et læringsmiljø, kan knyttes til det Drijvers et al. (2010) omtaler som instrumentell orkestrering.

Av analysen fremkommer det likevel en bekymring fra flertallet av lærerne som handler om hvordan de skal knytte programmering og matematikk sammen. Denne bekymringen virker å grunne i ulike utfordringer avhengig om lærerne underviser på barneskolen eller ungdomsskolen. Lærerne på ungdomsskolen reflekterer i større grad over utfordringer direkte knyttet til programmeringsverktøyets funksjoner, mens lærerne fra barneskolen forteller om utfordringer med å knytte programmeringsaktiviteter direkte til temaer innenfor matematikken. Utfordringene ser dermed ut til å være mer rettet mot selve verktøyene og deres funksjoner på ungdomsskolen, og mot lærernes didaktiske aktivitet på barneskolen. I læreplanen for matematikk i grunnskolen sier kompetansemålet for tiendeklasse at målet for opplæringa er at eleven skal kunne «utforske matematiske

egenskaper og sammenhenger ved å bruke programmering» (Utdanningsdirektoratet, 2019a). Etter ti år skal altså alt elevene har lært om programmering i matematikkundervisning resultere i at de når dette målet. Lærerne i studien fra ungdomsskoler forteller om utfordringer knyttet til variabelfunksjonen ved blokkbaserte programmeringsspråk som Scratch og Micro:bit, og knyttet til hvordan de kan få tekstbaserte programmeringsspråk som Python til å fungere som verktøy for å forstå matematikken bedre. I lys av teori om instrumentell genesis kan det virke som at disse utfordringene bunner i instrumenteringsprosessen av programmeringsverktøyene. Det vil si at verktøyene kanskje ikke har blitt forvandlet til fullverdige matematiske instrumenter enda.

Lærere vil altså befinne seg i en svært kompleks prosess ved bruk av programmeringsverktøy i matematikkundervisning. Ikke bare må de utvikle bruksskjemaer og personlige instrumenter, de skal også sørge for at elever gjør det samme. Dette foregår gjennom en prosess av profesjonell genesis der lærere utvikler et profesjonelt instrument for deres didaktiske aktivitet, som er å lære elevene matematikk. Lærerne fra barneskoler i denne studien uttrykket ingen utfordringer eksplisitt knyttet til å bruke verktøyene. Det kan tyde på at de har blitt utviklet til personlige instrumenter. De forteller derimot om utfordringer knyttet til å koble verktøyene til matematikk, noe som kan forklares med at verktøyene ikke har blitt utviklet til profesjonelle instrumenter. Den doble instrumentelle genesis som er forventet av lærere referer både til deres personlige genesis ved bruk av digitale verktøy på generell basis og til deres profesjonelle genesis ved bruk av digitale verktøy for å undervise i matematikk for elevene deres. Selv om den profesjonelle genesis kan virke avgjørende for å kunne oppfylle læreplankravene, hevder jeg på grunnlag av det studien viser, at en god personlig genesis er en forutsetning for å utvikle profesjonell genesis. Å se bort fra den personlige genesis kan være feil av to grunner. For det første kan læreres teknologiske ferdigheter variere betydelig, og de vil derfor ha svært ulike forutsetninger for å kunne forvandle verktøy til instrumenter for deres didaktiske aktiviteter. For det andre, på grunn av den raske utviklingen av digitale verktøy, virker det usannsynlig å tro at en personlig genesis vil bli til av seg selv. Det kan implisere et behov for at lærere først må oppfatte teknologiens potensial og dens mulige bidrag til undervisning og læring i matematikk, for at de senere skal kunne ta den i bruk og begynne å bruke den personlig som et matematisk instrument.

For å møte utfordringen med å knytte programmering og matematikk sammen, kan det potensielt være et behov for å forbedre læreres instrumentelle genesis. Dette er en stor oppgave siden hvert nye artefakt krever individuell oppmerksomhet. Thomas og Palmer (2014) har også merket seg dette behovet når de oppsummer hindringer og muligheter i undervisning med digital teknologi. De mener forbedringen må gjøres gjennom faglige utviklingsprogrammer som promoterer veiledning av lærere som kan fungere som mentorer for andre lærere, fortrinnsvis på samme skole. En annen mulighet kan være erfaringsdeling mellom lærere på en skole, der lærere med god instrumentell genesis kan dele av sine erfaringer og veilede andre som ønsker å ta i bruk lignende verktøy. Det forutsetter at lærere selv tar initiativ til det og at settes av tilstrekkelig med tid. Men kanskje kan det å virkelig se hvordan programmering kan fremme matematikk bare være mulig når programmering har blitt en håndgripelig del av hverdagen? Kilhamn et al. (2021) har også stilt spørsmål til hva hovedhensikten med å lære programmering i matematikkfaget er og burde være. De gjør oppmerksom på at dersom hovedhensikten er å lære å bruke et nytt verktøy, spør det om tiden brukt på programmering er bortkastet fra et matematikkundervisningsperspektiv. Da datamaskinene ble introdusert som nye verktøy i skolen, var for eksempel det å lære og bruke tastatur og søkemotorer ikke inkludert i matematikkpensumet. Sånn sett kunne man argumentert for at programmering

burde vært et eget fag i skolen, for å få på plass det mest grunnleggende for at det senere kan bli et mer verdifullt verktøy for å lære matematikk. På den andre siden kan prosessen med å bli kjent med et verktøy gjennom problemløsning og utforskning i matematikkfaget, bidra til å utvikle strategier og fremgangsmåter som kan være verdifulle på flere områder. Kanskje vil det derfor være nødvendig med en diskusjon om elever skal lære programmering for matematikkens skyld, eller om programmering for programmeringens skyld kan være nyttig for å utvikle nye måter å tenke og arbeide på.

6 Avslutning

I denne masteroppgaven har jeg undersøkt hva som kjennetegner fem grunnskolelæreres PTK ved bruk av programmeringsverktøy i matematikkundervisning. Selv om oppgaven min dreier seg om en spesifikk kontekst, er resultatene likevel av verdi for forskningsfeltet da den belyser en aktuell problemstilling, undervisningen av programmering i grunnskolematematikk fra lærernes perspektiv. Det er også verdifullt å ha studert lærernes forhold til programmering kort tid etter læreplanimplementeringen høsten 2020. Oppgaven kan og bidra til kunnskap om hvordan grunnskolelærere forholder seg til undervisning i programmering. I dag har grunnskolelærere tilgang på en rekke ulike verktøy og materiell når de underviser. Mye er tilgjengelig gjennom diverse læringsplattformer ulike skoler benytter, og en rekke norske kommuner har gått til innkjøp av klassesett med programmeringsverktøy som blant annet Micro:bit. Likevel vil det være avgjørende å bruke det tilgjengelige materialet riktig. For å gjøre det er det nødvendig med tilstrekkelig fagkunnskap om både programmering og matematikk.

Det er nok langt flere aspekter som kan belyse lærernes PTK enn de som fremgår av min analyse, men de kan likevel bidra til å gi et visst inntrykk av hva som kjennetegner den i grove trekk. Innsikt i lærernes PTK kan gjøre det enklere å få grep om hva som virker å fungere og hva som er utfordrende ved bruk av programmeringsverktøy i matematikkundervisning. Av analysearbeidet virker det som at det er deltakernes positive orienteringer som danner grunnlaget for at de velger å benytte programmeringsverktøy i matematikkundervisning, og for at de vier tid til å jobbe mot kompetansemålene som gjelder programmering. Når lærerne deretter blir kjent med verktøyenes funksjoner og potensial i undervisning, involveres de i en prosess av instrumentell genesis. Funn tyder på at en god instrumentell genesis må være på plass for at programmeringsverktøy kan bli instrumenter for matematiske aktiviteter. Her virker det som det er et behov for å hjelpe lærerne med å videreutvikle deres instrumentelle genesis. Faglige utviklingsprogrammer og erfaringsdeling kan være hensiktsmessig i denne prosessen. En god instrumentell genesis virker også å være nødvendig for å utvikle en god profesjonell genesis, der de samme verktøyene blir instrumenter for deres didaktiske aktivitet. I denne prosessen spiller også deltakernes MDKT inn, områder med kunnskap som er nødvendig for å lykkes med integreringen av teknologi matematikkfaget. Her tyder funn i studien på at lærerne har kunnskap om læreplaninnhold og om hvordan elevene deres tenker og hva de synes er vanskelig.

Det virker altså som det er et behov for pedagogisk innsats for å synliggjøre sammenhengen mellom matematisk innhold og programmering for grunnskolelærere, for eksempel i form av konkrete undervisningsopplegg og aktiviteter eller pedagogiske praksiser. De som jobber med lærere, lærerutdanningen og produksjon av læremateriell spiller en viktig rolle i dette kontinuerlige arbeidet.

6.1 Videre forskning

Programmering har nå vært en del av kompetansemål i matematikkfaget i to år. Det betyr at flere lærere og elever begynner å få erfaring med å bruke programmeringsverktøy. Med utgangspunkt i farten digitaliseringen i verden foregår, er det ingenting som tilsier at mengden nye verktøy i klasserommet kommer til å avta med det første. Elevene lærer verken dårligere eller bedre med verktøyene i seg selv, og det er derfor nødvendig å fortsette og undersøke hva slags kunnskap lærere trenger for å undervise med programmeringsverktøy og samtidig lære elevene matematikk.

Som forklart i kapittel 3.3.1, førte ulike årsaker til at utvalget i denne studien bestod av matematikklærere spredt fra 2. – 10.klasse. I videre forskning på samme tema kunne det vært aktuelt å rette fokuset mot enten småtrinnet, mellomtrinnet eller ungdomsskolen for å gi et mer detaljert bilde av en mer avgrenset kontekst. Det kunne også vært hensiktsmessig å gjennomført en lignende studie, men endret forskningsmetoden. Observasjon sammen med intervju kunne gitt et tydeligere bilde over hva deltakere sier at de gjør og hva som faktisk skjer i klasserommet. En kvantitativ undersøkelse kunne også vært aktuelt for å gi målbare resultateter av læreres PTK.

Til slutt vil jeg si at PTK rammeverket er interessant å bruke i forskning fordi det kan belyse hvor komplekse prosesser lærere befinner seg i når de underviser med teknologi i matematikkundervisning. Likevel er det omfattende å analysere et datamateriale med utgangspunkt i tre teoretiske komponenter, så en idé kunne vært å ta utgangspunkt i én av komponentene for å undersøke denne grundigere. Innsikt om aspekter ved læreres kunnskap kan videre utnyttes for å utvikle faglige utviklingsprogrammer angående integrering av programmeringsverktøy i matematikk. Senere kunne man undersøkt effekten av slike program på for eksempel læreres undervisningspraksis.

Referanser

- Balanskat, A., & Engelhardt, K. (2015). *Computing our future - Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Brussel: European Schoolnet.
http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03
- Ball, D. L., Thames, M. H., & Phelps, G. (2008). Content knowledge for teaching: What makes it special? *Journal of teacher education*, 59(5), 389-407.
<http://dx.doi.org/10.1177/0022487108324554>
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: some findings of design research in England. *Digital Experiences in Mathematics Education*, 3(2), 115–138. <https://www.doi.org/10.1007/s40751-017-0028-x>
- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group.
<https://doi.org/10.17471/54007>
- Bozkurt, G., Uygan, C., & Turgut, M. (2018). Instrumental genesis of a preservice mathematics teacher: Instrumented actions on the perpendicular line construction. I H.-G. Weigand, A. Clark-Wilson, A. Donevska-Todorova, E. Faggiano, N. Grønbaek, & J. Trgalova. Proceedings of the Fifth ERME Topic Conference (ETC 5) on Mathematics Education in the Digital Age (MEDA) (s. 43–50). Copenhagen, Denmark: University of Copenhagen.
- Bray, A., & Tangney, B. Technology usage in mathematics education research - A systematic review of recent trends. *Computers & Education*, 114, 255–273.
<https://doi.org/10.1016/j.compedu.2017.07.004>
- Bryman, A. (2016). *Social research methods* (5. Utg.). Oxford university press.
- Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning*, 25(2), 170-185.
<https://doi.org/10.1080/10986065.2020.1779012>
- Clark-Wilson, A., Robutti, O., & Thomas, M. (2020). Teaching with digital technology. *ZDM Mathematics Education*, 52(7), 1223-1242. <http://dx.doi.org/10.1007/s11858-020-01196-0>
- Clark-Wilson, A., Aldon, G., Cusi, A., Goos, M., Haspekian, M., Robutti, O., & Thomas, M. O. J. (2014). The challenges of teaching mathematics with digital technologies- The evolving role of the teacher. I P. Liljedahl, C. Nichol, S. Oesterle & D. Allan (Red.). *Proceedings of the Joint Meeting of PME 38 and PME-NA 36* (pp. 87-166). PME.
- Drijvers, P., Doorman, M., Boon, P., Reed, H., & Gravemeijer, K. (2010). The teacher and the tool: Instrumental orchestrations in the technology-rich mathematics classroom. *Educational Studies in Mathematics*, 75(2), 213-234. <https://doi.org/10.1007/s10649-010-9254-5>
- Drijvers, P., Doorman, M., Boon, P., & van Gisbergen, S. (2009). Instrumental orchestration: theory and practice. *Proceedings of the Sixth Congress of the European Society for Research in Mathematics Education*, 1349-1358.

- Gjøvik, Ø., & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten–tidsskrift for matematikkundervisning*, 30(3), 31-37.
- Guin, D., & Trouche, L. (1998). The complex process of converting tools into mathematical instruments: the case of calculators. *International Journal of Computers for Mathematical Learning*, 3(3), 195–227. <https://doi.org/10.1023/A:1009892720043>
- Haraldsrud, D. A., Løvold, H. H. & Sveinsson, A. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Haspekian, M. (2005). An “Instrumental Approach” to study the integration of a computer tool into mathematics teaching: The case of spreadsheets. *International Journal of Computers for Mathematical Learning*, 10(2), 109–141. DOI:10.1007/s10758-005-0395-z
- Haspekian M. (2014) Teachers’ Instrumental Geneses When Integrating Spreadsheet Software. I A. Clark-Wilson, O. Robutti, N. Sinclair. (Red.) *The Mathematics Teacher in the Digital Era. Mathematics Education in the Digital Era*, 2, 241-275. Springer, Dordrecht. https://doi.org/10.1007/978-94-007-4638-1_11
- Herheim, R. (2019). Programmering og algoritmisk tenking? *Tangenten – tidsskrift for matematikkundervisning*, 30(4).
- Herheim, R., & Severina, E. (2020). Scratch programming and student’s explanations. *Mathematics Education in the Digital Age (MEDA)*, 45.
- Heintz, F., Manila, L., & Färnqvist, T. (2016). A review of models for introducing Computational Thinking, Computer Science and Computing in K–12 education. Proceedings of 2016 IEEE Frontiers in Education Conference, 1–9.
- Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education*, 4(1), 48–69. DOI:10.1007/s40751-017-0038-8
- Hong, Y. Y., & Thomas, M. O. J. (2006). Factors influencing teacher integration of graphic calculators in teaching. *Proceedings of the 11th Asian Technology Conference in Mathematics*, 234-243. <https://www.atcm.mathandtech.org/EP/2006/2006P152/fullpaper.pdf>
- Hoyles, C., & Noss, R. (2015). A computational lens on design research. *ZDM*, 47(6), 1039-1045. <https://doi.org/10.1007/s11858-015-0731-2>
- Kvale, S., & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg.). Gyldendal Akademisk.
- Johansen, A.-K. (2020, 07 11). Programmering vil bli en utfordring for lærere. <https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold>
- Johannessen, A., Tufte, P. A., Christoffersen, L. (2016). *Introduksjon til samfunnsvitenskapelig metode* (5.utg.). Abstrakt forlag.
- Johnsen-Høines, M. & Alrø, H. (2019). Forskningsmetodologi og undervisningsmetodologi i utdanningsforskning. I K, R. Breivega & T, E. Rangens (Red.), *Demokratisk danning i skolen* (s. 34-48). Universitetsforlaget.
- Koschmann, T. (1997). Logo-as-Latin Redux. *The Journal of the Learning Sciences*, 6(4), 409–415. <http://www.jstor.org/stable/1466780>
- Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: a literature-informed Delphi study. *Research in Mathematics Education*, 23(2), 159-187.

- Kilhamn, C., Bråting, K., & Rolandsson, L. (2021). Teachers' arguments for including programming in mathematics education. I G.A. Nortvedt, N.F. Buchholtz, J. Fauskanger, F. Hreinsdóttir, M. Häikiöniemi, B. E. Jessen, ..., A. Werneberg (Red.) Bringing Nordic mathematics education into the future. Papers from NORMA 20. *Preceedings of the Ninth Nordic Conference on Mathematics Education*, 169–176. NCM & SMDF.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Misfeldt, M., Szabo, A., & Helenius, O. (2019). Surveying teachers' conception of programming as a mathematical topic following the implementation of a new mathematics curriculum. I U.T Jankvist, M. Van den Heuvel-Panhuizen, & M. Veldhuis (Red.), Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education CERME11 (s.2713-2720). Freudenthal Group & Freudenthal Institute, Utrecht University and ERME.
- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: a framework for teacher knowledge. *Teachers College Record*, 108(6), 1017–1054.
- Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, 33(3). <https://doi.org/10.14742/ajet.3521>
- National Council of Teachers of Mathematics (2000). Principles and Standards for School Mathematics. Reston, VA: NCTM
- Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and computers* (Vol. 17). Springer Science & Business Media.
- 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>
- NESH (2021, 16. desember). Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora. <https://www.forskningsetikk.no/globalassets/dokumenter/4-publikasjoner-som-pdf/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora.pdf>
- Olsson, H., Sörensen, S., & Bureid, G. (2003). *Forskningsprosessen: Kvalitative og kvantitative perspektiver*. Oslo: Gyldendal akademisk.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful ideas*. New York: Basic Books.
- Pörn, R., Hemmi, K., & Kallio-Kujala, P. (2021). "Programming is a new way of thinking" – teacher views on programming as a part of the new mathematics curriculum in Finland. I Y. Liljekvist, L. Björklund Boistrup, J. Häggström, L. Mattsson, O. Olande, H. Palmér (Red.), Sustainable mathematics education in a digitalized world. Proceedings of MADIF12. SMDF.
- Rabardel, P. (2002). People and technology: a cognitive approach to contemporary instruments. Université Paris 8. Hal01020705
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.
- Ringdal, K. (2013). *Enhet og mangfold* (3.utg.). Fagbokforlaget.

- Sevik, K. m. f. (2016). Programmering i skolen. I S. f. I. i. utdanningen (Red.). Utdanningsdirektoratet. Hentet fra https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Shulman, L. S. (1986). Those Who Understand: Knowledge Growth in Teaching. *Educational Researcher*, 15(2), 4-14. <https://doi.org/10.3102/0013189X015002004>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Stenseth, B., Kaufmann, O. T. & Forsström, S. E. (2019). Programmering og matematikk. *Tangenten – tidsskrift for matematikkundervisning*, 30(2), 7-12
- Stephens, M., & Kadujevich, D. M. (2020). Computational/Algorithmic Thinking. I S. Lerman (Red.), *Encyclopedia of Mathematics Education* (s. 117-123). London: Springer.
- Sundberg, M. (2015). A Study of Mathematics Teachers Conceptions of Their Own Knowledge of Technological Pedagogical Content Knowledge (TPACK). I *Views and Beliefs in Mathematics Education* (s.159-169). Springer Spektrum, Wiesbaden.
- Tabach M., Trgalová J. (2019) The Knowledge and Skills that Mathematics Teachers Need for ICT Integration: The Issue of Standards. I G. Aldon & J. Trgalová (Red.) *Technology in Mathematics Teaching. Mathematics Education in the Digital Era*, 13, 183-203. Springer, Cham. https://doi.org/10.1007/978-3-030-19741-4_8
- Thagaard, T. (2018). *Systematikk og innlevelse: en innføring i kvalitative metoder* (5.utg.). Fagbokforlaget.
- Thomas, M. O. J., & Hong, Y. Y. (2005b). Teacher factors in integration of graphic calculators into mathematics learning. I H. L. Chick & J. L. Vincent (Red.), *Proceedings of the 29th conference of the international group for the psychology of mathematics education* 4, 257-264. Melbourne: University of Melbourne.
- Thomas, M. O. J., & Palmer, J. M. (2014). Teaching with digital technology: Obstacles and opportunities. I A. Clark-Wilson et al. (Red.), *The Mathematics Teacher in the Digital Era* (s.71-89). Springer International Publishing.
- Thurm, D., Barzel, B. (2022) Teaching mathematics with technology: a multidimensional analysis of teacher beliefs. *109(1)*, 41-63. <https://doi.org/10.1007/s10649-021-10072-x>
- Tjora, A. (2021). *Kvalitative forskningsmetoder i praksis* (4.utg.). Gyldendal akademisk.
- Trouche, L. (2004). Managing complexity of human/machine interactions in computerized learning environments: Guiding students' command process through instrumental orchestrations. *International Journal of Computers for Mathematical Learning*, 9, 281-307. <https://doi.org/10.1007/s10758-004-3468-5>
- Utdanningsdirektoratet. (2019a). *Læreplan i matematikk 1.-10. trinn (MAT01-05)*. <https://www.udir.no/lk20/mat01-05?lang=nob>
- Utdanningsdirektoratet. (2019b) *Kompetanse i fagene*. <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/kompetanse-i-fagene/?lang=nob>
- Utdanningsdirektoratet. (2019c, 27. mars). *Algoritmisk tenkning*. <https://www.udir.no/kvalitetog-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2021, 31. august). *Hvordan ta i bruk nye læreplaner?* <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvordan-ta-i-bruk-lareplanen/>

- Verillon, P., & Rabardel, P. (1995). Cognition and artifacts: A contribution to the study of thought in relation to instrumented activity. *European Journal of Psychology of Education, 10*(1), 77-101. <http://www.jstor.org/stable/23420087>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
- Yelland, N. (1995). Mindstorms or a storm in a teacup? A review of research with Logo. *International Journal of Mathematical Education in Science and Technology, 26*(6), 853-869. <https://doi.org/10.1080/0020739950260607>

Vedlegg

Vedlegg 1 – Godkjenningsbrev fra NSD

12.05.2022, 11:05

Meldeskjema for behandling av personopplysninger



Vurdering

Referansenummer

340909

Prosjekttittel

Masterprosjekt lærerutdanning

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for samfunns- og utdanningsvitenskap (SU) /
Institutt for lærerutdanning

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Melih Turgut, melih.turgut@ntnu.no, tlf: 73558959

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Signe, signe.sandengenk@hotmail.com, tlf: 47247866

Prosjektperiode

01.01.2022 - 30.06.2022

Vurdering (1)

22.12.2021 - 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 22.12.2021, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 30.06.2022.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra foresatte til behandlingen av personopplysninger om barna. 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 foresatte kan trekke tilbake. Barna vil også gi sin tilslutning til deltakelse.

For alminnelige personopplysninger vil lovlig grunnlag for behandlingen være foresattes samtykke, jf. personvernforordningen art. 6 nr. 1 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 foresatte/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 og deres foresatte 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/foresatt tar kontakt om sine/barnets 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).

Ved bruk av databehandler (spørreskjemaleverandør, skylagring eller videosamtale) må behandlingen oppfylle kravene til bruk av databehandler, jf. art 28 og 29. Bruk leverandører som din institusjon har avtale med.

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

MELD VESENTLIGE ENDRINGER

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

OPPFØLGING AV PROSJEKTET

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

Kontaktperson hos NSD: Silje Fjelberg Opsvik

Lykke til med prosjektet!

Vedlegg 2: Informasjonsskriv og samtykkeerklæring

Vil du delta i forskningsprosjektet

Hva er læreres pedagogiske teknologiske kunnskap (PTK) i en programmeringskontekst i grunnskolen?

Jeg heter Signe Sandengen Kveseth og jobber med en masteroppgave i matematikdidaktikk ved lærerutdanningen på NTNU Trondheim. Prosjektet handler om programmering i grunnskolen, og hvilke erfaringer, orienteringer og kunnskap lærere har om programmering i matematikkundervisning. Formålet er å undersøke lærerkonstruksjon av PTK (pedagogisk og teknologisk kunnskap) for å kunne si noe om hvordan programmering integreres i matematikkundervisningen og hvilke faktorer som påvirker dette. For å finne svar på dette, ønsker jeg intervju med lærere som har erfaring med programmering i matematikkundervisning i grunnskolen.

Det er masterstudent og veileder ved NTNU som vil ha tilgang til personidentifiserbart data (blant annet navn i forbindelse med samtykkeerklæring og lydopptak). Alle navn og personidentifiserbart materiale vil bli anonymisert i oppgaven og i publikasjonen.

Hvem er ansvarlig for forskningsprosjektet?

Behandlingsansvarlig for forskningsprosjektet er Institutt for lærerutdanning ved Norges teknisk-naturvitenskapelige universitet, Trondheim.

Hva innebærer det for deg å delta?

Deltakelse i et 20-40 minutters intervju. Intervjuet kommer til å være semistrukturert. Det betyr at jeg allerede har forberedt noen hovedspørsmål innenfor ulike kategorier relevant til masteroppgavens problemstilling. I tillegg kan jeg stille oppfølgingsspørsmål til interessante og relevante innspill. Jeg vil bruke en digital lydopptaker i intervjuet. Lydopptak slettes etter transkribering.

Intervjuet inneholder spørsmål om dine erfaringer, orienteringer og pedagogisk og matematisk kunnskap om undervisning. Alt knyttet til programmering og programmeringsverktøy.

Dersom du ønsker å få tilsendt intervjuguiden på forhånd så kan du be om det, men det er ingen krav om å stille forberedt til intervjuet.

Det er frivillig å delta

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

Ditt personvern – hvordan 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.

- Det er masterstudent og veileder ved institutt for lærerutdanningen, NTNU i Trondheim som vil ha tilgang
- Navnet og kontaktopplysningene dine vil jeg erstatte med en kode som lagres på egen navneliste adskilt fra øvrige data
- Annet datamateriale vil jeg lagre på NTNUs forskningsserver

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er rundt mai/juni 2022. Lydopptak vil slettes etter transkribering, mens andre personopplysninger vil slettes etter prosjektslutt. Datamaterialet er anonymisert i oppgaven og dermed også ved publikasjon.

Dine rettigheter

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

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

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra institutt for lærerutdanning, NTNU Trondheim har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?

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

- Institutt for lærerutdanning, NTNU ved Melih Turgut, 73558959 eller melih.turgut@ntnu.no
- Vårt personvernombud: Thomas Helgesen, 93079038 eller thomas.helgesen@ntnu.no

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

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

Med vennlig hilsen

Melih Turgut
(Forsker/veileder)

Signe Sandengen Kveseth
(Student)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet *hva er læreres pedagogiske teknologiske kunnskap i en programmeringskontekst på barneskolen?* og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i intervjuet
- at [Signe Sandengen Kveseth] kan gi opplysninger om meg til prosjektet – hvis aktuelt

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

(Signert av prosjektdeltaker, dato)

Vedlegg 3: Intervjuguide

	Spørsmål	Hensikt
Innledende	Introduksjon til forskningen og forklaring av målet med intervjuet. Forklare hvilke rettigheter deltakeren har og få personen til å bekrefte at det er greit med lydopptak.	For å få praten i gang med deltakeren og sørge for at han/hun forstår hensikten med intervjuet
	<ul style="list-style-type: none"> - Hva er din alder? - Hva er din høyeste utdanningsgrad? Master? Evt. tema - Hvor lenge har du jobbet som lærer, eventuelt vikar? - Hvor mange år har du undervist i matematikk? - Har du undervist i andre fag? Hvor lenge? 	Skaffe grunnleggende informasjon om deltakeren
Teknologisk og pedagogisk	<ul style="list-style-type: none"> - Bruker du programmeringsverktøy i matematikkundervisning? For eksempel Scratch, Python, Micro:bit etc. - Når ble du introdusert for programmering og hvor lærte du deg det? - Hva var det mest utfordrende mens du lærte deg å bruke programmeringsverktøyene? - Har du klart å møte noen av disse utfordringene? Hvordan? - Bruker du heller programmeringsverktøy enn undervisning? For eksempel ved beregninger. - Hvordan organiserer du timene dine når du bruker programmering i matematikkundervisning. Hvorfor på denne måten? - Hva er de største utfordringene elevene møter på underveis? Hvorfor akkurat disse tror du? - Hva er de største utfordringene ved å bruke programmeringsverktøy i matematikkundervisning? - Hvordan vet du om elevene forstår idéene eller konseptene du forsøker å lære bort gjennom programmering? Kan du begrunne valg av vurderingstyper? 	For å kartlegge hva deltakeren bruker i undervisning og hvordan vedkommende har blitt introdusert for disse og programmering generelt
Refleksjon	<ul style="list-style-type: none"> - Hvis du ser tilbake på hvilke verktøy du har brukt og hvordan du har lagt opp undervisning i programmering, hva oppfattet du som positivt/nyttig? - Hva oppfattet du som negativt/utfordrende? - Hvilke av dine identifiserte fordeler og utfordringer er mest relevant/viktig for deg og ditt lærerskap? Hvorfor? 	For å få deltakeren til å evaluere og reflektere over egen undervisning fra positive og negative sider, der han/hun har brukt programmering.

	Spørsmål	Hensikt
Personlig	<ul style="list-style-type: none"> - Kan du beskrive hva du legger i begrepet programmering? - Hvilke kompetanser mener du programmering er knyttet til i matematikkundervisning? - Hvilke muligheter ser du for programmering i skolen? - Etter din mening, hva er hovedgrunnen til at elever bør lære seg grunnleggende programmering? - Hva tenker du om programmering som en del av læreplanen i matematikk? 	For å få en dypere forståelse av deltakerens forhold mellom verdi og teknologi/programmering
Avsluttende	<ul style="list-style-type: none"> - Er det noe du ønsker å legge til det vi har snakket om? - Er det noen temaer eller spørsmål du hadde forventet å bli spurt om, som jeg ikke har gjort i løpet av intervjuet? 	Sørge for at deltakeren kanskje kommer på flere innspill og tanker som kan bidra til forskningen

