

Erik Atle Løken

Læreres bruk av programmering i grunnskolen

En kvalitativ studie om matematikklæreres bruk
av programmering i grunnskolen

Masteroppgave i matematikdidaktikk (5-10)

Veileder: Øistein Gjøvik

Mai 2022

Erik Atle Løken

Læreres bruk av programmering i grunnskolen

En kvalitativ studie om matematikklæreres bruk av
programmering i grunnskolen

Masteroppgave i matematikdidaktikk (5-10)
Veileder: Øistein Gjøvik
Mai 2022

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



Kunnskap for en bedre verden

Sammendrag

Programmering ble en obligatorisk del av læreplanen og matematikkfaget fra og med skoleåret 20/21 gjennom fagfornyelsen. En av grunnene til å inkludere programmering var at det ble ansett for å være en viktig digital ferdighet å mestre for fremtidens samfunn. Det er gjort lite forskning på hvordan programmering blir brukt, og hvilken rolle programmering har i matematikkfaget i norsk kontekst. Derfor har jeg inntatt et lærerperspektiv, og undersøkt hvordan lærere bruker programmering. Formålet med studien var å beskrive kjennetegn ved undervisningstilnærmingen til et utvalg matematikklærere i programmering

Problemstillingen ble belyst ved hjelp av en kvalitativ tilnærming. Utvalget bestod av tre matematikklærere fra både mellomtrinn og ungdomstrinn. Gjennom studiens teoretiske rammeverk ble lærerne vurdert til å være kompetente i didaktisk bruk av programmering. Datainnsamlingen bestod av semi-strukturerte intervjuer med hver enkelt av lærerne. Datamaterialet ble analysert gjennom deduktiv tematisk analyse.

Funnene viser at de intervjuede lærernes undervisningstilnærming til programmering kjennetegnes ved at de ser en epistemologisk verdi i programmering, fremfor å bare se programmering som et verktøy med pragmatisk verdi. De tilstreber også utforskende programmeringsarbeid, med lite begrensende rammer, der metodefrihet og elevenes autonomi individuelt eller i gruppe er sentralt. Lærerne gir også uttrykk for en tilnærming til programmering der synet på lærerrollen kan beskrives som «liberalt», fordi det bryter med det tradisjonelle synet på lærerrollen.

Abstract

Programming became a compulsory part of the curriculum and mathematics subject from the school year 20/21 through «fagfornyelsen». One of the reasons for including programming were that it was considered an important digital skill to master for the future. There is scarce research on how programming is used, and what role programming has in the mathematics subject in the Norwegian context. Therefore, I have taken the perspective of teachers, and investigated how teachers use programming in mathematics. The purpose of this study is to describe the characteristics of a selection of mathematics teachers approach to teaching programming.

The study had a qualitative approach. The sample consisted of three mathematics teachers from both intermediate and lower secondary school. Through the study's theoretical framework the teachers were considered to be competent in the didactic use of programming. The data collection consisted of semi-structured interviews with each of the teachers which were later transcribed. The data material was analysed through deductive thematic analysis.

The study shows that the teachers' approaches to teaching programming are characterized by the fact that they see an epistemological value in programming, rather than just seeing programming as a tool with pragmatic value. They also strive for exploratory work in programming, where freedom of method and students' autonomy individually or in groups is a key part. The teachers also expressed an approach to programming where the view of the teacher's role can be described as "liberal", because it breaks with the traditional view of the teacher role.

Forord

Studietiden går mot slutten, og innleveringen av masteroppgaven symboliserer overgangen til en ny fase i livet. Gjennom de snart 5 årene jeg har bodd i Trondheim har jeg møtt mange fantastiske mennesker fra hele landet, og det er jeg utrolig takknemlig for. Det har vært både oppturer og nedturer gjennom studietiden, men helt klart flest oppturer. Erfaringene mine fra de siste 5 årene kommer jeg til å ta med meg videre i livet, og det vil utvilsomt være en positiv påvirkning i jobb og livet ellers.

Jeg vil rekke en stor takk til veilederen min, Øistein Gjøvik, for konstruktive tilbakemeldinger som har vært til stor hjelp. Jeg ønsker også å takke lærerne som bidro til denne studien med sine tanker og erfaringer. Videre har jeg satt pris på gjengen på lesesalen, dere har gjort skriveprosessen lettere. Til slutt vil jeg takke venner, familie og kjæreste for oppmuntrende ord, jeg hadde ikke klart å skrive oppgaven uten dere!

Trondheim, Mai 2022

Erik Atle Løken

Innholdsfortegnelse

1	Innledning	1
1.1	Programmering i skolen	1
1.2	Læreres bruk av programmering	2
1.3	Forskningsspørsmål	3
1.4	Bidrag til forskningsfeltet	3
1.5	Studiens oppbygning	3
2	Teori	4
2.1	Fagfornyelsen	4
2.2	Programmering i skolen – En oversikt	5
2.2.1	LOGO	5
2.2.2	LEGO Mindstorms	5
2.2.3	First LEGO League	6
2.2.4	Blokk-basert programmering	6
2.3	Programmeringsbegrepet	7
2.4	Teoretisk rammeverk	8
2.4.1	Pedagogisk teknologikunnskap	9
2.4.2	Matematisk Kunnskap for Undervisning	10
2.4.3	Personlige holdninger og verdier	10
2.4.4	Instrumentell genesis	10
2.4.5	Dobbel genesis	12
2.5	Tidligere forskning	13
2.5.1	Programmering i skolen	13
2.5.2	Teknologi og lærere	14
3	Metode	15
3.1	Metodisk tilnærming	15
3.1.1	Paradigme	15
3.1.2	Kvalitativ tilnærming	15
3.1.3	Fenomenologi	16
3.2	Utvalg	16
3.3	Datainnsamling	17
3.3.1	Intervju som datainnsamlingsmetode	17
3.3.2	Semi-strukturert intervju	17
3.3.3	Utforming av intervjuguide	18
3.4	Dataanalyse	19

3.4.1	Tematisk analyse	19
3.4.2	Analyseprosessen	20
3.5	Studiens Kvalitet.....	22
3.5.1	Validitet og reliabilitet	22
3.5.2	Etikk	23
3.5.3	Forskerrollen	24
4	Analyse	25
4.1	Lærernes pedagogiske teknologikunnskap.....	25
4.1.1	Janne	26
4.1.2	Lars	27
4.1.3	Simen	29
4.1.4	Lærernes pedagogiske teknologikunnskap.....	30
4.2	Programmeringens epistemologiske verdi	30
4.2.1	Janne	31
4.2.2	Lars	32
4.2.3	Simen	32
4.3	Utforskende programmeringsarbeid og metodefrihet	34
4.3.1	Janne	34
4.3.2	Lars	35
4.3.3	Simen	36
4.4	Et liberalt syn på lærerrollen	36
4.4.1	Janne	37
4.4.2	Lars	37
4.4.3	Simen	38
4.5	Oppsummering av funn	39
5	Drøfting.....	41
5.1	Programmeringens epistemologiske verdi	41
5.2	Utforskende undervisning	43
5.3	Liberalt syn på lærerrollen	45
5.4	Programmering og matematikk.....	46
6	Avslutning	49
6.1	Konklusjon.....	49
6.2	Studiens bidrag til forskningsfeltet	50
6.3	Forslag til videre forskning	51
	Litteraturliste:	52
	Vedlegg	56

Figurer

Figur 2.1: Et eksempel på et blokk-basert program laget i Scratch.	6
Figur 2.2: Oversikt over algoritmisk tenkning (Utdanningsdirektoratet, 2019).....	8
Figur 2.3: Oversatt modell av PTK (Thomas & Hong, 2013, s. 70).	9
Figur 2.4: Oversatt modell av instrumentell genesis (Guin & Trouche, 1998, s. 202). ..	11
Figur 3.1: Utforming av intervjuguide (egne oversettelser) (Bryman, 2016, s. 470). ...	18
Figur 3.2: Kodene analysen tok utgangspunkt i.....	21

Tabeller

Tabell 3.2: Oversikt over antall datasegmenter i hver av kodene.....	22
--	----

1 Innledning

1.1 Programmering i skolen

Teknologi blir en stadig større del av samfunnet, og på bakgrunn av den teknologiske utviklingen så Europakommisjonen et behov for å hjelpe utdanningsinstitusjoner i Europa med å tilpasse seg et samfunn i rask endring. De lanserte en aksjonsplan der digitale verktøy skulle brukes for å forbedre undervisning, og elever skulle få relevante kunnskaper og ferdigheter for å kunne mestre overgangen til en helteknologisk hverdag (Bocconi et al., 2018). Begrepet «21st century skills» blir brukt av blant annet OECD, og presenterer hvilke ferdigheter som blir viktige i fremtidens jobbmarked (Schleicher, u.å.). Ferdighetene blir ansett som å være grunnleggende for elever i den moderne skolen, og inneholder blant annet kritisk tenkning, samarbeid, kreativitet og teknologisk kyndighet. Introduksjonen av 21st century skills viste at skolen, i likhet med resten av samfunnet, opplever rask endring der det stilles nye krav til hva en person må kunne. En av de nye kunnskapsområdene som ble innført i skoler i Europa var programmering. Selv om Papert (1980) forsøkte å introdusere programmeringens potensial for læring i matematikk allerede tidlig på 80-tallet, ble ikke programmering en obligatorisk del av matematikkfaget i Norge før fagfornyelsen trådte i kraft.

I Norge ble det i 2016 fremmet et forslag om å fornye læreplanene i skolen (Meld. St. 28 (2015-2016), s.9). Forslaget hadde rot i jobben som ble gjort av «Ludvigsen-utvalget» (NOU 2015:8), og utvalgets oppgave var å vurdere hvorvidt innholdet av de gjeldende læreplanene var oppdaterte, og gav elever relevant kompetanse. De konkluderte med at fagene måtte fornyes for å «møte fremtidige kompetansebehov i arbeids- og samfunnslivet» (NOU 2015:8, s. 8). Som følge av forslaget startet arbeidet med fagfornyelsen, og i den forbindelse oppnevnte Utdanningsdirektoratet «Sanne-utvalget». Sanne-utvalget (2016) skulle undersøke teknologi i skolen, og blant annet vurdere hvordan teknologi ble utnyttet, og gi anbefalinger for hvorvidt programmering skulle innføres som en del av læreplanen. De konkluderte med at programmering var en fremtidsrettet ferdighet, og at det burde bli en obligatorisk del av læreplanen gjennom fagfornyelsen (Sanne et al., 2016, s. 76). Til tross for at anbefalingen fra utvalget var å opprette et eget teknologifag (som inneholdt programmering), ble det bestemt at programmering skulle innføres gjennom matematikk, naturfag, musikk og kunst og håndverk. I matematikk ble det lagt inn et kompetansemål som eksplisitt omtalte programmering fra og med 5. trinn til og med 10. trinn (Kunnskapsdepartementet, 2019). Enkelte av kompetansemålene for 2. trinn, 3. trinn og 4. trinn omhandler riktignok arbeid med stegvise instruksjoner, variabler, løkker og vilkår.

Innføringen av programmering i matematikk hadde naturlig nok implikasjoner for matematikklærere, og førte med seg nye krav til kompetanse. Programmering har vært en del av Norske skoler i forkant av fagfornyelsen gjennom ulike prosjekter (FIRST® LEGO®, u.å.; Lær Kidsa Koding, u.å.). Prosjektene førte til et kunnskaps- og erfaringsmessig sprik, ettersom noen matematikklærere hadde tatt del i prosjektene, og hadde derfor erfaring og kompetanse knyttet til programmering, mens andre matematikklærere hadde sitt første møte med programmering påtvunget gjennom fagfornyelsen. I tiden før og etter innføringen av fagfornyelsen har det blitt stilt

spørsmålstegn ved matematikklæreres kompetanse i møtet med programmering (Bjørkeng, 2020; Johansen, 2020; Moreau, 2021).

1.2 Læreres bruk av programmering

Læreplanen gir få føringer for hvordan programmering skal praktiseres i matematikkfaget, og det er i stor grad opp til lærerne å avgjøre hvordan programmeringen skal brukes, og hva deres mål med programmering i matematikkundervisning er. Ettersom fagfornyelsen trådte i kraft så sent som skoleåret 21/22 (for 10.trinn), er det lite forskning på hvordan matematikklærere bruker programmering i sin undervisning i norsk kontekst. Dermed er det vanskelig å si noe om hvilken motivasjon lærere har for å bruke (eller ikke bruke) programmering, og hvilke fordeler eller muligheter de ser med programmering. Hva gjør lærere som bruker programmering som et didaktisk verktøy jevnlig? Hvilke koblinger ser de til matematikkfaget? Videre er det lite forskning i norsk kontekst på hvorfor noen lærere valgte å bruke programmering før det ble en obligatorisk del av læreplanen, eller hvorfor noen lærere fortsatt avstår fra å bruke programmering i matematikk, til tross for at læreplanen er tydelig på at det er en obligatorisk del av matematikkfaget.

Selv om læreplanen inneholder kompetansemål der programmering er nevnt eksplisitt, kommer det ikke klart frem i kompetansemålene hvilken rolle programmering er tiltenkt i faget. Noen kompetansemål knytter programmering sammen med matematikk, mens andre omhandler programmering som ferdighet. Det åpner for at lærerne selv kan tolke hvordan, og i hvilken grad, de bruker programmering i matematikk. En matematikklærer kan derfor velge å bruke programmering for å lære elevene om programmeringsspesifikke konsepter som løkker, vilkår eller variabler, der det er et poeng i seg selv at elevene skal lære hvordan de programmerer. Læreplanen åpner også for at programmering kan brukes som et digitalt hjelpemiddel der hensikten er at elever lærer matematikk ved å bruke programmering. Valget om å bruke programmering for å lære programmeringsspesifikke konsepter eller som en inngang for å forstå matematikk bedre er i dag i stor grad avhengig av lærerne som underviser.

Det er gjort lite forskning i norsk skole på læreres oppfatninger og bruk av programmering i matematikk. I Sverige var mange lærere var bekymret for sin egen kompetanse i forkant av deres innføring av programmering i læreplanen (Vinnervik, 2020). Den korte tiden fra avgjørelse til implementering hadde gjort det utfordrende for skolen og lærerne å holde følge med de nye kompetansekravene. Mange Svenske lærere så en sammenheng mellom matematikk og programmering, men gav uttrykk for å ikke ha tilstrekkelig kompetanse for å undervise i programmering (Misfeldt et al., 2019). Kilhamn et al (2021) gjennomførte En kvalitativ studie med 20 såkalte «early adopters» (lærere som er tidlig ute med ny teknologi) i svensk grunnskole hadde som formål å beskrive hvilke argumenter utvalget deres bruker for å argumentere for programmering som en del av skolen (Kilhamn et al., 2021). De fant ut at lærernes argumenter kunne deles i fire ulike kategorier: utvikle algoritmisk tenkning, øke elevens engasjement i faget, lære matematikk og programmering som et kraftig verktøy. De konkluderer med at den digitale revolusjonen skolen opplever i dag kan være med på å endre matematikkfaget i form av at det blir mer utforskende og praktisk. Denne endringen og den stadig økende bruken av teknologi spås som en stor utfordring for lærere som ikke er «early adopters» (Kilhamn et al., 2021).

1.3 Forskningsspørsmål

Det er naturlig å tenke at utfordringene som ble belyst av studiene i forrige delkapittel vil gjelde for norske lærere og. Ettersom mye av forskningen på læreres bruk av programmering omhandler lærere som har liten erfaring med programmering, har vi lite kunnskap om hvordan lærere som er vant med programmering bruker det. Derfor ønsker jeg å undersøke hvordan lærere som er kompetente i programmering benytter seg av teknologien i sin undervisning. Dette ønsker jeg å gjøre gjennom å belyse problemstillingen:

Hva kjennetegner undervisningstilnærmingen til et utvalg lærere med høy pedagogisk teknologikunnskap i programmering?

For å belyse denne problemstillingen vil jeg ta for meg to forskningsspørsmål:

- 1) *Hvordan gir et utvalg lærere uttrykk for sin egen pedagogiske teknologikunnskap?*
- 2) *Hva kjennetegner et utvalg lærernes undervisningstilnærming i programmering?*

I studien har jeg intervjuet tre erfarne matematikklærere som hadde brukt programmering i undervisning i forkant av fagfornyelsen. Datamaterialet består av semi-strukturerte intervjuer som hadde som formål å gi et bilde av lærerens tanker, holdninger og opplevelser med programmering, og samtidig gi et bilde av hvilke tanker de hadde om hvordan programmering ble brukt i deres undervisning.

1.4 Bidrag til forskningsfeltet

Dette er en enkeltstående kvalitativ studie, men kan bidra med å skape et utgangspunkt for å diskutere hvordan programmering kan utnyttes i undervisning. Ønsker vi at elevene skal reprodusere kode, ønsker vi at de skal utvikle sin problemløsningskompetanse, eller skal det brukes som en alternativ løsningsmetode i matematikk? Innsikten fra denne studien kan bidra til forskningsfeltet gjennom å beskrive hvordan programmeringskyndige lærere tilnærmer seg programmeringsarbeid i matematikk, og hvordan de utnytter programmering som en ressurs i sin undervisning. Resultatene kan også bidra gjennom å være et utgangspunkt for diskusjon blant lærere, lærerstudenter eller ansvarlige for profesjonsutvikling rundt om i skolen. Programmering er fortsatt en ny arbeidsmåte for mange, og for å utnytte programmeringens muligheter best mulig er det nødvendig at både nye og etablerte lærere kan bruke det i sin matematikkundervisning. På den måten kan resultatene fra denne studien være med på å forbedre kompetansehevingen på lærerutdanningen, slik at lærerne som begynner i yrket har et bilde av hva programmering kan og bør være. Det kan også være viktig innsikt for å forbedre alle former for etterutdanning eller profesjonsutvikling, der etablerte lærere bygger videre på sin kompetanse.

1.5 Studiens oppbygning

I kapittel 2 presenteres teori knyttet til fagfornyelsen og innføringen av programmering. Videre tar jeg for meg programmering i en skolekontekst, og legger frem forståelsen av programmeringsbegrepet studien tar utgangspunkt i. Videre vil jeg i kapittel 3 presentere og begrunne de metodiske valgene knyttet til utvalg, datainnsamling og analyseprosessen. Analysen presenteres i kapittel 4, og deretter drøftes funnene fra analysen i lys av det teoretiske rammeverket og relevant forskning i kapittel 5. I siste kapittel presenterer jeg konklusjonen, en vurdering av studiens bidrag til forskningsfeltet og forslag til videre forskning.

2 Teori

Kapittelet tar for seg fagfornyelsen, og hvordan programmering ble en obligatorisk del av grunnskolen gjennom blant annet matematikk. Deretter vil jeg presentere en kort historisk oversikt over programmering i skolekontekst. Oversikten er ikke en fullstendig oversikt alle programmeringsplattformer som har vært prøvd ut, men den tar for seg de viktigste frem til nå. Videre vil jeg gjøre rede for forståelsen av programmeringsbegrepet denne studien baserer seg på. Mot slutten av kapittelet presenteres studiens teoretiske rammeverk, og deretter tar jeg for meg tidligere forskning på området.

2.1 Fagfornyelsen

I 2016 fremmet Kunnskapsdepartementet et forslag (Meld. St. 28 (2015-2016), s.9) om å fornye læreplanene i skolen. Etter forslaget startet lærere, pedagoger og andre fagfolk arbeidet med *fagfornyelsen*. Fagfornyelsen hadde som formål å utvikle nye læreplaner som «forbereder elevene på fremtiden» (Utdanningsdirektoratet, 2021b). De nye læreplanene for 1. – 9. trinn ble innført fra skoleåret 2020/2021, og året etter ble læreplanene for 10. trinn tatt i bruk. Fornyelsen startet med å utvikle kjerneelementer i alle fag. Kjerneelementene beskriver sentrale «kunnskapsområder, metoder, begreper, tenkemåter og uttrykksformer» i faget (Utdanningsdirektoratet, 2021a). De nye læreplanene videreførte de grunnleggende ferdighetene fra LK06, men la et særskilt ansvar til samfunnsfag (digitale ferdigheter), norsk (lese- og skriveferdigheter) og matematikk (regneferdigheter) (Kunnskapsdepartementet, 2018)

I matematikkfaget består kjerneelementene av utforskning og problemløsning, modeller og anvendelser, resonnering og argumentasjon, representasjon og kommunikasjon, abstraksjon og generalisering og matematiske kunnskapsområder (Kunnskapsdepartementet, 2019). Fagfornyelsen innførte også programmering som en del av matematikk, naturfag, kunst & håndverk og musikk. Programmering nevnes ikke som et eget kjerneelement i de nye læreplanene for matematikkfaget, og det er heller ikke nevnt som del av kjerneelementene. Til tross for dette er programmering nevnt eksplisitt i kompetansemål fra og med 5. trinn til og med 10. trinn i matematikk.

Inkluderingen av programmering i matematikken var ikke et åpenbart valg. Sanne-utvalget ble oppnevnt av Utdanningsdirektoratet for å blant annet «beskrive hva gruppa mener grunnopplæringen bør gi av relevant kompetanse i teknologi i det 21. århundret og komme med forslag til endringer i grunnopplæringen» (Sanne et al., 2016, s.3). Utvalget ønsket å opprette et eget fag dedikert til å utvikle elevenes kyndighet med teknologi på bakgrunn av det de beskrev som et «misforhold mellom den teknologiske virkeligheten elevene møter, og det de lærer på skolen» (Sanne et al., 2016, s. 77). I rapporten anbefalte de at programmering skulle innføres i skolen gjennom et eget teknologifag, og at elevene behøvde «kompetanse i teknologi på egne premisser og ikke kun for å lære fag» (Sanne et al., 2016, s.75). Det argumenteres riktignok for at teknologi og programmering kan innføres som en del av de etablerte fagene, men at dette forutsetter at elevene får en praktisk innføring i programmering (Sanne et al., 2016, s. 76). Utvalget trakk frem lærerkompetanse, dedikerte timer og ressurser, og engasjement fra skoleeier og ledelse som sentrale faktorer for at innføringen av teknologi og programmering i skolen skal oppnå suksess. Lærerkompetanse i programmering er en

av fokusområdene i denne studien. Det argumenteres for at lærere behøver generell teknologisk kompetanse, og kompetanse i programmering, for å kunne undervise i et fag slik de anbefaler (Sanne et al., 2016, s. 77). Gjennom fagfornyelsen ble programmering som nevnt innført gjennom blant annet matematikk, og derfor vil kravene til programmeringskompetanse nå tillegges matematikklærere over hele Norge.

Mellom 5. – 10. trinn ble det inkludert ett kompetansemål som tok for seg programmering for hvert årstrinn. Kompetansemålene for 5. trinn og 8. trinn fokuserer på at elevene skal lære programmeringskonsepter som variabler, vilkår, løkker og utvikling av algoritmer. På de resterende trinnene det trukket en tydeligere kobling til matematikkfaget. Elevene skal gjennom programmering utforske geometriske figurer og mønstre (6. trinn), utforske data i tabeller og datasett (7. trinn), simulere tilfeldige forsøk og beregne sannsynlighet (9. trinn) og utforske matematiske egenskaper og sammenhenger (10. trinn) (Kunnskapsdepartementet, 2019).

2.2 Programmering i skolen – En oversikt

2.2.1 LOGO

Seymour Papert (1980) argumenterte for at elevene kunne lære matematikk gjennom programmeringsspråket LOGO. Han mente programmering åpnet opp for utforskende arbeid med matematikk, der læreren fungerer som en veileder, fremfor en underviser eller foreleser, gjennom «pensumfri læring» (egen oversettelse) (Papert, 1980, s. 31). LOGO er et programmeringsspråk som er designet rundt en skilpaddefigur som brukeren programmerer til å bevege seg på skjermen ved hjelp av forskjellige kommandoer. Når skilpadden beveget seg kunne den legge igjen et spor, slik at bevegelsen resulterte i en tegning. Skilpadden kunne for eksempel få kommandoen «RIGHT 90» (der skilpadden roter 90 grader mot høyre), eller «FORWARD 100» (der skilpadden beveget seg 100 enheter i retningen fronten pekte). På den måten kunne LOGO for eksempel brukes for å tegne geometriske figurer.

Med utgangspunkt i Piagets konstruktivistiske læringsteori hevdet Papert (1980) at datamaskinene og programmeringens inntog i verden åpnet for et nytt matematisk miljø, der elevene kunne «samhandle» med matematikken. Gjennom å utvikle geometriske mønstre eller figurer kunne elevene lære matematikk på en måte de aldri hadde gjort før. Han mente også at elevene kunne dra nytte av å tenke som en datamaskin for å forstå deler av den formelle matematikken. Papert lanserte LOGO som et verktøy med et stort didaktisk potensial i matematikk. I tiden etter at han gav ut boken «Mindstorms» (Papert, 1980) var det flere som trodde at programmering ville ha stor innvirkning på matematikkfaget (Misfeldt & Ejsing-Duun, 2015, s. 2525). Implementeringen av LOGO stod ikke til forventningene, og dette var et resultat av at elevene ikke fikk med seg den matematiske kunnskapen ut av arbeidet, og aktivitetene i LOGO ble derfor ikke-matematiske (Noss & Hoyles, 1992, Ainley et al., 2006; sitert i Misfeldt & Ejsing-Duun, 2015, s. 2525)

2.2.2 LEGO Mindstorms

LEGO og LOGO startet et samarbeid der de skulle kombinere programmeringen fra LOGO med roboter fra LEGO Technic (Üçgül, 2013). Gjennom flere år med utvikling ble LEGO Mindstorms lansert i 1998. Papert (1993; sitert i Üçgül, 2013) så aktiviteter og programmering med roboter som en god innfallsvinkel for å realisere læringsprinsippene fra Piagets konstruktivismen. Etter lanseringen i 1998 tok det ikke lang tid før LEGO

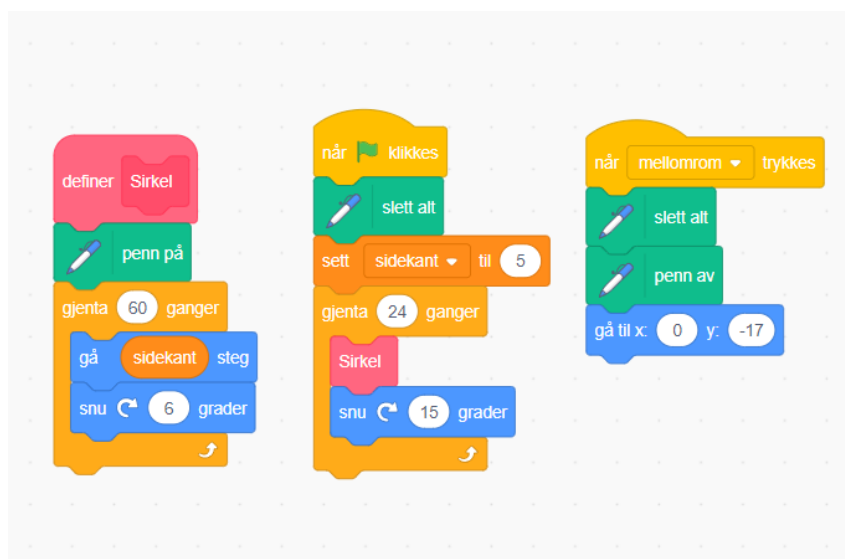
Mindstorms ble testet ved flere skoler i USA. Gjennom starten av 2000-tallet ble det gjennomført flere forskningsprosjekter som undersøkte læringspotensialet til LEGO Mindstorms, men mange av studiene opplevde blandede resultater (Üçgül, 2013). LEGO Mindstorms ble videreutviklet og oppdatert i årene etter lanseringen, og etter hvert utviklet LEGO egne versjoner som var ment for bruk i skolen.

2.2.3 First LEGO League

I First LEGO League Challenge (FLL) (FIRST® LEGO®, u.å.) skal elever programmere og bygge en robot med LEGO Mindstorms. Roboten skal løse en utfordring som er inspirert av virkelige problemer, og gruppene konkurrerer om å finne den beste løsningen. Konseptet ble etablert i 1999, og har i tiden etter blir eksportert til over 70 land. Formålet med FLL er å øke elever interesse for vitenskap og teknologi, samt fremme elevs kreativitet, innovasjon og samarbeid. I Norge blir FLL arrangert årlig med blant annet Equinor og Utdanningsdirektoratet som sponsorer. I forbindelse med fagfornyelsen utformet FLL et dokument (Hjernekraft, u.å) som sammenfattet hvilke kompetansemål i den nye læreplanen som arbeidet omfattet. FLL tilbys elever i tre ulike aldersgrupper (4-6 år, 6-9 år og 10-16 år), og for mange elever er dette deres først møte med programmering i skolen.

2.2.4 Blokk-basert programmering

I blokk-basert programmering (BBP) får brukeren tilgang til ulike blokker som representerer forskjellige kommandoer. Programmer bygges ved at brukeren kombinerer blokker med hverandre, og ulike blokker skiller gjerne med farger som indikerer hvilken kategori av kommandoer blokken tilhører (f.eks løkker, vilkår eller variabler) (Se figur 2.1). For å unngå feil i syntaks benytter BBP seg av en slags «puslespill-metamor» (Lin & Weintrop, 2021, s. 2). Elevene får en forståelse for hvilke blokker som kan kombineres, og hvilke blokker som ikke kan samhandle med hverandre (feil i syntaks). BBP gir dermed elever en tilbakemelding på hvilke kommandoer som brukes til hva, og hvordan disse samhandler med hverandre, og vil derfra kunne bygge forståelse for kode og prinsipper i programmering.



Figur 2.1: Et eksempel på et blokk-basert program laget i Scratch.

Andersen (2021) og hans kollegaer hevder BBP senker terskelen for koding, og at det gjør programmering tilgjengelig for en større del av elevgruppa. Til tross for at BBP har

eksistert siden midten av 80-tallet, har bruken av BBP-plattformer i skolen økt betraktelig som følge av utdanningsrettete BBP-plattformer som Scratch (MIT Media Lab, u.å) og Micro:Bit (Microsoft, u.å). I læreplanen for matematikk (Kunnskapsdepartementet, 2019) står det ikke spesifisert at det skal benyttes BBP, men det er en målsetning at elever skal arbeide med BBP på mellomtrinnet, og gjøre overgangen til tekstbasert programmering i løpet av ungdomsskolen (Kaufmann & Stenseth, 2021, s. 1033).

2.3 Programmeringsbegrepet

Begrepet *programmering* omfavner en hel rekke arbeidsmåter, konsepter og innhold på tvers av flere fagfelt. Begrepet blir brukt på ulikt vis, og har ulike meninger avhengig av hvilket fagfelt man er innenfor, og dermed blir det naturlig å presentere hvilken forståelse av begrepet denne studien baserer seg på. Forsström og Kaufmann (2019, s. 19) definerer programmering som «the process related to the development and implementation of instructions for computer programs so the computer can perform specific tasks, solve problems, and support human interactions». De hevder videre at programmering krever at elever har kunnskap om utforming, vurdering, implementering og analyse av algoritmer. Forsström og Kaufmanns (2019, s. 19) definisjon av programmering beskriver på mange måter det som ofte kalles «koding», nemlig arbeid med å utforme egen kode for å lage programmer. Koblingen mellom programmering og kode er åpenbar, der koding kan sees på som en del av samlebegrepet programmering, og i enkelte fagfelt vil mange kunne anse programmering og koding som det samme, nemlig å skrive eller utvikle sin egen kode eller egne programmer. Programmering i skolesammenheng innebærer utforming av egen kode, testing av kode, utforske hvordan algoritmer lages, endring av andres kode (remikse), feilsøking og tolking av kode.

I arbeidet med fagfornyelsen beskrev Sanne-utvalget (2016, s. 25) programmering som «en måte å forstå de grunnleggende mekanismene i digital teknologi» og «en tilegnelse av algoritmisk tenkning», og dermed utvider de definisjonen av programmering til å omhandle mer enn utforming av programmer (koding). Utvalget mener programmering er «en måte å forstå de grunnleggende mekanismene i digital teknologi, og en tilegnelse av algoritmisk tenkning som har likhetspunkter med matematisk logikk» (Sanne et al., 2016, s. 25). Algoritmisk tenkning (se figur 2.2) er den norske oversettelsen av begrepet *computational thinking*. Algoritmisk tenkning ble innført som en del av kjerneelementet «utforsking og problemløsning» i læreplanen (Kunnskapsdepartementet, 2019), og kan derfor sies å være en del av de sentrale kunnskapene og ferdighetene i matematikkfaget.



Figur 2.2: Oversikt over algoritmisk tenkning (Utdanningsdirektoratet, 2019)

Programmeringsbegrepet vil i denne studien være et samlebegrep for en rekke ulike programmeringsspråk og programmeringsplattformer. Videre i studien vil programmering bli sett på som en form for teknologi, med en tilhørende kompetanse. Lærere eller elever som har tilegnet seg denne programmeringskompetansen kjenner til grunnleggende konsepter i programmering (løkker, vilkår, variabler) på et eller flere av programmeringsspråkene eller plattformene de bruker. Arbeid med programmering vil innebære arbeid med å utvikle elevers programmeringskompetanse, utvikle matematisk kompetanse ved hjelp av programmering og utvikling av kompetansene som er beskrevet i algoritmisk tenkning ved hjelp av programmering. Dette er min tolkning for at forståelsen av begrepet skal tilpasses en matematikdidaktisk kontekst. På den måten omfavner «arbeid med programmering i matematikk» en hel rekke tilnærminger og aktiviteter. Dette anser jeg som viktig for at studien kan åpne for en bred forståelse av programmering. På den måten vil alle tilnærminger og til programmering kunne fanges opp.

2.4 Teoretisk rammeverk

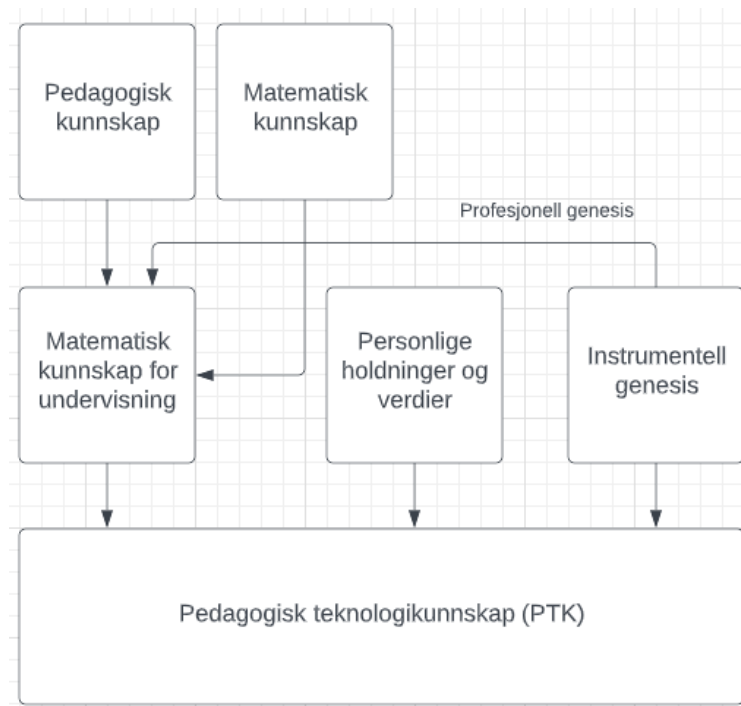
Tidlig i forskningsprosessen ønsket jeg å forklare hvordan lærerne tilnærmet seg undervisning med programmering gjennom rammeverket instrumentell genesis (f.eks Guin & Trouche, 1998), men med særlig vekt på Haspekians (2005, 2011, 2014) begrep «dobbel genesis». Utfordringen med rammeverket var at det ikke gav et godt nok bilde av lærernes kompetanse og undervisningstilnærming i programmering. Det kunne beskrive veien fra artefakt til didaktisk instrument, men tok ikke stilling til alle aspektene ved en læreres praksis. Jeg anså det derfor som nødvendig å ta i bruk et rammeverk som omfattet flere aspekter ved lærernes tilnærming. I studier som tar for seg læreres bruk av teknologi benyttes ofte rammeverk TPACK (f.eks Kramarski & Michalsky, 2015) rammeverket. I TPACK beskrives ulike kunnskaper som inngår i lærerens bruk av teknologi i en pedagogisk sammenheng. Kunnskapstypene har som formål å male et detaljert og fullstendig bilde av lærere «teknologiske pedagogiske fagkunnskap» (egen oversettelse) (Kramarski & Michalsky, 2015, s. 89). Jeg ønsket at lærerne skulle gi uttrykk for deres holdninger og tanker knyttet til innføringer av programmering, fordi jeg

hadde en formening om at påvirkningen var stor. TPACK beskriver ikke denne delen av praksisen, og jeg ble derfor nødt til å finne et annet rammeverk.

Til slutt falt valget på PTK-rammeverket som er utviklet av Thomas (Thomas & Hong, 2013; Thomas & Palmer, 2014) og hans kollegaer. Rammeverket beskriver læreres matematiske kunnskap for undervisning, deres personlige holdninger og verdier når det gjelder teknologi og en instrumentell genesisprosess (dobbel genesis), som sammen gir et bilde av lærerens pedagogiske teknologikunnskap (PTK). Derfor vil jeg benytte meg av PTK-rammeverket for å gi et bilde av lærerne, og hvordan PTK påvirker deres tilnærming til undervisning med programmering i matematikk. I tillegg vil jeg bruke instrumentell genesis rammeverket, og særlig dobbel genesis (Haspekian, 2005, 2011, 2014). Jeg vil komme tilbake til hvorfor dobbel genesis kan inkluderes i PTK-rammeverket senere i kapitlet.

2.4.1 Pedagogisk teknologikunnskap

Thomas og Hong (2005) problematiserte bruken av graf-kalkulatorer i matematikk. De mente det var et didaktisk potensial som ikke ble utnyttet, og at implementeringsgraden var lav til tross for at graf-kalkulatoren hadde vært et tilgjengelig hjelpemiddel i 15 år. De utviklet et rammeverk som hadde som formål å adressere utfordringer for læreres implementering og videre bruk av teknologi i matematikkundervisning (Thomas & Hong, 2013, s. 69). Rammeverket beskriver læreres *pedagogiske teknologikunnskap* (PTK) (egen oversettelse). PTK-rammeverket kan ifølge Thomas & Hong (2013, s. 69) brukes til å forklare lærers bruk av teknologi som en støtte for læring i matematikk, gi innsikt i læreres syn på teknologi som en del av matematikkundervisning og gi delvis innsikt i deres kjennskap til en gitt teknologi. En lærers pedagogiske teknologikunnskap er satt sammen av faktorene *matematisk kunnskap for undervisning* (egen oversettelse) (MKT), *personlige holdninger og verdier* (PO)(egen oversettelse), og *instrumentell genesis* (Thomas & Palmer, 2014) (Se figur 2.3).



Figur 2.3: Oversatt modell av PTK (Thomas & Hong, 2013, s. 70).

2.4.2 Matematisk Kunnskap for Undervisning

Faktoren *matematisk kunnskap for undervisning* (MKT) er satt sammen av to underfaktorer, *pedagogisk kunnskap* og *matematisk fagkunnskap* (egne oversettelser) (Thomas & Hong, 2013, s. 70). Ideen om lærerens pedagogiske kunnskap er hentet fra Shulmans (1987, s. 8) begrep «general pedagogical knowledge», som omhandler lærerens generelle pedagogiske strategier på tvers av fag. Det omfatter lærerens syn på hvordan elever lærer, og hvilke pedagogiske prinsipper som kjennetegner en lærers praksis. Faktoren matematisk fagkunnskap inneholder lærerens kunnskap om matematikk. Faktorene utgjør lærerens matematiske kunnskap for undervisning, som Thomas og Hong (2013) har hentet fra Hill og Ball (2004). Lærerens MKT kan for eksempel komme til syne i arbeid med funksjoner. Hva et bunnpunkt er, eller hvordan man finner stigningstallet til en graf, er lærerens matematiske fagkunnskap. Derimot hvorvidt hva som gir best læringsutbytte av å jobb individuelt eller i par for en gruppe elever vil omhandle lærerens pedagogiske kunnskap. Faktoren kan på mange måter sees på som «ryggraden» hos matematikklærere.

2.4.3 Personlige holdninger og verdier

Thomas og Palmer (2014, s.76) argumenterte for at læreres holdninger og verdier påvirker læreres praksis i større grad enn man tror, og at de derfor fortjente mer oppmerksomhet i forskning. De tar stilling til disse aspektene gjennom faktoren *personlige holdninger og verdier*. Lærerens personlige holdninger og verdier inneholder lærerens mål, holdninger, formeninger, verdier, preferanser, syn på matematisk kunnskap, syn på teknologi i matematikk og syn på hvordan matematikk bør læres (Clark-Wilson & Hoyles, 2019, s. 347). Faktoren tar utgangspunkt Schoenfelds (2011) rammeverk (*Resources, Orientations, og Goals*), der undervisnings sees på som en målstyrt aktivitet, og rammeverket forsøker å forklare hvorfor lærere tar valgene de gjør (Thomas & Palmer, 2014, s. 76). Faktoren kan komme til syne på ulikt vis.

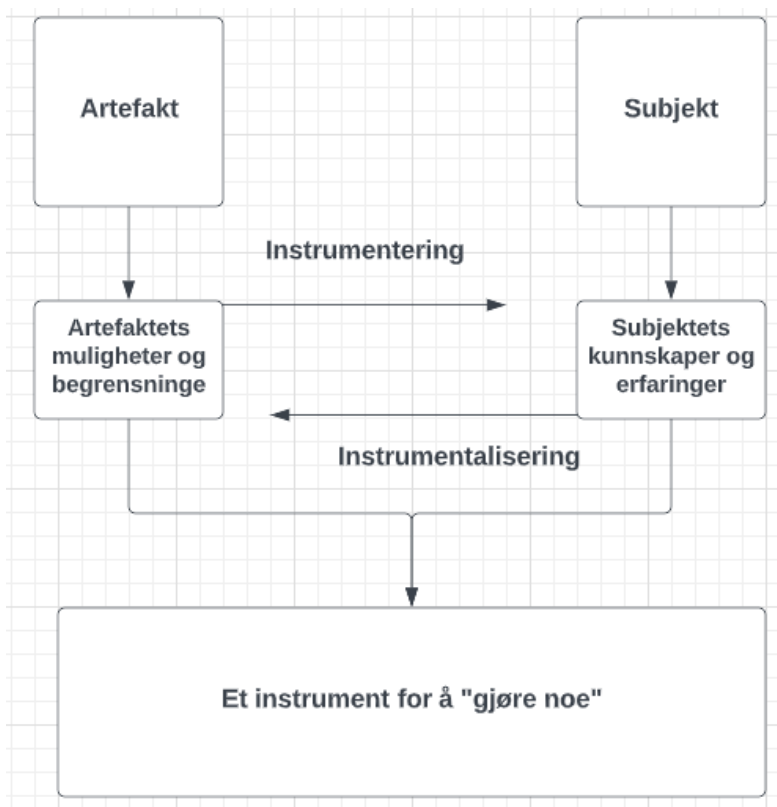
For eksempel vil en mer konservativ og en mer progressiv lærers erfaring med ny teknologi kunne være ulik, der den mer progressive læreren er mer tilbøyelig til å ta i bruk ny teknologi. Ved inntoget av en ny teknologi vil lærerens PO påvirke deres PTK. Den erfarne læreren vil kanskje ha en preferanse for penn og papir, og et syn på hvordan matematisk kunnskap tilegnes som ikke er forenelig med bruk av ny teknologi. Det kan føre til at teknologien ikke benyttes av den erfarne lærere, eller at den benyttes i noen få tilfeller. På den andre siden vil den mer progressive læreren tilnærme seg den nye teknologien med en formening om at teknologi er en nyttig didaktisk støtte i matematikk, og at matematikk kan læres gjennom å bruke teknologi i like stor grad som ved penn og papir. Da er det naturlig å tenke at den nye teknologien brukes mer. Resultatet vil i de fleste tilfeller være at den mer progressive læreren har høyere PTK sammenlignet med den mer konservative læreren.

2.4.4 Instrumentell genesis

Den siste faktoren i PTK-rammeverket er lærerens *instrumentelle genesis*. Instrumentell genesis er prosessen der et subjekt utvikler et *instrument* (et verktøy for å løse et problem) fra et *artefakt* (et menneskelaget symbolsk eller materielt objekt) (Guin & Trouche, 1998). Tanken om instrumentell genesis bygger på arbeidet til Verillon og Rabardel (1995) om instrumentert aktivitet. De skilte mellom artefakt og instrument, og poengterte at selv artefakter som var designet for å løse en spesifikk oppgave ikke ble et instrument før subjektet viste hvordan et problem kunne løses gjennom artefaktet. Det vil for eksempel si at en kalkulator, som er designet for blant annet multiplikasjon, forblir

et artefakt for elevene frem til de har utviklet kalkulatoren til et instrument for multiplikasjon. Siden starten av 2000-tallet har rammeverket blitt hyppig brukt i matematikdidaktisk forskning om bruk av digitale verktøy. Genesisprosessen forklares ut ifra delprosessene *instrumentering* og *instrumentalisering* (Se figur 2.4).

Trouche (2020a, 2020b) gir uttrykk for et forhold mellom prosessene, der instrumentering virker mot subjektet, og instrumentalisering virker mot artefaktet. I instrumenteringsprosessen utvikler subjektet *instrumenterte aksjonsskjemaer* (egen oversettelse). Instrumenterte aksjonsskjemaer utvikles kontinuerlig, og kan sees som handlinger som lar subjektet løse gitte oppgaver (Artigue, 2002, s. 250). Bozkurt et al (2018; sitert i Buteau et al., 2020, s. 373) mente at subjektet ikke hadde tilegnet seg artefaktet som et matematisk instrument før det hadde utviklet instrumenterte aksjonsskjemaer. Resultatet av instrumenteringsprosessen er på mange måter at subjektet vet hvilke problemer instrumentet kan løse, og hvilke teknikker som kreves. Det kan for eksempel være at en person lærer seg hvordan de kan bruke Python (tekstbasert programmeringsplattform) til å sjekke hvorvidt et vilkårlig tall er et primtall. Subjektet vil da ha instrumenterte aksjonsskjemaer som gjør det mulig å løse problemet «er X et primtall». Når subjektet lærer hvordan de bruker et instrument foregår det instrumentalisering. Subjektet utvikler *brukerskjemaer* (egen oversettelse) som omhandler hvordan artefaktet håndteres og brukes (Trouche, 2004; sitert i Buteau et al., 2020, 372), i form av grunnleggende handlinger på artefaktet (skru på, finne en spesifikk funksjon). Når subjektet har utviklet brukerskjemaer for artefaktet, og instrumenterte aksjonsskjemaer som lar den løse gitte problemer gjennom artefaktet, kan man si at subjektet har utviklet et matematisk instrument fra artefaktet.



Figur 2.4: Oversatt modell av instrumentell genesis (Guin & Trouche, 1998, s. 202).

For å belyse hvordan en instrumentell genesis prosess kan se ut vil jeg tar for meg et eksempel der en elev tilegner seg programmering (i form av blokkprogrammering) som et verktøy for å simulere sannsynlighet. Innledningsvis er programmeringen helt ukjent for elevene, og er da utelukkende et artefakt for elevene. Gjennom samhandlingen utforsker eleven programmeringsplattformen, og etter hvert endres elevens brukerskjemaer. Da lærer eleven hvordan programmeringsplattformen «funger», og hvordan den manøvrer seg på plattformen. Dette kan være å lære seg hvordan man setter ulike blokker sammen eller hvordan man finner spesifikk blokker og endrer de. Denne endringen blir «synlig» på artefaktet, og den viser derfor instrumentaliseringprosessen.

Videre vil eleven forsøke å benytte artefaktens muligheter og sin egen kunnskap for å få til en sannsynlighetssimulering med gitte betingelser. Gjennom denne samhandlingen utvikler elevene instrumenterte aksjonsskjemaer for hvordan eleven kan «simulere sannsynlighetsutfall X». Endringen av elevens instrumenterte aksjonsskjemaer tyder på en instrumenteringsprosess. Dette eksempelet viser på veldig generell basis hvordan en instrumentell genesisprosess kunne foregått.

2.4.5 Dobbel genesis

Dobbel genesis bygger på ideen om instrumentell genesis, men beskriver prosessen der et artefakt utvikler seg til å være et didaktisk verktøy for en lærer. Haspekian (2014) hevder at lærere konstruerer to ulike instrument fra ett og samme artefakt, et didaktisk instrument (profesjonell genesis) og et instrument for personlig bruk (personlig genesis). Ut ifra den forståelsen vil lærerens personlige genesis være svært lik elevenes instrumentelle genesis, der det handler om å utvikle et instrument for å løse problemer. Personlig- og profesjonell genesis utgjør Haspekians (2014) dobbel genesis, og beskrives gjennom begrepene instrumentalisering og instrumentering, slik som i instrumentell genesis, men begrepene innhold endrer seg noe når det kommer til profesjonell genesis.

Et artefakt instrumentaliseres gjennom at læreren oppdager artefaktets didaktiske potensial (Haspekian, 2011, s. 2301). Prosessen skjer kontinuerlig, og når en lærer bruker et artefakt som et didaktisk instrument vil den kunne oppdage nye didaktiske potensialer (og potensielt utvikle et nytt didaktisk instrument av det samme artefaktet). I instrumenteringsprosessen må læreren opprette nye undervisningsskjemaer, eller endre allerede eksisterende undervisningsskjemaer, for å gjøre artefaktet til en del av undervisningen (Haspekian, 2011, s. 2301). Læreren bygger en forståelse av når og hvordan verktøyet passer inn i sin undervisning, og vil gradvis justere undervisningsskjemaene.

For å belyse en dobbel genesis prosess kan man for eksempel se for seg en lærer som bruker blokkprogrammeringsplattformen «Scratch» for første gang. Blokk-basert programmering vil ikke passe inn i noen av lærernes etablerte undervisningsskjemaer, og må derfor gjennom en instrumenteringsprosess. Samtidig som læreren tester programmeringen vil den gjennom instrumentaliseringen kontinuerlig oppdage didaktiske potensial ved Scratch. Etter hvert instrumenterer læreren Scratch, og inkluderer det som en egnet i undervisning med sannsynlighet. Mens denne profesjonelle genesisen pågår tilegner også læreren seg et personlig instrument gjennom personlig genesis (slik det ble beskrevet i ordinær instrumentell genesis). Læreren sitter igjen med Scratch som et didaktisk instrument for å støtte elevens læring i sannsynlighet, og et personlig instrument for å løse problemer i sannsynlighet.

Innslaget av dobbel genesis vises gjennom den doble pilen på modellen av PTK. Thomas og Hong (2013, s. 69) forklarer dobbel genesis i PTK slik:

PTK grows as teachers progress through the stages of instrumentalisation and instrumentation of the tool, gaining a personal appreciation of its role in learning mathematics, and importantly, of ways in which students may be assisted through various teaching approaches to emulate their instrumental genesis of the technological tool. (Thomas & Hong, 2013, s. 69)

Genesisprosessen påvirker lærerens PTK direkte, ved å øke deres evner til å bruke en gitt teknologi (personlig genesis), men også indirekte, ved å endre lærerens undervisningsskjemaer (profesjonell genesis) og didaktiske «verktøykasse». Ideen om dobbel genesis som en del av PTK i denne studien er inspirert av Clark-Wilson & Hoyles (2019).

2.5 Tidligere forskning

Forskningen som presenteres vil fordeles på to delkapitler. Det første delkapittelet presenterer forskning på programmering i en skolekontekst. Deretter omhandler neste del læreres implementering og bruk av teknologi i matematikk.

2.5.1 Programmering i skolen

Stigberg & Stigberg (2020) undersøkte hvordan fire svenske lærere integrerte programmering i sin undervisning kort tid etter det ble en del av læreplanen. Lærerne i studien gir uttrykk for at kunnskapen deres i programmering stammer fra et introduksjonskurs, ressurser på internett og kollegialt samarbeid. Videre underviste lærerne til de yngste elevene for det meste i ulike un-plugged aktiviteter. Lærerne til de eldre elevene brukte blokk-basert programmering (ScratchJr), og noen tok i bruk Python for de aller eldste elevene. Stigberg & Stigberg (2020) mener lærerne har god kunnskap om den nye læreplanen, men at de trenger støtte for å bygge sin undervisningskompetanse i programmering og deres forståelse av forholdet mellom programmering og matematikk.

Forsström & Kaufmann (2019) gjennomførte en litteraturstudie av 15 studier fra ulike land i Europa, som undersøkte programmering i matematikkundervisning for elever i grunnskolen. De ønsket å undersøke hvilke potensial programmering som en del av matematikken kunne ha i skolen. De identifiserte av en rekke studier trakk frem programmering som en motiverende faktor, men at resultatene ikke var mulig å generalisere. Det samme var tilfellet for elevenes prestasjoner i matematikk, der flere studier rapporterte positive effekter av programmering, men resultatene var ikke generaliserbare. Forsström og Kaufmann peker også på viktigheten av lærerens rolle, og særlig inn mot arbeid i grupper. De viser at lærere som en «guide» fremmer problemløsning i grupper i større grad enn læreren som en foreleser. Det er uklart hvorvidt gruppearbeid fører til økt læring i programmering. De hevder det trengs mer forskning på kollektive læringsprosesser i arbeid med programmering og hvordan lærerens rolle påvirker elevenes læringsprosesser.

Andersen et al (2021) undersøkte elevers kollektive læringsprosesser i realfagene gjennom arbeid med blokk-basert programmering i grupper. Utvalget deres bestod av totalt 43 elever mellom 12 og 16 år, fra 3 ulike skoler. De rapporterte at elever ikke trengte avansert programmeringskompetanse for å løse problemer med blokk-basert programmering. De mener dette skyldes blokkprogrammeringens visuelle og intuitive

design som minsker utfordringer knyttet til syntaks. Programmering trekkes også frem som et egnet utgangspunkt for utforskende arbeid i grupper, ved at programmeringen hadde en medierende og konkretiserende effekt.

Popat og Starkey (2019) undersøkte læringsutbytte elever hadde av koding. De gjorde en litteraturstudie av 10 artikler og konkluderte med at koding hadde et stort læringsutbytte forbi selve kodingen for elevene. Utbyttet omhandlet blant annet problemløsningsferdigheter i matematikk, kritisk tenkning, sosiale ferdigheter og autonomi.

2.5.2 Teknologi og lærere

Misfeldt et al (2019) undersøkte læreres forståelse av sammenhengen mellom programmering og matematikk. Utvalget deres bestod av 133 svenske matematikklærere. De fant ut at lærerne i studien jevnt over mente det var tydelig sammenheng mellom programmering og matematikk, og at mange av lærerne ønsket å bruke programmering i matematikkundervisning, men de følte seg for lite forberedt. De konkluderer med at utvalget ikke er stort nok til å generalisere resultatene, men at resultatene kan gi en indikasjon på læreres tanker. De problematiserer at så mange lærere gir uttrykk for å ikke være forberedt på innføringen av programmering, og Misfeldt et al (2019) foreslår å utvikle læreres programmeringskompetanse med utgangspunkt i matematikkfaget.

Bray og Tangney (2017) gjennomførte en litteraturstudie der de undersøkte bruken av teknologi i matematikk. I studien undersøkte de totalt 139 artikler med formål om å beskrive trendene i teknologibruk. De konkluderer med at gode økter med teknologi kjennetegnes blant annet av utforsking, samarbeid og en lærer som fungerer som en tilrettelegger. De trekker også frem at oppgavene som brukes ofte baserer seg på et sosialkonstruktivistisk syn på læring, der samarbeid og problemløsning ligger til grunn.

Clark-Wilson & Hoyles (2019) undersøkte læreres bruk av *dynamisk matematisk teknologi* (egen oversettelse) (DMT) gjennom prosjektet «Cornerstone Maths». DMT beskrives som «technology with multiple, dynamically linked, mathematical representations that can be manipulated» (Clark-Wilson & Hoyles, 2019, s. 343). Utvalget deres bestod av 111 lærere fra 42 ulike skoler i London. De mener at det er behov for å utvikle en «verktøykasse» som lærerne kan støtte seg på i profesjonsutvikling knyttet til DMT. De mener dette kan overkomme de vanligste utfordringene lærerne gir uttrykk for (mangel på kunnskap og ressurser). De konkluderer med at tidlig i implementeringen kan det være hensiktsmessig at læreren er spesifikk og mindre ambisiøs, men at dette endrer seg med erfaring.

3 Metode

Gjennom hele forskningsprosessen er det tatt mange valg knyttet til forskningsdesign, utvalg, planlegging og gjennomføring av datainnsamling, analyse og etiske betraktninger. Først vil jeg presentere studiens metodiske tilnærming, og ta stilling til hvordan dette påvirker forskningsprosessen. Deretter presenteres utvalget og datainnsamlingen. I påfølgende delkapittel presenterer jeg deduktiv tematisk analyse som analysemetode, og gjør rede for hvordan analyseprosessen ble gjennomført. Til slutt tar jeg for meg etiske betraktninger og min påvirkning som forsker.

3.1 Metodisk tilnærming

Innledningsvis vil jeg presentere metodologien som studien tar utgangspunkt i, og drøfte hvordan valgene knyttet til metodologi kan påvirke forskningsprosessen.

3.1.1 Paradigme

Et paradigme kan sees på som forskerens «syn på verden» (Mackenzie & Knipe, 2006), og valg av paradigme påvirker hele forskningsprosessen (Kivunja & Kuyini, 2017, s. 26). Studien har som formål å beskrive kjennetegn ved lærernes undervisningstilnærming gjennom erfaringene, tankene og holdningene til praktiserende matematikklærere. På bakgrunn av dette mener jeg det fortolkende paradigmet er passende for min studie. Det fortolkende paradigmet har som hensikt å forstå deltakernes syn på det som undersøkes (Creswell, 2003 i Mackenzie & Knipe, 2006). Ved å ta utgangspunkt i det fortolkende paradigmet baserer jeg meg på en subjektiv epistemologisk posisjon. En subjektiv epistemologisk posisjon innebærer at forskeren gir mening til datamaterialet ut ifra egne erfaringer og meningen er farget av forskeren interaksjon med deltakerne (Kivunja & Kuyini, 2017, s. 33). Påvirkningen mine erfaringer og formeninger kan ha for resultatene diskuteres i kapittel 3.4.3 Forskerrollen. Valget av paradigmet medfører også en relativistisk ontologi, som vil si at forskeren arbeider ut ifra at det finnes flere virkeligheter, og at virkeligheten som undersøkes er et resultat av deltakernes oppfatninger (Kivunja & Kuyini, 2017, s. 33). Studier i det fortolkende paradigmet baserer seg ofte på kvalitative data fra intervjuer, observasjon eller dokumentanalyse (Mackenzie & Knipe, 2006). Denne studiens baserer seg data fra intervjuer med lærerne. I lys av denne studien vil det fortolkende paradigmet fortone seg gjennom at lærernes tanker, holdninger og erfaringer når det gjelder programmering som en del av matematikkfaget vil tolkes av meg, og deretter presentert. De gjør at lærernes meninger vil farges av min tolkning av deres meninger.

3.1.2 Kvalitativ tilnærming

En kvalitativ studie baserer seg ofte på ord fremfor tall-basert data (Bryman, 2016, s. 375). Tilnærmingen er ellers lite definert, og kan være vanskelig å gi en entydig definisjon, men den kan forklares slik:

[...] an inquiry process of understanding based on distinct methodological traditions of inquiry that explore a social or human problem. The researcher builds a complex, holistic picture, analyzes words, reports detailed views of informants, and conducts the study in a natural setting. (Creswell, 1998, s. 15)

Kvalitativ forskning baserer seg ofte på få tilfeller, og mange variabler, til forskjell fra kvantitativ forskning som ofte undersøker flere tilfeller og få variabler. Datamaterialet jeg skal undersøke stammer fra deltakernes utalte erfaringer, tanker og holdninger om programmering i matematikkundervisning, og dermed faller studien under den kvalitative tilnærmingen. Bryman (2016, s. 375) trekker frem at kvalitative studier ofte har et epistemologisk og ontologisk utgangspunkt som passer med det fortolkende paradigmet, og et induktivt syn på forholdet mellom teori og forskning. I denne studien baserer jeg meg riktignok på en deduktiv tematisk analyse.

3.1.3 Fenomenologi

Fenomenologi er en tilnærming innen kvalitativ forskning der formålet er å undersøke én eller flere individers opplevelse av et felles konsept eller fenomen (Creswell, 1998, s. 51). Innen den fenomenologiske tilnærmingen er det sentralt at forskeren skal beskrive den underliggende meningen til deltakernes erfaringer og tanker rundt et gitt konsept. Tjora (2018, s. 31) hevder det er to hovedspørsmål i fenomenologien: 1) hva betyr den sosiale verden for meg 2) hva betyr den sosiale verden for den jeg undersøker. Det første spørsmålet er sentralt å besvare i denne studien, etter som jeg baserer meg på et fortolkende paradigme. Det medfører at konklusjoner fra denne studien må sees i lys av min egen forståelse av verden. Jeg gjør rede for min påvirkning på studien senere i kapitlet. Denne studien ønsker å tolke lærernes beskrivelser av hvordan de ser verden for å belyse problemstillingen. Derfor vil Tjoras (2018, s.31) andre spørsmål besvares gjennom denne studien.

3.2 Utvalg

Kvalitativ forskning baserer seg ofte på et målrettet utvalg (Bryman, 2016, s. 410). Et målrettet utvalg består av deltakere som er valgt på bakgrunn av problemstillingen, og kriteriene for deltakerne har rot i at datamaterialet skal kunne besvare problemstillingen. På bakgrunn av de metodiske valgene jeg har presentert, og studien formål, er det passende å basere seg på et målrettet utvalg. Jeg ønsket at deltakerne skulle ha erfaring som matematikklærere både før og etter fagfornyelsen. Deltakerne måtte også ha erfaring med programmering fra før fagfornyelsen (helst fra matematikk, men jeg godtok erfaring fra for eksempel valgfag programmering). Bakgrunnen for dette var en målsetning om å finne matematikklærere med mye erfaring med programmering. Foruten disse to kriteriene så jeg ikke et behov for flere kriterier for å sikre data som kunne svare på forskningsspørsmålet.

Det er utfordrende for en forsker å avgjøre hvor stort utvalget bør være for å ha tilstrekkelig med data for å svare på forskningsspørsmålet (Bryman, 2016, s. 416). Størrelsen på utvalget i denne studien er valgt på bakgrunn av tid og tilgjengelighet. Med tidsrammen denne oppgaven er tildelt, er det nødvendig for studiens kvalitet at datamaterialet ikke blir så stort at det ikke kan gjennomarbeides. På samme måte kan det være utfordrende å finne deltakere som oppfyller kriteriene i en tid der behovet for deltakere er spesielt høyt som følge av antallet masteroppgaver som skrives i samme tidsrom som denne. Utvalget i denne studien består av tre lærere som oppfyller kravene som er nevnt tidligere. Alle deltakerne er ansatt ved ulike skoler i Trøndelag. Utvalget består av to menn og én kvinne. For å bevare deltakernes anonymitet har jeg tildelt hver deltaker et pseudonym. Her beskrives deltakernes i studien i korte trekk:

Janne (mellomtrinn):

Den første læreren underviser på mellomtrinnet i Trøndelag. Vedkommende har 16 års erfaring som matematikklærer, og underviser også i naturfag. Læreren fortalte at hun hadde brukt programmering i fagene sine før fagfornyelsen, blant annet gjennom LEGO League. Denne læreren blir tildelt pseudonymet «Janne».

Lars (ungdomsskole):

Den andre læreren underviser på ungdomstrinnet ved en ungdomsskole i Trøndelag. Vedkommende har 10 års erfaring som matematikklærer. Læreren beskriver seg selv som kyndig i programmering, og har bakgrunn som ingeniør. Han underviser i tillegg i valgfaget programmering. Denne læreren blir tildelt pseudonymet «Lars».

Simen (mellomtrinn):

Den tredje læreren underviser på mellomtrinnet ved en skole i Trøndelag. Vedkommende har 16 års erfaring som matematikklærer. Læreren gir uttrykk for å være en av de kyndige i programmering ved sin skole, og har erfaring med både blokk-basert og tekst-basert programmering. Denne læreren blir tildelt pseudonymet «Simen»

3.3 Datainnsamling

Studien baserte seg på data fra intervju med lærerne. Det ble gjennomført ett intervju med hver lærer. I dette delkapittelet presenterer jeg intervju som datainnsamlingsmetode, og begrunner hvorfor det ble valgt. Til slutt gjør jeg rede for hvordan intervjuguiden ble utformet.

3.3.1 Intervju som datainnsamlingsmetode

Intervju er en av de mest brukte datainnsamlingsmetodene i kvalitativ forskning (Bryman, 2016), og noe av grunnen til dette er at intervju er en fleksibel datainnsamlingsmetode. Intervjuer i kvalitativ forskning er ofte mindre strukturerte enn intervjuer i kvantitativ forskning, som gjerne følger en fastsatt intervjuguide. En kvalitativ forsker har derfor større mulighet til å vike fra intervjuguiden og stille relevante oppfølgings spørsmål som anses som interessante i intervjuøyeblikket (Bryman, 2016).

Med utgangspunkt i forskningsspørsmålet og studiens metodologi, så jeg det som hensiktsmessig å benytte intervju som datainnsamlingsmetode. Jeg ønsker å undersøke læreren opplevelser, tanker og meninger, og intervju som metode gir meg muligheten til å få rike beskrivelser fra deltakerne. Det var ikke et stort behov for standardiserte svar i studien, og derfor ikke et behov for en fastsatt intervjuguide. Jeg valgte likevel å utforme en intervjuguide som et utgangspunkt, samtidig som jeg åpnet opp for å stille oppfølgings spørsmål som kunne oppklare eller utforske deltakernes opplevelser, tanker og meninger. På bakgrunn av dette har jeg valgt å bruke et *semi-strukturert intervju* som datainnsamlingsmetode.

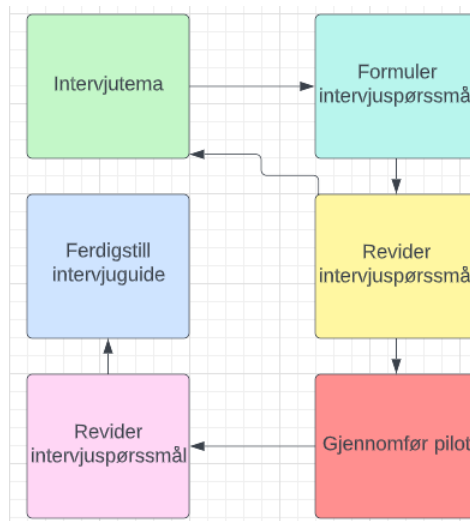
3.3.2 Semi-strukturert intervju

Innen kvalitative intervjuer skiller man ofte mellom ustrukturerte intervjuer og semi-strukturerte intervjuer (Bryman, 2016, s. 468). I et ustrukturert intervju vil intervjuet kunne sees på som en samtale mellom forskeren og deltakeren om utvalgte temaer, men som navnet tilsier er det ingen planlagte spørsmål, og intervjuet går den retningen

deltakeren tar det (Bryman, 2016, s. 468). I et semi-strukturert intervju baserer forskeren seg på noen planlagte spørsmål innen avgrensede temaer, men åpner opp for å stille spørsmål som ikke er skrevet i intervjuguiden hvis deltakeren kommer med interessante innspill forskeren mener bør utforskes i større grad (Bryman, 2016, s. 468). I semi-strukturerte intervjuer stilles de forberedte spørsmålene tilnærmet likt til alle som blir intervjuet (Bryman, 2016, s. 468), som medfører en viss form for standardisering. I studien har jeg benyttet en intervjuguide, der spørsmål og enkelte oppfølgingsspørsmål er utformet i forkant av intervjuene. Formålet med denne tilnærmingen var å kunne benytte det semi-strukturerte intervjuets fleksibilitet, samtidig som det åpnet opp for å kunne sammenligne deltakernes svar i etterkant. Jeg vil gjøre rede for utformingen av intervjuguiden i neste delkapittel.

3.3.3 Utforming av intervjuguide

I semi-strukturerte intervjuguider skal forskeren formulere spørsmål som tar for seg de sentrale temaene i forskningsspørsmålet, og spørsmålene skal gi deltakerne mulighet til å beskrive sin opplevelse av verden (Bryman, 2016, s. 468). For å formulere gode spørsmål foreslår Bryman (2016, s. 470) at forskeren stiller seg selv spørsmålet: «Hva trenger jeg å vite for å svare på forskningsspørsmålet mitt?» (egen oversettelse). Videre fulgte jeg hans stegvise beskrivelse av utformingen intervjuguiden (se figur 3.1).



Figur 3.1: Utforming av intervjuguide (egne oversettelser) (Bryman, 2016, s. 470).

Jeg startet med å velge ut intervjutemaer ut ifra faktorene i PTK-rammeverket. Videre utformet jeg et førsteutkast som ble brukt i et pilotintervju. Det var utfordrende å skaffe en lærer som oppfylte utvalgskravene på kort tid, så jeg gjennomførte piloten med en medstudent som hadde kompetanse i matematikk og programmering. I etterkant av piloten skrev jeg om flere av spørsmålene, før jeg ferdigstilte intervjuguiden.

Intervjuguiden (se vedlegg 1) baserte seg på spørsmål jeg kategoriserte i temaene «generell info om læreren», «lærerens holdninger til programmering i matematikkundervisning», «profesjonell instrumentell genesis», «elevers instrumentelle genesis» og «programmering i matematikkundervisning». Jeg ønsket ikke bare å få lærere til å beskrive sin egen og elevers kunnskaps- og kompetansebygging i programmering, men også få et bilde av deres holdninger rundt programmerings inntog i matematikkfaget. Hensikten å skape et så komplett bilde av lærernes pedagogiske teknologikunnskap som mulig.

3.4 Dataanalyse

Jeg baserte meg på en deduktiv tematisk analyse, som vil si at det teoretiske rammeverket som ble presentert i forrige kapittel ble brukt gjennom hele analyseprosessen, fra utforming av de første kodene til å beskrive resultatene. Gjennom delkapittelet gjør jeg rede for valgene som ble tatt og presenterer alle fasene i analyseprosessen.

3.4.1 Tematisk analyse

Tematisk analyse er en fleksibel og systematisk fremgangsmåte for å genere koder og tema ut ifra et kvalitativt datamateriale (Clarke & Braun, 2017). Analysen er fleksibel når det kommer til utvalgsstørrelse, forskningsspørsmål, datainnsamlingsmetode og teoretisk rammeverk (Clarke & Braun, 2017, s. 297). Tilnærmingen har som formål å identifisere, analysere og tolke mønster eller temaer i datasettet, og passer godt for forskere med liten erfaring innen kvalitativ forskning. Temaene som generes i analysen deles ofte i to hovedkategorier, *semantiske* og *latente* temaer, og tematiske analyser fokuserer gjerne på å finne temaer som passer inn i én av kategoriene (Braun & Clarke, 2006, s. 84). Hvis man inntar en semantisk tilnærming til analysen søker man etter temaer som stammer fra den eksplisitte informasjonen man får fra datamaterialet, og man ønsker ikke å «gå forbi». Det vil altså si at datamaterialet forstås ut ifra hva lærerne har sagt, og en det er overflatebetydningen som ligger til grunn for de ulike tema som finnes. Derimot hvis forskeren ønsker å finne latente temaer i datamaterialet, tolker de datamaterialet forbi akkurat det som står, og forsøker få tak en mening bak formuleringene. Da er hensikten å utforske de underliggende meningene. Tematisk analyse kan beskrives gjennom seks faser: 1) Bli kjent med datamaterialet 2) generere koder 3) søke etter temaer 4) revidere potensielle temaer 5) definere og navngi temaene 6) presentere analysen (Braun & Clarke, 2006, s. 87).

Fase 1: Bli kjent med datamaterialet

I den første fasen forsøker man å skape noen tidlige analytiske tanker om datamaterialet, men det er viktig at forskeren ikke begynner å konstruere temaer for tidlig, ettersom dette kan føre til at man håndplukker data som passer til sine oppfatninger (Terry et al., 2017, s. 25). Prosessen med å bli kjent med datamaterialet består ofte av gjentatt lesing av transkripsjonene og notere idéer for koding ved intervju som datainnsamlingsmetode. Ettersom jeg brukte deduktiv tematisk analyse (der kodene ble hentet fra rammeverket), var det ikke behov for å notere ned ideer til koder, og derfor bestod denne delen av prosessen for meste av å lese transkripsjonene

Fase 2: Generere koder

I den andre fasen skal forskeren generere de første kodene fra datamaterialet, og kodene skal beskrives en semantisk eller latent mening i datamaterialet som forskeren anser som interessant for å besvare forskningsspørsmålet (Braun & Clarke, 2006, s. 88). De første kodene baserte seg på PTK-rammeverket, og jeg forsøkte å lage et sett med koder som beskrev alle faktorene i rammeverket tilstrekkelig. Jeg var også åpen for å opprette koder etter behov underveis i kodingen.

Fase 3: Søke etter temaer

I den tredje fasen starter arbeidet med å bygge temaer fra datamaterialet ved å sortere kodene, og kodene må sorteres i hovedtemaer og undertemaer (Braun & Clarke, 2006, s. 89-90). Forskningsspørsmålet fungerer som en rettesnor i denne fasen, slik at forskeren

sikrer seg at søket vil gi relevante temaer. Når denne fasen er ferdig skal man sitte igjen med såkalte kandidat-temaer, som forskeren senere må revidere for å avgjøre om temaet skal bestå i sluttproduktet. I denne fasen sorterte jeg kodene ut ifra forskningsspørsmålene, og undersøkte om noen koder kunne kombineres eller fjernes. Flere av datasegmentene viste både lærernes PTK og kjennetegn på undervisningstilnærming, og ble derfor brukt i begge delene.

Fase 4: Revidere potensielle temaer

Etter den tredje fasen har forskeren en rekke kandidat-temaer, som i denne fasen skal revideres. Braun & Clarke (2006, s. 91) beskriver revideringsprosessen gjennom to nivåer av revidering. På det første nivået skal forskeren se på kodesegmentene i hvert kandidat-tema, og vurdere hvorvidt de henger sammen og danner et mønster. Om kodene danner et mønster, kan forskeren starte på nivå 2 der forskeren skal vurdere hvorvidt temaene er gyldige opp mot hele datasettet. Hvis temaene skal være gyldige, må de «presist reflektere meningene i datasettet som helhet» (egen oversettelse), og det teoretiske rammeverket i deduktive tematiske analyser er med på å avgjøre hvilke temaer som reflekterer presist (Braun & Clarke, 2006, s. 91). Etter fase tre satt jeg igjen med flere potensielle temaer, men jeg vurderte det til at flere av temaene ikke var interessante for denne studien på bakgrunn av problemstillingen. Jeg kommer tilbake til dette i neste delkapittel.

Fase 5: Definere og navngi temaene

I den femte fasen skal forskeren definere og navngi temaene. Definerings av et tema omhandler å beskrive essensen av innholdet i hvert tema, og navngivingen handler om å gå fra midlertidige arbeidsnavn til de offisielle navnene som skal brukes i sluttproduktet, og ved endt analyse bør man kunne beskrive hva temaene er og ikke er med få setninger (Braun & Clarke, 2006, s. 92). I denne fasen navnga jeg temaene med den hensikt å få frem essensen av hvorfor de er interessante i en matematikdidaktisk kontekst

Fase 6: Presentere analysen

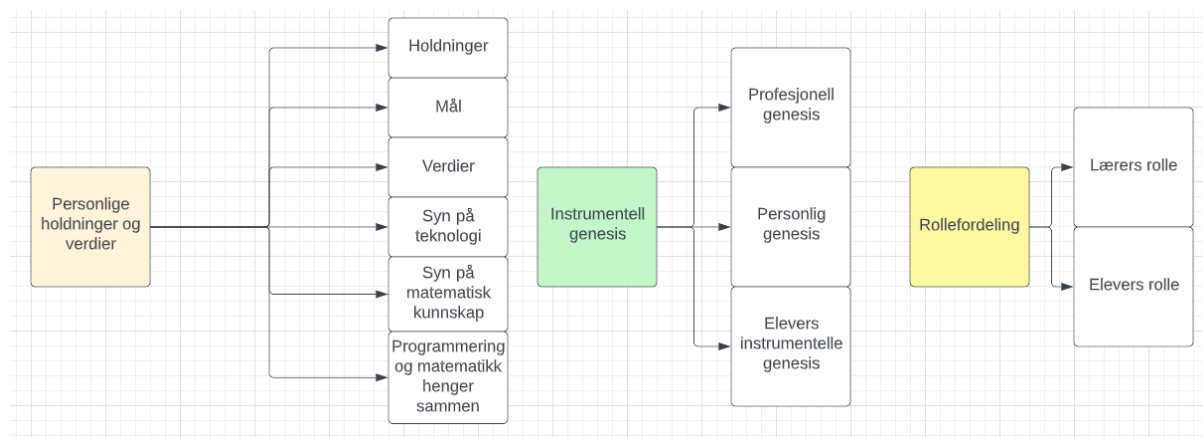
For å starte på den siste fasen krever det at forskeren har gjort analysen grundig og helt ferdig, og sitter med temaene som skal presenteres. Presentasjonen skal vært presis, og inneholde datasegmenter som eksemplifiserer et tema godt (Braun & Clarke, 2006, s. 93).

3.4.2 Analyseprosessen

Jeg brukte en deduktiv tematisk analyse som baserte seg på PTK-rammeverket. En deduktiv tematisk analyse bruker et rammeverk som en rød tråd gjennom analyseprosessen, fra generering av koder til tolkning og utforming av tema. Før kodingen startet lagde jeg kodene *personlige holdninger og verdier* (med underkoder mål, holdninger, verdier, syn på teknologi, syn på matematisk kunnskap og programmering og matematikk henger sammen), *instrumentell genesis* (med underkoder profesjonell genesis, personlig genesis og elevers instrumentelle genesis), *rollefordeling* (med underkoder lærers rolle og elevers rolle) (se figur 3.2). Koden som omhandlet rollefordelingen til elever og lærer i programmeringsundervisning stammet ikke fra rammeverket, men jeg så det som et viktig moment å fange opp for å forklare lærernes undervisningstilnærming.

Matematisk kunnskap for undervisning var ikke en del av kodene jeg startet med, ettersom jeg så det som utfordrende å få lærerne til å sette ord på sin matematiske- og

pedagogiske kunnskap. Ettersom MKT omhandler matematikk på generell basis, og lærerens pedagogiske tilnærminger, så er faktoren ikke rettet spesielt mot programmering og teknologibruk. Derfor er faktoren ikke relevant på samme måte som lærernes holdninger, tanker og erfaringer og grad av genesis. Jeg arbeidet ut ifra en tanke om at lærerne hadde god matematisk kunnskap for undervisning ettersom alle hadde over 10 års erfaring som matematikklærere. Bakgrunnen for denne vurderingen springer ut fra en antakelse om at erfaring som matematikklærer kan bygge selvtillit, som gjør at læreren blir trygg i en klasseromsituasjon. Tryggheten mener jeg kan føre til at lærere får integrert ukjent teknologi enklere, fordi de har tro på at deres tilnærming vil fungere. Derfor vurderte jeg det til at rammeverket fortsatt kunne brukes til tross for at intervjuet ikke gikk i dybden i lærerens matematiske kunnskap for undervisning.



Figur 3.2: Kodene analysen tok utgangspunkt i.

Jeg startet kodingsprosessen med disse kodene, men jeg opprettet i tillegg en *diverse* kode som skulle fange opp eventuelle viktige funn som falt utenfor kodene fra PTK-rammeverket. Etter kodingen var det totalt 12 datasegmenter uten særlig sammenheng i diverse-koden, og jeg har derfor valgt å ikke ta den med i tabellen under. Koden *instrumentell genesis* ble utformet for å fange opp lærerens tilegnelse av det didaktiske og personlige verktøyet, men også eventuelle betraktninger som kunne tolkes som å omhandle elevenes instrumentelle genesis. Etter den første kodingen så fordelingen av datasegmenter i de ulike kodene slik ut:

Personlige holdninger og verdier	
Kode	Antall
Holdninger	18
Mål	15
Verdier	2
Syn på teknologi	12
Syn på matematisk kunnskap	3
Programmering og matematikk henger sammen	15
Sum	65
Instrumentell genesis	
Kode	Antall
Profesjonell genesis	16
Personlig genesis	2
Elevens instrumentelle genesis	11

Sum	36
Rollefordeling	
Kode	Antall
Lærers rolle	17
Elevers rolle	7
Sum	24

Tabell 3.1: Oversikt over antall datasegmenter i hver av kodene.

Etter den første kodingsprosessen var ferdig startet jeg å søke etter temaer i datamaterialet. Søkingen medførte at jeg lagde koden *didaktisk potensial* (med underkoder positiv påvirkning, negativ påvirkning, epistemologisk verdi og pragmatisk verdi). Bakgrunnen for dette var et jeg så det som viktig å ta stilling til hvordan lærerne beskrev effekten av å ha en ny arbeidsmåte i matematikkfaget, og hvordan de beskrev potensialet de så som følge av det. Etter andre gjennomgang var det totalt 25 datasegmenter i koden *didaktisk potensial*.

Deretter innledet jeg fase 4, der jeg reviderte potensielle temaer. Denne fasen resulterte i 7 mulige tema:

1. Utforskende undervisning
2. Syn på lærers programmeringskompetanse
3. Epistemologisk og pragmatisk verdi
4. Kontinuitet i programmeringsarbeid
5. Programmering som en fremtidsrettet kompetanse
6. Programmering som et verktøy for elevers algoritmiske tenkning
7. Programmering har en klar sammenheng med matematikk

Etter jeg hadde en oversikt over de mulige temaene, gikk jeg igjennom datamaterialet på nytt for å undersøke hvorvidt temaene kunne beskrives ut ifra rammeverket, og i hvilken grad de svarte på problemstillingen. Gjennomgangen medført at enkelte temaer falt bort, da jeg vurderte de til å ikke gi et tydelig svar på problemstillingen. Etter revideringsprosessen satt jeg igjen med temaene «Epistemologisk verdi», «utforskende programmeringsarbeid og metodefrihet» og «liberalt syn på lærerrollen», og de blir presentert i analysekapittelet.

3.5 Studiens Kvalitet

Studios kvalitet vil belyses gjennom validitet- og reliabilitetsbegrepene (tilpasset en kvalitativ kontekst). Valgene knyttet til disse to tar i stor grad utgangspunkt i Bryman (2016). Deretter presenterer jeg de etiske betraktninger som er lagt til grunn for denne studien. Til slutt vil jeg presentere refleksjoner knyttet til min rolle som forsker, og påvirkningen det kan ha på studiens utfall.

3.5.1 Validitet og reliabilitet

Kvaliteten til forskning beskrives ofte ut ifra *validitet* og *reliabilitet*. Validitet omhandler hvorvidt datamaterialet sammenfaller med resultatene forskeren presenterer, og i hvilken grad resultatene er generaliserbare, og reliabilitet sier noe om i hvilken grad studien er mulig å replisere med samme resultat, og hvorvidt det er flere forskere involvert som er enig om resultatene (Bryman, 2016). Det problematiseres hvorvidt disse kriteriene er passende for å vurdere kvaliteten til kvalitativ forskning, og som et alternativ presenterer Bryman (2016) begrepet *troverdighet* (eng: trustworthiness) fra Lincoln og Guba (1985, 1994). Troverdighet består av begrepene kredibilitet (eng:

credibility), overførbarhet (eng: transferability), pålitelighet (eng: dependability) og bekreftbarhet (eng: confirmability) (egne oversettelser).

Studiens kredibilitet sikres ved at forskeren har fulgt prinsippene for god forskningspraksis, og at deltakerne i studien får innsyn i resultatene for å kunne bekrefte at forskeren har forstått dere perspektiv på rett vis (Bryman, 2016). Etter endt analyse fikk deltakerne innsyn i resultatene, og hadde muligheten til å komme med innspill dersom de opplevde at beskrivelsene ikke gav et rett bilde av virkeligheten.

Ettersom kvalitative studier ofte baserer seg på små utvalg innen en gitt kontekst, kan det være utfordrende å generalisere eller overføre resultatene til en annen kontekst. For å sikre studiens overførbarhet i størst mulig grad, bør forskeren beskrive konteksten forskningen er gjort i så godt som mulig. I denne studien har jeg gjort rede for kriteriene for deltakelse, og forsøkt å beskrive deltakere og konteksten intervjuene ble gjennomført i så godt som mulig (samtidig som anonymiteten ivaretas).

Pålitelighet kan sikres ved å ha et *audit trail* (Lincoln & Guba, 1985, 1994 i Bryman, 2016). Et audit trail betyr at forskeren tar vare på dokumentasjon fra hele forskningsprosessen, slik at studien kan ettergås i den grad det er mulig ved kvalitative studier. Ved å ha dokumentasjon for hele prosessen, vil det i større grad være mulig å vurdere hvorvidt studiens resultater er «korrekte». For å øke studiens pålitelighet vil jeg ta vare på dokumentasjon i den grad jeg kan. Det vil si at transkripsjon, koding og analyse vil bli tatt vare på, og datamaterialet tas vare på så lenge prosjektet aktivt (lydfilene slettes innen prosjektslutt).

Bekreftbarhet har tilknytning til forskerens objektivitet. Det er ikke mulig for en forsker å oppnå fullstendig objektivitet ovenfor datamaterialet og den påfølgende analysen (Bryman, 2016), men forskeren kan øke bekreftbarheten ved å ikke la personlige overbevisninger påvirke studien i stor grad (jeg har gjort rede for min påvirkning på studien i 3.5.3 Forkerrollen), slik at forskningsprosessen er transparent, og resultatene sees i lys av forskerens utgangspunkt.

3.5.2 Etikk

Jeg har basert meg på Diener og Crandalls (1978; sitert i Bryman, 2016, s. 125-134) fire etiske prinsipper for å ivareta forskningens etiske grunnlag. Prinsippene omhandler 1) hvorvidt forskningen er til skade for deltakerne 2) hvorvidt det har forekommet et informert samtykke fra lærerne 3) hvorvidt forskeren har respektert privatlivets fred og 4) hvorvidt forskningen har vært redelig med sitt formål ovenfor deltakerne.

Studiens etiske grunnlag ble sikret gjennom å følge NSDs (Norsk Senter for Forskningsdata) retningslinjer. Prosjektet ble godkjent av NSD før oppstart, ettersom datamaterialet inneholdt sensitive personopplysninger om deltakerne. Før intervjuene ble gjennomført ble deltakerne presentert med et samtykkeskjema vedlegg der de ble informert om studiens overordnede formål og deres rettigheter som deltaker i studien. Deltakerne ble også informert om lagring og behandling av data underveis i prosjektet og etter prosjektets slutt. Samtykkeskjema sikret at deltakerne gav et informert samtykke. Når intervjuene ble gjennomført var problemstillingen formulert annerledes, og lærerne ble informert om at prosjektet ville ta for seg programmeringens rolle i deres undervisning, og derfor er samtykkeskjemaet sentret rundt den formuleringen av problemstillingen. I etterkant ble lærerne informert om endringen av problemstillingen når de fikk innsyn i resultatene. For å sikre at datamaterialet ble samlet inn og lagret i henhold til NSDs retningslinjer ble det brukt diktafon fra NTNU ved fysiske intervjuer.

Lydopptaket ble kryptert før lagring. Ved digitale intervjuer ble opptaksfunksjonen på Zoom brukt, og lydfilen ble kryptert før lagring.

Studien sikret derfor deltakernes anonymitet og deltakelsen skjedde på bakgrunn av et informert samtykke. Intervjuene ble planlagt og gjennomført i arbeidstid via e-post, og dermed har jeg respektert privatlivets fred. Deltakerne var kjent med studiens overordnede formål i forkant av deltakelse, og det har ikke vært noe form for bedrag eller skade for deltakerne.

3.5.3 Forskerrollen

I en kvalitativ studie er det viktig at forskeren er bevisst på sin egen rolle, og hvordan hans tanker, meninger og holdninger kan påvirke datainnsamling, dataanalyse og presentasjon av resultat. Bryman (2016) trekker frem dette som et moment som kritiseres hyppig i kvalitativ forskning. Han hevder kvalitativ forskning påvirkes i stor grad av «usystematiske vurderinger av hva som anses som viktig» (s. 398) (egen oversettelse). Derfor mener jeg det er nødvendig å gjøre rede for eventuelle hypoteser jeg selv hadde om programmering i forkant av arbeidet, slik at forskningsprosessen sees i kontekst av disse meningene. Før arbeidet startet hadde jeg et bilde om at en betydelig del av lærerstanden ikke opplevde å ha tilstrekkelig kompetanse i programmering, og at dette medførte at det ble noe lærerne forsøkte å unngå å benytte seg av det i undervisning. Videre hadde jeg en hypotese om at programmering ikke var blitt en del av matematikk på mange skoler i Norge, og at «implementeringsgraden» var lav. Jeg hadde også et bilde av at det hersket stor tvil rundt hvilken rolle programmering var tiltenkt i matematikkfaget, som også var grunnen til denne vinklingen på oppgaven. Disse holdningene er et resultat av inntrykk jeg har fått i praksisperioder, samtaler med faglærere, kronikker og artikler. Disse holdningene er spesielt viktig å gjøre rede for ettersom datamaterialet skal analyseres gjennom en tematisk analyse, der mye avhenger av hva forskeren selv anser som viktig å belyse.

I tillegg til å ta hensyn til mine innledende hypoteser er det viktig å reflektere rundt min dobbeltrolle som forsker i denne studien. Jeg har designet intervjuguide, gjennomført intervjuene og analysert transkripsjonene i etterkant, noe som vil kunne påvirke resultatet av studien. Det vil være en særlig stor påvirkning i gjennomføringen av intervjuene, ettersom de var semi-strukturerte og jeg avgjorde hvilke utsagns som skulle utdypes hos informantene. Det er utfordrende å peke ut akkurat hvordan mine hypoteser, gjennomføring og analyse påvirker resultatet, men studien gjort i det fortolkende paradigmet vil påvirkes av dette. Formålet med å presentere hypoteser og min egen påvirkning er at forskningsprosessen skal være så transparent som mulig.

4 Analyse

Intervjuene ble analysert gjennom en deduktiv tematisk analyse. Analysen resulterte i tre kjennetegn ved lærernes undervisningstilnærming i programmering og eksempler på hvordan lærerne gav uttrykk for sin pedagogiske teknologikunnskap. Temaene har som formål å besvare problemstillingen:

Hva kjennetegner undervisningstilnærmingen til et utvalg lærere med høy pedagogisk teknologikunnskap i programmering?

For å belyse denne problemstillingen vil jeg i dette kapittelet ta for meg mine to forskningsspørsmål.

- 1) *Hvordan gir et utvalg lærere uttrykk for sin egen pedagogiske teknologikunnskap?*
- 2) *Hva kjennetegner et utvalg læreres undervisningstilnærming i programmering?*

Kapittelet er todelt, der hver del tar for seg et forskningsspørsmål. Det første delkapittelet, «4.1 Lærernes pedagogiske teknologikunnskap» beskriver hvordan de tre lærernes PTK kommer frem i intervjuene. I tillegg vil jeg gjøre rede for hvorfor lærernes PTK er vurdert til å være høy i en programmeringskontekst. Videre i delkapitlene «4.2 Programmeringens epistemologiske verdi», «4.3 Utforskende programmeringsarbeid» og «4.4 Liberalt syn på lærerrollen» presenteres de tre kjennetegnene ved utvalgets undervisningstilnærming i programmering. Datamaterialet vil presenteres gjennom utdrag fra transkripsjonene. I transkripsjonene vil korte pauser symboliseres ved «...» og hopp i samtalen symboliseres ved «[...]».

4.1 Lærernes pedagogiske teknologikunnskap

Ettersom denne studien er basert på et kvalitativt datamateriale, er det vanskelig å gi en objektiv vurdering av hvilket nivå lærernes pedagogiske teknologikunnskap er på. Studien støtter seg på en tolkning fra meg som forsker, og derfor vil jeg først gjøre rede for hva jeg legger i «høy pedagogisk teknologikunnskap». Jeg har tolket det slik at alle tre lærerne har høy PTK når det gjelder programmering. Vurderingen tar utgangspunkt i Thomas og Palmers (2014, s. 85-86) generelle beskrivelse av en lærer med høy PTK:

[They] relate the technology to linking multiple representations of constructs, understanding of ideas, generalization and moving from step-by-step processes to an overview. They have advanced to the point where they are competent in instrumentation of the technology and are able to focus on other important aspects, such as the linking of graphical, tabular, algebraic, ordered pair and other representations. With high PTK they see digital technology as having a wider application than simply calculation. They feel free to loosen control and encourage students to engage with conceptual ideas of mathematics through individual and group exploration, investigation of mathematical ideas, and the use of methods, such as prediction and testing. For these teachers the mathematics rather than the technology has come to the foreground, and technology [...] They feel free to loosen control and encourage pupils to engage with conceptual ideas of mathematics through individual and group exploration, investigation of mathematical ideas, and the use of methods, such as prediction and testing. (Thomas & Palmer, 2014, s. 85-86)

I denne beskrivelsen kommer det frem en del kjennetegn knyttet til lærere med høy PTK sin praksis. Vurderingen av lærernes PTK vil ta utgangspunkt i en tolkning som baserer seg på disse kjennetegnene, men jeg vil også inkludere deres holdninger knyttet til programmering og erfaring. Så med høy PTK mener jeg at lærerne kan bruke programmering, ikke bare selv, men i klasserommet som en didaktisk ressurs. De har positive holdninger knyttet til programmering, og gir uttrykk for at de ser en verdi av programmering som en del av matematikken. De vil også beskrive å ha en trygghet i rollen som matematikklærer som gjør at de kan «miste kontrollen» slik Thomas og Palmer (2014) beskrev.

Videre i dette delkapittelet vil jeg presentere hver lærer, og gjøre rede for hvordan lærernes PTK kommer frem. En av faktorene i rammeverket er *matematisk kunnskap for undervisning*, som består av lærerens pedagogiske- og matematiske kunnskap. I intervjuet var ikke dette en av fokusområdene da fokuset lå på programmering og ikke matematikk generelt. I analysen tok jeg utgangspunkt i at lærernes lange erfaring som matematikklærere gav de en god pedagogisk- og matematisk kunnskap.

4.1.1 Janne

Janne gir uttrykk for å være motivert for å bruke programmering i matematikkundervisning. På spørsmål om hva Jannes motivasjon var for å bruke programmering svarte hun følgende:

Janne: Den er ganske høy (latter), ja. Jeg er ikke noe sånn ... når jeg skal holde på med programmering eller se hva som passer, så må jeg, jeg er ikke så flink i programmering at jeg «bare tar det» på strak hånd, jeg må ofte forberede meg, og skrive litt sånn steg for steg hva jeg må ... hvordan jeg skal introdusere det og hvor jeg skal ungene få lov til å forske selv, og hva er jeg nødt til å ... ja, jeg skriver et lite sånn «manus» for meg selv da.

Jeg tolker utdraget som at Janne ønsker en god faglig progresjon i programmering, og at hun derfor støtter seg på en form for planleggingsdokument som hjelper henne å legge opp undervisning best mulig. Det tyder derfor på at Janne gjør en del didaktiske refleksjoner i forkant av undervisningen, og at hun har et klart med hva elevene skal få ut av hver fase i undervisningen. Jannes didaktiske refleksjoner mener jeg kan knyttes til dobbel genesis gjennom instrumenteringsprosessen i profesjonell genesis. Ut ifra Jannes beskrivelser tolker jeg det slik at hun er kommet langt på vei i sin profesjonelle instrumentelle genesis i de programmeringsplattformene hun bruker i sin undervisning.

Senere i intervjuet sa hun at det var hun som hadde ansvaret for programmeringen på trinnet, og at hun hadde bakgrunn fra teknologi og entreprenørskap. På spørsmål om hvordan hun utviklet sin egen programmeringskunnskap fortalte hun at hun brukte ressurser på nett for å skaffe seg en oversikt, men at mye av utviklingen lå i å bruke verktøyet selv, og teste ulike løsninger. I det neste utdraget gir Janne uttrykk for at programmering hadde beriket matematikkundervisning, og på spørsmål om hun kunne gi eksempler på hvorfor hun tenkte dette svarte hun:

Janne: Nei, istedenfor å ... nå er ikke jeg sånn som bare sitter og regner i boka uansett (latter), men om en skal tenke seg det da, så er det jo, altså det skaper enda mer variasjon, at du får til å bruke programmering og PC og sånt. I tillegg så er det jo det her de kommer til å møte, mange av dem, i voksenlivet og kommer høyere opp. Det forbereder dem, de kommer til å sitte mer og mer på data. Så ... ja, jeg vil si at det beriker fordi at de ikke bare sitter med blyant og penn og løser oppgaver eventuelt på Chromebooken, de gjør noe med det. Det er litt sånn praktisk da egentlig.

Ut ifra utdraget tolker jeg det til at Janne har positive holdninger til programmering, og at hun ser en klar nytte av det, både i matematikkfaget, men også utenfor skolen. Det er på bakgrunn av Jannes beskrivelser av det didaktiske potensialet programmering har (hun nevner at det skaper variasjon). Beskrivelsen gir inntrykk av at Janne mener programmering gir et annet didaktisk potensial enn andre teknologier gjør. Det kan tyde på at Jannes holdninger til programmeringens muligheter og nytteverdi er positive, som passer med kjennetegnene for høy PTK. På spørsmål om hvorfor noen matematikklærere ikke brukte programmering hevdet Janne at dette kunne stamme fra at programmering ikke var en del av standardiserte prøver, og at lærerne ikke så nytten av det:

Janne: Jeg tror ikke de ser nytten av matematikken i det på den samme måten som jeg gjør. De ser på det som et eget tema, og noe som de ikke kan. Og da slipper de å ta det, fordi det blir heller ikke målt.

Janne sier selv at hun ser nytten av programmering, til tross for at det ikke er en del av «det som blir testet». Dette kan tyde på at Janne ser en verdi som går forbi det som måles i for eksempel nasjonale prøver, og at dette motiverer Janne til å bruke programmering til tross for det ikke testes. I PTK faktoren personlige holdninger og verdier kan dette knyttes til både synet på teknologi, men også Jannes mål. Hun ser en verdi av programmering, og selv om det ikke blir testet, bruker Janne programmering, til tross for at hun kunne unngått å bruke det til en viss grad. Derfor mener jeg dette kan tyde på at Janne har et mål om at elevene skal få jobbe med programmering i matematikk.

Videre gir hun også uttrykk for at programmering hører hjemme i matematikken:

Janne: Ja, altså ... verktøy i form av at ... hva skal jeg si? Det er jo en motiverende faktor, men at de er nødt til å bruke matematikken ... for å få til programmering, og motsatt. Ja, så at det på en måte, det er linket, veldig sånn flettet sammen.

Både Thomas & Palmers (2014) beskrivelse og Haspekians (2014) studie trekker fram læreres evne til å se en sammenheng mellom teknologien som brukes og matematikk, og ut ifra dette utdraget tolker jeg det slik at Janne har et klart bilde av hvordan de to kan henge sammen. Hun gav uttrykk for at programmering hørte hjemme i matematikken senere i intervjuet. Ut ifra dette tolker jeg det slik at Jannes syn på programmering og matematikkfaget påvirker hennes PTK, og at dette hjelper henne til å bruke programmering som en støtte i matematikk.

4.1.2 Lars

Han gir uttrykk for å mestre flere programmeringsspråk og «måten å tenke programmering på». Det kommer frem i dette utdraget:

Lars: Ja, først og fremst så, det er noe jeg fremmer til elevene og, det er jo sånn at når du har lært deg å tenke programmering, som det er veldig mye fokus på med algoritmisk metode, så er det jo først og fremst å finne ut hva du ønsker å få til, skrive pseudo -kode, og så er det jo ... jeg sammenligner det med språk da, det er en del sånne gloser. Jeg vet hvordan man itererer noe, men hvordan gjør de det i dette programmet og hvordan kan du vise frem noe.

Ut ifra utdraget tolker jeg det som at Lars har et klart bilde av hvordan man går frem i programmering, fordi han nevner viktigheten av planlegging og pseudo-kode (pseudo-kode er en form for uformell og forenklet kode, som beskriver hva brukeren ønsker å

gjøre med et program, og som kan «oversettes» til mange ulike programmeringsspråk). Det kan tyde på at Lars er en erfaren programmerer. Det mener jeg også kommer frem i form av Lars sin parallell mellom ulike handlinger i kode og gloser. Det tolker jeg som at flere ting innen programmering forstår Lars på et mer generelt nivå, og derfor er det ikke noe særlig problem å ta det med inn i et ukjent programmeringsspråk, fordi han forstår de generelle prinsippene. Det underbygges også av formuleringen hans om å «tenke» programmering, som jeg mener tyder på at programmering er tilegnet som mer en kode for Lars, det er en generell fremgangsmåte og en tilnærming til mange ulike problemer.

Senere i intervjuet forteller Lars om kompetanseutvikling gjennom kurs:

Lars: [...] Og når vi er på kurs, når vi hadde kurs sist, så gikk jo jeg rundt og hjalp de fleste. Alt de skulle lære i kurset kunne jeg, så da kunne jeg være en ressurs når de stod fast, da kunne jeg gi de tips.

Tidligere i intervjuet gav han uttrykk for at han har et ønske om å kunne holde programmeringskurs for sine kollegaer, som tyder på at Lars innehar en programmeringskompetanse som gir han overskudd og overblikk for å kunne støtte de rundt seg underveis i kurset. Fra utdraget tolker jeg det som at Lars er kommet mye lengre på vei i både instrumenteringen og instrumentaliseringen sin sammenlignet med sine kollegaer, og kan derfor være en støtte for dem. I tillegg til å bruke programmering i matematikkundervisning underviser Lars i valgfag programmering. Basert på dette tolker jeg det slik at Lars er kommet langt på vei i sin personlige- og profesjonelle genesis, og at dette kan tyde på at Lars har høy PTK.

På spørsmål om hva som kjennetegner en lærer som mestrer å bruke programmering i matematikk svarte Lars følgende:

Lars: Det er vel, først og fremst å se at det har plass i samfunnet, og at det er folk som kan programmere. Og at du selv ønsker å la eleven stifte bekjentskap med det nettopp fordi ... ikke sånn, «det er et kompetansemål, så da må du lære det», men tenke «Kult, her kan jeg jo få vist», og at du ser den muligheten.

Lars ser en verdi i programmering utover at det er et kompetansemål. Det tolker jeg som at Lars sin motivasjon er mer en ytre motivasjon som følge av at programmering er blitt en del av læreplanen, og kanskje kan det tyde på at Lars ville brukt programmering i sin undervisning uavhengig om programmering var en del av kompetansemålene i matematikk. Formuleringene hans tyder på at Lars ser en nytteverdi av programmering. Ut ifra dette tolker jeg det som at Lars har et positivt syn på programmering som en del av matematikkfaget, og dette sammenfaller med kjennetegnene på lærere som har høy PTK. Derfor mener jeg at dette kan tyde på at Lars har høy PTK.

Senere i intervjuet beskriver Lars programmering som «en forlenget arm» i mange matematiske tema. Jeg har tolket det som at Lars mente at programmering var et verktøy der man anvender matematikk til å løse problemer, og at programmering derfor kan vise hvordan matematikk anvendes. Dette tyder på et positivt syn på programmering og mulighetene det gir. Han knytter også inn problemløsning som en kobling mellom programmering og matematikk. Videre på spørsmål om hvordan han utvikler sin egen programmeringskompetanse beskriver Lars en sammenheng mellom programmering og matematikk:

Lars: Ja, det er jo å sitte og knote litt, se gjennom noen eksempler, tenk «hvordan kan jeg vise frem dette i matematikk?», hvis jeg skal vise frem statistikk, diagrammer osv. Kan jeg bruke programmering til å vise frem

det her? Og så «ja det kan jeg jo faktisk gjøre». Og så har jeg sagt til en del lærere at regneark ble veldig ofte brukt før, til å vise frem ting som programmering kan vise bedre, som terningkast og sjanse og sannsynlighet.

Ut ifra utdraget tolker jeg at Lars ser noe av den samme verdien i programmering i form av det didaktiske potensialet programmering har i matematikkfaget. På bakgrunn av Lars sine beskrivelser av programmering i matematikk tolker jeg det også som at han ser en klar sammenheng mellom programmering og matematikk, som i Thomas og Palmers (2014) beskrivelse kan tyde på at Lars har PTK. Ut ifra utdraget tolker jeg også at Lars sitt syn på hvilke didaktiske muligheter programmering tyder på at han har positive holdninger til programmering. Jeg tolker også Lars sine beskrivelser av egen kompetanseutvikling for å vise instrumentering (han reflekterer rundt hvordan han kan få programmering inn i sin undervisning) og instrumentering (oppdage nye didaktiske potensial som følge av refleksjonene) i hans profesjonelle genesis.

4.1.3 Simen

Simen underviser på mellomtrinnet, og benytter seg derfor av hovedsakelig BBP i sin undervisning. I tillegg har han erfaring med Python etter å ha undervist i fag på universitet. På spørsmål om hvordan han utvikler sin egne programmeringskompetanse gir Simen uttrykk for sin kompetanse med BBP:

Simen: Nei altså det har jo ... det er en liten utfordring fordi jeg må jo ... i hvert fall hvis du skal inn på noe Scratch og noe sånt der kan jeg det meste, men det er jo det der å ... Kompetansen er jo både fagkompetanse og den didaktiske kompetansen. Det er jo to forskjellige. På Scratch tror jeg jo du får til det meste, blokkprogrammering og sånt noe.

Simen gir uttrykk for å kunne få til det meste i en blokk-basert kontekst, og jeg tolker det som at Simen referer til sin personlige genesis, i form av at gir uttrykk for å kunne det meste. Han forteller videre om at kompetansen hans var lavere på tekstbasert programmering, men at dette ikke var et problem ettersom det ikke ble brukt på mellomtrinnet. Det kan tyde på at Simen har kommet langt på sin personlige- og profesjonelle genesis på den programmeringsplattformen han bruker i sin undervisning. Det sammenfaller med beskrivelsen av høy PTK. Simens grad av genesis kommer også frem når han trekker et skille mellom fagkompetanse og didaktisk kompetanse. Han beskriver utviklingen av den didaktiske kunnskapen slik:

Simen: [...] Mens det didaktiske er nå egentlig mer sånn daglige refleksjoner. Hva kan jeg bruke det til, hvordan skal jeg få det til å funke, hvordan skal du få med alle.

Simens beskrivelser av utviklingen av sin didaktiske kompetanse mener jeg kan knyttes opp mot profesjonell genesis. Han trekker frem spørsmål han reflekterer rundt for å utvikle sin pedagogiske bruk av programmering, som omhandler hva og hvordan man kan gjøre undervisningen mest didaktisk, som kan knyttes til instrumentalisering (oppdage hvordan programmering kan brukes) og instrumentering (endre sin praksis slik at programmering blir en hensiktsmessig del) i profesjonell genesis. Simens eksplisitte skille mellom didaktiske- og fagspesifikke ferdigheter mener jeg tyder på at han er godt i gang med sin profesjonelle genesis. Senere i intervjuet forteller Simen at han har tatt programmeringsøkter for sine kollegaer, fordi de opplevde å ikke ha tilstrekkelig kompetanse. Dette kan tyde på at Simen har kommet langt med sin profesjonelle genesis med BBP.

I det neste utdraget beskriver Simen motivasjonen sin for å bruke programmering:

Simen: Nei altså en ting er nå det at står i læreplanen da, det er jo en viktig motivasjon da. Også, men så syns jo jeg at det er litt artig selv da. Så sånt sett har jeg en indre motivasjon, ikke bare en ytre som blir pushet på meg [...] Jeg ser jo at det er mye ting man kan bruke det til i matten, og det er jo det som er motivasjonen min, at de skal kunne det, og så skal jeg forhåpentligvis ha denne gjengen neste år også, når de går i 7. trinn. Og da kanskje vi kan bruke det når det kommer til sannsynlighet og statistikk og den typen ting, og simulere. Vi har brukt det litt i geometri. Så det er litt sånn, det er flere ting jeg har som motivasjon for å bruke det.

Han attribuerer motivasjonen sin til læreplanen, men trekker også frem en indre motivasjon, som tyder på at Simen (i likhet med Lars) er motivert utover at det har blitt en del av læreplanen, som jeg mener tyder på at Simen ser en verdi av programmering i matematikk. Dette underbygges av at Simen sier at han ser mange bruksområder for programmering i matematikk. Formuleringen mener jeg også tyder på at Simen har instrumentert programmering, og oppdaget mange didaktiske muligheter. Senere på spørsmål om hvorvidt programmering har beriket undervisningen i matematikk trekker Simens frem programmeringens muligheter for variasjon som en berikelse:

Simen: Ja det vil jeg jo si at det har. Eh ... det er jo som jeg sa, det er jo enda en måte å vise frem noe på. Du ønsker jo å ha variasjon og du ønsker å ha flere måter å vise frem ting på. Og så tenker jeg at det er en berikelse når du får til det.

I utdraget forteller Simen at programmering åpner opp for variasjon og flere representasjoner, og at dette er en berikelse for faget. Jeg har tolket utdraget til at Simen ser innføringen av programmering som noe positivt for matematikkfaget. Det kan tyde på at Simen har positive tanker om programmering i matematikk, som igjen kan tyde på høy PTK.

4.1.4 Lærernes pedagogiske teknologikunnskap

I utdragene har jeg belyst eksempler der lærernes PTK kommer til syne gjennom deres formuleringer. Som jeg nevnte innledningsvis hadde intervjuguiden som formål å undersøke PO og instrumentell genesis i rammeverket PTK. Jeg har tolket lærernes formuleringer til å tyde på at samtlige ser en nytteverdi i programmering, og at det er viktig å arbeide med programmering i skolen. Det mener jeg viser at alle lærerne har positive holdninger når det gjelder programmering i matematikkfaget. Det underbygges av at lærerne attribuerer motivasjonen deres for å bruke programmering til mer enn at det er blitt en del av læreplanen. Videre har belyst utdrag der jeg har vurdert lærernes grad av profesjonell genesis når det kommer til programmering til å ha kommet lenger enn den gjennomsnittlige matematikklærer. Jeg mener utdragene viser at alle lærerne kan vurderes til å ha høy PTK når det kommer til programmering.

4.2 Programmeringens epistemologiske verdi

En av kjennetegnene ved lærernes undervisningstilnærming var at de så og utnyttet en epistemologisk verdi i programmering i matematikk. Som nevnt tidligere kan teknologi ha både epistemologis- og pragmatisk verdi (Artigue, 2002). Teknologi som har epistemologisk verdi kan støtte elevers læringsprosesser i matematikk, og utvikle deres forståelse av matematiske objekter (Thomas & Hong, 2013, s. 70). Teknologi som har en

pragmatisk verdi, vil kunne støtte læring av matematikk ved stor regnekapasitet eller effektive utregninger. I dette delkapittelet presenteres utdrag der lærerne gir uttrykk for at programmering ikke bare kan brukes for å være et konkret verktøy i matematisk aktivitet, men at programmering kan være en støtte for elevenes konseptuelle matematiske forståelse. Lærerne gir uttrykk for at programmering kan støtte elevers læringsprosesser i arbeid med variabler, prosent, problemløsning og matematisk resonnering og argumentasjon.

4.2.1 Janne

I utdraget fortalte Janne at hun så en epistemologisk verdi i programmering som følge av de mulighetene det gir i arbeid med variabler. På spørsmål om hun så et potensial for å lære noe nytt svarte Janne følgende:

Janne: De har jo ikke vært borti det jeg holder på med (*variabler) nå før da. Men altså, det er nå helt nytt for dem. Så ... jeg tror ikke de hadde skjønnet hva en variabel er for noe uten det programmet. Det tror jeg ikke. Fordi at nå ser de at variabelen endrer på seg, de ser at det er en variabel. Hadde de ikke sett det, at det hadde kommet opp som en sånn «boks» i hjørnet, så tror jeg at de hadde sett på variabelen som et svar som de får ut, tror jeg.

Aktiviteten hun sikter til gikk ut på at klassen hennes programmerte sine egne kalkulatorer. Kalkulatoren hadde som formål å regne mellom norske kroner og den fiktive valutaen som finnes i Harry Potter universet. Ut ifra utdraget tolker jeg at Janne ser en epistemologisk verdi i programmering som åpner for at elevene kan få en bedre eller ny forståelse som elevene ikke ville fått ved andre arbeidsmåter. Det kan tyde på at Janne ser et epistemologisk potensial i form av at programmering åpner for at elevene kan «samhandle» med matematiske konsepter slik Thomas og Palmer (2014, s. 85-86) beskrev det. Jeg mener derfor at Jannes formulering tyder på at hun mener programmering har en epistemologisk verdi.

I det neste utdraget gir Janne uttrykk for hvorfor hun beskrev aktiviteten der elevene skulle programmere sin egen kalkulator som god:

Janne: For det første så lager de seg egentlig ... også jobbe med, litt sånn, generelle regler da, altså regneregler da. Bevis og ... den biten der også. Så de er nødt til å finne noe generelt. Altså egentlig jobber de med algebra, altså de jobber med masse aspekter her. De jobber med algebra, og må argumentere med hvorfor det her stemmer, og de får jobbet masse med den biten der på forhånd da. [...] Altså jeg vil jo at de skal teste en og en variabel først, og så sjekke, stemmer det her? Og de får jobbet ... når de går fra kroner til de her tre pengemyntene i Harry Potter, så får de jo repetert det uten at det blir kjedsommelig repetisjon, fordi de jobber med algebraen eller det generelle på ulike måter. De er nødt til å finne ut av det alle sammen.

I utdraget trekker Janne frem flere ulike ferdigheter i matematikk som hun mener utvikles gjennom oppgaven. Jeg mener flere av ferdighetene hun trekker frem (argumentasjon, bevis, generalisering) både krever og utvikler elevers forståelse, og derfor tyder det på at Janne anser programmeringsaktiviteten å fremme elevenes forståelse innen de nevnte aspektene, og at hun dermed ser en epistemologisk verdi i programmering gjennom arbeidet. Jeg tolker Jannes utsagn som at programmering åpner i større grad for at elevene kan samhandle direkte med matematikken, og at programmeringstilnærmingen og konteksten gjør at matematikken i større grad blir noe elevene anvender, og at Janne anser dette som positivt. Det kan tyde på at

programmering åpner for at elevene bygger en forståelse som gjør det mulig å anvende matematiske konsepter for å løse utfordringene de møter i konteksten.

4.2.2 Lars

I liket med Janne gir Lars uttrykk for at programmering kan ha en epistemologisk verdi inn mot variabler i matematikk. I det første utdraget blir Lars bedt om å ta stilling til påstanden «elever får en bedre forståelse av matematikk gjennom å bruke programmering»:

Lars: Ja, det kan nok gi det. Når du ser at, som for eksempel i algebra i programmering, at du begynner å bruke det. Så går du kanskje fra et snevert begrep til at A-en er en verdi, til at «å, det er variabel», vi kan endre på den. Og så plutselig «oi, nå begynner jeg å skjønne formler, det er jo derfor formler». Formler er jo på en måte bare den oppskriften, og så må du sette inn forskjellige tall.

Jeg tolker det som at Lars mener at programmering gjør det mer naturlig for elevene å anvende variabler for å løse et problem eller designe en funksjon, og at det kan konkretisere konseptet sånn at elevene i større grad forstår. Tanken om at programmering kan brukes til å bygge en forståelse av matematikk tyder på at Lars ser en epistemologisk verdi hos programmering. Det kan også vise til at Lars sin PTK gjør at han ser en klar sammenheng mellom algebra og programmering, og at han derfor gir uttrykk for å se en epistemologisk verdi inn mot det.

Senere i intervjuet ble Lars spurt om han så et potensial for å lære noe nytt i matematikk. Da knyttet Lars læreres evne til å realisere programmerings epistemologiske potensial til lærerens tankesett:

Lars: Det blir jo litt som ved de tidligere spørsmålene, det går på tankesettet ditt. Hvis du ser en mulighet til å utvikle, gjennom å lære noe nytt, ja så selvsagt så vil du få nye muligheter, du vil kanskje utvikle nye opplegg, du vil kanskje endre praksisen din til det bedre.

Utdraget tolker jeg som at Lars mener at det er muligheter for å lære seg nye fagstoff i programmering, men at læreren må ha et tankesett som lar dem se den muligheten for at den skal utnyttes. Det vil i så fall kunne knyttes til lærerens PTK. Ut ifra en PTK vinkling kan en lærers tankesett knyttes opp faktoren PO, der delfaktorer som f.eks lærerens syn på teknologi eller mål for undervisningen påvirker hvilke muligheter læreren ser. Utviklingen av lærerens tankesett kan også ha rot i profesjonell genesis, gjennom instrumentalisering, der læreren stadig oppdager et didaktisk potensial hos programmering, eller instrumentering, der læreren endrer undervisningskjemaene sine.

4.2.3 Simen

Simen gir i likhet med de to andre lærerne uttrykk for at han ser en epistemologisk verdi i programmering. Det vil jeg belyse i dette første utdraget, der Simen blir spurt om et eksempel på hvorfor han mener programmering passer inn i matematikken:

Simen: For eksempel kan du, som vi sa i stad da, med sannsynlighet for eksempel, å kunne simulere ting ... eh ... lage et program som kaster en terning. Det er jo forholdsvis enkelt ...eh ... og som gir elevene en annen forståelse da. Det er jo det vi i matten ... så er vi jo ute etter at elevene skal forstå. Og da er det ikke sånn at det bare er en måte å få de til å forstå, så du kan jo godt kaste terning i tillegg, men så er det jo fint å ... Når du må programmere noe, så må du forstå det på en eller annen måte, og den erfaringen jeg satt igjen med i fjor, fordi da programmerte jeg, lagde jeg

en god del sånne simuleringer på ... da underviste jeg i [matematikkfag] på [universitet]. Og da lagde vi en masse sånn dere, og jeg forsto det bedre selv også, selv om jeg egentlig føler at jeg kunne det fra før.

Jeg har tolket det som at Simen omtaler et epistemologisk potensial i programmering, ettersom han snakker om elevenes forståelse. Utdraget mener jeg tyder på at Simen har opplevd programmeringens epistemologiske verdi selv, gjennom arbeidet hans på universitetet der programmering var en del. I utdraget mener Simen at elevene kan få en annen forståelse gjennom om å programmere en terning (fremfor å kaste en fysisk terning), og jeg tolker det som at han mener at elevene får en større forståelse fordi de må «lage» verktøyene de skal bruke. Det betyr at de må forstå hva matematikken som ligger til grunn for en terning, fremfor å bare akseptere at en terning fungerer slik de gjør. Det kan også tyde på at Simen opplever at programmering tilbyr et didaktisk potensial som er ulikt andre arbeidsformer, og som utnytter en epistemologisk verdi i programmering som bygger elevenes matematiske forståelse.

I intervjuet beskrev Simen en aktivitet der elevene hadde programmert ved hjelp av BitBot. Bitbot er tilleggsutstyr til Micro:Bit-robotene, som lar brukeren koble roboten sammen med et sett med hjul, slik at man har en slags radiostyrt bil som man programmerer. Oppgaven gikk i grove trekk ut på at elevene skulle lage et program som instruerte den radiostyrte bilen til å kjøre og svinge etter ønske. I programmet kunne de ikke basere svingkommandoene på rotasjon i grader, men måtte bruke prosent og tid. I neste utdrag beskriver Simen hvordan denne typen arbeid kan brukes for å støtte elevers forståelse av matematikk fra tidlig alder:

Simen: Men der er det masse greier med å få den til å snu riktig, få den til å kjøre riktig tilbake igjen, ha riktig prosent på hvor mye kraft du skal bruke og sånt noe. [...]Og det kan jo for så vidt være et eksempel på det du sa i stad med å lære seg noe nytt. For det her tipper jeg du kunne gjort til en viss grad på andre, tredje og fjerdeklasse, og fått den til å snu tilbake igjen, og så prate om det her «hva er det den prosenten betyr for noe?», «Hvorfor kan man ikke kjøre i 120% prosent fremover? Hvorfor går ikke det an?». Det er det en god del ting man kan prate om, ut ifra en programmeringsøkt uten at det nødvendigvis er noe de kan fra før.

Jeg tolker Simens utsagn som at han mener programmering har et epistemologisk potensial som muliggjør at elever kan samhandle med matematiske ideer tidligere. Det kan tyde på at Simen mener programmering er ulikt annen teknologi, fordi programmering har epistemologisk verdi som følge av at elevene får samhandle med en matematisk kontekst gjennom et problem, og at dette åpner for matematisk samtale. Jeg tolker det også som at Simen mener at det oppstår kognitive konflikter som følge av at elevene må utforme programmene selv, og blir derfor konfrontert gjennom hele prosessen (for eksempel hvorfor kan ikke en BitBot kjøre på 120% kapasitet), og at dette er et godt utgangspunkt for å utvikle elevenes matematiske forståelse. Derfor kan utdraget sees som et eksempel på at Simen ser en epistemologisk verdi.

I det neste utdraget svarer Simen på hvorfor han mente at aktiviteten med BitBotene var en god økt:

Simen: Nei det er jo når du går rundt og hører på de, fordi det er her er en sånn aktivitet du bare kan sette i gang i klasse og bare høre på, høre på de prate, og du hører de bruker ... det er ikke nødvendigvis at de bruker matematiske begreper, men du hører at de de prater om egentlig er matte. De bruker ord, og de forklarer ting, og de argumenterer for hvorfor de må ha mindre og større tall her. Og der er det kanskje mange som

tenker at det her ikke er en mattetime, men at dette bare er programmering og lek, men jeg tenker at det ligger masse matte inni her.

I utdraget gir Simen uttrykk for at programmeringsarbeidet gav et utgangspunkt for faglig samtale for elevene. Jeg mener den faglige samtalen Simen beskriver kan knyttes opp mot kjerneelementet «resonnering og argumentasjon» fra læreplanen (Kunnskapsdepartementet, 2019), og kan dermed regnes som utvikling av sentrale matematiske ferdigheter for elevene. Derfor tolker jeg at Simen ser en epistemologisk verdi i programmering, ettersom aktiviteten oppfordrer til og muliggjør en faglig diskusjon der elevenes forståelse av matematiske konsepter kan utvikles. Derfor mener jeg utdraget viser at Simen ser en epistemologisk verdi i programmering i form av å utvikle sentrale ferdigheter i matematikk knyttet til resonnering og argumentasjon.

4.3 Utforskende programmeringsarbeid og metodefrihet

Et annet kjennetegn hos lærernes undervisningstilnærming var at de gav uttrykk for at de ønsket at elevene skulle arbeide med programmering på en måte som jeg har valgt å beskrive som *utforskende*. Det innebærer at elevene eksplisitt blir bedt om å utforske, for eksempel ved å arbeide med programmering uten tydelige rammer, eller finne løsninger helt selv. Lærerne tilstreber også en arbeidsmåte der elevene opplever metodefrihet, og elevenes autonomi utvikles individuelt eller i gruppe. Lærerne anerkjente behovet for at elevene behøvde et ferdighetsgrunnlag tidlig i programmeringen, noe som ofte krevde en mer strukturert og stegvis fremgangsmåte, men de understreket viktigheten av å at elevene fikk utforsket selv i det innledende arbeidet.

4.3.1 Janne

Janne så et behov for at elevene på mellomtrinn måtte tilegne seg grunnleggende programmeringsferdigheter for å kunne bruke det, men hun gav uttrykk for at hun ville at elevene skal utforske og prøve seg frem:

Janne: Eh, jeg har med sånn steg for steg for meg selv, hva skal jeg si, og hvilke spørsmål er det lurt ... altså hvor skal elevene få lov til å gruble og utforske litt, og når er det jeg skal ta dem inn igjen. Og hva er neste steg og neste steg.

Jeg tolker Jannes utsagn som at det utforskende aspektet ved programmeringsarbeidet er viktig for henne, og progresjonen hun ønsker for elevene, og at hun derfor inkluderer det i planleggingsdokumentet sitt. Dette kan tyde på at Janne har et mål om å arbeide utforskende i programmering, og at Janne så en styrke i at elevene fikk mulighet til å arbeide uten begrensende rammer. Det underbygges videre i neste utdrag.

Senere intervjuet forteller Janne at hun lar elevene få muligheten til å arbeide ut ifra en «oppskrift» eller ferdig løsning, men tilstreber at eleven prøver seg frem selv, og hun ønsket ikke at elevene skulle være en kopi av henne selv:

Janne: [...] Så jeg vil ... når de har satt sammen noen blokker så vil jeg at de skal teste det med ulike ting. Så utforske litt da. Så jeg vil jo ikke at de skal ... ja, jeg gir de en sånn oppskrift, men jeg vil ikke at de skal være en sånn tro kopi av meg, de står fritt i enkelte ting, og jeg vil alltid at de skal teste.

Ut ifra utdraget tolker jeg det som at Janne anser det som viktig at elevene prøver seg frem, og ikke bare følger en oppskrift hun gir dem. Det underbygger også tolkningen som ble gjort tidligere om at Janne har et mål om at programmeringsarbeid skal preges av

autonomi for elevene. Dette kan tyde på at utforskende arbeid og metodefrihet er viktig i Jannes undervisningstilnærming i programmering. Jeg tolker det også som at Jannes intensjon med å gi elevene en oppskrift (selv om hun gir uttrykk for at hun håper de finner en egen løsning) er at de skal ha et utgangspunkt, som gjør at elevene har noe å falle tilbake på hvis de opplever utfordringer i utforskningen og testingen. Det kan også gjøre det enklere for Janne å støtte elevene utforskning. Derfor mener jeg det kan tyde på at utforskende programmeringsarbeid er en målsetning med programmeringsarbeidet i Jannes undervisning, og at det derfor kan anses å være et kjennetegn ved hennes undervisningstilnærming i programmering.

4.3.2 Lars

I det første utdraget gir Lars uttrykk for at han hadde et stort behov for å ha kontroll tidlig i sin lærerkarriere:

Lars: [...] Jeg kan huske tilbake til den første timen jeg hadde og, fy flate altså. Jeg syns jeg var en drittdårlig lærer. Fordi jeg hadde det behovet for å ha kontroll, men det behovet for kontroll, det førte jo til at rammene, mulighetene til elevene ble begrenset. Hvis du holder noen i et tau, så får de ikke til å gå så veldig langt.

Jeg tolker Lars sin metafor om å holde noen i tau som at han mener at mange rammer, og lite rom for utforskning, vil begrense elevenes læring i matematikk. Det kan derfor tyde på at Lars mener at mindre rammer, og at elever får rom for å utforske og teste selv, vil kunne påvirke elevenes læring positivt. På bakgrunn av det tolker jeg at Lars ønsker en undervisningstilnærming i programmering som åpner for utforskning, metodefrihet og autonomi individuelt eller som gruppe for elevene. Derfor mener jeg at Lars tilstreber utforskende programmeringsarbeid i matematikk.

Den utforskende tilnærmingen til programmeringsarbeid kommer også frem i neste utdrag, der Lars beskriver formålet sitt med programmering i sin undervisning:

Lars: [...] Så når jeg forbereder undervisning så er det ikke ... formålet er ikke å drive forelesning, formålet er å fortelle om metoder, hvordan man kan utvikle program, skjønne at løkker kan være et kraftig verktøy, og skjønne at if-, else-, nestet if- kan gjøre at du kan gjøre ganske mye med korte linjer med kode når du skjønner. Hva er det her for noe? Hvordan kan jeg for eksempel løse noe med å skrive ut de første 25 primtallene.

Ut ifra dette tolker jeg det som at Lars ønsker å gi elevene de nødvendige verktøyene for å løse et problem, men at det er viktig at hvordan de ulike verktøyene tilegner elevene seg selv, og ikke kopierer han. Verktøyene i denne konteksten vil være generelle programmeringskonsepter som vilkår (slik Lars trekker frem i utdraget) eller løkker, som elevene videre må anvende på spesifikke problemer. Formålet slik han beskriver virker å være at elevene skal få erfare programmering på skolen, men for at de skal kunne bruke programmering må de tilegne seg programmering som en form for fremgangsmåte i møte med ulike problemer. Det kan tyde på at Lars anser en slik tilnærming som hensiktsmessig for størst mulig læringsutbytte for elevene. Fra et dobbel genesis perspektiv kan det virke som at Lars gjennom sin profesjonelle genesis tenker at en utforskende tilnærming til programmeringsarbeidet er det mest hensiktsmessige for å støtte elevenes instrumentelle genesis. Lars sin tanke om å gi elevene tilgang til generelle programmeringsprinsipper, men tilstrebe at de løser problemer uten begrensende rammer, kan også tyde på at han anser elevenes evne til selvstendig problemløsning som en ferdighet det er viktig å bygge i programmering. Ut ifra hvordan

Lars gir uttrykk for å ikke ha behov for å ha «kontroll» over elevene i arbeidet, og hvordan han beskriver sitt formål med programmeringsarbeid i matematikk, tolker jeg det som at Lars sin undervisningstilnærming til programmering i matematikk kan sies å være utforskende.

4.3.3 Simen

Når Simen ble spurt om å gi et eksempel på en god programmeringsaktivitet han hadde brytt trakk han frem en aktivitet der elevene arbeidet med BBP-plattformen Micro:Bit gjennom BitBot-robotene som ble beskrevet tidligere. I utdraget beskriver han aktiviteten kort:

Simen: Ja, vi brukte de her BitBotene nå da. Og det er kanskje et sånt eksempel der det ikke er så lett å finne matematikken i det. Du kan ikke bestemme at den skal snu 90 grader: Du kan si at den skal svinge til venstre med en viss prosent og en viss tid. Og så må du etter hvert da finne ut ... du finner ut at «hvis du har den tiden og den prosenten så svinger den 90 grader», og så svinger den 180 grader og ... ja. Så der har vi gjort noe sånt ... de første oppgave er ofte da at den skal kjøre 1 meter, snu 180 grader, og så kjøre 1 meter tilbake. Og så må de lage et program som tilsvarer det her da

Aktiviteten Simen beskriver legger opp til at elevene selvstendig eller i grupper skal utvikle et program som gjør det mulig å styre bitbot-roboten. Ut ifra måten Simen beskriver hvordan elevene fant ut hvordan de fikk bilen til å svinge mener jeg det tyder på at de ikke fikk en oppskrift, men prøvde seg frem med ulike verdier for prosenten og tiden bilen skulle svinge mens de utviklet programmet. Programmeringsarbeidet mener kan beskrives som utforskende, ettersom elevene ikke arbeidet etter en oppskrift fra læreren, som de skulle reprodusere, men fikk frie tøyler til å utforme et program som løste problemet slik de selv ønsket. Videre når Simen ble spurt om hvorfor han mente dette var en god økt fortalte han at dette var en økt han bare kunne sette i gang og deretter høre på elevene snakke og argumentere matematikk (Se utdrag i delkapittel 4.2 Programmeringens epistemologiske verd).

Ut ifra den beskrivelsen tolker jeg det som at elevene fikk arbeide og utforske selvstendig eller i grupper, uten noen begrensende rammer fra læreren. Basert på at Simens eksempel på en god programmeringsøkt bærer preg av en utforskende og testende tilnærming, der det oppfordres til metodefrihet frem reproduksjon av kode, mener jeg utforskende programmeringsarbeid og metodefrihet er en av kjennetegnene ved Simens undervisningstilnærming i programmering i matematikk.

4.4 Et liberalt syn på lærerrollen

Lærerne beskriver et syn på lærerrollen der læreren ikke nødvendigvis er den som kan mest i klasserommet, og at læreres kompetanse i programmering ikke behøver å være så stor for å bruke programmering hensiktsmessig i sin undervisning. Jeg valgte å beskrive synet på lærerrollen som liberalt fordi jeg mener det bryter med et mer tradisjonelt syn på lærerrollen. Tradisjonelt sett har rollefordelingen i klasserommet basert seg på at læreren er den kompetente, som deler sin kunnskap, og elevene er de uvitende som skal tilegnes seg lærerens kunnskap. Lærerne forteller at programmering åpner for at lærere i større grad kan bruke elevene som en ressurs, og basere seg på kollektiv læring (der læreren er inntar rollen som lærende sammen med elevene).

4.4.1 Janne

Når Janne ble spurt om hva hun mente var kjennetegn hos lærere som mestret å bruke programmering i matematikkundervisning svarte hun følgende:

Janne: Litt sånn ... ja... litt nysgjerrig på nye ting, og ikke være så redd for å prøve. Ikke være så redd for at man begir seg ut på noen man ikke vet så mye om. En som ikke synes det er så farlig å stå fast, eller gjøre feil.

Ettersom en av kjennetegnene Janne trekker frem er at læreren ikke er redd for å begi seg ut på noe de ikke vet så mye om, tolker jeg det som at Janne mener lærere ikke behøver å være «eksperter» for å bruke programmering i sin matematikkundervisning. Det kan i så fall tyde på at Janne ser det som fullt mulig for en lærer å bruke programmering som et hensiktsmessig didaktisk instrument selv med lite erfaring, og at programmering er en teknologi som åpner for en lærer som starter sin personlige genesis omtrent likt som elevene. Videre argumenterte Janne for hvorfor hun mente egenskapene kjennetegnet lærere som mestret å bruke programmering i matematikk:

Janne: Nei ... ofte så er det jo sånn at du kommer til timen, og så skal du jo kunne mer enn ungene (latter). Og visst du er helt fersk i programmering, så må du tåle å si til ungene at «det her veit jeg ikke, men det kan jeg sjekke til neste gang» eller «vi kan teste det ut sammen» eller ... være litt åpen på at du kan ikke alt, du er ikke noe ekspert, men at dere går veien sammen.

Hun trekker frem at læreren «ofte skal kunne mer enn elevene», men at i programmering er ikke nødvendigvis det tilfellet, og særlig ikke hvis du er uerfaren. Dette tolker jeg som at Janne mener det kan være en utfordring for lærere å kunne mye mer programmering enn elevene, og ha rollen som «ekspert», fordi det er en ny del av faget, og mange elever har stort kompetanse i programmering fra fritiden. Videre gir hun uttrykk for læreren bør være åpen om å ikke kunne alt, og at programmering er noe læreren kan lære sammen med elevene. Det kan tyde på at avstanden mellom elevene og læreren når det kommer til kunnskap er mindre i programmering sammenlignet med mer etablerte arbeidsformer, og at det kan føre til at læreren må ha et mer liberalt syn på sin rolle i undervisningen.

4.4.2 Lars

I intervjuet fortalt Lars om hvordan han presenterte programmering som en del av matematikkfaget for lærere på en konferanse, og at han forsøkte å kommunisere at programmering litt forenklet kunne sees på som å «få en dum ting til å gjøre det du vil». Han gjennomførte en un-plugged programmeringsaktivitet der lærerne skulle instruere hverandre. Resultatet fra aktiviteten var ifølge Lars at flere lærere oppdaget et potensial i programmering, og at terskelen for å bruke det i undervisning ble tilsynelatende senket. I det neste utdraget forteller Lars om dette:

Lars: [...] Så erfaringen min gjør at jeg kan da tenke mye lenger enn en som kanskje er litt usikker. Jeg vet at jeg møter folk her, og de er litt sånn ... ikke eksperter på programmering, jeg må få de til å tenke «det her er ikke farlig», og så kan de gjøre det med elever og. Hva er det her faget for noe? Det er et fag med mange muligheter, det er opp til dere.

Ut ifra Lars sin beskrivelse av programmering, tolker jeg det som at Lars mener programmering er en arbeidsmåte som ikke nødvendigvis krever så stor formell kompetanse for å bruke. Det kan virke som at Lars tenker at det viktigste er at lærerne ikke er redde for å prøve seg frem i programmering, og at ved å prøve seg frem vil lærerne etter hvert bruke det i undervisning. Fra et dobbel genesis perspektiv kan det

forklares som at læreren ikke vil starte på sin profesjonell genesis uten å samhandle med artefaktet «programmering», og at uavhengig av hvilken grad lærerne har instrumentert og instrumentalisert programmering er de nødt til å starte en plass. Jeg tolker det som at det er en liknende tanke Lars forsøkte å kommunisere på konferansen. En slik tilnærming vil i tilfelle åpne for at læreren kan ta i bruk programmering tidlig i sin profesjonelle genesis, og omforme artefaktet til et personlig instrument samtidig som elevene, samtidig som den profesjonelle genesisen utvikler et bedre og mer fleksibelt didaktisk instrument for læreren. Denne måten å se på lærerens krav til kompetanse er på mange måter lik som den Janne gav uttrykk for, der de mener lærere kan bruke programmering i matematikk uten å være noen «eksperter». Jeg tolker det som at Lars tenker at terskelen for å bruke programmering er lavere enn det de mindre erfarne lærerne tror, og at det handler om å prøve det ut så man er tørr å teste det ut i undervisning, og på den måten utvikle kunnskapen. I neste utdrag gir Lars uttrykk for at lærere som ikke kan programmering tror de må ha god kompetanse for å bruke det i sin undervisning:

Lars: Ja, for en som ikke kan programmering, så vil det være sånn at de blir stressa av å se hvor mange programmeringsspråk det er, for da tenker du at du må en veldig god kompetanse i alle for å undervise i det. Jeg er så trygg på meg selv, på den måten jeg har lagt opp matematikk- og programmeringsfaget ... at jeg vil at eleven skal få utforske.

I utdraget sammenligner han uerfarne lærere med seg selv, og peker på at hans trygghet i programmering gjør at han kan la elevene utforske. Jeg tolker det som at Lars tenker at erfaringen hans gir han en trygghet og et overblikk som gjør at han inntar en rolle der han er komfortabel med at han ikke må kunne alt, fordi han er trygg på den generelle tilnærmingen han har lagt opp til i matematikkfaget. Det kan tyde på at Lars har en oppfatning om at uerfarne lærere tror de må kunne alle programmeringsspråk for å bruke programmering, og at dette er et stressmoment Lars ikke kjenner på fordi hans tilnærming og PTK åpner for det. Han fortalte senere at han oppfordret elevene til å prøve ut programmeringsspråkene de selv ønsket, selv om det var språk han ikke kunne, fordi arbeidet munnet uansett ut i «en kompetanse som heter programmering». Dette tolker jeg som at Lars tilnærmer seg programmering med en tanke om at han ikke må være den som kan mest av alle om alt, og at han kan støtte elevens utvikling selv på språk han ikke kan, fordi han forholder seg til en mer generell programmeringskompetanse. Lars sine beskrivelser maler et bilde av lærerrollen som noe annet et «kunnskapssenter» eller ekspert, der lærere kan benytte seg av ukjent teknologi gitt at de har en god didaktisk tilnærming til arbeidet.

4.4.3 Simen

Simen gav tidlig i intervjuet uttrykk for at programmering var noe nytt, som i mange tilfeller stilte nye krav til lærerne. I det første utdraget beskriver Simen hvordan programmering har påvirket inngangen til undervisning:

Simen: [...] Jeg liker ikke å gå inn i en undervisningsøkt og ikke kunne helt, og ikke helt vite hva skal, men sånn er det jo med programmering da, fordi du veit jo aldri hva elevene finner på å gjøre. Plutselig så finner de på en helt annen måte å gjøre ting på enn det jeg ville gjort, og da må du på en måte prøve å sette deg inn i deres utgangspunkt på hvordan de skal gjøre noe.

Jeg tolker det som at Simen mener at endringen i lærerrollen kommer som følge av at programmering er en teknologi som åpner for mange ulike fremgangsmåter for elevene. Det gjør at læreren risikerer å stå overfor en løsning de ikke har forberedt seg på, og at kompetansekravet til læreren blir umulig å oppnå dersom de ønsker å oppfylle den

tradisjonelle lærerrollen som klasserommets «ekspert». Derfor tolker jeg fra utdraget at Simen mener programmering er en såpass ny teknologi, men at lærere kan bli kjent med teknologien, og starte sin profesjonelle genesis uten lang erfaring. Det introduserer i likhet med de andre et liberalt syn på hvilken rolle læreren skal fylle, og Kompetansen som kreves for å ta i bruk programmering i sin undervisning. Ut ifra Simens beskrivelser mener jeg det tyder på at hans tilnærming til programmering preges av et syn på sin egen rolle i klasserommet som gjør at Simen benytter seg av programmering selv om han ikke kan alt.

Videre i intervjuet åpnet Simen for at lærere kunne støtte seg på elevene, og utnytte deres kunnskaper som en ressurs for sin undervisning:

Simen: Og så kommer det litt an på hvor høyt opp du gjør det her da. Fordi ... eh ... jeg tenker at jo større elevene er, jo enklere er det for å si at «jeg veit ikke helt det her», og kanskje vi må lære oss det her sammen, også finner vi ut av det her sammen. Hvis du er på jobb på ungdomsskolen, så er det nesten sikkert at noen har gjort et eller annet før, og kan et eller annet ... eh ... også får man det til.

I dette utdraget gir Simen uttrykk for at læreren kan bruke elevene som en ressurs om de ikke har den nødvendige kunnskapen selv. Simen beskriver med det et syn på lærerens rolle som har mange likheter med det Janne beskrev, der det åpnes for at læreren kan utvikle et matematisk instrument gjennom sin personlige genesis samtidig som elevene, og samtidig utvikle det didaktiske instrumentet gjennom profesjonell genesis. Dette kan tyde på Simen er åpen for at læreren kan bruke programmering i sin matematikkundervisning uten å ha lang erfaring eller noen særlig formell programmeringskompetanse, som jeg mener kan anses å være en liberal holdning til læreres krav til kompetanse. Det kan også tyde på en stor endring i lærerrollen der læreren kan innta en lærende rolle på lik linje som elevene. Synet på lærerrollen underbygges senere i intervjuet hevder Simen at en lærer ikke behøver å kunne alt som inngår i en programmeringsoppgave får å benytte den i sin undervisning, så lenge oppgaven det arbeides med er innenfor rekkevidden til elevene.

4.5 Oppsummering av funn

I analysekapittelet tok jeg utgangspunkt i Thomas og Palmers (2014, s. 85-86) generelle beskrivelse av en lærer med høy PTK, som beskriver trekk ved lærerens holdninger og praksis. Deretter presenterte jeg utdrag fra datamaterialet der jeg viste hvordan lærernes PTK kom til syne, og hvorfor jeg mente de hadde høy PTK. Lærere gav uttrykk for sin PTK i hovedsak gjennom positive holdninger og tanker rundt programmeringspotensial og nytteverdi i matematikk. Videre tolket jeg deres grad av personlig- og profesjonell genesis ut ifra utdragene, og vurderte det til at lærerne var kommet langt på vei i sine genesisprosesser. Faktoren *matematisk kunnskap for undervisning* (MKT) ble ikke undersøkt i stor grad i intervjuet, på bakgrunn av at jeg vurderte det utfordrende å få lærerne til å fortelle om det. Lærerne hadde mellom 10 og 16 års erfaring, og basert på deres erfaring vurderte jeg det til at lærerne hadde en matematisk og pedagogisk kunnskap som var forenelig med at de hadde høy PTK. Erfaringen mener jeg kan føre til at lærerne har en trygghet i sin rolle som lærer som gjør at de lettere kan integrere ukjent teknologi i sin undervisning. Denne antakelsen mener jeg ble underbygget av kjennetegnene *utforskende undervisning og metodefrihet* og *liberalt syn på lærerrollen*. Ut ifra kriteriene jeg oppga for at en lærer skulle ha høy PTK vurderte jeg alle tre lærerne til å være kyndige i programmering og positive til påvirkningen programmering kunne

ha. Derfor konkluderte jeg i første del av analysen med at alle tre lærerne i utvalget hadde høy PTK i programmering.

I den andre delen av analysen presenterte jeg tre kjennetegn på lærernes tilnærming til undervisning med programmering. Kjennetegnene omhandlet programmering som en støtte for elevenes forståelse av matematikk, en utforskende undervisningstilnærming der metodefrihet og elevenes autonomi var i sentrum, og et syn på lærerrollen og lærers kompetanse som bryter med det tradisjonelle synet på læreren.

Det første kjennetegnet viste at lærerne i utvalget så en epistemologisk verdi i programmering. Lærerne knyttet den epistemologiske verdien i hovedsak til arbeid med variabler og prosent. Funnene støttes av Artigue (2002), som hevder at erfarne og kyndige teknologibrukere ofte ser et epistemologisk potensial, i motsetning til mindre erfarne som ofte kun utnytter det pragmatiske potensialet. Det passer også med Thomas og Palmers (2014, s. 85) generelle beskrivelse av lærere med høy PTK. Der trekkes det frem at lærere med høy PTK lar matematikken stå i sentrum selv når arbeidet baserer seg på teknologi.

Funnene fra analysen viser at lærerne i utvalget ønsker en utforskende tilnærming til programmeringsarbeidet, der elevene oppfordres til å teste og prøve seg frem. Tilnærmingen kjennetegnes også gjennom at lærerne ikke ønsker begrensende rammer, og metodefrihet og elevenes autonomi står sentralt. Tilnærmingen passer godt med Thomas og Palmers (2014, s. 85) beskrivelse av lærere med høy PTK, der lærere lar eleven utforske og samhandle med matematiske konsepter individuelt eller i gruppe.

Lærerne beskriver en endring i lærerrollen. I analysen gir de uttrykk for at programmering ikke krever mye formell kompetanse for å bruke i sin undervisning, og to av lærerne åpner opp for at læreren kan lære «med» elevene, og utnytte dyktige elever som en ressurs. Jeg mener dette kan tyde på at programmering kan introdusere en endring i den pedagogiske tilnærmingen til matematikklærere, noe jeg vil komme tilbake til i neste kapittel.

5 Drøfting

I dette delkapittelet drøftes funn fra analysen for å belyse problemstillingen: «Hva kjennetegner undervisningstilnærmingen til et utvalg lærere med høy pedagogisk teknologikunnskap i programmering?».

Kjennetegnene fra forrige kapittel drøftes i hvert sitt delkapittel i lys av relevant forskning og rammeverket PTK med dobbel genesis. Avslutningsvis gjør jeg rede for min oppfatning av sammenhengen mellom programmering og matematikk.

5.1 Programmeringens epistemologiske verdi

I analysen kommer det frem at programmering ikke bare hadde en pragmatisk verdi for lærerne, men at de også så et potensiale for å bygge elevenes forståelse av matematikk. Lærere med høy PTK ser ifølge Thomas og Palmer (2014, s. 85) at programmering kan anvendes utover sin pragmatiske verdi, og at i deres bruk av teknologi ligger fokuset i hovedsak på matematikken, fremfor teknologien. Disse poengene gjenspeiles i datamaterialet, ettersom alle lærerne gir uttrykk for å se programmering som noe mer enn et digitalt hjelpemiddel som gjør at elevene kan gjøre oppgaver raskere. Et spørsmål som dukker opp som i forbindelse med funnet er hvorvidt det er ønskelig at matematikklærere utnytter en epistemologisk verdi i programmering. Det er naturligvis ønskelig med lærere med høyest mulig PTK, men er hensikten med innføringen av programmering at elevene skal bruke programmering for å lære matematikk, eller for å tilegne seg programmering som et verktøy på bakgrunn av det pragmatiske potensialet? Ifølge Artigue (2002) og Thomas og Palmer (2014) bruker mindre erfarne lærere i større grad det pragmatiske potensialet i teknologien. Det kan tyde på at lærere må utvikle sin PTK og profesjonelle genesis slik at de kan utnytte den epistemologiske verdien av programmering. Hvis det skal være en målsetning om at matematikklærere skal ha teknologikunnskapen til å utvikle elevenes matematiske forståelse gjennom programmeringsarbeid kan det være behov for utbredt etterutdanning og kompetanseheving. Popat og Starkey (2019, s. 368) hevder at programmering egner seg for å utvikle elevenes problemløsningskompetanse. Ettersom problemløsning inkluderes som et av matematikkfagets kjerneelementer i den nye læreplanen (Kunnskapsdepartementet, 2019), vil problemløsning etter min mening være en potensiell innfallsvinkel for uerfarne lærere som ønsker at elevene skal lære sentrale matematiske ferdigheter, i tillegg til hvordan de programmerer. Gjennom å bruke programmering jevnlig vil disse uerfarne lærerne starte utviklingen av sin PTK og profesjonelle genesis, og vil etter hvert ha muligheten til å utnytte programmeringens epistemologiske potensial over store deler av pensum i matematikk.

Et viktig aspekt å belyse i forbindelse med problematikken i forrige avsnitt er hvordan kompetansemålene for programmering i matematikk er formulert. For eksempel skal elevene på 6. trinn «bruke variabler, løkker, vilkår og funksjoner til å utforske geometriske figurer og mønstre», og elevene på 10. trinn skal «utforske matematiske egenskaper og sammenhenger ved å bruke programmering» (Kunnskapsdepartementet, 2019). Derfor mener jeg at enkelte av kompetansemålene kan sies å legge opp til at programmering skal brukes for å utforske matematikken, og på den måten utnytte en epistemologisk verdi i programmering. Det vil i tilfellet si at norske matematikklærere

skal utnytte den epistemologiske verdien slik lærerne i denne studien har trukket frem. Dermed blir kjennetegnet noe mer enn en didaktisk mulighet som dyktige lærere utnytter i programmeringsarbeid, det blir noe alle matematikklæreres undervisningstilnærming i programmering i prinsippet burde kjennetegnes av (avhengig av hvilke trinn det undervises på). Videre mener jeg det kan underbygge Misfeldt et al (2019) forslag om at lærernes programmeringskompetanse burde utvikles med utgangspunkt i matematikkfaget. På den måten vil kompetanseutviklingen ha en tett tilknytning til den gjeldende læreplanen, og veien for lærere til å bruke programmering som en hensiktsmessig del av sin undervisning er kanskje kortere.

En annen utfordring med å utnytte det epistemologiske verdien av programmering stammer fra at programmering er en relativt ny del av faget. Programmering ble en obligatorisk del av læreplanen i skoleårene 20/21 (1. – 9. trinn) og 21/22 (10. trinn). Etter snaut to år i skolen er det naturlig å tenke at deler av lærerstanden kan ha manglende erfaring, og en PTK som gjør det utfordrende å arbeide på denne måten. En studie (Vinnervik, 2020) som undersøkte læreres opplevde utfordringer med programmering i forkant av innføringen i den svenske skolen viste at mangel på tid og dårlige forhold for profesjonell utvikling var en utfordring. Det tenkelig at norske lærere kan oppleve liknende utfordringer. Derfor kan det være vel ambisiøst å forvente at lærere med lav programmeringskompetanse skal designe økter som går forbi å lære programmering som en ferdighet, der elevene samhandler med matematiske konsepter. Spørsmålet blir da: hva er målsetningen til den norske skolen? Er det en målsetning at samtlige matematikklærere til slutt skal være i stand til, og ha et ønske om, å utnytte programmerings epistemologiske potensial, eller er hensikten med innføringen av programmering at elevene skal lære seg grunnleggende konsepter i koding? Dette kan være en viktig presisering å kommunisere til matematikklærere, slik at flest mulig vet hva som forventes av dem. Hvis ønsket er at elevene skal forstå matematikk gjennom programmering, kan erfaring og grad av profesjonell genesis spille en viktig rolle, slik det virker å ha gjort for lærerne i denne studien. Er målsetningen derimot at programmering er innført for å gi elevene en digital ferdighet, som er viktig å kunne i seg selv, er det liten grunn til at programmering ikke skal være en del av klasserom over hele Norge.

Elevenes manglende erfaring med programmering kan vise seg å være en utfordring på samme måte som manglende kompetanse hos lærerne. Programmering kort fartstid i skolen, og for mange elever kan deres første møte med programmering være gjennom matematikken. For at elevene skal kunne bruke programmering til å jobbe med matematikk må de naturligvis ha en viss programmeringskompetanse. Dette kan være en hindring for at programmering brukes i matematisk arbeid, særlig for de av lærerne som selv ikke har startet med programmering (der vil kanskje lærer og elever ha behov for grunnopplæring). I rapporten til Sanne-utvalget (2016, s. 77) poengterte de at det var stofftrengsel i den norske skolen. Dette kan virke hemmende på implementeringen av programmering, ettersom lærere kan prioritere andre kompetansemål eller andre teknologier, slik at programmering benyttes i svært liten grad, eller ikke i det hele tatt. I en slik situasjon er det sannsynlig at programmering vil brukes for sitt pragmatiske potensial, og at timene brukes utelukkende for å lære seg å kode. Det er mulig at en slik utfordring vil løses av seg selv, ettersom læreres erfaring og kompetanse vil øke for hvert år programmering er en del av læreplanen, gitt at de faktisk benytter seg av det. Det riktignok et problem for skolen hvis lærere ikke opplever å ha tilstrekkelig kompetanse for å benytte seg av programmering i matematikk, og det er tilsynelatende ikke tiltak som adresserer eventuelle kompetansemangler. Da vil ikke programmering

oppleve en gradvis positiv utvikling for hvert år, slik jeg beskrev, og programmering vil kunne utvikle seg til å bli en større og større utfordring.

Til tross for at programmering nylig ble en del av læreplanen, virker ikke lærerne i denne studien å påvirkes av utfordringene som jeg har beskrevet i stor grad. Jeg mener påstanden underbygges av synet på lærerrollen og lærerens programmeringskompetanse som drøftes i siste delkapittel. Hva er bakgrunnen for dette? Naturligvis har disse lærerne som nevnt tidligere erfaring med programmering fra før fagfornyelsen. En annen forskjell fra lærerne i dette utvalget og lærere som opplever programmering som en utfordring er hvorvidt de ser en klar sammenheng mellom matematikk og programmering. Janne forteller at hennes kolleger ikke ser nytten av matematikken i programmering. Lars og Simen knytter programmering til prosent, algebra, sannsynlighet og geometri. Jeg mener dette aspektet ved deres PTK kan være svært utslagsgivende. Det vil være vanskelig for en lærer å bruke programmering hensiktsmessig i matematikk hvis de ikke ser noen tydelig sammenheng mellom de to. I Stigberg og Stigbergs (2020) studie er læreres syn på sammenhengen mellom programmering og matematikk lagt frem som en utfordring. De forteller at lærerne i studien har god kjennskap til læreplanen, men hevdet at lærerne behøvde støtte for å utvikle undervisningskompetanse i programmering og deres forståelse av sammenhengen mellom matematikk og programmering. Det kan tyde på at det er viktig å se en klar sammenheng for lærere, og jeg tror at lærere som ikke ser en sammenheng vil ha lav motivasjon og vilje til å integrere programmering i sin undervisning. Hvis det er tilfellet tror jeg lærerne unngår programmering, som naturligvis er en utfordring skoleverket bør adressere. Da bør de etter min mening gjøre lærerne bevisste på hvordan matematikk og programmering henger sammen, og når den «grunnmuren» er bygget kan den epistemologiske verdien utnyttes.

5.2 Utforskende undervisning

Lars trekker frem at lærere må være villig til å «miste kontrollen» for å ikke begrense elevene. Jeg tolket det som at Lars bruker «miste kontrollen» som en slags motsetning til en undervisningstilnærming preget av tydelige rammer og instruksjon fra læreren. I PTK-rammeverket er lærerens evne til å miste kontrollen og la elevene samhandle med teknologien og matematikken en del av hva som beskriver en lærers pedagogiske teknologikunnskap. Thomas og Palmer (2014, s. 85) knytter behovet for å ha kontroll til lærere med lav PTK, og mener lærere med høy PTK i større grad lar elevene arbeide uten begrensende rammer. Lærerne gir uttrykk for at gode økter med programmering har elementer av utforskning og en «mangel» på kontroll. I analysen hevder Lars at han ikke klarte å miste kontrollen og la elevene utforske kort tid etter han begynte som lærer. Lars sine beskrivelser av seg selv som ung lærer tyder på at erfaring kan være en faktor som er svært avgjørende for hvorvidt en lærer klarer å slippe taket og fjerne rammene. Erfaringsargumentet kan videre knyttes til lærerens selvtillit som pedagog og faglærer. Men hva skal uerfarne, eller kanskje nyutdannede lærere gjøre? Hvordan skal man erstatte erfaringskomponenten slik at selv uerfarne lærere får utviklet sin pedagogiske teknologikunnskap slik at de kan arbeid utforskende? Clark-Wilson og Hoyles (2019) mener at målrettet profesjonsutvikling kan være en løsning for å løfte de lærerne som ikke opplever å ha de nødvendige kunnskapene for å bruke programmering. For å løfte kompetansen på etablerte og nyutdannede lærere, og ha muligheten til kollektiv utvikling mener jeg det kan være hensiktsmessig å etablere faggrupper som utvikler eksempler på hvordan programmering kan være en støtte for elevenes forståelse av matematikk eller eksempler på gode undervisningsopplegg der elevene får arbeide utforskende og oppdagende. På den måten kan uerfarne lærere nyte av erfaringen og refleksjonene til

lærere med høy PTK, slik som lærerne i denne studien. Det kan være med på å gjøre programmering til en fullverdig del av matematikkfaget for lærere raskere.

Funnene i analysen belyser lærernes fokus på utforskende undervisning i arbeid med programmering. Som jeg nevnte tidligere blir undervisningen beskrevet som utforskende fordi lærerne brukte begrepet eksplisitt, eller fordi det er undervisning der det åpnes for flere fremgangsmåter, prøving og feiling og høy grad av autonomi for enkeltelever eller grupper av elever. I programmering er det ofte mulig å løse et problem gjennom en hel rekke ulike fremgangsmåter. Dette mener jeg kan være noe av forklaring for lærernes tilnærming til programmeringsarbeidet. Janne presiserer for eksempel at hun gir elevene en fremgangsmåte, men at hun ikke ønsker at elevene skal kopiere denne, og hun ønsker at elevene tester selv, som tyder på at Jannes gir elevene ferdig fremgangsmåte som et utgangspunkt for utforskning. Noen av de samme fokusområdene kommer frem gjennom utdragene fra Lars. Han maler også et bilde om at programmeringskompetansen deres ikke bør «overføres» (i form av forelesning fra læreren) til elevene. Dette utsagnet tolker jeg som at Lars støtter en undervisningstilnærming i programmering der elevene må oppdage kunnskapen selv. Bray og Tangney (2017, s. 23) presenterer fellestrekk ved teknologibasert undervisning. De trekker frem undervisningstilnærminger som oppfordrer til utforskning og samarbeid som et kjennetegn ved god praksis. Andersen et al (2021, s. 133) hevdet i sin studie at programmering kunne være en god inngang til utforskende undervisning i matematikk. Lærere med høy PTK tilstreber i større grad at elevene skal få rom til å «samhandle med konseptuelle matematiske ideer gjennom utforskning individuelt eller i gruppe» (egen oversettelse) (Thomas & Palmer, 2014, s. 85). Dette kan tyde på at bruk av teknologi ofte løses best ved en åpen tilnærming, der metodefrihet og autonomi står sterkt. Derfor tror jeg programmeringens «natur» oppfordrer til en utforskende tilnærming, og at lærerne i utvalget mener at programmeringskompetanse ikke bygges best gjennom forelesning, men gjennom utforskning, prøving og feiling og oppdagelse.

Programmering handler om å utvikle programmer, som er stegvise instruksjoner som får en datamaskin eller robot til å utføre en oppgave. Da mener jeg det er naturlig å tenke at en undervisningstilnærming der elevene for eksempel reproducerer effektiv kode er hensiktsmessig. Effektiv kode i denne sammenhengen betyr kode som løser problemet enklest og raskest mulig. Da jobber elevene tett på «løsningen», men vil i stor grad reproducere andres kode, og det er lite som åpner for utforskning eller egne fremgangsmåter for elevene. Det blir derfor naturlig å stille spørsmålet: hvorfor skal man jobbe utforskende med programmering? Hvis målet med arbeidet for eksempel er at elevene skal lære seg å kode for å løse problemer, hvorfor bør de ikke kopiere effektiv og godt designet kode? Jeg mener at lærernes innfallsvinkel kan delvis forklares gjennom dobbel genesis. Personlig genesis er en lang og komplisert prosess, der læreren omformer et artefakt til et matematisk instrument for personlig bruk. Profesjonell genesis omhandler prosessen der læreren omformer et artefakt til et didaktisk instrument, men den hensikt å støtte elevenes instrumentelle genesis. Sammen utgjør de to Haspekians (2014) dobbel genesis. En forklaring på hvorfor lærerne tilstreber å fjerne rammer, og la elevene prøve seg frem, kan finnes i deres personlige genesis. Hvis lærerne i sin personlige genesis opplevde at en utforskende tilnærming resulterte i større grad av instrumentering og instrumentalisering, er det naturlig at de gjennom sin profesjonelle genesis tilstreber å støtte elevenes instrumentelle genesis på samme vis. Det kommer frem i Lars sin beskrivelse av sin kompetanseutvikling, der han forteller at «knoting» er en sentral del av det. Knoting i denne sammenhengen tolker jeg som utforskning og testing på egenhånd. Det vil i så fall kunne tyde på at programmering, til

tross for at det i hovedsak er stegvise instruksjoner, der læreren kan lage en «oppskrift», krever mer enn reproduksjon for at elevene skal sitte igjen med et matematisk instrument. Hvis det er tilfellet, vil det ha implikasjoner for hvordan lærere bør tilnærme seg programmering.

5.3 Liberalt syn på lærerrollen

I analysen kommer det frem at lærernes syn på hvilken kompetanse en lærer behøver for å bruke programmering, og hvilken rolle en lærer kan i programmeringsarbeid kan beskrives som *liberalt*. Det kan beskrives som liberalt i den forstand at synet på lærerrollen i programmeringsarbeid bryter med en mer tradisjonell rollefordeling mellom lærer og elev, der læreren er «kunnskapscenteret» og eleven er en passiv mottaker av denne kunnskapen. Lærerne i denne studien beskriver en ny måte å se på lærerrollen, og kompetansen en lærer behøver. Janne og Simen gir uttrykk for at programmering er en arbeidsmåte der særlig uerfarne lærere må akseptere at de ikke nødvendigvis har rollen som «den kompetente». De åpner opp for at programmeringsarbeid kan gjøre at læreren er ferdighetsmessig likestilt eller underlegen sine elever. Når dette skjer kan lærerens rolle som foreleser eller kunnskapscenter endres, slik at læreren i større grad fungerer som en veileder og tilrettelegger for elevenes læring. Ut ifra lærernes beskrivelser av rollefordelingen tolket jeg det til at programmering var en arbeidsmåte der elevene kunne løse en oppgave på så mange forskjellige måter at det ikke var mulig for læreren å alltid være «den kompetente». Bray og Tangney (2017, s. 4) hevder at den tradisjonelle lærerrollen slik den har blitt beskrevet ikke nødvendigvis passer godt sammen med økt bruk av teknologi, og de underbygger bildet av læreren som en tilrettelegger fremfor et kunnskapscenter. Jeg mener derfor økt bruk av teknologi, og rask teknologisk utvikling i klasserommet, kan utfordre den tradisjonelle lærerrollen. Det vil i så fall kunne medføre at lærere i dag kan oppleve en konflikt mellom deres «vanlige» pedagogiske tilnærming og kompetansen de innehar i programmering, der kompetansen deres ikke er tilstrekkelig for å oppfylle rollen som en ekspert for elevene. Janne og Simens beskrivelser kan tyde på at programmering kan bringe med seg en endring i lærerrollen. Hvordan kan endringen forklares? Programmering brukes ofte sammen med andre teknologier, og det vil stilles store krav til kompetanse hos lærerne, som stadig må videreutvikles. Kravet til oppdatert kompetanse kan gjøre det vanskelig for lærere å holde på den tradisjonelle rollefordelingen i klasserommet. En konsekvens kan være at lærere i større og større grad må innta rollen som veileder eller tilrettelegger for å kunne støtte elevenes læringsprosesser over et stadig større utvalg av digitale hjelpemidler. Veilederrollen trekkes frem som et av kjennetegnene på god undervisning med teknologi av (Bray & Tangney, 2017, s. 23). Med utgangspunkt i dette mener jeg at hyppigere bruk av teknologi kan endre hvordan lærere tilnærmer seg sin undervisning, og at det liberale synet som lærerne i dette utvalget gir uttrykk for kan være en indikator for hvilken retning utviklingen av lærerrollen beveger seg.

Et interessant aspekt ved dette funnet er at lærerne, som jeg har tatt utgangspunkt i at er svært kompetente, mener at lærere kan bruke programmering hensiktsmessig uten lang erfaring med programmering eller formell kompetanse. Til tross for erfaringen og den pedagogiske teknologikunnskapen de har bygget opp, maler de et bilde av at programmering ikke krever så veldig mye forkunnskaper for å benytte seg av det i sin undervisning. Læreres selvtillit med en gitt teknologi er tett knyttet opp mot deres PTK (Clark-Wilson & Hoyles, 2019; Thomas & Hong, 2013). Selvtillit kan påvirke hvorvidt lærere tar steget mot å ta i bruk programmering, og at en manglende selvtillit kan være med på å forklare hvorfor enkelte lærere ikke bruker programmering i sin undervisning,

til tross for at det er en obligatorisk del av læreplanen. Lærerne i denne studien er erfarne pedagoger, og har en teknologikunnskap som gjør at de bruker programmering, men også utforsker og tester nye måter det kan være et hjelpemiddel for dem. Fra et dobbel genesis (Haspekian, 2014) perspektiv kan dette forklares gjennom instrumentalisering, der læreren stadig oppdager et artefakts didaktiske potensial. Hvordan kan man gå frem for å bygge selvtillit og utvikle profesjonell genesis hos lærere som ikke har erfaringsgrunnlaget og teknologikunnskapen som lærerne i dette utvalget har bygget opp? Ifølge Clark-Wilson og Hoyles (2019, s. 354) kan en innfallsvinkel være å bygge et sortiment av spesifikke eksempler på hvordan en teknologi kan brukes i klasserommet, som kan brukes som grunnlag for faglig diskusjon og utvikling i et profesjonsfelleskap. Thomas og Hong (2013, s. 82) trekker frem at støtte fra kollegaer kan påvirke en lærers selvtillit i stor grad, og at dette vil forplante seg videre og øke lærerens PTK gjennom deres personlige holdninger og verdier. Lærerne som hadde lav selvtillit i Thomas og Palmers (2014) studie gav uttrykk for at de utviklet sin kompetanse i hovedsak gjennom håndbøker, nettsteder, workshops eller elevene. Lærerne med medium og høy selvtillit utviklet og tilegnet seg sin kompetanse i mye større grad fra kollegaer. På bakgrunn av dette tenker jeg at kompetanseutvikling i et profesjonsfelleskap på skolene kan være en nøkkel for å få flest mulig lærere til et ferdighets- og selvtillitsnivå som gjør at de bruker programmering i sin undervisning. En fordel med en slik løsning, sammenlignet med for eksempel workshops eller kurs, er at det åpner opp for en kontinuerlig utvikling i et felleskap, der hver enkelt lærers profesjonelle genesis kan bidra inn i en form for «kollektiv profesjonell genesis». En kollektiv profesjonell genesis vil la en gruppe lærere sammen omforme et artefakt til et didaktisk instrument. Da vil gruppa ha muligheten til å utnytte andres erfaringer og kunnskaper, og prosessen fra artefakt til instrument går fortere. Det kan for eksempel være at matematikklærerne på 6. trinn går sammen for å gjøre Scratch til et didaktisk instrument i geometri, eller at lærere i 9. klasse diskuterer hvordan de kan la elevene få erfaring med sannsynlighets simuleringer gjennom å programmere i Python. På den måten vil det være enkelt å støtte nye og uerfarne matematikklærere med å bruke programmering som en hensiktsmessig del av sin undervisning.

5.4 Programmering og matematikk

Selv om sammenhengen mellom programmering og matematikk ikke har vært et av fokusområdene for denne studien, mener jeg likevel det er viktig å drøfte. Sanne-utvalget (2016, s. 25) knyttet programmering til tilegnelse av algoritmisk tenkning, og mente det hadde likheter med matematisk logikk. Algoritmisk tenkning er en del av kjerneelementet *utforskning og problemløsning* (Kunnskapsdepartementet, 2019), og derfor kan sammenhengen mellom programmering og matematikk blant annet knyttes til elevers problemløsningskompetanse, generaliseringsevne, autonomi og samarbeid. Gjennom denne studien har jeg fått innsikt i ulike måter programmering brukes i matematiske kontekster som følge av lærernes beskrivelser. Jeg tenker at programmering i en skolekontekst bør sees på som noe mer enn koding, og at algoritmisk tenkning bør stå sentralt. Ferdighetene som beskrives i algoritmisk tenkning er viktige fordi de er verdifulle ferdigheter for elevene utenfor matematikkfaget også. Flere av momentene i algoritmisk tenkning mener jeg har likhetstrekk med ferdighetene som trekkes frem i *21st century skills* (Schleicher, u.å), som beskriver hvilke elementære ferdigheter som blir viktige i fremtiden. Derfor mener jeg at sammenhengen mellom programmering og matematikk kommer frem gjennom algoritmisk tenkning, og

sammenhengen viser hvorfor programmering er et viktig didaktisk verktøy, som matematikklærere bør benyttes seg av.

Når Papert (1980) introduserte programmeringens didaktiske potensial, ble sammenhengen mellom matematikk og programmering i hovedsak vist gjennom geometri. Flere programmeringsplattformer (særlig BBP) er inspirert Paperts (1980) programmeringsspråk LOGO. Sammenhengen mellom programmering og matematikk kan komme til syne i arbeid med geometri i LOGO. Hvis elever skal tegne geometriske figurer, kan de se figurene ut ifra to ulike perspektiver: *skilpaddeperspektivet* og *kartperspektivet*. Når elever skal tegne ut ifra skilpaddeperspektivet vil de være i planet, og kan visualisere figuren gjennom å «gå» tegningen. Da vil elevenes tankerekke og diskurs ta utgangspunkt i at de skal gå en lengde, og deretter svinge til høye eller venstre. Skilpaddeperspektivet tar i større grad utgangspunkt i elevenes intuitive matematiske forståelse. Elevene kan også se geometriske figurer i kartperspektivet. I kartperspektivet ser elevene figurene fra «luften», og elevene tar utgangspunkt i koordinater og rotasjon for å beskrive hvordan den geometriske figuren blir tegnet. Kartperspektivet er perspektivet elevene møter i den formelle geometrien i skolen, og det baserer seg i større grad på lært matematisk kunnskap. Jeg mener at arbeid med LOGO åpner for at elevene kan kombinere de to perspektivene. Samtidig kan arbeid med LOGO støtte overgangen fra skilpaddeperspektiv til kartperspektiv, altså en overgang fra en intuitiv forståelse av en geometrisk figur, til en formell matematisk forståelse av figuren. Samtidig er den matematiske konteksten et utgangspunkt for å utvikle elevenes programmeringskompetanse. På den måten beriker programmering og geometri hverandre. Derfor mener jeg at arbeid med geometri ved hjelp av LOGO kan belyse en sammenheng mellom matematikk og programmering.

Programmering har et særtrekk som man ikke finner i så mange andre teknologier: brukeren må utvikle funksjonene selv. Det mener jeg gjør at elevene tvinges til å samhandle med matematiske ideer og fremgangsmåter, og på den måten vil matematikken «bak» konteksten være sentral, gitt at det arbeides med matematiske problemer. Annen teknologi som gir elevene ferdige kommandoer vil i større grad begrense seg til å være et pragmatisk verktøy, og tidligere i denne studien har jeg tatt for meg hvorfor det kan være viktig å utnytte den epistemologiske verdien. Jeg tror programmering både som en ferdighet isolert sett, men også som et didaktisk verktøy for lærere vil bli mer og mer toneangivende i det moderne matematikkfaget. Jeg støtter opp under Lars sin formulering om at læreres tankesett er avgjørende for hvordan programmering brukes, og at det ikke finnes noen matematiske tema der programmering ikke passer inn i noen grad, selv om tema som geometri og sannsynlighet vil være naturlige innfallsvinkler tidlig i løpet.

Programmering er en ny teknologi i den norske skolen, og for lærere og elever uten erfaring med programmering kan det være vanskelig å se en sammenheng mellom matematikk og programmering. I geometri, sannsynlighet og algebra er det kanskje lettere å se en kobling matematikken, og det gjenspeiles i kompetansemålene fra læreplanen i matematikk (Kunnskapsdepartementet, 2019). Jeg trukket frem viktigheten av å utvikle elever som algoritmiske tenkere. Ferdighetene som er beskrevet i algoritmisk tenkning mener jeg er svært viktig for elevene å mestre, slik at de er forberedt på samfunnsutviklingen som er spådd, og jeg mener at programmering er den best egnede teknologien for å utvikle elevens algoritmiske tenkning. Jeg mener at sammenhengen mellom programmering og matematikk kan derfor sies å være

fagkunnskap (geometri, algebra og sannsynlighet), men også ferdigheter (problemløsning, generalisering og samarbeid).

6 Avslutning

Studien hadde som formål å belyse problemstillingen «hva kjennetegner undervisningstilnærmingen til et utvalg lærere med høy pedagogisk teknologikunnskap i programmering?».

For å belyse denne problemstillingen valgte jeg å undersøke to forskningsspørsmål:

- 1) *Hvordan gir et utvalg lærere uttrykk for sin egen pedagogiske teknologikunnskap?*
- 2) *Hva kjennetegner et utvalg læreres undervisningstilnærming i programmering?*

I dette kapitlet presenteres konklusjonen på problemstillingen, og deretter vurderes kvaliteten på studien. Avslutningsvis vil jeg komme med forslag til videre forskning basert på resultatene av denne studien.

6.1 Konklusjon

Lærerne gav uttrykk for sin egen PTK i hovedsak gjennom deres beskrivelser av programmeringsverdi i matematikkfaget, og deres egen grad av personlig- og profesjonell genesis med ulike programmeringsspråk. Lærerne gav uttrykk for å se en verdi av programmering i matematikk, og beskrev på ulikt vis deres grad av personlig og profesjonell genesis. Jeg tok utgangspunkt i Thomas & Palmers (2014, s. 85) generelle beskrivelse av lærere med høy PTK, og gjorde jeg rede for hvorfor jeg mener at lærerne i utvalget har høy PTK når det gjelder programmering. I analysen presenterte jeg også tre kjennetegn ved lærernes tilnærming til programmeringsundervisning.

Det første kjennetegnet omhandlet lærernes syn på programmering som et didaktisk verktøy med en epistemologisk verdi, altså at programmering har et potensial for å utvikle elevenes forståelse av matematiske konsepter og ideer. Det andre kjennetegnet viste at lærerne hadde en utforskende tilnærming til programmeringsarbeid. Det vil si undervisning som åpnet for metodefrihet og testing, som utviklet elevenes autonomi og hadde få begrensende rammer for elevene. Denne tilnærmingen passet med kjennetegnene som ble trukket frem i den generelle beskrivelsen av lærere med høy PTK (Thomas & Palmer, 2014, s.85). Det siste kjennetegnet viste at lærerne hadde et liberalt syn på lærerrollen, der læreren ikke nødvendigvis oppfyller rollen som «ekspert» i klasserommet når det kommer til programmeringsarbeid, og læreren kan i større grad utvikle sin kompetanse sammen med elevene. Bakgrunnen for at synet ble beskrevet som liberalt var at jeg mener det bryter med et mer tradisjonelt syn på lærerrollen.

Det virker som at lærerne i denne studien har en selvtillit med programmering som gjør at de ikke har problemer med å bruke programmering i matematikk. Selvtilliten kan være et resultat av lang erfaring som matematikklærere, og høy pedagogisk teknologikunnskap. Ut ifra PTK-rammeverket er selvtillit en sentral faktor i læreres PTK (Thomas & Hong, 2013, s. 83), og det vil derfor påvirke hvordan og hvor hyppig en lærer bruker programmering. Hvis det er en målsetning om at norske matematikklærere skal utnytte programmerings epistemologiske verdi, arbeide utforskende uten begrensende rammer og bruke programmering til tross for at de ikke opplever seg selv som eksperter, mener jeg de må benytte seg av profesjonsutvikling for å trigge en form for «kollektiv profesjonell genesis». En kollektiv profesjonell genesis, der det er grunnlag for en faglig diskusjon, og erfarne lærere med høy PTK kan dele sin kunnskap med mindre erfarne

lærere. Det mener jeg vil være det nærmeste man kommer å «erstatte» mange år med erfaring, slik at ferske lærere kan utvikle sin PTK raskere. Videre mener jeg det er sentralt at lærerne utvikler sin profesjonelle genesis, slik at programmering blir et best mulig didaktisk instrument. Dette krever at lærerne har høy grad av instrumentalisering, som kan utvikles gjennom jevnlig kurs der fokuset er på å gi gode eksempler på programmering som et didaktisk instrument i matematikk. Profesjonsutvikling i kollegiet og kurs støttes opp av tidligere studier (Clark-Wilson & Hoyles, 2019; Thomas & Hong, 2013; Thomas & Palmer, 2014). En mer kollektiv kompetanseutvikling vil etter min mening kunne håndtere flere av utfordringene jeg har drøftet i denne studien. Hvis det er ønskelig at undervisningstilnærmingen til lærerne i denne studien skal bli normen, vil det kunne kreve at matematikklærere målbevisst utvikler sin PTK og profesjonelle genesis når det kommer til programmering. Det vil også kunne bli et behov for å «endre» læreres personlige holdninger og verdier, slik at de ser den didaktiske verdien programmering kan ha i deres undervisning. Thomas & Hong (2013, s. 82) mener at et støttende profesjonsfellesskap kan bidra til å utvikle læreres pedagogiske teknologikunnskap når det gjelder denne faktoren.

Programmering mener jeg fortsatt bør anses som noe nytt i skolen, og som med alt nytt vil det møte motstand. Lærerne i denne studien har gitt et innblikk i hvordan de tilnærmer seg programmering. De har lang erfaring, og det er ikke slik at man kan forvente at alle lærere er på deres nivå over natten. En «løsning» som har blitt drøftet flere ganger er profesjonsutvikling. Lærerne i dette utvalget ønsker å bruke programmering fordi de ser en reell verdi i teknologien. De er dyktige og erfarne, og hvis skoler rundt om hadde opprettet faggrupper som hadde som formål å kollektivt utvikle programmeringskompetanse ville kunnskapene, ferdighetene og erfaringen som lærere som Janne, Lars og Simen sitter med kunne komme andre lærere til gode.

6.2 Studiens bidrag til forskningsfeltet

Underveis har studien arbeidet ut ifra ulike formuleringer på problemstilling og tilhørende forskningsspørsmål, men jeg har hele veien arbeidet med samme problemområde: Grunnskolelæreres bruk av programmering i matematikk. Utvalget i denne studien består av tre matematikklærere som arbeider ved ulike skoler i Trondheim. På bakgrunn av utvalgets størrelse og den geografiske begrensningen vil konklusjonene fra denne studien være vanskelig å generalisere. Studien har basert seg på en kvalitativ fenomenologisk tilnærming, der formålet er å beskrive informantenes syn på verden, og derfor mener jeg at det manglende grunnlaget for generalisering ikke bryter med studiens overordne formål. Ettersom studien har undersøkt lærernes tilnærming til programmering spesifikt, er det ikke nødvendigvis slik at undervisningstilnærmingen deres vil være lik i bruk av annen teknologi. Gjennom hele forskningsprosessen har jeg tatt valg knyttet til utvalg, datainnsamling, analyse av data, tolkning av data og presentasjon og drøfting av analysefunn. I lys av dette vil studiens funn måtte sees i lys av mine hypoteser og tanker som ble presentert i delkapittel 3.5.3 Forskerrollen. Jeg mener likevel at studiens konklusjoner kan bidra til forskningsfeltet.

Studien har beskrevet kjennetegn på undervisningstilnærmingen i programmering i matematikk til tre lærere med høy pedagogisk teknologikunnskap. Kjennetegnene kan fungere som en inspirasjon for lærere og lærerstudenter som har lite erfaring med å bruke programmering i matematikk. Jeg har også vist hvordan rammeverket PTK (Thomas & Hong, 2013) kan brukes for å beskrive faktorene som utgjør læreres PTK. Studien belyser hvordan lærernes holdninger og verdier knyttet til en gitt teknologi og

grad personlig og profesjonell genesis er sentrale i deres PTK. Studien har også vist hvordan ideen om dobbel genesis kan brukes for å beskrive hvordan lærere kontinuerlig utvikler et personlig og didaktisk instrument fra det samme artefaktet. Jeg har også belyst hvordan begge genesisprosessene inngår i lærernes PTK, og derfor kan påvirke lærernes evne til å bruke programmering pedagogisk i matematikkundervisning. Skillet mellom prosessene i dobbel genesis understreker at det ikke nødvendigvis er tilstrekkelig for lærere å utelukkende utvikle sin egen programmeringskompetanse for å bruke programmering som en didaktisk støtte i matematikk.

6.3 Forslag til videre forskning

Utforskende undervisning ble trukket frem som et kjennetegn ved lærernes tilnærming til programmeringsarbeid. I forrige kapittel presenterte jeg en hypotese om at bakgrunnen for at lærerne ønsket utforskende undervisning var fordi de mente at programmering ble tilegnet i større grad gjennom oppdagelse fremfor reproduksjon. Elevers læringsutbytte i utforskende programmeringsarbeid sammenlignet med reproduserende programmeringsarbeid kan være interessant å undersøke videre. Utfallet kan fortelle noe om hvordan lærere «bør» tilnærme seg programmeringsarbeid i matematikk.

Gjennom studien har programmering blitt omtalt som et didaktisk instrument med epistemologisk verdi for lærerne i utvalget. En naturlig videreføring vil være å undersøke hvorvidt arbeid med programmering i matematikk gir elevene bedre forståelse av visse tema i matematikk. Forsström og Kaufmann (2019) poengterte at det var lite forskning på programmering i skolen. Flere av studiene de undersøkte viste bedre læringsutbytte, men resultatene var ikke mulig å generalisere. Derfor vil jeg foreslå å undersøke hvorvidt programmering gir elever bedre matematisk forståelse

Med nye læreplaner følger nye læreverk. En av drøftingspunktene fra forrige kapittel var sammenhengen mellom programmering og matematikk, og tidligere i oppgaven argumenterte jeg for at lærerne i denne studien så en sammenheng mellom de to. For videre forskning mener jeg det kan være interessant å undersøke hvordan programmering omhandles i nye læreverk, og hvorvidt læreverkene legger opp til at elever ser en sammenheng mellom å programmere og å gjøre matematikk.

I denne forrige kapittel presenterte jeg kollektiv genesis som en mulig tilnærming for å støtte uerfarne læreres kompetanseutvikling i programmering. Mange lærere i skolen har utviklet sin programmeringskompetanse i hovedsak gjennom organiserte kurs, der lærerne blir opplært av kursholdere. Derfor mener jeg det vil være interessant å undersøke hvilken effekt denne typen kursing har på læreres programmeringskompetanse, sammenlignet med kollektiv profesjonsutvikling mellom lærerne ved hver enkelt skole. Kunnskapen vil kunne bidra med å forbedre kompetanseutviklingen til praktiserende lærere og lærerstudenter.

Litteraturliste:

- Andersen, R., Mørch, A. I., & Litherland, K. T. (2021). Learning Domain Knowledge Using Block-Based Programming: Design-Based Collaborative Learning. I D. Fogli, D. Tetteroo, B. R. Barricelli, S. Borsci, P. Markopoulos, & G. A. Papadopoulos (Red.), *End-User Development* (s. 119–135). Springer International Publishing. https://doi.org/10.1007/978-3-030-79840-6_8
- Artigue, M. (2002). Learning Mathematics in a CAS Environment: The Genesis of a Reflection about Instrumentation and the Dialectics between Technical and Conceptual Work. *International Journal of Computers for Mathematical Learning*, 7(3), 245-274. <https://doi.org/10.1023/A:1022103903080>
- Bjørkeng, K. P. (2020, 10. september). Nesten ingen lærere kan faget de nå må lære bort. *Aftenposten*. <https://www.aftenposten.no/kultur/i/OpLb33/nesten-ingen-av-laererne-kan-faget-de-naa-skal-laere-bort>
- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). *THE NORDIC APPROACH TO INTRODUCING COMPUTATIONAL THINKING AND PROGRAMMING IN COMPULSORY EDUCATION*. CNR Edizioni. <https://doi.org/10.17471/54007>
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- Bray, A., & Tangney, B. (2017). 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.
- Buteau, C., Muller, E., Mgombelo, J., Sacristán, A. I., & Dreise, K. (2020). Instrumental Genesis Stages of Programming for Mathematical Work. *Digital Experiences in Mathematics Education*, 6(3), 367–390. <https://doi.org/10.1007/s40751-020-00060-w>
- Clarke, V., & Braun, V. (2017). Thematic analysis. *The Journal of Positive Psychology*, 12(3), 297–298. <https://doi.org/10.1080/17439760.2016.1262613>
- Clark-Wilson, A., & Hoyles, C. (2019). A research-informed web-based professional development toolkit to support technology-enhanced mathematics teaching at scale. *Educational Studies in Mathematics*, 102(3), 343–359. <https://doi.org/10.1007/s10649-018-9836-1>
- Creswell, J. W. (1998). *Qualitative Inquiry and Research Design: Choosing Among Five Traditions*. Sage Publications Inc. <https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=63251&site=ehost-live>
- FIRST® LEGO®. (u.å.). *What is FIRST® LEGO® League?*. About. Hentet 1. april 2022 fra <https://www.firstlegoleague.org/about#>
- Forsström, S. E., & Kaufmann, O. T. (2019). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18-32. <http://ijlter.org/index.php/ijlter/article/view/1297>
- 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>
- 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.
<https://doi.org/10.1007/s10758-005-0395-z>
- Haspekian, M. (2011). THE CO-CONSTRUCTION OF A MATHEMATICAL AND A DIDACTICAL INSTRUMENT. I Pytlak, M., Rowland, T., & Swoboda (Red.), *Seventh Congress of the European Society for Research in Mathematics Education* (s. 2298–2307). <https://hal.archives-ouvertes.fr/hal-01273866>
- 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: An International Perspective on Technology Focused Professional Development* (s. 241–275). Springer Netherlands.
https://doi.org/10.1007/978-94-007-4638-1_11
- Hill, H. C., & Ball, D. L. (2004). Learning Mathematics for Teaching: Results from California's Mathematics Professional Development Institutes. *Journal for Research in Mathematics Education*, 35(5), 330–351.
<https://doi.org/10.2307/30034819>
- Hjernekraft. (u.å). *Fagfornyelsen – nye læreplaner 2020*. FIRST® LEGO® League Scandinavia.
<https://hjernekraft.org/media/dokumenter/fll/fagfornyelsen/kompetansemaal-fll-2021.pdf>
- Johansen, A. K. (2020, 11. juli). Programmering vil bli en utfordring for lærere. *Forskning.no*. <https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold-matematikk/programmering-vil-bli-en-utfordring-for-laerere/1711838>
- Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 52(7), 1029–1048. <https://doi.org/10.1080/0020739X.2020.1736349>
- Kilhamn, C., Bråting, K., & Rolandsson, L. (2021). *Teachers' arguments for including programming in mathematics education*. (s. 169-176).
- Kivunja, C., & Kuyini, A. B. (2017). Understanding and Applying Research Paradigms in Educational Contexts. *International Journal of Higher Education*, 6(5), 26-41.
<https://doi.org/10.5430/ijhe.v6n5p26>
- Kramarski, B., & Michalsky, T. (2015). Effect of a TPCK-SRL Model on Teachers' Pedagogical Beliefs, Self-Efficacy, and Technology-Based Lesson Design. I C. Angeli & N. Valanides (Red.), *Technological Pedagogical Content Knowledge: Exploring, Developing, and Assessing TPCK* (s. 89–112). Springer US.
https://doi.org/10.1007/978-1-4899-8080-9_5
- NOU 2015:8. (2015). *Fremtidens skole – Fornyelse av fag og kompetanser*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001/>
- Kunnskapsdepartementet. (2018, 26. juni). *Fornyer innholdet i skolen* [Pressemelding]. <https://www.regjeringen.no/no/dokumentarkiv/regjeringen-solberg/aktuelt-regjeringen-solberg/kd/pressemeldinger/2018/fornyer-innholdet-i-skolen/id2606028/>
- Kunnskapsdepartementet. (2019). *Læreplan i matematikk (MAT01-05)*. Fastsett som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020.
- Lin, Y., & Weintrop, D. (2021). The landscape of Block-based programming: Characteristics of block-based environments and how they support the transition

- to text-based programming. *Journal of Computer Languages*, 67, 1–18.
<https://doi.org/10.1016/j.cola.2021.101075>
- Lær Kidsa Koding. (u.å.). *Om Lær Kidsa Koding*. Lær Kidsa Koding! Hentet 6. april 2022 fra <https://www.kidsakoder.no/om-lkk/>
- Mackenzie, & Knipe, S. (2006). Research dilemmas: paradigms, methods and methodology. *Issues in Educational Research*, 16(2), 193–205.
- Meld. St. 28. (2015-2016). *Fag – Fordypning – Forståelse—En fornyelse av Kunnskapsløftet*. Kunnskapsdepartementet.
<https://www.regjeringen.no/contentassets/e8e1f41732ca4a64b003fca213ae663b/no/pdfs/stm201520160028000dddpdfs.pdf>
- Microsoft. (u.å.). *Makecode—Micro:Bit*. Micro:Bit. Hentet 31. mars 2022 fra <https://makecode.microbit.org/>
- Misfeldt, M., & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. I K. Krainer & N. Vondrová (Red.), *CERME 9—Ninth Congress of the European Society for Research in Mathematics Education* (s. 2524–2530). Charles University in Prague, Faculty of Education and ERME. <https://hal.archives-ouvertes.fr/hal-01289367>
- Misfeldt, M., Szabo, A., & Helenius, O. (2019). Surveying teachers' conception of programming as a mathematics topic following the implementation of a new mathematics curriculum. I U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Red.), *Eleventh Congress of the European Society for Research in Mathematics Education* (Bd. TWG15, Nummer 13). Freudenthal Group.
<https://hal.archives-ouvertes.fr/hal-02417074>
- MIT Media Lab. (u.å.). *Scratch*. Scratch. Hentet 31. mars 2022 fra <https://scratch.mit.edu>
- Moreau, H. N. (2021, 30. desember). Skal lære elevene koding, men forstår det ikke selv. NRK. <https://www.nrk.no/innlandet/laerere-trenger-hjelp-til-a-knekke-koden-pa-koding-1.15781343>
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas* (2.utg). BasicBooks.
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365–376.
<https://doi.org/10.1016/j.compedu.2018.10.005>
- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G., & Voll, L. O. (2016). *Teknologi og programmering for alle*, 1-91.
- Schleicher, A. (u.å.). *The case for 21st-century learning—OECD*. The case for 21st-century learning. <https://www.oecd.org/general/thecasefor21st-centurylearning.htm>
- Schoenfeld, A. H. (2011). *How we think: A theory of goal-oriented decision making and its educational applications* (s. XVII, 245). Routledge
- Shulman, L. (1987). Knowledge and Teaching: Foundations of the New Reform. *Harvard Educational Review*, 57(1), 1–23.
<https://doi.org/10.17763/haer.57.1.j463w79r56455411>
- Stigberg, H., & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures in Education*, 18(4), 483–496. <https://doi.org/10.1177/1478210319894785>
- Terry, G. Hayfield, N. Clarke, V. & Braun, V. (2017). *Thematic analysis*. I C. Willig & W. Rogers (Red.), *The SAGE Handbook of Qualitative Research in Psychology* (s. 17-37). SAGE Publications. <https://doi.org/10.4135/9781526405555>

- Thomas, M. O. J., & Hong, Y. Y. (2005). Teacher factors in integration of graphic calculators into mathematics learning. *Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education*, 4, 257–264.
- Thomas, M. O. J., & Hong, Y. Y. (2013). Teacher Integration of Technology into Mathematics Learning. *International Journal for Technology in Mathematics Education*, 20(2), 69–84.
- Thomas, M. O. J., & Palmer, J. M. (2014). Teaching with Digital Technology: Obstacles and Opportunities. I A. Clark-Wilson, O. Robutti, & N. Sinclair (Red.), *The Mathematics Teacher in the Digital Era: An International Perspective on Technology Focused Professional Development* (s. 71–89). Springer Netherlands. https://doi.org/10.1007/978-94-007-4638-1_4
- Tjora, A. (2018). *Viten skapt: Kvalitativ analyse og teoriutvikling* (1. utgave). Cappelen Damm akademisk.
- Trouche, L. (2020a). Instrumentalization in Mathematics Education. I S. Lerman (Red.), *Encyclopedia of Mathematics Education* (s. 392–403). Springer International Publishing. https://doi.org/10.1007/978-3-030-15789-0_100013
- Trouche, L. (2020b). Instrumentation in Mathematics Education. I S. Lerman (Red.), *Encyclopedia of Mathematics Education* (s. 404–412). Springer International Publishing. https://doi.org/10.1007/978-3-030-15789-0_80
- Utdanningsdirektoratet. (2019, 27. mars). *Algoritmisk tenkning*. Algoritmisk tenkning. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2021a, 22. september). *Slik ble læreplanene utviklet*. <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/slik-ble-lareplanene-utviklet/>
- Utdanningsdirektoratet. (2021b, 4. oktober). *Evaluering av fagfornyelsen – hva, hvorfor og hvordan*. <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/evaluering-av-fagfornyelsen/fagfornyelsen-hva-skal-evalueres/>
- 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. <https://doi.org/10.1007/BF03172796>
- Vinnervik, P. (2020). Implementing programming in school mathematics and technology: Teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*. <https://doi.org/10.1007/s10798-020-09602-0>
- Üçgül, M. (2013). History and Educational Potential of LEGO Mindstorms NXT. *Mersin University Journal of the Faculty of Education*, 9(2), 127–137.

Vedlegg

Vedlegg 1: Intervjuguide

Vedlegg 2: Samtykkeskjema

Vedlegg 3: Godkjenning for behandling av personopplysninger fra NSD

Vedlegg 1: Intervjuguide

Intervjuguide - Masterprosjekt

Generell info

- Hvor lenge har du undervist i matematikk?
- Hvilke programmeringsverktøy har du brukt i matematikkundervisning?
- Hvor hyppig vil du selv si du bruker programmering?

Holdninger

- Hva er din motivasjon for å bruke programmering i matematikkundervisning?
- Hva tenker du om at programmering ble innført gjennom allerede etablerte fag fremfor å opprette et eget programmerings- eller teknologifag/IKT-fag?

Kan du gi noen eksempler på hvorfor det hører hjemme i matematikken?

Kan du gi noen eksempler på hvorfor du ville foretrukket et eget fag?

- Har programmering etter din mening beriket undervisning matematikk?

Kan du gi eksempler på dette?

P-IG

- Hadde du noen erfaring med programmering i forkant av fagfornyelsen?
- Hva var ditt første møte med programmering?
- Hvordan går du frem for å utvikle din egen programmeringskunnskap?
- I et tenk scenario, der du skulle bruke et programmeringsverktøy du ikke hadde noen kjennskap til i undervisningen din. Kan du beskrive hvordan du ville forberedt deg til deg økten?

Hvorfor ville du forberedt deg slik?

- Hvis en lærer skal bruke et programmeringsverktøy som et undervisningsverktøy, hvilke kunnskaper må h*n ha?

- Hva kjennetegner en lærer som mestrer å bruke programmering som et undervisningsverktøy i matematikkundervisning?

- Har ditt syn på programmering som en del av matematikk endret seg fra fagfornyelsen ble innført og til nå?

E-IG

- Ta stilling til påstanden «elever får en bedre forståelse av matematikk ved å bruke programmering?»

Hvorfor mener du dette stemmer/ikke stemmer?

- Kan du gi noen eksempler på typiske elevutfordringer i møtet med programmering i din matematikkundervisning?
- Hvordan har elevene tatt i bruk programmering i din undervisning?
- Hva kjennetegner elever som etter din mening «mestrer» programmering i din matematikkundervisning?

Programmering i matematikkundervisning

- Kan du gi et eksempel på en god programmeringsaktivitet du har brukt i matematikkundervisning?
Hvorfor var dette en god aktivitet?
- På hvilken måte har programmering som en del av skolematematikken et potensial for læring hos elevene?
- Er det viktig at elevene får erfaring med programmering i matematikk?
- Ser du et potensial for å lære seg noe nytt når du arbeider med programmering?
- Hvilke temaer og arbeidsmåter i programmering er også viktig i matematikk?

Vedlegg 2: Samtykkeskjema

Masterprosjekt: Programmering i skolen

Dette er et spørsmål til deg om å delta i forskningsprosjektet «programmering i skolen». I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Jeg er masterstudent ved instituttet for lærerutdanning på NTNU. Våren 2022 skal jeg skrive en masteroppgave i matematikkdiridaktikk, og jeg skal i forbindelse med dette intervjuere lærere i norsk skole. Formålet med prosjektet er beskrive programmering som en del av matematikkundervisning ut ifra læreres perspektiv. Opplysningene som samles inn, skal ikke brukes til noe annet formål enn mitt forskningsprosjekt.

Hvem er ansvarlig for forskningsprosjektet?

Øistein Gjøvik (Universitetslektor ved instituttet for lærerutdanning, NTNU) er ansvarlig for prosjektet gjennom sin rolle som veileder for masterprosjektet.

Hvorfor får du spørsmål om å delta?

Forskningsprosjektet har som formål å undersøke læreres bruk av programmering i matematikkundervisning. Derfor ønsker jeg bruke lærere som underviser i matematikk på ungdomstrinnet. Utvalget vil bestå av lærere som underviser ved skoler i Trondheim. Det er planlagt at utvalget vil bestå av fire lærere.

Hva innebærer det for deg å delta?

Lærere som deltar i forskningsprosjektet, vil bli intervjuet av meg. Intervjuet vil handle om deltakernes erfaringer med bruk av programmering i matematikkundervisning, læreres erfaringer med programmering og samtale rundt eventuelle utfordringer med bruk av programmering i skolen. Intervjuene vil bli tatt lydopptak av, slik at samtalen kan transkriberes og analyseres. Intervjuet vil følge en intervjuguide som er formulert i forkant av intervjuet av meg. Varigheten av intervjuet vil bestemmes av deltakerens svar, eller deltakerens tilgjengelige tid. Informasjon om deltakere, og innholdet av intervjuet vil bli anonymisert, og kun behandles av meg og min veileder.

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.

Deltakerne i studien vil få mulighet til å lese mine betraktninger etter endt analyse, og får mulighet til å komme med innspill eller presiseringer rundt deres bidrag til prosjektet.

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. Dine opplysninger vil kun behandles av meg og min veileder. I oppgaven vil deltakernes bidrag anonymiseres gjennom en kode (i form av tilfeldig valgte fornavn). Deltakernes kontaktopplysninger og tilsvarende kode vil lagres adskilt fra datamaterialet fra intervjuet. Deltakerne i studien vil ikke kunne gjenkjennes i masteroppgaven.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene vil være anonyme i prosjektet, og etter prosjektet er avsluttet, vil deltakernes personopplysninger, lydopptak fra intervju og eventuelle planleggingsdokument slettes. Prosjektet er etter planen ferdig 25. mai.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

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

Dine rettigheter

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

innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene

å få rettet opplysninger om deg som er feil eller misvisende

å få slettet personopplysninger om deg

å sende klage til Datatilsynet om behandlingen av dine personopplysninger

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

Masterstudent

Erik Atle Løken

Tlf: 40468423

E-post: erikalok@stud.ntnu.no

Veileder

Øistein Gjovik

Tlf: 73559416

E-post: oistein.gjovik@ntnu.no

Personvernombud – NTNU

Thomas Helgesen

Tlf: 93079038

E-post: 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: 53 21 15 00.

Med vennlig hilsen

Øistein Gjøvik
(Forsker/veileder)

Erik Atle Løken
(Masterstudent)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet [sett inn tittel], og har fått anledning til å stille spørsmål. Jeg samtykker til:

å delta på intervju

å presentere en undervisningsøkt med programmering (skriftlig/muntlig)

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

Vedlegg 3: Godkjenning for behandling av personopplysninger fra NSD

Behandlingen av personopplysninger er vurdert av NSD. Vurderingen er:

Det er vår vurdering at behandlingen vil være i samsvar med personvernlovgivningen, så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet 06.01.2022 med vedlegg, samt i meldingsdialogen mellom innmelder og Personverntjenester. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 25.05.2022.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake.

Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER

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

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles 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

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).

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

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

FØLG DIN INSTITUSJONS RETNINGSLINJER

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

Nettskjema og Microsoft er databehandler i prosjektet. NSD legger til grunn at behandlingen oppfyller kravene til bruk av databehandler, jf. art 28 og 29.

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: nsd.no/personverntjenester/fulle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema

Du må vente på svar fra NSD før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

