



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2021 30 stp
REALTEK

Hva kjennetegner matematikklæreres holdninger rundt undervisning av programmering i teoretisk matematikk?

What Characterizes the Attitudes of Mathematics
Teachers Regarding the Teaching of Programming
in Theoretical Mathematics?

Hana Zukanovic
Lektorutdanning i realfag

Innholdsfortegnelse

FORORD	5
SAMMENDRAG	7
ABSTRACT	8
INNLEDNING	9
1.1 FORSKNINGSSPØRSMÅL	10
1.2 FAGFORNYELSEN OG PROGRAMMERING	10
1.3 AKTUALISERING	11
2. TEORI	12
2.1 LÆRERES HOLDNINGER	12
2.2 PROGRAMMERING I SKOLEN	14
2.3 PROGRAMMERING I MATEMATIKKFAGET	15
2.4 PROGRAMMERING OG UNDERVISNING	18
3. METODE	23
3.1 FORSKNINGSDESIGN OG METODE	23
3.2 UTVALG	24
3.3 SEMISTRUKTURERTE INTERVJUER	26
3.4 ANALYSEPROSESSEN	28
3.5 STUDIENS KVALITET	32
3.6 FORSKNINGSETIKK	34
4. RESULTATER	35
4.1 AFFEKTIVE RESPONSER	35
4.2 ATFERDSMESSIGE RESPONSER	38
4.3 KOGNITIVE RESPONSER	40
5. DISKUSJON	43
5.1 HVILKE FØLELSER UTTRYKKER INFORMANTENE KNYTTET TIL UNDERVISNING AV PROGRAMMERING I 1T?	44
5.2 HVORDAN UNDERVISER LÆRERNE PROGRAMMERING I 1T?	49
5.3 HVA TENKER INFORMANTENE RUNDT UNDERVISNING AV PROGRAMMERING I 1T?	51
5.4 LÆRERNES UTTRYKTE HOLDNINGER RUNDT UNDERVISNING AV PROGRAMMERING I 1T	56
6. AVSLUTTENDE REFLEKSJONER OG VIDERE FORSKNING	57
REFERANSER	59

VEDLEGG	62
---------	----

VEDLEGG 1: INFORMASJONSSKRIV	62
VEDLEGG 2: INTERVJUGUIDE	66
VEDLEGG 3: GODKJENNELSE FRA NSD TIL BEHANDLING AV PERSONOPPLYSNINGER	70

Forord

Det er flere jeg ønsker å takke i forbindelse med arbeidet.

Marte Bråtalien, tusen hjertelig takk for all hjelp! Jeg føler meg heldig som fikk en så grundig og dyktig hovedveileder, og jeg gir mine sterke anbefalinger videre. Morten Munthe var min biveileder, og jeg takker for alle de flotte innspillene om programmering og matematikk. I tillegg vil jeg takke for gode råd og tips omkring forskning på temaet.

Jeg vil også takke informantene for å frivillig tatt seg tid til å bli intervjuet, til tross for det hektiske skoleåret. Denne studien hadde ikke vært mulig å gjennomføre uten deres bidrag.

Lykke til videre, alle sammen!

Oslo, våren 2021



Hana Zukanovic

Sammendrag

Programmering implementeres i skrivende stund inn i teoretisk matematikk på videregående skoler og lærerne skal trene elever i både algoritmisk tenkning og problemløsning blant annet gjennom programmering. Det har florert i nyhetsbildet det siste året om at lærere mangler både kompetanse, erfaringer og tid til å implementere programmering inn i undervisningen, og rundt omkring i verden forskes det på programmering i matematikkfaget. Lærernes holdninger er viktige når implementeringer iverksettes, men likevel finnes det få studier omkring læreres holdninger til programmering i matematikk, og enda mindre om holdningene til undervisning av programmeringen. I dette studiet forsøkes det å svare på følgende forskningsspørsmål: «Hva kjennetegner matematikklæreres holdninger rundt undervisning av programmering i teoretisk matematikk?»

Kvalitativ case-metode ble anvendt som forskningsdesign, i et forsøk på å oppnå en dypere forståelse av holdningene til lærere, med utgangspunkt i trekomponentmodellen (Raaheim, 2019). Det ble gjennomført semistrukturerte intervjuer med fire matematikklærere i 1T, som datainnsamlingsmetode. I tillegg ble rammeverket til Anker (2020) brukt som analyseprosess, med abduktiv analyse i fokus. Analyseprosessen bestod av fire analysefaser, og ble brukt som et verktøy i et forsøk på å belyse forskningsspørsmålet.

Analysen viser at informantenes holdninger kan tendere mot en ambivalent holdning til å undervise programmering i 1T. Det ser ut til at informantene ser nytteverdien av å undervise programmering i 1T, men deres holdninger preges også av flere utfordringer. Utfordringene kan knyttes til faktorer som manglende kompetanse blant både lærere og elever, ny eksamensform i 1T og tekniske problemer i timen, og informantene gir uttrykk for både usikkerhet og frustrasjon. Det ser ut til at det er et gap mellom informantenes visjoner og atferd, og en mulig konsekvens er at informantene ikke har nok ressurser til å i best mulig grad gi elevene mulighet til å trene seg i algoritmisk tenkning og problemløsning. Derimot kan det virke som at informantene er håpefulle for fremtiden, hvor de ser for seg at elevers forhåndskompetanse vil øke med tiden.

Abstract

Programming is at this very moment implemented into theoretical mathematics in high schools, and, as such, the teachers are expected to teach the students both algorithmic thinking and problem-solving. During the past year, news about teachers lacking both competence, experience, and time to implement programming into the mathematics subject has flourished in great numbers, and all around the world there is research being done on programming when it comes to the subject of mathematics. The teacher's attitudes is important when implementing programming in school subjects, yet there is a lack of studies around the topic, and even less research exist around teacher's attitudes towards teaching programming in the subject of mathematics. This study attempts to answer the following research question: "What characterizes mathematics teachers attitudes when it comes to teaching programming in theoretical mathematics?"

Qualitative case method was used as the research design, with the three-component model, as described by Raaheim (2019), to acquire a better understanding regarding the attitudes of the participants in this study. Four mathematic teachers were put through semi-structured interviews in order to collect data. In addition, Anker's (2020) four step analysis process was utilized during the analytic process, and abductive analysis has been performed as one of the main steps in the process.

The findings show that the participants of the study lean towards ambivalent attitudes around teaching programming in theoretical mathematics in high school. It seems like the participants see the usefulness of teaching mathematics through programming, but the challenges stem from a lack of competence from both teachers and students, challenges evolving a new type of exam in theoretical mathematics this year, as well as technical problems during class, and the participants seem to feel both unsecure and frustrated about the situation. There seems to be a rift between the participants visions and their behavior, making for a likely consequence in that the participants do not have enough time to train the students in algorithmic thinking and problem-solving. On the other hand, it seems like the participants are hopeful for the students' competence will increase in the future.

Innledning

Dersom du leser denne oppgaven grunnet din interesse for programmering i skolen, har du kanskje hørt om, eller selv testet ut, en robot kalt Lego Mindstorms. Roboten er oppkalt etter boken *Mindstorms* (MIT media lab, 2016), av Seymour Papert, professoren som kan sees på som en av de første forkjemperne for å implementere programmering inn i matematikkfaget på skolen (Misfeldt & Ejsing-Duun, 2016). Dette var starten på en av mange bølger innenfor programmering i skolen som har kommet og gått (Bostrøm, Bø, Langmyhr, & Rydland, 2008), men høsten 2020 ble programmering obligatorisk i skolen i Norge gjennom fagfornyelsen og særlig er det matematikkfaget som har fått hovedansvaret om opplæringen av programmering (Tellefsen, 2021). Algoritmisk tenking og problemløsning er to store temaer som man kan finne i de nye læreplanene i matematikk 1T (Utdanningsdirektoratet, 2020a), og programmering kan bidra til at elever kan trene seg i problemløsning (Sevik, 2016) og potensielt algoritmisk tenkning (Stephens & Kadjevich, 2020). Det er lærerne som til syvende og sist får ansvaret om å implementere programmering inn i matematikken, gjennom sin undervisning. Læreres holdninger til matematikk (og derav programmering) kan påvirke matematikkundervisningen, ved at ulike syn på programmering kan gi ulike utfall i klasserommet (Ernst, 1989). I tillegg kan lærerens holdninger til matematikk også påvirke elevers holdninger til matematikk (Goodykoontz, 2008). Videre beskrives det at elevers læring av matematikk kan bli påvirket av elevenes holdninger til matematikk, og dette kan ha en overføringsverdi når vi snakker om å lære matematikk gjennom programmering. Til sammen peker dette mot at lærerens holdninger til det å undervise programmering i matematikk, kan ha stor påvirkning på elevenes læring.

I Sverige startet implementeringen av programmering i matematikk for grunnskolen allerede høsten 2018, og en kvalitativ studie har funnet flere bekymringsverdige faktorer: Lærere er usikre på faginnholdet, det kan være ulik faglig utviklingsmulighet for lærerne, det rapporteres om mangel på læremateriell og tilbakevendende problemer med skolens IT-infrastruktur (Vinnervik, 2020). Denne typen usikkerhet og utfordringer kan ha påvirkning på lærerens holdninger til å undervise programmering i matematikk, som igjen kan påvirke deres faktiske undervisning.

1.1 Forskningsspørsmål

Det finnes foreløpig lite forskning på holdninger blant norske lærere i henhold til programmering i matematikk, og enda mindre omkring holdninger til undervisning av programmering i matematikk. En måte å beskrive holdninger på er at det er en evaluerende tendens mot en enhet (Zeigler-Hill & Shackelford, 2020). Enheten i mitt tilfelle vil være *programmering* i IT, og holdningene dreier seg om undervisning av enheten. Jeg har dermed valgt å fordype meg i følgende forskningsspørsmål;

Hva kjennetegner matematikklæreres holdninger rundt undervisning av programmering i teoretisk matematikk (1T)?

En modell for å forsøke å måle læreres holdninger på er å undersøke hva lærerne føler, gjør og tenker omkring å undervise programmering i 1T, basert på trekomponentmodellen til Raaheim (2019) som vil utdypes i kapittel 2.1. På bakgrunn av trekomponentmodellen valgte jeg også å støtte meg på tre følgende underspørsmål i mitt forsøk på å nærme meg et svar på forskningsspørsmålet;

1. Hvilke følelser uttrykker lærere knyttet til undervisning av programmering i 1T?
2. Hvordan underviser lærerne programmering i 1T?
3. Hva tenker lærerne rundt undervisning av programmering i 1T?

De tre underspørsmålene vil forsøkes å besvares i diskusjonsdelen, under kapittel 5.

1.2 Fagfornyelsen og programmering

I Norge har det skoleåret 2020/2021 kommet en gradvis implementering av nye læreplaner i alle fag på alle trinn, kalt fagfornyelsen. I denne prosessen er målet at i løpet av 2023 skal alle trinn ha begynt å bruke de nye læreplanene (Utdanningsdirektoratet, 2020b). Fagfornyelsen bygger på Stortingsmelding nr. 28 og programmering skal inn blant annet i matematikk, naturfag, samfunnsfag og kunst- og håndverk. Argumentene for å gjeninnføre programmering i skolen igjen er mange, blant annet å tilpasse elevene til det fremtidige jobbmarkedet ved å sørge for at de ikke bare kan bruke teknologi, men også forstå hele prosessen bak ulike teknologier (Kafai & Burke, 2013). Kunnskapsdepartementet (2014) har satt søkelys på noen

nøkkelferdigheter elever burde få muligheten til å kunne oppnå, som kreativitet, innovasjon, kritisk tenkning, metakognisjon, kommunikasjon, samarbeid, digital kompetanse, medborgerskap, karriere og arbeidsliv, og det er ønskelig at programmering skal bidra til å øve slike nøkkelferdigheter. I tillegg vil man få muligheten til å øve elever i *computational thinking* (Kafai & Burke, 2013), som vil si at man tar i bruk blant annet abstrahering (å redusere enkeltfenomener under ett begrep), ideer, nedbrytning, heuristiske resonnementer (løse problemer ved å ta i bruk rimelige og sannsynlige løsninger), og man må kunne jobbe med store datasett (Wing, 2006). Sevik (2016) mener at det kan styrke metakognisjon (lære seg å lære) hos elever, ved at de blant annet må prøve og feile, og underveis evaluere produktet sitt for å enten forbedre det, eller videreutvikle det. I tillegg hevdes det å kunne gi elevene gode verktøy til å bli bedre med problemløsning. Jeg vil under delkapittel 2.2.1 utdype begrepene *problemløsning* og *algoritmisk tenkning*, som jeg har valgt ut som hovedpunktene under den nye læreplanen i teoretisk matematikk.

1.3 Aktualisering

Det er lærerne som til syvende og sist får jobben med å videreformidle nye læreplaner i fagfornyelsen til elevene, og det stoles på at de gjør en profesjonell jobb med å tolke læreplanen og velge ut lærerike aktiviteter (Vinnervik, 2020). Er det slik at lærerne tolker læreplanen med samme intensjoner som skaperne av læreplanene, og har lærerne tilstrekkelig med kunnskap om programmering i matematikk til å kunne utføre slike tolkninger, i tillegg til å designe og evaluere aktiviteter i klasserommet? Det hevdes at det er begrensede kunnskaper om hvordan lærerne både tolker og implementerer læreplanene, og hvordan læreplanene påvirker undervisningen (Vinnervik, 2020). Dersom man ønsker suksess med nye læreplaner, er det viktig å sette seg inn i læreres erfaringer og responser til læreplanene og forsøke å forstå de (Hargreaves, 2005). Store endringer i læreplanene tar tid, og det er mange lærere som mangler programmeringskompetanse (Humble, Mozelius, & Sällvin, 2020). Å lære seg programmering er en tidkrevende prosess. Ifølge Humble, Mozelius og Sällvin (2020) er det utfordringer knyttet til prosessen med implementeringen, slik som at lærere med lite forkunnskaper om programmering skal lære seg programmering, og hvordan man skal bruke programmering i matematikk.

Vi står ovenfor en endring i læreplanen til blant annet teoretisk matematikk på videregående, og anbefalingene til hvordan man kan undervise matematikk gjennom programmering kan kanskje for noen oppleves som annerledes enn den konvensjonelle måten å undervise matematikk på. Endring er en kompleks prosess bestående av mange ulike komponenter, og en av komponentene er lærerens holdninger (Kennedy & Kennedy, 1996).

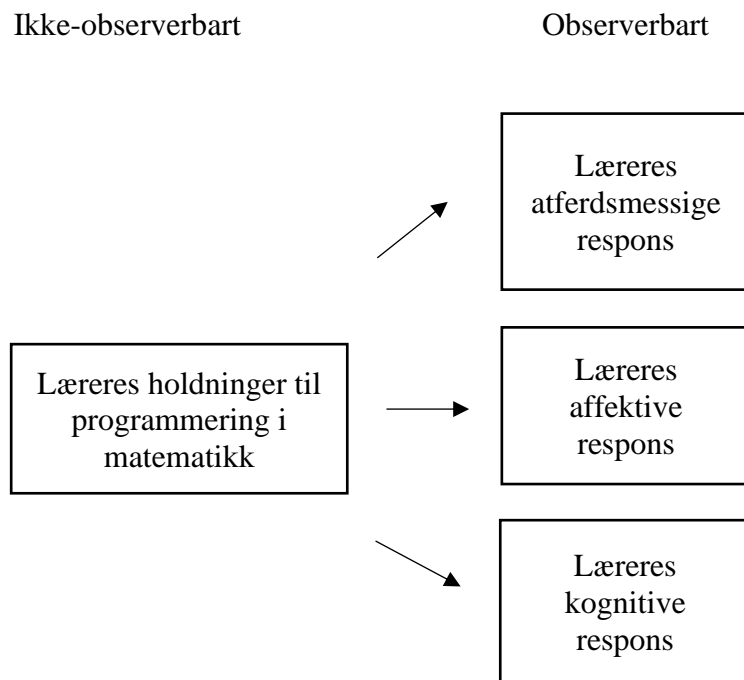
2. Teori

2.1 Læreres holdninger

En holdning kan beskrives som en evaluerende tendens mot en enhet (Zeigler-Hill & Shackelford, 2020). En enhet er noe som har en selvstendig eksistens, som for eksempel en hendelse, en ny læreplan, eller en del av den – slik som programmering. Videre deles tendensene opp i hva en tror om enheten, hva slags følelser som assosieres med enheten og til slutt tanker om tidligere atferd knyttet til enheten (Zeigler-Hill & Shackelford, 2020), som for mitt forskningsspørsmål blir hva lærerne tenker om å undervise programmering i matematikk 1T, hvilke følelser de assosierer med å undervise programmering i 1T, og hva forteller de meg om tidligere erfaringer med å undervise programmering i 1T. Produktet av tendensene vil samlet forsøke å gi et bilde av holdningene til et menneske. Dette kan være utfordrende, ettersom tanker, følelser og atferd kan være motstående, og det kan gi en ambivalent holdning (Zeigler-Hill & Shackelford, 2020), til for eksempel å undervise programmering i matematikk 1T. En ambivalent holdning kan føre til at man tar ubesluttsomme handlinger (Zeigler-Hill & Shackelford, 2020). I tillegg kan vi mennesker (og derav lærere også) ha flere holdninger til en enhet (Wilson, 2000), noe som kompliserer det ytterligere. Fortidens holdninger kan tas med seg videre inn i fremtidens oppdaterte holdninger. Vi kan til og med konstruere nye holdninger mens vi reflekterer rundt nåtidens holdninger (Zeigler-Hill & Shackelford, 2020).

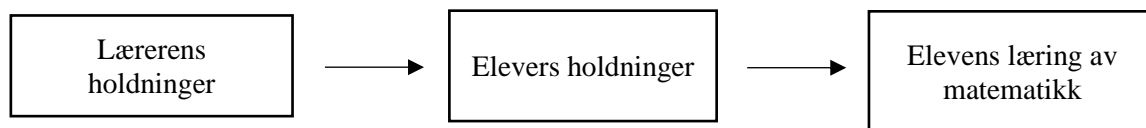
Holdninger er ikke noe vi kan observere direkte, fordi det er i menneskets indre, men vi må høre på det personene sier og se på det de gjør (Raaheim, 2019). En mulig modell for å måle holdninger er ifølge Raaheim (2019) trekomponentmodellen, og i figur 2-1 er det illustrert et eksempel hvor enheten som har blitt brukt er programmering i matematikk. Modellen viser at

de observerbare ytre faktorene ved menneskers holdninger kan deles opp i personens kognitive- affektive- og atferdsmessige respons. Her kan det observeres likheter med Zeigler-Hill & Shackelfords (2020) skildringer av holdninger, hvor kognitiv respons vil være hva en tror om enheten, affektiv respons vil være hva en føler om enheten, og atferdsmessige respons vil kunne være hva du forteller at du har gjort tidligere knyttet opp mot enheten.



Figur 2-1: Trekomponentmodellen for lærerens holdninger, inspirert av Raaheim (2019)

Det hevdes at den amerikanske talk-show vertinnen Oprah Winfrey en gang skal ha sagt at *tidenes største oppdagelse er at en person kan forandre sin fremtid bare ved å endre sin holdning*. Om hun faktisk har uttalt seg slik og om dette er vitenskapelig bevist er usikkert, men det vi vet er at elevers læring av matematikk kan påvirkes av elevers holdninger til matematikk (Goodykoontz, 2008). Det kan virke som et dynamisk, men komplekst samspill mellom holdninger og oppnåelse. Holdningen kan påvirke hva man mestrer, og mestring kan påvirke holdninger (Goodykoontz, 2008). Videre er lærerens holdninger til matematikk en av faktorene som kan påvirke elevers holdninger til matematikk, vist i figur 2-1. Jeg velger å tenke at dette kan være overførbart også når vi snakker om å lære matematikk gjennom programmering, og dermed blir det viktig å utforske lærerens holdninger til å lære matematikk gjennom programmering.



Figur 2-2: Inspirert av Goodykoontz (2008), hvor pilene viser påvirkninger

Læreres holdninger er også viktige i henhold til endringer i skolen (Kennedy & Kennedy, 1996). Vi står ovenfor flere endringer skoleåret 2020/2021, blant annet en endring i læreplanen til teoretisk matematikk ved at programmering skal implementeres. Lærerne spiller en viktig rolle i om implementeringen av nye læreplaner kommer til å lykkes eller ikke (Wang & Cheng, 2009). Implementeringen av læreplaner er ingen enkel sak, og det er ikke sikkert at skaperne av læreplanene har klart å forutse hvordan lærerne kommer til å implementere læreplanen (Wang & Cheng, 2009). I tillegg er det fare for at det motsatte kan skje; lærerne forstår ikke hvordan skaperne ønsker at læreplanene skal settes ut i liv i klasserommene.

Oppsummert har jeg presentert to viktige grunner til å undersøke læreres holdninger; de kan ha en innvirkning på elevers læring av matematikk og i tillegg kan holdningene påvirke om implementeringen av læreplanen i IT kommer til å lykkes. I neste delkapittel skal jeg dykke dypere inn i en av de store endringene i fagfornyelsen; programmering.

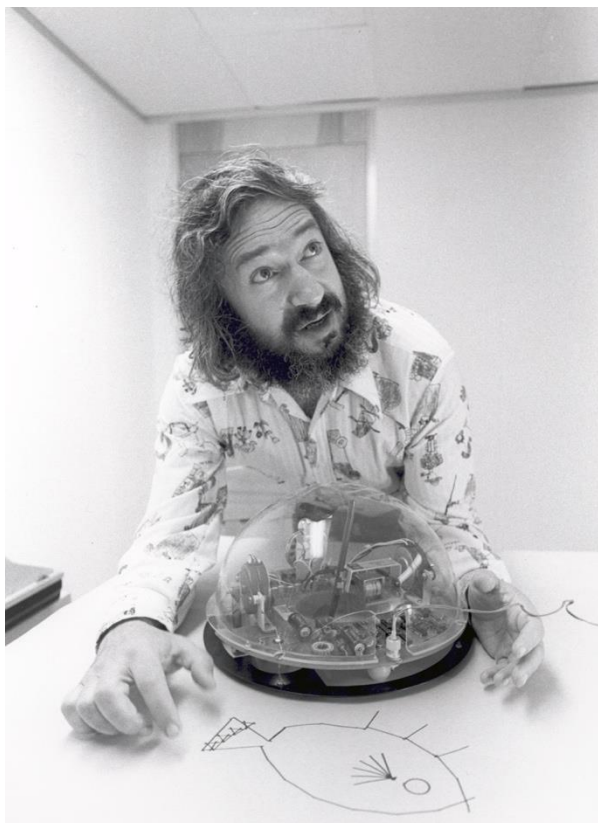
2.2 Programmering i skolen

Programmeringens implementering i skolen gjennom fagfornyelsen er en endring i skolen, men det er ikke første gang at det blir gjort forsøk på å introdusere programmering inn i den norske skolen. Dette kan faktisk spores så langt tilbake som til 1968, da Herman Ruge var den første til å holde en undervisning i elektronisk databehandling (EDB) ved Forsøksgymnaset (Bostrøm, Bø, Langmyhr, & Rydland, 2008). I 1981 hadde en artikkel i Norsk Skoleblad overskriften «Nå haster det med EDB i grunnskolen», hvor det ble ytret at dersom det ventes bare et år til med å innføre datalære, vil Norge ligge langt bak andre land i den teknologiske verden. Programmering skulle inn i valgfaget EDB, og argumentene for å fremme EDB i skolen var å forberede elever til et samfunn hvor EDB har mer og mer innflytelse, innsikt i muligheter og

begrensninger av EDB, samt å kunne diskutere hvordan EBD kunne og burde brukes. Dette kan minne om dagens argumenter for å gjeninnføre programmering igjen i skolen rundt omkring i verden, hvor det blant annet anses å som viktig å ikke bare kunne bruke teknologi, men også vite hvordan hele prosessen bak teknologien foregår (Kafai & Burke, 2013), for å blant annet møte behovet for flere folk som kan utvikle morgendagens teknologi (Sevik, 2016). Det kan også argumenteres for at dagens argumenter er mer innviklet, gjennom at programmering også skal tilrettelegge for at elevene skal øve nøkkelferdigheter som kreativitet, innovasjon, kritisk tenkning, metakognisjon, kommunikasjon, samarbeid, digital kompetanse, medborgerskap, karriere og arbeidsliv (Kunnskapsdepartementet, 2014). Eksperimenteringen med integrering av programmering i skolen utgikk rundt omkring i verden på 1990-tallet, blant annet på grunn av mangel på kompetanse blant lærere, mangel på integrering av programmering inn i fagene, og visstnok spørsmål om formålet med det (Kafai & Burke, 2013). Det ble blant annet mer fokus på å lære elever å surfe på nettet, enn at elevene skulle forstå hvordan internett fungerer. Det var ulike meninger blant lærerne, som at det var unødvendig å lære bort programmering, mens andre mente det var for vanskelig å lære seg og i tillegg lære det bort (Kafai & Burke, 2013). I skrivende stund er situasjonen annerledes, og mange land har bestemt seg for å gi integreringen av programmering i skolen en ny sjanse. I 2015 var det 12 land i Europa som allerede hadde integrert programmering inn i læreplanen, i tillegg til 7 land som planla å gjøre det i fremtiden (Balanskat & Engelhardt, 2014) og i Norge har vi som nevnt tidligere fagfornyelsens gradvise implementering skoleåret 2020/2021.

2.3 Programmering i matematikkfaget

Som nevnt i introduksjonen, regnes ofte Seymour Papert som pioneren når det kommer til undervisning av programmering i matematikkfaget på skolen, og han var med på å utvikle programmeringsspråket LOGO i et forsøk på å fremme dette inn i barneskolen (Papert, 1980). LOGO var innlemmet inn i en robot kalt Turtle (*se figur 2-3*), en *skilpadde* man kunne styre til å tegne på papir. Det argumentertes for at Turtle kunne brukes til å trene på matematiske temaer i skolen, som geometri (Papert, 1980).



Figur 2-3: Seymour Papert presenterer Turtle-roboten, 2016, av Wikimedia Commons. (https://commons.wikimedia.org/wiki/File:Seymour_Papert.jpg). CC BY-NC-ND 2.0.

I Norge, i skrivende stund, er det matematikkfaget som har fått hovedansvaret for opplæringen av programmering i skolen (Tellefsen, 2021). Ifølge Utdanningsdirektoratet (2020a) er det fokus på blant annet at elever skal bli gode til å løse problemer, i tillegg til at algoritmisk tenkning (*algorithmic thinking*) blitt mer synliggjort og programmering skal brukes for å fremme algoritmisk tenkning. I matematikk 1T kan man observere nettopp disse endringene i kompetansemålet som sier at; «mål for opplæringen er at eleven skal kunne formulere og løse problemer ved hjelp av algoritmisk tenkning, ulike problemløsningsstrategier, digitale verktøy og programmering» (Utdanningsdirektoratet, 2020a, s. 5). Det finnes mange ulike argumenter som forsøker å gi grunner til å lære matematikk gjennom programmering, og jeg har valgt å lage to hovedkategorier; algoritmisk tenkning og problemløsning, inspirert av kompetansemålet i 1T. Dette er to komponenter som læreren nå skal kjenne til og kunne hjelpe elevene med gjennom undervisning i 1T, og Utdanningsdirektoratet kompetansepakke for matematikk på videregående skal gi en innføring i spesielt disse to komponentene (Utdanningsdirektoratet, 2021)

Algoritmisk tenkning

I matematikken kan algoritmisk tenkning betraktes som en form for matematisk argumentasjon, som anses som nødvendig for å kunne forstå, teste, forbedre eller utforme en algoritme (Stephens & Kadijevich, 2020). En algoritme kan beskrives som en oppskrift, eller en fremgangsmåte, og dersom den utføres presist og systematisk vil den føre fram til den ønskede konklusjonen. Det er ulike måter å definere kjernekomponentene som anses som essensielle for å kunne øke ens ferdigheter i algoritmisk tenkning, men ifølge Stephens & Kadijevich (2020) vil det være evner i *dekomponering* (å kunne bryte ned et komplekst problem til mindre, mer løsbare problemer), *abstrahering* (å redusere enkeltfenomener under ett begrep) og *algoritmisering* (evnen til å kunne omdanne en prosess, prosedyre eller beskrivelse om til en algoritme). Det er i skrivende stund begrenset med forskning på bruken av programmering til å fremme algoritmisk tenkning i matematikk, men det påstås at det kan bidra til et større fokus på algoritmisk tenkning og dets kjernekomponenter, og slik fremme algoritmisk tenkning (Stephens & Kadijevich, 2020).

Problemløsning

Problemløsning kan beskrives som en aktivitet som tas i bruk med hensikt om å løse en oppgave vedkommende ikke har løst før (Svartdal, Problemløsning, 2019). En kan skille mellom reflekterte og ureflekterte strategier, hvor man enten forsøker å omformulere problemet slik at det blir lettere å løse, eller prøve og feile til man kommer frem til et svar. En kan ta i bruk algoritmer, som er fastsatte oppskrifter på fremgangsmåter som kan garantere suksess, eller heuristikk (også kalt tommelfingerregler), som kan beskrives som raskere snarveier som ikke kan garantere suksess. La oss ta et eksempel, hvor du er på vei ut døra, men har mistet nøklene dine. Du kan enten begynne å lete fra et hjørne til et annet, uten å hoppe over en eneste krok, eller så kan du tenke igjennom hvor det er mest sannsynlig at du hadde nøklene sist. Den første vil være en algoritmisk fremgangsmåte, og tar lengre tid, men vil gi deg løsningen til slutt; nøklene må være et sted i huset ditt, ellers hadde du ikke kommet deg inn. Den andre vil være en heuristisk fremgangsmåte som er raskere og mer effektiv, men den kan ikke garantere at du finner svaret på løsningen.

Det hevdes at programmering kan føre til økte problemløsningsevner (Kaufmann & Stenseth, 2020). Koblingen her er at man i computational thinking tar i bruk en rekke av problemløsningsmetoder for å komme frem til ønsket løsning. Videre indikerte Feurzeig, Papert, Bloom, Grant, & Solomon (1970) i sine funn at elever som fikk programmerere i matematikk ble mer trent til å kunne skille mellom når de skulle ta i bruk algoritmisk tenkning og når de burde ta i bruk heuristiske fremgangsmåter. Hva vet vi i nyere tid om sammenhengen mellom problemløsning i matematikk og programmering? En studie ifra 2009 av elevers problemløsningsevner ved bruk av LOGO i programmering fant ut at evnene varierte fra oppgave til oppgave, mens en annen studie som tok i bruk Scratch programmering fra 2017 fant ut at elever klarte å utvikle programmering som presenterte problemet og evner i logisk tenkning (Kaufmann & Stenseth, 2020). Noen forskere mener, ifølge Kaufmann & Stenseth (2020), at det kan ha pedagogisk nytteverdi å forsøke å koble matematikk til programmering gjennom blant annet problemløsning, men at det er magert med forskning. Kaufmann & Stenseth (2020) har selv gjennomført et kvalitativt forsøk hvor resultatene deres viste elever som hadde progressjon i argumentene sine mens de jobbet med en problemløsningsoppgave.

2.4 Programmering og undervisning

Programmering implementeres i skrivende stund inn i den norske skolen i 1T, og høsten 2020 ble eksempeloppgaver tilpasset fagfornyelsen offentliggjort. «Lærere med hovedfag klarer ikke løse nye eksamensoppgaver for videregående i matematikk», lyder en overskrift fra Stavanger Aftenblad (Fosse, 2020). Realfagsansvarlig ved Sandnes videregående skole, Thomas Landmark, beskriver en følelse av å være overrumplet og maktesløs, i møtet med oppgavene. Flere er bekymret for om lærere har tilstrekkelig med kompetanse for å kunne undervise programmering i matematikk (Fosse, 2020; Johansen, 2020; Utdanningsforbundet, 2020) og Landmark uttrykker i Aftenposten i likhet med flere andre bekymringen for elevers kompetanse; «Programmering er et nytt fagfelt. Dette skal bygge på det elevene egentlig skulle ha lært på ungdomsskolen. Men elevene i videregående kommer inn i et løp der de ikke har denne bakgrunnen» (Fosse, 2020).

Angående kritikken rundt læreres manglende kompetanse svarer Utdanningsdirektoratet til Aftenposten at lærerne har visst at programmeringen skal inn i matematikken lenge, og at det

er opp til skoleeiers ansvar å tilrettelegge for kompetanseøkning (Fosse, 2020). I tillegg uttrykker de at eksempeloppgavene er åpne for diskusjon og tilbakemeldinger. På Utdanningsdirektoratet sine nettsider skrives det at ansvaret for nye læreplaner ligger på den enkelte skole, men skal ikke ligge på den enkelte læreres ansvar alene (Utdanningsdirektoratet, 2020c). Videre tilbys det en modulbasert kompetansepakke for grunnskolen og videregående på Utdanningsdirektoratet sine nettsider (Utdanningsdirektoratet, 2021), hvor formålet er at lærere skal forstå hva programmering er og hvilke muligheter det kan gi i fagene deres. Det påpekes at dette ikke er et komplett programmeringskurs. Det virker som at det er opp til hver enkelt skole angående hvordan de har tenkt å øke kompetansen for sine lærere i programmering og didaktiske tilnærminger til fagfornyelsen.

Det er dessverre per dags dato begrenset med norsk forskning omkring læreres holdninger til programmering i matematikk i videregående skoler, men det finnes et par studier i Sverige som kan hjelpe til å belyse temaet. Den svenske regjeringen varslet i 2016 at programmeringen skulle inn i grunnskolen (Sevik, 2016) og i 2017 skulle programmering implementeres inn i matematikk (Vinnervik, 2020). Ut fra medieoppslagene som ble presentert i Norge kan man få et inntrykk av at mange av lærerne har gjennomgått eller trenger å gjennomgå en kompetanseheving. I flere svenske studier belyser lærerne et prioriteringsspørsmål, hvor lærerne forteller at det ikke er tid til å lære elevene både nok programmering og matematikk, og man må tenke over hva som skal sløyfes (Mozelius, Ulfenborg & Persson, 2019; Vinnervik, 2020). En lærer forteller at «more and more things are being added, but without mentioning what should be removed. This creates a stress factor in teachers» (Mozelius, Ulfenborg, & Persson, 2019, s. 704). I den samme studien nevnes det også utfordringer med å tolke læreplanene og nasjonale retningslinjer. Retningslinjene og læreplanene blir beskrevet som til tider utydelige, og at læreplanene kan tolkes på mange ulike måter. I tillegg belyses vanskeligheter som at det er utfordrende å lære programmering og å innlemme det i matematikken, grunnet dårlig tid (Humble, Mozelius, & Sällvin, 2020). Det oppleves mangel på instruksjoner for integreringen, i tillegg til for lite tid til profesjonell utvikling. Selv om Humble, Mozelius & Sällvin (2020) sin studie omhandler lærerens *oppfatninger*, kan det argumenteres for at likhetene mellom oppfatninger og holdninger er såpass store at funnene likevel vil være aktuelle for problemstillingen i min studie.

En annen utfordring som lærerne uttrykker i en kvalitativ studie angående undervisning av programmering i matematikk er tekniske problemer (Vinnervik, 2020), hvor det var snakk om

å ha problemer med både maskinvare, programvare og nettverkskoblinger. Eksempelvis kan installasjoner ikke ville installere seg likevel og man kan plutselig miste tilkoblingen til nettverket. En lærer uttrykte spesifikt at det tapper energien å forholde seg til tekniske utfordringer i undervisningen, når en forventer at utstyret skal fungere som planlagt. Selv om dette er forhold som ikke har med selve programmeringen å gjøre, kan de påvirke lærerens holdning til det å undervise programmering i matematikk. Flere funn i studien viste at lærerne kjente seg usikre og utilstrekkelige, blant annet fordi de hadde mangel på dybdekunnskap i programmering og for lite erfaring med å lære bort implementeringen. Tid ble et tema både omkring prioriteringsspørsmålet, og at det var tidkrevende å finne og evaluere læremateriell. Til tross for utfordringene så er lærerne positive til implementeringen av programmering i matematikk (Mozelius, Ulfenborg, & Persson, 2019). I deres kvalitative studie, hvor de så på åtte ungdomsskolelæreres holdninger til integreringen av programmering, beskrev tre av lærerne at kursing og kompetanseheving i programmering har fått de til å oppleve en positiv holdningsendring, i motsetning til tidligere, hvor de opplevde affektive faktorer som bekymring. Derimot var det tre andre lærere som verken hadde tatt noen kurs eller følt på bekymring, men flesteparten presiserte at det hastet med utdanning i programmering for å kunne undervise implementeringen på en god måte. Lærerne så nytteverdien av programmering i matematikk, og syntes det var en god ide for elever å lære programmering i en ung alder, og en lærer tenker at det kan være en nyttig måte å forbedre elevers logiske tenkning på.

En svensk kvantitativ studie har sett på om erfaringer med programmering hadde noe å si for holdningene til elevene, og det viste seg at å ha erfaringer med programmering kunne påvirke elevers angst mot implementeringen av programmering i matematikk (Martínez, 2019). Elever med lite erfaring med programmering før de startet på videregående, var oftere mer engstelige til at programmering skulle inn i matematikken enn elever med mer erfaring. Et annet interessant funn i den svenske studien var at hvor mye programmerings-erfaring elevene hadde, påvirket ikke hvor nyttig de syntes at programmering var for deres fremtidige utdanning eller yrkesliv. Jeg anerkjenner at dette ikke er direkte overførbart til min problemstilling, men mange av lærerne vil befinne seg i en situasjon hvor de må lære seg programmering for å kunne undervise i det. Derfor kan det tenkes at det kan dukke opp noen likheter i dataene.

Læreres holdninger til matematikk er også viktige i henhold til hvordan lærerne kommer til å undervise elever i matematikk (Ernst, 1989), og det er ingenting som tilsier at dette ikke gjelder også for programmering. Dette vil si at lærerne sine holdninger til programmering i matematikk

kan skinne gjennom i undervisningen. Ernst (1989) deler opp undervisningspsykologien i to deler; lærernes tankeprosesser og lærernes tankestrukturer. Lærernes tankeprosesser handler om å planlegge, ta beslutninger og reflektere, mens tankestrukturene handler om kunnskapen til læreren, tro og holdninger. Når det kommer til holdninger, mener Ernst (1989) at matematikklæreres holdninger kan smitte over på undervisningen til læreren;

The different philosophies of mathematics have practical classroom outcomes. For example, an active, problem solving view of mathematical knowledge can lead to the acceptance of children's methods and approaches to tasks. In contrast, a static Platonist or instrumentalist view of mathematics can lead to the teacher's insistence on there being a single 'correct' method for solving each problem. Again, a teacher's view of knowledge as integrated can lead to teaching in which mathematics and other subject matter areas are interrelated. The opposite view can result in an insistence that questions of mathematics and geography, for example, are dealt with separately during mathematics lessons and geography lessons (Ernst, 1989, s. 8).

Dette kan oppfattes som at lærere kan ha ulike filosofier rundt matematikk, hvor dette kan gi ulike utfall i klasserommet. Det kan være forskjell på en lærer med et syn på matematikk som orienterer seg rundt problemløsning, og en lærer som har et mer instrumentelt. Den første læreren vil muligens godta at det finnes flere veier til løsningen på et matematisk problem, mens den andre læreren kunne tenkes å insistere på å kun lære bort en «korrekt» metode til hvert problem. Ernst (1989) beskriver matematikklærere som lærer bort matematikk, som også gjelder mitt tilfelle, men hvor forskjellen er at de bruker programmering til å lære bort matematikk. Dermed vil det kunne tenkes seg at dersom matematikklærere har ulike filosofier rundt programmering, vil det muligvis kunne føre til ulike utfall i klasserommet. Hvis en lærer ser på programmering i matematikk som en ressurs for problemløsning, er det en mulighet for at denne holdningen vil komme til syne ved en åpen holdning til programmering i matematikk. Dersom en lærer har et instrumentelt syn på matematikk, kan mulige konsekvenser for elevene bli å lære seg *kun* det programmet som læreren har skrevet for dem, og dermed potensielt miste muligheten til dybdeforståelse. I tillegg kan mye av poenget ved å implementere programmering (som å lære seg problemløsning) forsvinne. En slik holdning, som å ha et instrumentelt syn, vil kunne komme til syne ved at læreren har en mer lukket holdning. Det siste eksempelet til Ernst (1989) handler om læreren har et smalt syn eller ikke, som vil si hvorvidt de føler fagene kan relateres til hverandre. Programmering og matematikk kan

relateres til hverandre for noen lærere, mens andre kan tenke motsatt; at programmering ikke har noe i matematikk å gjøre. Det kan tenkes at en lærer med et smalt syn vil kunne utvise mer negative holdninger til programmering i matematikk, fordi vedkommende ser de som to separate felt.

Ifølge Ernst (1989) blir lærerens tankestrukturer ofte oversett når nye reformer i matematikkundervisningen blir til, og Ernst var spesielt bekymret hvordan læreres holdninger til undervisning kan påvirke selve undervisningen deres. I skrivende stund skjer innføringen av fagfornyelsen i Norge, og denne engelske forskningen er med på å belyse viktigheten av å undersøke lærernes holdninger til undervisning av programmering i matematikk 1T. Med dette i bakhodet har jeg et ønske om å nærme meg svaret på mitt forskningsspørsmål;

Hva kjennetegner matematikklæreres holdninger til å undervise programmering i 1T?

3. Metode

3.1 Forskningsdesign og metode

I denne studien var målet å utforske matematikklæreres holdninger, og da spesifikt til å undervise programmering i matematikk 1T. Det finnes lite forhåndskunnskaper angående dette temaet, og det er spesielt begrenset forskning rundt norske matematikklæreres holdninger til å undervise programmering i matematikk 1T. Valget falt derfor på å bruke kvalitativ metode i studien, som kan gi deg en mulighet til å fange opp uttalelser som kan være utfordrende å tallfeste (Dalland, 2014). Videre skriver Dalland (2014) at kvalitativt orienterte studier er passende om man ønsker å gå i dybden på et fenomen, ønsker fleksibilitet, nærhet og forståelse ovenfor et fenomen. I og med at vi visste lite om holdningene til lærerne fra før av, ønsket jeg å gå i dybden på et fåtall informanter, for å kunne få en bedre innsikt i holdningene og forhåpentligvis øke forståelsen ovenfor dette fenomenet. Jeg valgte å gjennomføre en case-studie, hvor man blant annet ønsker å utvikle kunnskaper og forståelse av en enhet (Wæhle, Dahlum, & Grønmo, 2020), og enheten min er en gruppe matematikklærere. For å kunne gå i dybden i forskningsspørsmålet mitt var det passende å avgrense både tid og rom, slik man gjør i en case-studie (Postholm & Jacobsen, 2018), hvor jeg utførte alle intervjuene i løpet av en måneds tid, og intervjuet kun matematikklærere fra 1T.

Holdninger kan, som tidligere nevnt, forsøkes å avdekkes ved å høre på hva informantene sier (Raaheim, 2019). Derfor ønsket jeg å bruke intervju som metode, slik at jeg kunne høre på hva informantene hadde å fortelle meg, og slik identifisere deres affektive, atferdsmessige og kognitive responser. Valget falt på individuelle intervjuer, ettersom informanter kan påvirkes av hverandres meninger i et gruppeintervju (Postholm & Jacobsen, 2018), og jeg ønsket å unngå en slik gruppedynamikk. Ettersom temaet mitt for studien er lite utforsket, valgte jeg å gjennomføre semistrukturerte intervjuer, som vil si at man har noen spørsmål klare på forhånd, men at man også er åpen for at nye temaer kan bli introdusert av deltakerne (Postholm & Jacobsen, 2018). Jeg ønsket at lærerne skulle fortelle meg nokså fritt hva de følte og mente, men samtidig at det var et mål med det vi snakket om. I tillegg ønsket jeg at vi skulle skape forskningen sammen, ettersom det per dags dato dessverre er begrenset med forskning omkring mitt forskningsspørsmål.

3.2 Utvalg

Når man skal gjennomføre en kvalitativ studie er det ingen forhåndsbestemte fremgangsmåter til hvordan man velger utvalget sitt, men det er likevel hensiktsmessig å ha en plan (Postholm, 2005). Jeg ønsket å utforske læreres holdninger, og valgte dermed noen avgrensninger, som vil utdypes ytterligere. For å kunne brukes som informant, måtte læreren ha;

- undervist i teoretisk matematikk (1T) på videregående skole i skoleåret 2020/2021
- jobbet på en videregående skole på Østlandet
- gjennomgått minst ett programmeringskurs
- minst to års erfaring med å jobbe som lærer på videregående skole

Det strategiske valget med å gjennomføre studien med matematikklærere i 1T hadde flere årsaker. Jeg ønsket å avgrense studien til å kun omhandle et matematikkfag, i et forsøk på å få noen fellestrekk for utvalget, slik at jeg forhåpentligvis kunne komme mer i dybden på problemstillingen. I tillegg var det mye frem og tilbake med eksamensoppgavene til matematikk 1P og 1PY det skoleåret, om eksamenene skulle inneholde programmering eller ikke. I matematikk 1T var det mer stabilt og sikkert at det skulle komme programmering inn i læreplanen og på eksamen, og dermed kjentes det mer trygt å velge disse lærerne som hadde hatt en lengre periode til å forberede seg (både faglig og mentalt) til innføringen av programmering. I tillegg kan det tenke seg at matematikklærere i 1T har en mer solid matematisk bakgrunn enn lærere i 1P, og dermed i større grad ville sette ord på koblinger mellom programmeringen og matematikken.

Enhver by er forskjellig, et hvert fylke er ulikt og enhver skole er annerledes. Mitt valg falt på å rekruttere informanter fra Østlandet, i håp om å få lærere med mest mulig lik bakgrunn, ettersom bakgrunnen ikke skulle være et tema. Et annet fellestrekk jeg var opptatt av, var å rekruttere informanter som hadde gjennomført minst en kort kurs i programmering for lærere. Dette valget ble tatt basert på et ønske om å kunne snakke mest mulig om undervisning av matematikk med programmering, og jeg håpet på at lærere som hadde tatt kurs, kunne ha større sannsynlighet i å ha undervist matematikk gjennom programmering. I tillegg håpet jeg på at informantene hadde høyere sannsynlighet for å ha noen tanker rundt programmering i matematikk enn lærere som ikke har gått igjennom et kurs. Alle informantene hadde gjennomført et tre dagers kurs som omhandler programmering for lærere, ved et universitet, og

lærerne ble sendt på kurset gjennom arbeidsplassen sin. Til slutt er det en kjent sak at flesteparten av nyutdannede lærere må belage seg på å jobbe mer de to første årene som lærer, hvor det er mye nytt å sette seg inn i. Det kan tenke seg at det var en mulighet for at denne faktoren kunne forstyrret dataene mine, og jeg valgte dermed å rekruttere lærere som hadde mer enn to års erfaring i skolen.

Totalt ble 11 skoler kontaktet, hvorav 7 av skolenes avdelingsledere for realfag svarte at de var positive til å la lærere delta i studien min. Videre kontaktet jeg lærere på de utvalgte skolene som underviste i matematikk 1T, og det var til slutt fire matematikklærere som ville være med i studien. Jeg opplevde ikke at noen lærere sa nei, enten sa de ja eller så svarte de ikke på henvendelsen min. Både avdelingsledere og lærere ble kontaktet via epost med en forespørsel, og vedlagt i eposten var et informasjonsskriv (*se vedlegg 1*). Informantene jeg endte opp med hadde varierende erfaringer med programmering fra før fagfornyelsen i matematikk 1T, men ingen av de hadde fullverdig undervisningskompetanse i informatikk. Tre av informantene var etter min mening i midten av karrieren sin, mens en av informantene var i starten av karrieren sin. Informantene mine har blitt anonymisert, og jeg har valgt å kalle de informant A, B, C og D. Kjønn er ikke relevant i studien min, og derfor velger jeg å bruke det kjønnsnøytrale pronomenet *hen* når jeg omtaler informantene. I tabell 3-1 har jeg oppsummert bakgrunnskunnskaper om informantene, med antall år de har vært lærer og hva slags programmeringsbakgrunn de har.

Tabell 3-1: Bakgrunnskunnskap om informantene

	Antall år i yrket	Programmeringsfag på studiet	Annen programmeringserfaring
Informant A	20	Nei	Programmeringskurs for lærere (3 kvelder) og annet kort kurs.
Informant B	3	Ja	Programmeringskurs for lærere (3 kvelder) og annet kort kurs.
Informant C	12	Nei	Programmeringskurs for lærere (3 kvelder). Videreutdanning.
Informant D	15	Ja	Programmeringskurs for lærere (3 kvelder). Teknologi og forskningslære.

3.3 Semistrukturerte intervjuer

Som en forberedelse til intervjuene startet jeg med et prøveintervju med en lærer, og dette anbefales for å utforske hvordan dialogen åpner seg opp (Postholm & Jacobsen, 2018). I prøveintervjuet brukte jeg lydopptaker, og utførte transkripsjon og en kort analyse. Dette var en lærerik øvelse for å kunne både forbedre min rolle som intervjuer, og reflektere over formålet med intervjuene.

Jeg planla å gjennomføre intervjuene på arbeidsplassen til informantene, slik at de kunne føle seg mest mulig komfortable (Postholm & Jacobsen, 2018). Dessverre måtte jeg ty til plan B, som var å gjøre intervjuene på Zoom, grunnet Covid-19-pandemien. Valget falt på plattformen Zoom, hvor du kan gjennomføre digitale møter med video og lyd. Denne løsningen gjorde at jeg fortsatt kunne se og høre informantene, som jeg ville ha gjort på arbeidsplassen deres, og de kunne se meg. Jeg fikk med meg noe av kroppsspråket, men gikk for eksempel glipp av å

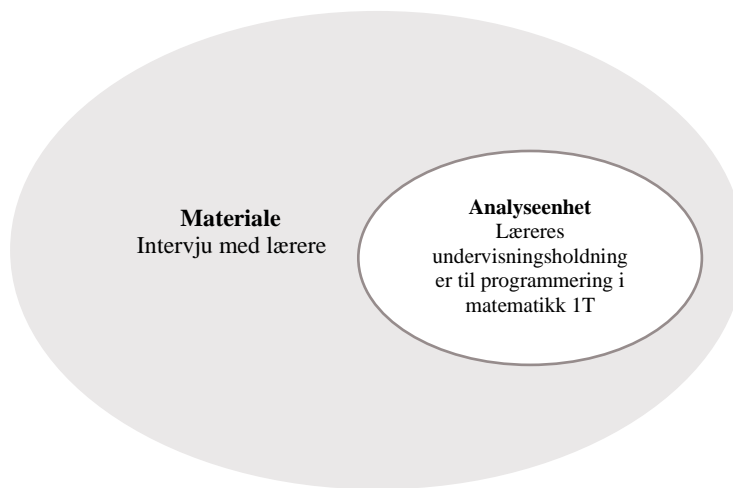
kunne se hvordan se satt på stolen, som jeg hadde fått med meg dersom vi hadde hatt fysiske intervjuer.

Alle intervjuene ble gjennomført i perioden november-desember 2020 over Zoom, og informantene valgte tidspunktene selv. Avtalen var at intervjuene skulle vare i 30 minutter. Som støtte hadde jeg på forhånd laget en intervjuguide (*se vedlegg 2*), som ble brukt under alle intervjuene. Ettersom intervjuene var semistrukturerte var jeg ikke innom alle spørsmålene med alle informantene, men jeg fokuserte på å få vært innom like temaer med alle informantene. Samtalene ble dreid inn mot temaer som hvilke erfaringer informantene hadde med programmering før fagfornyelsen, egen undervisning av programmering i matematikk 1T, tanker rundt eksempeloppgavene til 1T-eksamen (som hadde blitt publisert den høsten) og tanker rundt programmerings plass i matematikk 1T. Jeg benyttet meg av lydopptaker for å kunne ha muligheten til å transkribere intervjuene i ettertid, og informantene hadde samtykket til dette på forhånd ved å signere informasjonsskrivet jeg sendte ut på forhånd (*se vedlegg 1*).

Skoleåret 2020/2021 har en spesiell kontekst, og kan beskrives som et hektisk år for matematikklærere i teoretisk matematikk. Som nevnt tidligere starter arbeidet med å innføre fagfornyelsen for førsteklasingene på videregående, med blant annet ny overordnet del, tverrfaglige temaer, og alle læreplanene blir generelt fornyet (Utdanningsdirektoratet, 2020b), i tillegg til programmering i 1T (Utdanningsdirektoratet, 2020a). I tillegg skulle det for første gang være heldigital eksamen i 1T våren 2021 (Utdanningsdirektoratet, 2021), hvor det utover høsten 2020 ble sluppet eksempeloppgaver til ny eksamensform og gjort flere presiseringer underveis. Situasjonen informant A, B og C var i under intervjuet var at det hadde blitt sluppet eksempeloppgaver, og de ventet på justeringer. Informant D ble intervjuet et par dager etter den første presiseringen kom, hvor blant annet programmeringsoppgavene hadde blitt litt forenklet. Til slutt var vi også midt inni Covid-19-pandemien, som førte til tidvis mye digital hjemmeundervisning for lærerne. I februar 2021 avlyses både skriftlig og muntlig eksamen i matematikk 1T, ettersom en stor andel digital undervisning har gitt ulikt opplæringstilbud til elevene (Regjeringen, 2021), men det er viktig å presisere at dette skjer etter datainnsamlingen min.

3.4 Analyseprosessen

Selve ordet analyse stammer fra det gammelgreske ordet *analysis*, og betyr oppløsning (Tranøy & Tjønneland, 2020). Det er nettopp dette vi gjør når vi analyserer, vi løser opp datamaterialet vårt i flere biter, og forsøker å strukturere utvelgelsen av relevante biter (Anker, 2020). En stor del av prosessen handler også om å kunne velge ut hvilke deler av materialet ditt som skal forkastes, og hvilket som skal få bli. Jeg har inspirert meg av modellen til Geir Afdal (Anker, 2020), for å illustrere analyseenheten i forhold til datamaterialet, se figur 3-1 under.



Figur 3-1: Modell inspirert av Geir Afdal (Anker 2020), illustrerer materiale og analyseenheten

Analyseenheten består av datamaterialet, men det er en betydelig mindre enhet av den, og det er analyseenheten som det er ønskelig at datamaterialet skal bli redusert til, og som er av relevans for forskningsspørsmålet. Jeg har laget en oversikt med stegene i fremgangsmåten til min analyseprosess i figuren under, inspirert av Anker (2020) sine fire analysefaser (se figur 3-2), hvor hver av stegene vil bli utdypet underveis i delkapittelet. Analyseenheten ble mer synlig i steg 3 av analyseprosessen, hvor det startet en reduksjon i datamaterialet. De fire analysefasene gjør det derimot ikke til en lineær prosess, og jeg bevegde meg frem og tilbake mellom fasene underveis i analyseprosessen. Tanken bak de ulike fasene er at det skal være bli mer overkommelig og strukturert å utføre analysen (Anker, 2020)

Analyseprosessen

Analysefase 1: Materialinnsamling og tidlige analyser

Analysefase 2: Kondensering, koding og kategorisering

Analysefase 3: Å skrive ut analysene

Analysefase 4: Drøfting og teoretisering

Figur 3-2: De fire analysefasene (Anker, 2020)

I den første fasen av analyseprosessen skjer innsamling av data og her oppstår de første tankene og ideene rundt materialet (Anker, 2020). Jeg transkriberte lydopptakene fra de semistrukturerte intervjuene, som vil si at jeg overførte det jeg hørte på lydopptakeren til skriftlig tekst (Kvale & Brinkmann, 2015). Jeg tok bevisst med pauser og tilleggsinformasjon som ikke var tale, som for eksempel latter, for å få mer kontekst-nær transkripsjon, i tillegg til å utvikle en transkripsjonsnøkkel (*se figur 3-3*) som hjelpemiddel.

...	pause
[...]	lengre pause, over 3 sek
(tekst)	ikke-verbale faktorer (latter, sukk, e.l.)
()	utskriveren hører ikke hva som blir sagt
[tekst]	forklare kontekst
--	argument brutt
<i>kursiv</i>	ekstra trykk på et ord

Figur 3-2: Transkripsjonsnøkkel

Jeg tok hyppig i bruk tankenotater når jeg fikk ideer både etter intervjuene og under transkripsjonen, for å kunne lagre tankene mine til senere grundigere refleksjoner. Notatene kunne handle om alt fra mine helhetsinntrykk om informantenes holdninger, hva slags tonefall jeg opplevde informanten brukte under intervjuene, til ideer til senere temaer som jeg kunne bygge på i de neste analysefasene. I analysefase 1 starter også en overfladisk utvelgelse av

relevant datamateriale (Anker, 2020), hvor jeg valgte ut og merket meg utdrag fra transkripsjonen som jeg hadde noen tanker rundt at kunne være interessante for problemstillingen og omhandlet undervisning, uten å gå noe dypere i utdragene.

I den neste analysefasen kondenserte, kodet og kategoriserte jeg, slik som Anker (2020) foreslår, og denne fasen er grundigere og mer systematisk enn den forrige. Kondensering er et annet ord for meningsfortetning og kan forklares som en kort oppsummering av hva informanten har fortalt, hvor meningen i utsagnet skal frem (Kvale & Brinkmann, 2015). Dette valgte jeg å gjøre slik at arbeidet kunne bli mer effektivt, ettersom jeg da forhåpentligvis raskere kunne forstå hva utsagnet dreide seg i arbeidet med analyseprosessen. Å kode transkripsjonen handler om å sette merkelapper på materialet (Anker, 2020). Etter å ha lest teori, samlet inn dataene, transkribert og sortert materialet, hadde jeg allerede startet å gjøre meg opp en mening omkring ulike kategorier og koder. Samtidig så hadde jeg ikke bestemt meg for alle kategoriene på forhånd, ettersom dette er et lite utforsket felt, og det er begrenset med teori rundt mitt forskningsspørsmål. Derfor falt valget på å utføre en abduktiv analyseprosess, hvor det veksles mellom empirisk materiale og teori (Anker, 2020). Dette kan ses på som en blanding av empirinær koding og teoretisk koding, hvor man da blander induktiv og deduktiv koding. Jeg har derfor delvis konstruert kodene og kategoriene ved hjelp av teori jeg har lest, i tillegg til at det har dukket opp flere underveis, og resultatet mitt er et samspill av å gå frem og tilbake mellom empiri og teori.

Jeg forsøkte å svare på hva læreres holdninger er ved hjelp av tre underspørsmål og valgte å utføre en sortering (*se tabell 3-2 for eksempel*), før jeg gikk videre til å lage underkategorier under hver av hovedkategoriene. Sorteringen var basert på trekomponentmodellen til Raaheim (2019), med tre hovedkategorier; affektive, atferdsmessige og kognitive responser. For å kunne sortere ut affektiv respons så jeg etter om informantene brukte ord som uttrykte følelser og beskrev sinnsstemninger, i tillegg til tankenotatene omkring tonefallet til informantene i de ulike utsagnene. Atferdsmessig respons ble sortert ut ved å se etter ord og setninger som uttrykte hva informantene sa de gjorde, og for kognitiv respons var konsentrert rundt ord og setninger som uttrykte tanker og meninger.

Responsene kunne iblant være utfordrende å skille, for hvor går egentlig skillet mellom følelser og tanker? Dersom en informant forteller intervjueren av de *følte* noe, kjente på følelser eller uttrykte følelser som for eksempel glede, sinne eller irritasjon, valgte jeg å tolke dette som en

affektiv respons. Et eksempel på et utdrag fra transkripsjonen hvor jeg har valgt å tolke det som affektiv respons (understrekede ord ser jeg på som typiske for affekt), er følgende;

Informant A: Men jeg føler at jeg må bli litt tryggere. Jeg føler meg ikke trygg nok til å undervise i det her, egentlig.

Her har informant A brukt ordet *føler* to ganger, og forteller at hen ikke føler seg trygg nok til å undervise programmering i matematikk 1T, og føler et ønske om å bli tryggere. Men i dette eksempelet forteller også informant A om at hen ikke føler seg trykk nok til å undervise i det her, og det å undervise er jo noe man gjør. I dette tilfellet valgte jeg å la utsagnet gå under affektiv respons og ikke atferdsmessig, ettersom atferdsmessig respons var konsentrert rundt at informantene skulle fortelle meg spesifikt hva de gjør, og i dette tilfellet forteller informanten om hva hen føler om ting hen gjør, som jeg velger å kalle affektiv respons. Slik fortsatte jeg å sortere ut resten av responsene, og passet på å tenke nøye over skillene.

Tabell 3-2: Eksempel på sortering og meningsfortetning i analysearbeidet

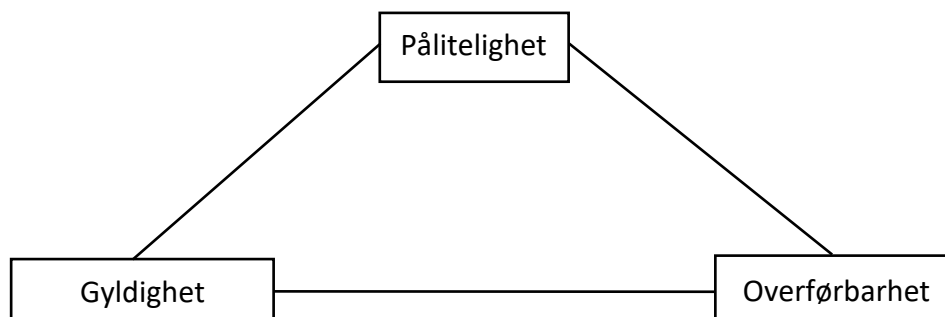
Merkelapp	Eksempel i transkripsjonen	Meningsfortetning
Affektiv respons (følelsesmessig)	... så jeg har prøvd, men jeg <u>føler</u> at det her, dette er jeg ikke noe god på (latter). I utgangspunktet <u>følte</u> jeg meg kanskje litt <u>alene</u> der fordi kollegaen min kunne ingenting (latter).	Føler seg ikke god på programmering Føler seg alene om programmeringen

Etter sorteringen og kodingen vokste det frem fem underkategorier; positiv affekt, negativ affekt, undervisningsatferd, nytteverdi og kompetanse. Underkategoriene vil bli presentert nærmere under kapittel 5. De ulike underkategoriene vokste frem som en blanding av teori og empiri, og jeg fokuserte på å finne felles holdepunkter mellom informantene i form av at alle skulle uttalt seg noe om de ulike kategoriene. Analysefase 3 handler om å skrive ut analysene,

noe jeg velger å gjøre under kapittel 4, hvor resultatene vil bli presentert. Til slutt handler analysefase 4 om drøfting og teoretisering, som vil bli gjort under kapittel 5.

3.5 Studiens kvalitet

Ifølge Anker (2020) kan man i vurderingen av kvaliteten til kvalitative studier ved å se på hvor pålitelig og gyldig studien er. Pålitelighet, eller *reliabiliteten*, handler om hvor troverdig forskningen er (Kvale & Brinkmann, 2015). Har jeg vært troverdig i min forskning, og vil en annen forsker oppnå helt andre resultater? Videre handler gyldighet, eller *validitet*, om hvor godt metoden min passer til det jeg skal undersøke, og om jeg måler det jeg tror jeg måler (Kvale & Brinkmann, 2015). Til slutt kan man også se på hvor generaliserbar en studie er, som kan beskrives som hvor overførbare funnene mine er til andre situasjoner (Postholm & Jacobsen, 2018). Dersom mine funn er generaliserbare vil for eksempel andre lærere kunne kjenne seg igjen i holdningene til mine informanter, eller deler av holdningene. Jeg har valgt å se reflektere over studiens kvalitet ved å se på pålitelighet, gyldighet og overførbarhet, som en slags hellig treenighet (se figur 3-4).



Figur 3-4: Den hellige treenighet angående studiens kvalitet

I et forsøk på å kvalitetssikre studien min har jeg forsøkt å være åpen om fremgangsmåten min i analyseprosessen, og begrunnet valgene for metoden. Jeg har lagt ved intervjuguiden så den kan brukes igjen, dersom det er ønskelig å gjenta studien. Under intervjuene har jeg forsøkt å holde meg så nøytral som mulig, for å ivareta min rolle som forsker. Transkripsjonen skjedde ordrett, i håp om å ivareta *virkeligheten* på best mulig måte, når overføringen fra muntlig tale til tekst foregikk. Semistrukturerte intervjuer ga meg en mulighet til å forsøke å forstå informantenes perspektiver (Postholm & Jacobsen, 2018), og valget av metoden kan ha styrket å få innblikk i holdningene til informantene. Det er utfordrende å sette seg inn i personers holdninger etter så korte intervjuer, og det en informant forteller trenger ikke nødvendigvis alltid samstemme med hva man faktisk gjør (Kennedy & Kennedy, 1996). Jeg fikk ikke muligheten til å utforske dypere om det informantene mine fortalte meg, faktisk stemte i praksis. I ettertid tenker jeg at jeg gjerne skulle kombinert intervjuene med observasjon, for å styrke troverdigheten til funnene. I tillegg er det en risiko for at informantene har fortalt meg det de tror at jeg vil høre, og ikke det som de faktisk tenker, føler og gjør.

Dersom en annen forsker skulle utført en liknende studie, kan det tenkes at forskeren kunne fått andre funn, men dette kan skyldes av at ulike forskere og ulike informanter vil kunne bringe med seg sin subjektivitet inn i forskningen (Postholm & Jacobsen, 2018). Min undersøkelse og rolle som forsker kan ha påvirket funnene mine ved at ens egne oppfatninger kan påvirke hva slags funn man ender opp med (Postholm & Jacobsen, 2018). Jeg vil også være åpen om at jeg jobbet som hjelpelærer på programmeringskurset jeg rekrutterte fra, og dette kan ha hatt en innvirkning på datamaterialet. Uansett hvor nøytral jeg har forsøkt å forholde meg, både til datainnsamlingen, analysen og drøftingen, har jeg egne oppfatninger at temaet, som kan ha smittet over på funnene mine.

Når man ser på generalisering vil fire informanter tenkes som et lavt tall dersom jeg skulle utforsket alle lærere i hele Norges holdninger. Målet mitt var å få et fylldig innblikk i utvalgets holdninger, og med begrenset tid vil det være fordelaktig å unngå å intervjuer for mange informanter (Kvale & Brinkmann, 2015). Yin (2013) argumenterer for at man kan øke generaliserbarheten ved å knytte resultatene opp mot tidligere teorier, og beskriver det som *analytisk generalisering*. Jeg har forsøkt, der det har vært mulig, å knytte resultatene mine opp mot tidligere teorier, og dette mener jeg at kan bidra til at deler av funnene mine kan være overførbare. For å ytterligere analytisk generalisere bør man også være åpen om hvordan man har kommet frem til resultatene, ved å blant annet gi leseren muligheten til å trekke sine egne

slutninger om funnene, og dette kan gjøres ved å gi leseren innblikk i datainnsamlingen, analysen og resultatene (Kvale & Brinkmann, 2015). Jeg har forsøkt å legge til rette for at leseren skal kunne være med på å vurdere om funnene er generaliserbare, og dette mener jeg også kan styrke overførbarheten.

3.6 Forskningsetikk

Når man driver med forskning, har man et etisk ansvar overfor forskningsdeltakerne, og som Kant en gang sa; «Du skal aldri ville bruke noe menneske som bare et middel» (Postholm & Jacobsen, 2018, s. 247). Jeg har intervjuet mennesker, og det er viktig at de blir ivaretatt i løpet av forskningsprosjektet ved frivillig informert samtykke, anonymisering og riktig fremstilling av data (Aker, 2020; Posthold & Jacobsen, 2018). Jeg var pliktig til å melde inn prosjektet mitt til Norsk senter for forskningsdata (NSD) grunnet valg av å bruke lydopptaker som en del av datainnsamlingsmetoden (*se vedlegg 3*). I rekrutteringsperioden fikk informantene tilsendt et informasjonsskriv (*se vedlegg 1*) for å sikre informert samtykke. Informantene kunne trekke seg fra forskningsprosjektet når som helst, og det skal være frivillig å delta (Postholm & Jacobsen, 2018). For å sikre anonymitet har jeg blant annet fjernet navnet på informantene, skolen de jobber på, kjønn og alder. Jeg har også forsøkt å være hensynsfull mot alle parter når jeg har transkribert, og tatt vekk alle navn på blant annet kollegaer, universiteter og kurs, for å anonymisere ytterligere. Data som inneholder beskrivelser av klassene til informantene har også blitt fjernet, av samme grunn. Jeg har iblant havnet i dilemmaer angående spennende og interessante utsagn som omhandler andre enn informantene selv, men som jeg har valgt vekk ettersom det kun er informantene som har samtykket til å være med i forskningsprosjektet.

Når det kommer til fremstilling av riktig data valgte jeg å transkribere ordrett, i et forsøk på å få med meg mest mulig av *virkeligheten* tilbake i tekstform. Når det kom til å legge frem dataene fra transkripsjonen, har jeg visse steder fjernet overflødig tekst slik at utsagnene er mer lesbare i tekstformat. Når vi snakker bruker vi et muntlig språk, og dette kan ende opp med å forvirre leseren når det oversettes til tekstformat (Anker, 2020). Et godt prinsipp som Anker (2020) legger frem er at man skal kunne se informantene i øynene etter å ha publisert oppgaven sin, og dette er et prinsipp jeg strevde etter å oppfylle under arbeidet.

4. Resultater

I dette kapittelet kommer analysefase 3 inn i bildet, hvor resultatene fra datainnsamlingen presenteres. Bakgrunnskunnskaper om informantene har jeg tidligere presentert i tabell 1. Jeg har valgt å dele inn resultatene etter de tre responsene fra trekomponentmodellen til Raaheim (2019). Resultatdelen skal helst være så tolkningsfri som mulig, men det må presiseres at jeg har tolket hvilken respons dataene hører til, som en konsekvens av sorteringen.

4.1 Affektive responser

Ordet *usikker* dukker ofte opp blant informantene når man ser på affektive responser, med visse variasjoner. Informant B uttrykker at hen føler seg usikker dette skoleåret, og på spørsmål om hen kan utdype hvorfor svares det at det er utfordrende å vite hvor mye tid man skal bruke på programmeringen i 1T. Hen forteller videre at det ikke er ønskelig at opplæringen av programmering skal gå på bekostning av annet fagstoff elevene skal lære i løpet av skoleåret i 1T. Informanten trekker frem en av grunnene til at det er utfordrende å vite hvor mye tid man skal vie til programmeringen er Utdanningsdirektoratet sine stadige presiseringer til hvordan den nye eksamen kommer til å bli, og alle informantene nevner utfordringer knyttet til dette skoleårets eksamen i 1T. Informant A beskriver eksamens-situasjonen som vanskelig, ettersom lærerne kun har fått et par eksempeloppgaver, og ikke et fullverdig sett enda, og snart har halve skoleåret gått. Det er, ifølge informanten, ønskelig å få et fullverdig eksempel på ny eksamen, i god tid i forveien, for å kunne føle at man har kontroll. Informant D beskriver skoleåret som «det er så mange usikre faktorer, særlig med eksamen», og trekker også frem heldigitaliseringen av eksamen som et bekymringsverdig problem.

Når informant A bruker ordet *usikker* i forbindelse med undervisningen av programmering i 1T får hen spørsmål om det er annerledes å undervise 1T med og uten programmering og svarer;

Informant A: (...) det har litt med det å så ikke gå inn med den tryggheten da, og at, man har jo aldri full kontroll, det har man jo ikke, det vil jo alltid komme opp problemstillinger også innenfor på en måte om vi skal kalle det for vanlige mattetimer, men man har i hvert fall litt mer faglig kontroll på

hva man holder på med og det føler jeg ikke at jeg har da og da må jeg forberede meg mer (...). Så jeg føler i hvert fall at det er mer forberedelser til det og mye mer usikkerhet.

Det kan virke som at det er mer usikkerhet rundt en matematikktime som inneholder programmering, blant annet fordi det blir brukt mer tid på forberedelser til timen. I tillegg beskriver informanten en følelse av manglende trygghet i en slik time. Hen utdyper også denne manglende tryggheten ved å fortelle at «Jeg føler jeg må bli litt tryggere. Jeg føler meg ikke trygg nok til å undervise i det her, egentlig.» Informant D, derimot, bruker ordet *trygg* opptil flere ganger når hen forteller om sin opplevelse av å undervise programmering i IT, blant annet i dette sitatet;

Informant D: Ja, for jeg har jo hatt programmering som fag og sånt, selv om det er lenge siden, så jeg er jo vant med tankegangen, og så syntes jeg jo sånt er litt gøy i utgangspunktet, og da hjelper det jo alltid da. Så, selv om man er vant med-, om man ikke får til alt med en gang, så er man liksom trygg på det da (...), når man har kjent på det et par ganger alt før så er man trygg på det da. Men folk som kanskje møter det her for første gangen kan jo kanskje bli litt satt ut.

Det virker som at informanten føler seg trygg på å undervise programmering ettersom hen har tidligere erfaringer med å undervise programmering i skolen, i tillegg til at hen føler det er gøy. Videre forteller informant C at hen behersker programmering på et anstendig nivå i henhold til undervisning av IT dette skoleåret;

Informant C: Ja, men jeg syntes jo fortsatt ikke at det er-, altså ja, for da tenker jeg at, okay, nå behersker jeg i hvert fall det som jeg skal, i hvert fall på et nivå som er sånn på det jeg skal lære bort, men jeg er jo fortsatt ikke noe, jeg hadde ikke noe god følelse i forhold til det [å undervise programmering], det har jeg jo ikke, men det handler vel så mye om det tekniske, som om selve programmeringen.

Det kan virke som at informanten har nok kompetanse til å undervise programmering dette skoleåret, men at det er andre ting som får hen til å kjenne på negative følelser. Hen forteller

om flere episoder hvor det har vært utfordrende å undervise grunnet tekniske problemer som oppstår i timene, når man skal ha programmering i IT, og ordet frustrert dukker opp flere ganger i løpet av samtaleemnet. I tillegg forteller hen om at det føles utrolig frustrerende når elever velger å ha Apple sin MacBook istedenfor datamaskiner som opererer med Microsoft Windows, fordi det er utfordrende å hjelpe elevene når informantene ikke har MacBook selv. Dette er noe informant D også nevner, at det kjennes utfordrende å hjelpe elever med en slik datamaskin. Videre forteller informant C om flere tekniske problemer som oppstår er at elevene iblant ikke er godt nok kjent med sin datamaskin:

Informant C: ... man tror jo at elevene er gode, at de kan sin Mac eller PC, men de har jo ikke peiling hvis de ikke får lastet ned det de skal laste ned, om det er en brannmur her eller der, ikke har fått lastet ned riktig pakke til programmeringsspråket, det er så mange ting som tar fokuset bort fra selve programmeringen, som er vanskelig.

Det kan virke som at utfordringer med elevers datamaskiner kan føles vanskelig ut for informanten. Andre tekniske problemer som dras frem er at elevene ikke får lastet ned det som er ønskelig å laste ned fra internett, elevene har mistet kontakt med serveren til nettleserprogrammet de har forsøkt å bruke og generelt at ting ikke fungerer som best mulig. Når informant A uttrykker frustrasjon, er det i forbindelse med at hen blir sendt på kurs for sent, og tankene omkring dette utdypes under kognitiv respons (*se kapittel 4.3*). Informant B sin frustrasjon dreier seg omkring den nye læreplanen, hvor det uttrykkes at «det skulle kortes litt ned på læremålene, og at det skulle være i dybden, men vi har jo egentlig fått enda mer arbeid nå siden vi må lære de programmering og sånt ...».

Informant A og B trekker også frem andre faktorer som har føltes utfordrende, og de bruker ordene *ensom* og *alene* når de forteller om slike situasjoner. Informant B forteller om å føle seg alene, ved å være den eneste læreren som underviser IT på sin skole;

Informant B: Ja, det har jo vært en del arbeid å sette seg inn i nye endringer og spesielt når vi har jo såpass liten skole at jeg har jo stått litt alene med å prøve å sette meg inn i dette og forstå, så det har jo gjort det litt utfordrende

Når informant A forteller om ensomhet, er det i forbindelse med undervisningstimen hen hadde sammen med en annen kollega, og forteller at «i utgangspunktet følte jeg meg kanskje litt aleine der også fordi kollegaen min kunne ingenting (latter).» Oppsummert bruker informantene affektive ord som usikker, frustrert, ensom og vanskelig, for å uttrykke seg rundt undervisning av programmering, men det dukker også opp ord som trygg hos noen informanter.

4.2 Atferdsmessige responser

Dersom en sammenlikner informantenes oppgitte tidsbruk til å undervise programmering i 1T, ser man at informant A og B har brukt vesentlig mindre undervisningstimer i 1T på programmering enn C og D. Informant A og B har kun brukt én fagdag totalt, mens C og D har brukt flere økter og blant annet gått igjennom et par moduler på lærebokforlaget sine nettsider. Ut fra det informant C og D forteller så virker det som at de har undervist programmering jevnlig i løpet av høsten, men at rødt nivå på skolen og hjemmeskole har satt en midlertidig stopper for programmeringen. Informant D er den eneste som bruker ordet *utforskende* når hen snakker om undervisning, bruker ordet gjentatte ganger og beskriver at det er fint å bruke programmering opp mot utforskende undervisning. Det virker ikke som at det er tid til å drive med utforskende arbeidsoppgaver i skrivende stund grunnet elevens begrensede forhåndskunnskaper i programmering, men at hen ser frem til en mulig fremtidig undervisningsatferd som kan inneholde mer utforskning i timene. Videre forteller informant D om at hen foretrekker samarbeid når elevene skal øve seg på programmering, men at dette har vært utfordrende grunnet hjemmeskole og hen ville dermed vente til skolene gikk over på gult nivå igjen. Intervjuet ble dog gjort et par dager før Microsoft Teams fikk funksjonen *breakout rooms*, hvor man kan dele opp elevene i mindre grupper slik at de kan samarbeide over Teams, noe informant D nevner at kan bidra til å kunne starte opp med programmering igjen i 1T. Samtidig så deler også informant D følgende tanker om det å la elevene samarbeide om programmeringsoppgaver;

Informant D: Men sånn, med programmering og gruppearbeid så er det litt lett for at det blir den som har skjønt mest på gruppa som sitter da og taster og sånt da. Og da er det lett for de andre å så slappe av. Når man holder på med programmering, spesielt i starten, så har jeg troa på at flest mulig skal gjøre det da. Det er jo et håndverk, det sitter i fingrene, du må komme

inn i den tankemåten, og da sitte og se på noen andre gjøre det, da-, du faller veldig fort av da. Så det å gjøre ting her, det er kjempeviktig, du må programmere, alle må programmere, hvis de skal lære seg det da.

Informanten uttrykker at alle elevene helst skal få muligheten til å programmere litt selv også, fordi programmering er et håndverk. Derimot så forteller hen videre at dette ikke helt lar seg gjøre i så stor grad ettersom hvis dette skal gjennomføres så blir det mer arbeidsomt for læreren. Alle elevene trenger hjelp i timen, forteller informanten, og det kan virke som at det er mer gjennomførbart å la elevene samarbeide i timene, for å rekke å hjelpe flest mulig elever med programmeringen. Det som informant D konkluderer med at hadde vært aller best var om elevene fikk jobbet en del med programmeringen alene på forhånd, og deretter jobbet sammen etterpå. «Da får man ideer og forståelse og det blir kanskje lettere å ta tak i de lure ideene, som de veldig flinke [elevene] kommer med», uttrykker hen. Informant C har også noen tanker rundt hvordan det har fungert best å undervise programmering i IT for hen;

Informant C: (...) det fungerer egentlig best hvis man sitter og programmerer i fellesskap, at de [elevene] taster akkurat det samme som du gjør. Men det er klart at det er jo litt begrensa av hva de lærer av det, de får jo ikke vært igjennom noe algoritmisk tankegang liksom (latter). Men, det som skjer når de skal jobbe alene er jo at det er så mange som skal ha hjelp hele tiden, så man rekker ikke over da. Pluss at jeg bruker-, det tar veldig lang tid å hjelpe, hvor er feilen, let i den høystakken, så jeg har vært veldig sånn, da må dere prøve å hjelpe hverandre, og så, men de er jo ikke så, den gruppa jeg har er det ingen som har programmert før. Ja.. Så ingen elever har noen erfaring.

Det virker som at det i praksis har fungert best når informanten og elevene programmerer samtidig, for når elevene skal prøve på egenhånd så blir det for arbeidsomt å rekke å hjelpe alle. Ulempen, ifølge informanten, er at elevene ikke får trent seg på algoritmisk tankegang, slik som det er ønskelig i fagfornyelsen. Videre forteller informanten om matematikklærere i praktisk matematikk, som har det hen kaller *panikk* over at de skal undervise programmering, og forteller litt om arbeidsmetoder man benytter i programmeringsundervisning som kan skille seg fra tradisjonell matematikkundervisning;

Informant C: (...) det er litt ulikt det vi vanligvis gjør [i matematikktimer]. De [elevene] har jo lært på ungdomsskolen at når vi skal løse en likning så skal vi jo ikke prøve og feile, man skal løse en likning og man skal komme fram til svaret. Så står jeg her og sier «jo men prøv og feil i vei», ikke sant, dere [elevene] må teste, og så må dere prøve å finne feilen, og så må dere modifisere litt-, de er vant til at de skal ha det riktig, og så skal de få svaret. De [elevene] skal ikke drive og sette inn nye ting og se om det fungerer, og hvis det ikke fungerer så kan man Google (latter). Det er på en måte arbeidsmetoder som de alltid har blitt advart mot, føler jeg (latter).

Informant A og B beskriver at de har fokusert på å ha en introduksjon til programmeringen, hvor de har installert programmet sammen med elevene og vist frem hvordan det fungerer. Når de beskriver programmerings-timen bruker begge informantene ordet *kalkulator*, hvor det var programmet som ble brukt som en kalkulator og elevene skulle øve seg på blant annet skrive variabler i programmet.

4.3 Kognitive responser

Når informantene blir spurt om programmeringens plass i matematikk 1T oppleves det som at alle informantene har noen tanker omkring hvorvidt det er nyttig å lære bort programmering i 1T eller ikke. Informant B forteller at hen gjerne skulle lært det i matematikken når hen selv var elev på videregående, og at det hadde vært nyttig å kunne til man begynte å studere. Informant C forteller at det er fint med programmering og algoritmisk tenkning, mens informant D forteller gjentatte ganger at programmering er et nyttig verktøy i 1T. For informant D er det et nyttig verktøy til blant annet utforskende arbeidsoppgaver, mens informant C utdyper dessverre ikke noe mer om hvorfor det er fint med programmering og algoritmisk tenkning. Informant A sin opplevelse av nytteverdi skiller seg spesielt ut fra de andre informantene sine opplevelser og hen forteller om en endring i tankegangen den siste tiden;

Informant A: Så, (latter), jeg har forandra veldig mening fra å føle at dette her er noe som bare tynger til noe som jeg ser kan bli et veldig godt verktøy. Det er

bare det at jeg er ikke helt der selv og jeg trenger mer tid. Så jeg skulle ønske jeg var mye flinkere til det der! (latter)

Hen forteller videre at til å begynne med så hen kun kommandoer, og tenkte tanker som «hva skal jeg med dette [programmering i matematikk 1T]?» og «hvorfors kan jeg ikke løse det [matematikkoppgaver] enklere med andre hjelpemidler?». Deretter beskriver hen en endring i tankegangen, hvor hen blant annet startet å tenke at «en dag kommer programmering til å ta over for GeoGebra», hvor man kunne gjøre samme tingene som man kan gjøre i GeoGebra, men «lage» GeoGebra selv og dermed få det som beskrives som «full kontroll» over programmet. Hen fikk en økt forståelse for bruksområdene til programmering i matematikkundervisningen og tenker at det er viktig å kunne se de. I tillegg til at læreren skal kunne se bruksområdene trenger også elevene å se de, legges det til av informanten. På spørsmål om hva som har bidratt til at hen nå ser nytten, trekker informant A frem programmeringskurset hen har deltatt på, sammen med samtaler med kolleger.

Videre er et annet ord som ofte blir nevnt av informantene, ordet *kompetanse*. Informant C forteller om følelser av frustrasjon i tiden før fagfornyelsen, hvor kognitiv respons tar over med tanker rundt kompetanse;

Informant C: (...) Det er veldig rart å få i fanget at du skal undervise noe du overhodet ikke er kvalifisert til. En annen ting er å si sånn, ja, nå skal vi putte inn geometri, sånn som de gjorde forrige gang. Da tenker man; «ja, det hadde jeg en gang for 25 år siden», og så kan man grave fram det. At du skal lære bort programmering, som folk tar en master i for å få kompetanse, det er litt sånn derre spesielt. Så, jeg tenkte, ja, da skal jeg lære det da, ved siden av jobben, på et år, og ikke nok med at jeg skal *lære* meg det, men jeg skal helst lære meg å bruke og få det inn i matematikken, (...)

Det kan virke som at hen tenker det er en ulempe å ikke ha tilstrekkelig med kompetanse i programmering og at det er meningen å skulle få økt kompetanse ved siden av jobben. Informant A uttrykker også tanker rundt heving av kompetanse i programmering, og som nevnt tidligere uttrykkes også affektiv faktor som frustrasjon i forbindelse med tanker om at skoleledelsen burde sendt lærerne på kurs *før* implementeringen;

Informant A: Det skal ikke komme, altså vi kan ikke lære parallelt med elevene, jeg kan ikke lære parallelt med elevene, jeg må være forberedt, jeg må vite hva jeg holder på med, og da må man ha det i forkant og ikke i etterkant, så vi har hatt litt (), men jeg syntes det er for lite tid (...)

I tillegg forteller informant A, som nevnt tidligere, at hen underviser i noe hen ikke kan selv, som igjen handler om kompetanse. I tillegg uttaler hen seg om at hen skulle ønske at hen var flinkere i programmering, og at tilgang til kurs burde kommet tidligere. Videre forteller informant D forteller om kollegaer sin kompetanse, og har noen tanker rundt kollegaers opplevelser i møtet med programmering i IT;

Informant D: Så jeg ser at det er mange som syntes det her er litt guffent og sånt, de har jo kjørt det samme kurset i 1P matematikk, og der er det mange av lærerne som har vært svært lite bort i programmering og da er det en «bug» nå første gangen (heh). Det er mange som er litt ukomfortable, får jeg inntrykk av. Men her jobber vi jo stort sett i team, så da har man støtte i teamet, og da kan man få med seg noen hvis man føler seg veldig usikker, så da er det litt sånn at vi er med på å dra hverandre opp. Men det er jo ikke sikkert at skoler som er veldig små og sånn kanskje, at man ikke har så mange i ryggen. Og da kan jeg tenke meg at det kan være tøft.

Hen forteller at det er mange av lærerne som har begrenset kompetanse i programmering, og for disse lærerne mener hen at det kan gå over i affektive faktorer som at de føler seg ukomfortable. Det virker som at det finnes fordeler på denne arbeidsplassen ifølge hen, at lærere med mindre kompetanse kan få støtte i teamet, men tenker at det kan være utfordrende på skoler hvor det ikke samarbeides like mye mellom kollegaer.

Informantene diskuterer også rundt elevers kompetanse og forhåndskunnskaper i henhold til programmering. Informant A nevner at elevene trenger med tid til å lære seg programmering, og at det kanskje kan bli enklere for elevene om et par år, når de allerede har hatt litt programmering tidligere i skoleløpet. Grunnen til at det kan bli enklere blir beskrevet som at lærerne «ikke må holde på med det helt enkleste, men at vi kan ta over noen [en klasse] som kan littegrann [programmering] fra før av».

Informant B forteller at et fåtall av elevene har sett programmering før 1T og det blir en stor overgang for elevene å skulle ha programmering i 1T dette skoleåret. Hen legger også til at det vil, ifølge hen, også være en stor overgang for elevene et par år fremover i tid, men at hen håper at dette vil bli bedre om et par år. Informant C peker på at det er problematisk at elevene ikke kan programmere fra før av, ettersom man i skrivende stund må bruke tiden på å lære elevene programmering først, men hen uttrykker også et håp om at det kan bli bedre om et par år. Informant D sier seg også enig i at situasjonen vil kunne forbedre seg om et par år og uttrykker dette med jevne mellomrom, blant annet her;

Informant D: Men, det blir jo spennende å se når elevene kommer fra ungdomsskolen med litt mer erfaring etter hvert da. Da kanskje det går litt mer sånn sømløst (...)

Oppsummert har informantene snakket om programmeringens plass i matematikkundervisningen, og det har blitt presentert ulike tanker omkring kompetanse. Ordene som går igjen er blant annet *nyttig* og *kompetanse*, når informantene deler sine tanker i intervjuene.

5. Diskusjon

Hensikten med dette kapittelet er å nærme seg svaret på problemstillingen; *hva kjennetegner læreres holdninger rundt undervisning av programmering i IT?* Fra analysen av intervjuene vokste som presentert tidligere fem hovedkategorier frem (i analysefase 2), som er fordelt under affektiv, kognitiv og atferdsmessig respons: usikkerhet, frustrasjon, undervisningsatferd, nytteverdi og kompetanse. De følgende delkapitlene tar for seg hver av de tre responsene, med fokus på hovedkategoriene og sammenhenger innenfor de ulike kategoriene. I delkapittel 5.1-5.3 forsøker jeg å svare på underspørsmålene som ble presentert innledningsvis og i delkapittel 5.4 forsøker jeg å nærme meg svaret på selve forskningsspørsmålet.

5.1 Hvilke følelser uttrykker informantene knyttet til undervisning av programmering i 1T?

Under affektive responser vokste det frem to kategorier, som jeg har valgt å kalle for *usikkerhet* og *frustrasjon*. Mitt valg av kategoriene begrunnes med at dette var følelser som gikk mest igjen samlet sett i datamaterialet, men med visse variasjoner blant informantene. Til tross for variasjonene, og at ikke informantene har uttrykt seg likt under responsene, mener jeg fortsatt at denne inndelingen er et godt forsøk på å få frem hva informantene felles *føler* i størst mulig grad, omkring undervisning av programmering i 1T. Usikkerhet og frustrasjon kan plasseres under negativ affektiv respons og merkelappene under kategoriene hadde negative ord i seg. Derimot vil jeg presisere at det ikke kun var negative responser blant informantene. Visse informanter uttrykte også positive responser, som *grad* av trygghet, og dette vil utdypes ytterligere. Positive og negative ord er dog kontekstavhengige. Eksempelvis vil en person som fryser kunne se på varme som noe positivt, mens på en annen side vil en person i en hetebølge kunne se på varme som noe negativt. Dermed ønsker jeg å legge til at under analyseprosessen ble i tillegg fokusert på konteksten til ordene informantene valgte å uttrykke seg med, i tillegg til at tankenotatene omkring tonefallet til informantene også hadde en innvirkning i tolkningene.

I kategorien usikkerhet tenker jeg på den affektive betydningen av ordet, som kan beskrives som en følelse av manglende trygghet og sikkerhet, hvor et synonym er utrygghet (Det Norske Akademi for Språk og Litteratur, 2021). Jeg har fokusert på ord som *usikker* og *utrygg*, samtidig som jeg har sett på utsagnene i kontekst og gjort opp mine egne tolkninger. Informantene uttrykte, dog i noe ulik grad, en følelse av usikkerhet når det kom til å undervise programmering i 1T. Det kan virke som at en av grunnene er den nye eksamensformen i 1T og uforutsigbarheten informasjonen gitt fra Utdanningsdirektoratet har ført med seg angående eksempeloppgavene til eksamen. Ettersom informantene hadde mange tanker omkring eksamen og hvordan dette påvirker undervisningen, kan det argumenteres for at eksamen kan virke litt styrende på undervisningen. Ifølge informant D bør man undervise slik at elevene er forberedt til eksamen. Hos informant B virker det som at eksamen bidro til usikkerhet omkring hvor mye tid hen skulle bruke på å undervise programmering i 1T, i likhet med Mozelius, Ulfenborg & Persson (2019); Vinnervik (2020) sine funn, og informanten begrunner det med usikkert omkring hvordan eksamen kom til å bli til slutt. I likhet med Mozelius, Ulfenborg & Persson (2019) sine funn, kan det virke som at det kan til tider være utfordrende å tolke nasjonale retningslinjer.

I dette skoleåret var ikke eksamen helt klar når intervjuene ble gjennomført i perioden november-desember, og det virket ikke som at informantene var sikre på hvordan den endelige eksamen kom til å se ut, og det kan argumenteres for at dette var en vesentlig faktor som bidro til følelsen av usikkerhet blant informantene. En kan dermed argumentere for at det kan være viktig å gi lærere nye eksempler til eksamener i god tid i forveien, og gjerne før skoleåret starter, hvor dette kanskje kan bidra til å redusere usikkerhet rundt hvor mye programmering man skal undervise i 1T. Det kan til og med tenkes at det vil være ekstra viktig å gi slik informasjon i god tid i startfasen av en implementering, ettersom lite erfaringer med å lære bort implementeringen gjør at lærere føler seg usikre og utilstrekkelige (Vinnervik, 2020). Informantene virker opptatt av å være godt forberedt til skoleåret, og dersom man får eksamensoppgavene tidligere kan det tenkes at det kan bidra til å redusere usikkerhet. Derimot kan tenkes at eksamen blir mer forutsigbar med årene, når det har vært utarbeidet flere eksamener man kan forholde seg til, og ikke bare eksempeloppgaver.

Videre uttrykker informant C og D en viss følelse av trygghet når det kommer til egen kompetanse, i motsetning til hva lærere har uttrykt under den tidligere implementeringen av programmering på 90-tallet, hvor det mentes at det var for vanskelig å lære bort programmering (Kafai & Burke, 2013). Tidligere var det mangel på kompetanse som var et av problemene. Informantene som uttrykker størst grad av trygghet i henhold til sin egen kompetanse er informantene som jeg oppfatter har høyest grad av kompetanse i programmering, og da snakker jeg om informant C og D. I dette tilfellet kan det tenkes at økt kompetanse har bidratt til å øke trygghetsfølelsen rundt undervisningen av programmering i 1T, i tråd med Martínez (2019) sin forskning. Informant A har det jeg oppfatter som minst kompetanse i programmering, og hen virker minst trygg til å undervise programmering i 1T, noe som står i tråd med at jo mindre kompetanse du har, jo mer engstelig blir du for implementeringen av programmering i matematikk (Martínez, 2019). Informanten forteller selv at hen ikke er trygg nok til å undervise programmering i 1T, og det er et behov for hen å bli tryggere. Dette kan peke på at det er viktig at lærerne får nok opplæring i programmering, hvor gevinsten blant annet kan bli å bidra til å redusere usikkerhet hos lærere, og få de til å føle seg tryggere.

Men, er det nødvendigvis slik at grad av trygghet skyldes erfaringer med å undervise programmering og/eller videreutdanning, eller kan andre faktorer også ha en betydning? Det kan også være relevant at de tryggeste informantene i studien jobbet på samme arbeidsplass.

Dermed kan graden av tryggheten til informant C og D kan skyldes andre faktorer enn egen kompetanse, hvor det virker som at informantene har en grad av et velfungerende kollegialt samarbeid på jobben. Dette inntrykket baserer seg på at informantene uttrykker gjentatte ganger at de føler de har et godt samarbeid på jobben med kollegaene de jobber i team med. Jeg opplever informant A og B som minst trygge til å undervise programmering i 1T, og de uttrykker ensomhet som en affektiv respons når de snakker om hvordan de føler seg når det gjelder å planlegge og undervise programmering i 1T. Informant B er den eneste læreren på jobben som underviser i 1T, mens informant A føler at møte-tiden på jobben blir brukt feil. Informantene virker minst trygge og har ikke uttrykt noen positive responser angående samarbeid på jobben, og ingen av de to informantene nevner noe betydelig kollegialt samarbeid omkring undervisningen av 1T. Ut fra datainnsamlingen kan det ikke fastslås at informantene som virker minst trygge, har et fravær av et godt kollegialt samarbeid på arbeidsplassen. Derimot kan det virke som at jo tryggere informantene er, jo mer forteller de om et godt samarbeid på jobben, og dette kan peke på at det kan være viktig å ha et godt samarbeid mellom lærerne på skolen, dersom man ønsker at lærerne skal bli tryggere og mindre usikre i å undervise programmering i 1T.

I likhet med usikkerhet er også frustrasjon en emosjonell respons, hvor synonymer til frustrasjon er blant annet ordene *skuffelse* og *oppgitthet* (Språkrådet, 2021). Frustrasjon kan oppstå som et følge av at man ikke får gjennomført det man har sett for seg (Svartdal, frustrasjon, 2021). Den affektive responsen *frustrasjon* dukket opp blant alle informantene, men det var ulike faktorer informantene var frustrert over. Informant A opplevdes som frustrert over å lære programmering samtidig som elevene skal undervises i programmering, hvor hen uttrykker blant annet at ««jeg kan ikke lære parallelt med elevene, jeg må være forberedt (...)» og at hen trenger å være kompetent i forkant, ikke i etterkant». Informant A, har i motsetning til resten av informantene, ikke hatt programmering på studiet, hatt noen tidligere kursing i dette eller tatt videreutdanning, og det kan virke som at hen har fått for lite tid til profesjonell utvikling, slik som lærerne i studien til Humble, Mozelius & Sällvin (2020) også opplevde. Informant A sine opplevelser kan bidra til å styrke bekymringene som har vært i media angående om lærerne har nok kompetanse til å undervise programmering (Fosse, 2020; Johansen, 2020; Utdanningsforbundet, 2020). Jeg tenker at det er ønskelig som lærer å ha mer kompetanse enn hva læreplanen tilsier, og ikke tilegne seg kompetanse samtidig som elevene. Det er mitt inntrykk at informant A også ønsker å kunne mer enn elevene i undervisningen sin. Det kan tenkes at frustrasjonen kan ha oppstått blant annet som et følge av at hen ikke får

gjennomført det hen ser for seg at en lærer sin jobb er, i likhet med hva Svartdal (2021) beskriver om frustrasjon.

I lys av informant A sine opplevelser omkring det som hen omtaler som «for sen» kursing kan man i tillegg argumentere for at Utdanningsdirektoratet sine uttalelser om at ansvaret for nye læreplaner skal ligge på den enkelte skole (Utdanningsdirektoratet, 2020c) kan tenke seg å være problematisk, ettersom noen skoleledere kan ta valget om å kurse lærerne sine parallelt med implementeringen. Dette kan tenkes å kunne sette lærere i en vanskelig situasjon, ettersom kan bli opp til skolelederne hvilke tidspunkter læreren skal kurses, i og med at det videre beskrives at ansvaret ikke ligger på den enkelte lærer. Dermed kan det oppstå en urettferdighet og skjev fordeling av kompetanse, hvor noen lærere har blitt sendt på kurs eller fått videreutdanning i god tid i forveien før implementeringen startet, mens andre lærere enten kurses samtidig som de skal undervise programmeringen, eller i verste fall ikke får kurs i det hele tatt og må lære seg dette ved siden av jobben. Det er viktig å få frem at dette er, ifølge (Utdanningsdirektoratet, 2020c), blant annet skoleleder sitt ansvar, og jeg tenker at skolelederne må ta ansvar for å heve kompetansen til lærerne i god tid før nye endringer skjer i skolen, hvis ikke kan det i visse tilfeller føre til negative affektive responser blant lærere, som frustrasjon.

Derimot virker informant D, og *spesielt* informant C, frustrerte over ulike tekniske problemer som oppstår når man bruker programmering i undervisningen, i likhet med Vinnervik (2020) sine lærere, og det kan virke som at informantene ikke føler seg kompetente og erfarne nok til å håndtere utfordringene. Her dukker det opp følelser som kan knyttes til mangel på kompetanse, slik som Humble, Mozelius og Sällvin (2020); Utdanningsforbundet (2020) og Johansen (2020) har uttrykt, men kanskje en annen type kompetanse enn det som har blitt beskrevet som et problem tidligere. Det krever altså både programmeringskompetanse og generell profesjonsfaglig digital kompetanse av lærerne, når de skal undervise programmering i matematikk. En konsekvens av stadige tekniske problemer i klasserommet kan føre til at lærere velger å unngå å bruke tekniske hjelpemidler i undervisningen sin, ettersom det blir vanskelig å stole på at utstyret kommer til å fungere etter planen (Vinnervik, 2020). Det er viktig at dette problemet fås frem i lyset ettersom konsekvensene kan bli at lærerne vegrer seg for, eller i verste konsekvens unngår, å implementere programmering i undervisningen grunnet frustrasjon rundt det tekniske. Informant A og B har ikke uttrykt seg omkring tekniske problemer i timen, men de har kun undervist én time med programmering i IT. Det kan være mulig at jo mer man underviser, jo mer sannsynlig er det å treffe på tekniske problemer i timen,

og at dette dermed ikke ble fokusert på i intervjuene. Eller så kan det igjen være korrelasjon mellom det faktumet at informant C og D jobber på samme arbeidsplass, og sannsynligvis i samme samarbeids-team, ettersom de begge har IT som fag dette skoleåret, og at tekniske problemer tilfeldigvis har blitt diskutert ekstra mye i teamet. Dette kan ha vært en årsak til at det ble et ekstra søkelys på tekniske problemer i intervjuene med informant C og D.

Resultater og teori kan peke på at informantene opplever negative affektive faktorer som usikkerhet og frustrasjon knyttet til å undervise programmering i IT, og noen av grunnene kan være mangelfull og sen informasjon fra Utdanningsdirektoratet omkring ny eksamensordningen i IT, manglende kompetanse omkring både programmering og tekniske problemer som oppstår under undervisningen, og manglende kollegialt samarbeid. Jeg ønsker å presisere at virkeligheten kan være mer kompleks, og at det så klart er flere faktorer som kan ha hatt en innvirkning, slik som heldigital skole, koronapandemien, fagfornyelsen, ledelsen på skolen og ulike forutsetninger rundt hva slags gruppe elever informantene hadde dette skoleåret, men at faktorene jeg har presentert var mest fremtredende og felles for informantene i studien. Derimot finnes det også grader av positive faktorer hos informant C og D, som det å føle en viss trygghet i undervisningssituasjoner når det kommer til egen kompetanse i selve programmeringen som skal gjøres i IT. Informantene som uttrykte høyest grad av trygghet hadde også høyest grad av bakgrunnskunnskaper, i likhet med Martínez (2019) sine funn, og omvendt. Dette kan tyde på at gode ressurser for opplæring for lærerne vil kunne være viktig for deres følelse av trygghet i undervisningssituasjonen og dermed tenke seg å kunne påvirke lærernes holdninger til programmering i matematikk IT. Resultater og teori kan tyde på at manglende kompetanse (og kursing) kan bidra til usikkerhet og frustrasjon, en negativ affektiv respons, hvor paralleller kan trekkes med Vinnervik (2020) sine funn, men at kompetansen ikke nødvendigvis alltid handler om selve programmeringen. Mangel på kompetanse som fører til usikkerhet kan også handle om å ikke ha nok kompetanse og erfaringer i å håndtere tekniske problemer som dukker opp underveis i undervisningen.

5.2 Hvordan underviser lærerne programmering i 1T?

Under atferdsmessig respons har jeg valgt å opprette kategorien *undervisningsatferd*. Denne kategorien handler om hvordan informantene forteller at de underviser programmering i 1T, og omtrent hvor mye tid de har brukt på å implementere programmering i 1T. Jeg stilte alle informantene spørsmål om de kunne fortelle meg om hvordan de har introdusert programmering i 1T, og ut fra slike beskrivelser fikk jeg også informasjon om omtrent hvor mye tid de har brukt på å undervise programmering frem til intervjuene ble gjort i november og desember 2020. Jeg ønsker å presisere igjen at informant A og B kun har undervist programmering i 1T én gang, og jeg vurderer dette til for erfaringer til at det blir hensiktsmessig å diskutere undervisningsatferden deres.

Informant C og D har, som tidligere beskrevet, undervist programmering i 1T jevnlig, og det kan virke som at oppleves annerledes å undervise en time med programmering enn en *klassisk* matematikktime for informantene. Informant C beskriver bruken av programmering i undervisningen sin som en mer *utradisjonell* måte å undervise på, hvor du må prøve, feile og modifisere om hverandre, i tillegg til å bruke Google som et redskap til å finne løsninger. Når hen snakker om å prøve, feile og modifisere, er vi inne på ett av de to store temaene i 1T; problemløsning. Dette kan peke mot at informanten kan tendere til å ha et mer åpent syn til matematikk og programmering, som ifølge Ernst (1989) vil kunne åpne opp for elevene kan selv utforske at det finnes flere fremgangsmåter knyttet til matematiske problemer. Samtidig forteller samme informant at det egentlig fungerer best å programmere i fellesskap, hvor læreren viser frem hvordan programmet skal skrives og elevene taster inn det samme som læreren taster, på sin egen datamaskin. Slike utsagn kan peke mot et mer lukket og instrumentelt syn på programmering i matematikk, slik som Ernst (1989) beskriver at læreren vil blant annet kunne insistere på å lære bort kun én *korrekt* måte å skrive et program på. Dette kan knyttes opp mot informantens egne tanker rundt at elevene kan gå glipp av algoritmisk tenkning ved å ha slik undervisning, men hen konkluderer med at denne type undervisning er den mest gjennomførbare dette skoleåret. Elevene trenger lite hjelp grunnet begrensede forhåndskunnskaper, og det er ikke nok kapasitet til å hjelpe mange nok elever med mindre man underviser på en slik måte. En annen utfordring som informanten trekker frem er at det tar lang tid å finne feilen til elevene, men en slik utfordring kan tenkes å bli mer overkommelig når informantene har fått mer erfaring omkring undervisning av programmering i 1T. Det kan virke som at informanten har en visjon om å benytte programmeringen til å fremme både

problemløsning og algoritmisk tenkning, men at faktorer som tid og lite erfaring med å feilsøke elevens koder kan føre til at det ofte blir undervist på en mer instrumentell måte. I dette tilfellet vil det også kunne tenke seg at elevene får mindre trening i algoritmisk tenkning og problemløsning, som kan virke som det motsatte av hva skaperne av læreplanen kanskje hadde ønsket seg i undervisningen.

Informant D er den andre informanten som har undervist programmering i 1T mer enn én gang, og forteller at hen ser frem til å drive mer utforskende i matematikk gjennom programmering. Dette kan i følge informanten skje om et par år, når elevene har mer forhåndskunnskaper, og trekker frem derivasjon som et tema for utforskende arbeid med programmering. Dette utsagnet kan gi en liten indikasjon på at informanten tenderer mot en mer åpen holdning til matematikk (Ernst, 1989), hvor utforskende arbeidsmetoder handler om å blant annet få lov til å prøve og feile og utforske selv. Jeg opplever informanten som tryggest av alle informantene når hen forteller om hvordan det er å undervise programmering i 1T, i tillegg til å virke som å være den informanten som har mest erfaring med å undervise programmering fra tidligere av. Det kan tenkes at erfaring kan ha noe å si på om du er komfortabel med å drive med utforskende undervisning ved hjelp av programmering eller ikke, i mine informanters tilfelle. Det virker også som at informant D ønsker, i likhet med informant C, at elevene skal kunne prøve seg på programmering på egenhånd. I likhet med informant C, virker det som at visjonen som oftest må utgå grunnet tidspresset i timen, hvor det blir mer effektivt å la elevene drive med samarbeidsoppgaver, får å kunne rekke å hjelpe flere elever i timen. Dette går på bekostningen av det som kan virke som visjonen til informant D, at elevene må få prøve seg på programmering på egenhånd, fordi «det er jo et håndverk», og elevene vil kanskje slappe litt for mye av dersom de er på en gruppe.

Informantene som uttrykte størst grad av trygghet var også de informantene som oppga å ha brukt mest skoletimer på å undervise programmering i 1T. Er det slik at tryggheten i seg selv fører til at informantene er mer komfortable og motiverte til å undervise per programmering, eller kan det være andre faktorer som spiller inn? Igjen kan det tenkes at et godt samarbeid på arbeidsplassen kan føre til at informantene føler seg tryggere og dermed ønsker å undervise mer i programmering. At trygge lærere velger å bruke flere matematikktimer med programmering enn utrygge lærere er bare en hypotese basert på mitt utvalg, som bør utforskes med langt høyere antall respondenter, og her er det mer behov for forskning. I tillegg kan det virke som, ut fra informantene som har fått undervist programmering jevnlig dette skoleåret, at elevene

trenger mer hjelp med programmering enn *klassisk* matematikk, og at tiden rett og slett ikke strekker til for å hjelpe alle elevene. Konsekvensen kan tenkes å bli at færre av elevene trenes i mindre opp i algoritmisk tenkning og problemløsning ettersom man må velge undervisningsmetoder som samarbeid og at elevene kopierer det læreren skriver på tavla.

Oppsummert kan det virke som at elevers begrensede forhåndskunnskaper i programmering kan påvirke hvordan informantene underviser programmeringen i 1T, hvor det virker som at tiden ikke alltid strekker til når mange elever trenger hjelp. Det kan virke som at informantenes visjoner angående ønsket undervisning ikke alltid lar seg gjennomføre, og i visse tilfeller kan det virke som at undervisningen blir mer instrumentell og mindre utforskende. Derimot kan det likevel virke som at informantene som oppfattes å ha høyere grad av trygghet i undervisningen av programmering i 1T, underviser oftere programmering i 1T enn informantene som oppfattes å ha en mindre grad av trygghet.

5.3 Hva tenker informantene rundt undervisning av programmering i 1T?

Under kognitive responser ble kategorien *nytteverdi* til, både inspirert av Martínez (2019) sin forskning, hvor hun fant ut at hvor mye programmerings-erfaring du har ikke påvirker hvor nyttig du syntes programmering i matematikk er, i tillegg til at jeg la merke til under analyseprosessen at alle informantene fortalte meg noe om nytten av programmering i 1T. Denne kategorien kan også ha blitt påvirket av at jeg stilte alle informantene spørsmål om tanker rundt programmerings plass i matematikk 1T. Martínez (2019) har brukt ordet *useful*, som jeg velger å oversette til det norske ordet *nyttig*. At noe er nyttig vil si at man kan utnytte det og bruke det til noe, og under nytteverdi valgte jeg å utforske hvorvidt informantene mine syntes det var nyttig å undervise programmering i 1T eller ikke.

Informantene i studien uttrykker at de ser nytteverdien i å undervise programmering i 1T og de kan virke positive til at programmering skal inn i matematikk 1T, i likhet med funnene til Mozelius, Ulfenborg & Persson (2019). Det uttrykkes ulike tanker rundt nytteverdien til programmering i matematikk er nyttig, som at programmering er fint å kunne i matematikk som en forberedelse til eventuelle fremtidige studier, kan brukes til utforskende arbeidsmetoder i

undervisningen og at det er positivt for elevene å lære seg algoritmisk tenkning. Informantene har ulik grad av programmeringserfaringer fra før av, men det kan virke som at dette ikke har påvirket tanker omkring nytteverdi, i likhet med funnene til Martínez (2019), hvor programmeringserfaringer ikke påvirket hvor nyttig man syntes var med programmering i matematikk. Dette kan tyde på en positiv endring fra 1990-tallet, da programmering utgikk i skolen blant annet på grunn av at lærerne ikke så nytteverdien av å skulle undervise det (Kafai & Burke, 2013). Informant A forteller, som presentert under kap. 4.3, om en endring hvor hen gikk fra å tenke at programmering bare var noe som tynget, til å begynne å se bruksområdene og tenke at det var et nyttig verktøy. Dersom et kort programmeringskurs over tre dager ikke nødvendigvis gir vesentlig økt programmeringskompetanse, og heller ikke uformelle samtaler med kollegaer, og kan man argumentere for at holdningsendringen skjedde uten noe stor form for økt kompetanse. Dermed finnes det muligheter for at informant A sine tanker fortsatt kan sammenliknes med Martínez (2019) sine funn om at programmeringskompetanse ikke har innvirkning på tanker omkring nytteverdi. Hva kan det da ha vært som førte til at informanten begynte å se på programmering som nyttig å undervise likevel? Tre av informantene i studien til Mozelius, Ulfenborg & Persson (2019) beskriver en positiv holdningsendring etter å ha blitt kurset i programmering, og kursingen informant A har vært igjennom kan dermed tenkes å ha bidratt til holdningsendringen likevel. Programmeringskurset hen deltok på kan ha fokusert på å skape positive holdninger til å undervise programmering, eller så kan de uformelle samtalene med kollegaer inneholdt noen nye innfallsvinkler. Å kunne se nytteverdien til å undervise programmering kan tenkes å ha en positiv effekt på elevene, hvor denne holdningen kanskje kan smitte over på elevene. Det er viktig at vi skaper gode holdninger hos lærerne, og dette er noe man kan ha i bakhodet når man utformer lærerkurs i programmering.

Kompetanse er den andre og siste kategorien som ble til under kognitiv respons. Utdanningsdirektoratet har, som nevnt tidligere, en egen kompetansepakke som gir en innføring i algoritmisk tenkning og programmering for lærere (Utdanningsdirektoratet, 2021), og kompetanse blir definert som;

Kompetanse er å kunne tilegne seg og anvende kunnskaper og ferdigheter til å mestre utfordringer og løse oppgaver i kjente og ukjente sammenhenger og situasjoner. Kompetanse innebærer forståelse og evne til refleksjon og kritisk tenkning. (Utdanningsdirektoratet, 2017, s. 9)

Kategorien *kompetanse* vokste frem av flere anledninger. Jeg la merke til at ordet kompetanse ble hyppig brukt av informantene under intervjuene, hvor mye av samtalene dreide seg rundt kompetanse i programmering. I tillegg har både tidligere teori og uttalelser i media fokusert en del på bekymringer rundt læreres såkalte *manglende* kompetanse i programmering. I kategorien *kompetanse* velger jeg å diskutere rundt flere definisjoner av kompetanse; informantenes kompetanse i programmering, informantenes kompetanse i å undervise programmering, informantenes kompetanse i å undervise matematikk gjennom programmering og til slutt kompetansen til elevene i henhold til programmering. Informantene sine forhåndskunnskaper i programmering har blitt presentert tidligere, og kan sees i tabell 3-1 under kapittel 3.2.

Informant B beskriver seg selv som *heldig* fordi hen har litt forhåndskunnskaper i programmering fra før av, i tillegg til å ha tatt et tre dagers kurs, hvor jeg tolker at hen tenker at det ikke er nok å bli sendt på et kort kurs for å være klar for å undervise programmering i 1T, for å oppnå nok kompetanse i programmering til å kunne undervise det i 1T. Når informant C forteller om at programmering er noe «som folk tar en master i for å få kompetanse», kan det virke som at hen tenker at programmering er en egen kompetanse i seg selv, som kan ta lang tid å oppnå. Videre kommer informant C inn på et interessant tema omkring kompetanse, hvor hen uttalte seg om at det er annerledes å lære seg programmering enn for eksempel mer klassiske matematiske temaer som geometri. Dette begrunner hen med at geometri er noe man kanskje lærte for 25 år siden, og kan lettere *grave* frem dette i hukommelsen, mens det virker som at med programmering så starter man med blanke ark. Fosse (2020), som nevnt tidligere, skrev at programmering er et nytt fagfelt, og flere har vært bekymret for om lærerne har tilstrekkelig med kompetanse for å kunne undervise programmering i matematikk (Fosse, 2020; Johansen, 2020; Utdanningsforbundet, 2020). Det finnes muligheter for at det er mer utfordrende å implementere programmering inn i læreplanen, enn andre klassiske matematiske temaer, dersom en har mindre forhåndskunnskaper om programmering. En kan argumentere for at programmering er et helt eget fagfelt, og at det er en større endring, ettersom et helt fagfelt skal inn i matematikken. På en side så kan det tenkes at ikke alle lærerne var klare for å undervise programmering i 1T dette skoleåret, og at det har vært for liten tid til å lære seg et helt nytt fagfelt. På en annen side trenger det ikke nødvendigvis være slik at man trenger kompetanse i programmering på bachelor/master-nivå, for å kunne implementere det inn i matematikk 1T.

Informant A fortalte at det er utfordrende å lære bort noe du ikke har kompetanse i selv, og siktet til programmering. Mangel på dybdekunnskap i programmering gjør at lærerne føler seg usikre og utilstrekkelige (Vinnervik, 2020). En kan argumentere for at det kan oppleves som ubehagelig å stå foran en klasse og lære bort noe du ikke er trygg på, og selv føler du ikke kan godt nok. Det kan hende at det er nærmest umulig å lære bort noe du ikke kan selv, og konsekvensene kan i dette tilfelle tenke seg å bli blant annet affektive faktorer som usikkerhet, frustrasjon og ubehag. Det virker som at det er en konsensus blant informantene at dersom man mangler kompetanse til å undervise programmering i IT, kan det bli ubehagelig å undervise de timene som inneholder programmering. Det virker også som at det kan være utfordrende å lære seg programmering ved siden av jobben, og spesielt utfordrende kan det kanskje oppleves dersom ledelsen ikke tilrettelegger for at man skal gjennomgå en kompetanseheving i god tid *før* man skal starte skoleåret med implementeringen, slik som informant A har opplevd. Derfor er det viktig at lærerne får god opplæring i programmering, hvordan undervise programmering og hvordan lære bort matematikk gjennom programmering. I tillegg kan det være viktig å tenke på å sende lærere på kurs gjerne *før* de skal undervise i programmering, slik at de kan slippe å lære parallelt med elevene.

Det kan også virke som, som nevnt tidligere av informant C, at implementeringen av programmering inn i matematikk kan være større prosjekt enn å *bare* få inn for eksempel geometri inn i matematikk, hvor man kan fiske frem gamle kunnskaper fra for eksempel lærerutdanningen. Derfor kan det være ekstra viktig at lærerne får grundig opplæring i både programmering og hvordan undervise både programmering og matematikk gjennom programmering. Det kan også tenkes at det kan finnes visse utfordringer av at ansvaret for implementeringen av nye læreplaner ligger på hver enkelt skole (Utdanningsdirektoratet, 2020c), hvor det da blir viktig at hver enkelt skole faktisk følger opp og gir lærerne en mulighet til kompetanseheving innenfor programmering. Dersom det er mer utfordrende å lære seg programmering, enn andre klassiske matematiske temaer, er det grunn til å tro at opplæringen blir enda viktigere.

Når det kommer til elevens kompetanse virker det som at flesteparten av elevene til informantene har lite erfaringer med programmering fra før de startet i IT, i likhet med bekymringene som har blitt uttrykt i mediebildet (Fosse, 2020). Gjennom informantenes utsagn kan det virke som at elevene ikke har tilstrekkelig med kompetanse for å kunne lære matematikk gjennom programmering, og informantene må drive med opplæring av programmering i IT,

fra starten av. Det kan virke som at matematikkfaget har fått hovedansvaret for opplæringen av programmering, slik som Tellefsen (2021) beskriver det. Hos informant B blir konsekvensen at det blir en stor overgang for elevene, når de ikke har sett programmering tidligere på ungdomsskolen. Derimot virker det som at det er et slags håp hos informantene om at elevene skal ha økte forhåndskunnskaper om et par år. Dette håpet kan tenke seg å skyldes at programmering skal inn i grunnskolen også, hvor informantene håper på at elevene har opparbeidet seg forhåndskunnskaper tidligere i grunnskolen, før elevene starter på videregående. Det kan tenkes at lærerne kan frigjøre tiden de nå bruker på opplæring av programmering, og at denne tiden kanskje kan brukes til å fokusere på å lære matematikk gjennom programmering. Dessverre finnes det muligheter for at elevene er engstelige for implementeringen av programmering inn i 1T, ettersom det virker som at de har lite forhåndskunnskaper, slik som Martínez (2019) sine funn viste. Forhåpentligvis vil denne mulige engstelsen bli mindre og mindre, jo mer opplæring elevene får i programmering fra grunnskolen av.

Når det kommer til informantenes tanker omkring undervisning av programmering i 1T, kan resultater og teori peke på at informantene ser nytteverdien av å undervise programmering, men at det er noen utfordringer knyttet til manglende kompetanse blant både lærere og elever. Det kan virke som at det er en konsensus blant informantene at det kanskje skal litt mer til for å oppnå kompetanse i programmering enn et kort kurs, og at programmering er noe større enn å *bare* få inn mer klassiske matematiske temaer inn i matematikkfaget. I tillegg virker det som at elevens begrensede forhåndskunnskaper kan føre til at informantenes visjoner angående undervisning ikke blir oppfylt i like stor grad som de ønsker selv.

5.4 Lærernes uttrykte holdninger rundt undervisning av programmering i 1T

Informantenes holdninger til å undervise programmering i teoretisk matematikk kan se ut til å være ambivalente, og dette ser ut til å påvirke informantenes visjoner og handlinger (*se figur 5-1*). Som nevnt tidligere i delkapittel 2.1, så vil en ambivalent holdning bety at følelser, handlinger og tanker kan være motstridende (Zeigler-Hill & Shackelford, 2020). På en side virker det som at informantene ser nytteverdien av å undervise programmering i 1T, men på en annen side oppstår det utfordringer som kan knyttes til ny eksamensform i 1T, og manglende kompetanse både lærere og elever, og dette ser ut til å påvirke atferden til informantene. Det kan virke som at informantene ikke får utført sin visjon i like stor grad som de selv ønsker, når det kommer til å undervise programmering i teoretisk matematikk. Noen informanter er til og med avventende når det kommer til å undervise matematikk gjennom programmering. Det å være ubesluttet er en mulig konsekvens av å ha ambivalente holdninger (Zeigler-Hill & Shackelford, 2020), og ettersom lærernes holdninger kan smitte over på elevenes holdninger (Goodykoontz, 2008) kan det tenkes at elever også kan sitte igjen med slike erfaringer. Dersom læreren er avventende med programmeringen, lærer kanskje elevene mindre programmering enn det som er ønskelig ut fra læreplanen.

Informant C er innforstått med at algoritmisk tenkning og programmering er en av kompetansene elevene skal opparbeide seg i teoretisk matematikk, men kan se ut til at den ambivalente holdningen fører til at det blir utfordrende å gjennomføre. Det er mulig det ikke er nok med tid til å drive opplæring i en slik type kompetanse, og en konsekvens kan bli at informantene ikke implementerer læreplanen slik som skaperne av læreplanen har sett for seg at det skal bli (Wang & Cheng, 2009). Det er mulig at lærerne trenger mer tid, støttestrukturer og opplæring, og følgende eksempel uttrykker informant C en oppsummering angående hva som kan ha gått galt.

Informant C: Jeg tror ikke de [Utdanningsdirektoratet] har tatt helt høyde for det [tekniske utfordringer i timen], eller jeg vet ikke jeg, de har jo innført noe ingen mattelærere kan, det er jo ganske interessant i seg selv (latter), først må alle oppdatere seg på programmering og så skal man ha det tekniske i tillegg.

Derimot kan det virke som at informantene er positive til fremtiden. Det ser ut til å være et slags håp blant lærerne om at elevers forhåndskompetanse kommer til å økes med tiden, og at dette kan føre med seg flere ulike gevinster i fremtidige undervisningssituasjoner.

6. Avsluttende refleksjoner og videre forskning

Informantenes holdninger til å undervise programmering i teoretisk matematikk kan se ut til å være ambivalente. På en side så virker det som at informantene ser nytteverdien av å undervise programmering i 1T, men utfordringer og grunner til frustrasjon og usikkerhet kan se ut til å påvirke atferden til informantene. Usikkerheten kan knyttes opp mot ny eksamensform i 1T, manglende kompetanse blant både lærere og elever og lite kollegialt samarbeid omkring implementeringen. Samtlige informanter uttrykkes positive affektive faktorer som grad av trygghet, hvor kompetanse og kollegialt samarbeid kan se ut til å ha spilt en rolle. Frustrasjonen informantene uttrykker handler om blant annet for sen kursing i programmering og utfordringer omkring tekniske problemer som kan oppstå i timene. Derimot virker det som at informantene delte et felles syn om at programmering i 1T var nyttig, men at noen utfordringer pekte på manglende kompetanse som er tidkrevende å oppnå. I tillegg kan det virke som at flertallet av informantenes elever har lite forhåndskunnskaper i programmering, som kan gjøre opplæringen av programmeringen både tidkrevende og sette begrensninger i undervisningen. Det kan virke som at usikkerheten, frustrasjonen og de presenterte utfordringene er det som kan påvirke atferden til informantene. Det finnes muligheter for at informantene ikke har nok ressurser til å i best mulig grad gi elevene mulighet til å trene seg i problemløsning og algoritmisk tenkning. Informantene får kanskje ikke får gjennomført det de mener er mest egnet i undervisningen av programmering i 1T, til tross for at informantene tenker at programmering er nyttig å undervise i 1T. Derimot virker det som at informantene er positive til fremtiden, med et håp om at elevers forhåndskompetanse kan komme til å øke med tiden.

Denne studien har for få informanter til at funnene kan generaliseres, men likevel mener jeg at jeg har bidratt med nyttige funn hvor deler kan være overførbare til andre lærere og liknende situasjoner. Jeg mener at skoleledere bør være oppmerksomme på å gi lærerne nok tid til kompetanseheving, for å unngå mulige konsekvenser som at lærerne unngår å undervise programmering. Jeg tenker også at det kan være nyttig for kursholdere i programmering å tenke

på at tekniske problemer kan gjøre lærerne frustrerte, og tenke over hvordan man kan øke en slik type kompetanse hos lærerne som skal undervise programmering i IT. I tillegg kan det virke som at et godt samarbeid på arbeidsplassen kan styrke tryggheten blant lærere når det kommer til innføringen av nye læreplaner, hvor skoleledere bør reflektere rundt hvordan samarbeidet fungerer for lærerne deres.

I videre studier kunne det vært interessant å utforske videre om programmeringserfaringen til læreren henger sammen med hvor mye av undervisningstiden læreren bruker på programmering i IT. Her kunne man også sett spesifikt på om lærere med lite erfaring bruker lite av undervisningstiden på programmering. Det kunne også vært interessant å undersøke om det er flere lærere som tenderer mot ambivalente holdninger til å undervise programmering i IT. Avslutningsvis hadde man også kunnet undersøkt om mangel på tid i undervisningen gjør at lærere ikke rekker å undervise elevene i algoritmisk tenkning og problemløsning i like stor grad som de ønsker.

Referanser

- Anker, T. (2020). *Analyse i praksis*. Oslo: Cappelen Damm Akademisk.
- Balanskat, A., & Engelhardt, K. (2014). *Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Brussel: European Schoolnet (EUN Partnership AISBL).
- Bostrøm, E., Bø, O., Langmyhr, D., & Rydland, T. (2008). *Informasjonsteknologifaget og skoleverket – en bakgrunn og handlingsplan for Norsk Informatikkråd*. Halden: Høgskolen i Østfold.
- Dalland, O. (2014). *Metode og oppgaveskriving*. Oslo: Gyldendal Akademisk.
- Det Norske Akademi for Språk og Litteratur. (2021). *Usikkerhet*. Hentet fra naob: <https://naob.no/ordbok/usikkerhet>
- Ernst, P. (1989). The knowledge, beliefs and attitudes of the mathematics teacher: A model. *Journal of education for teaching*, 15(1), ss. 13-33.
- Fosse, D.-H. (2020, September 26). *Lærere med hovedfag klarer ikke løse nye eksamensoppgaver for videregående i matematikk*. Hentet April 2021 fra aftenbladet: <https://www.aftenbladet.no/lokalt/i/41d8aR/laerere-med-hovedfag-klarere-ikke-loese-nye-eksamensoppgaver-for-videregaa>
- Goodykoontz, E. (2008). Factors that affect college student's attitudes toward mathematics. *The Research Repository at WVU*.
- Hargreaves, A. (2005). Educational change takes ages: Life, career and generational factors in teachers' emotional responses to educational change. *Elsevier*, ss. 968-983.
- Humble, N., Mozelius, P., & Sällvin, L. (2020, Juli 24). Remaking and reinforcing mathematics and technology with programming – teacher perceptions of challenges, opportunities and tools in K-12 settings. *The International Journal of Information and Learning Technology*.
- Johansen, A.-K. (2020, 07 11). *Programmering vil bli en utfordring for lærere*. Hentet fra forskning.no: <https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold-matematikk/programmering-vil-bli-en-utfordring-for-laerere/1711838>
- Kafai, Y. B., & Burke, Q. (2013, September 1). Computer programming goes back to school. *SAGE Journals*, ss. 61-65.
- Kaufmann, O., & Stenseth, B. (2020, Mars 25). Programming in mathematics education. *International Journal of mathematical education in science and technology*, ss. 1-20.
- Kennedy, C., & Kennedy, J. (1996). Teacher attitudes and change implementation. *Elsevier Science Ltd*, ss. 351-360.
- Kunnskapsdepartementet. (2014). *Elevenes læring i fremtidens skole*. Oslo: Norges offentlige utredninger.
- Kvale, S., & Brinkmann, S. (2015). *Det kvalitative forskningsintervju*. Gyldendal akademisk.
- Martínez, A. F. (2019, November). Gender and prior knowledge factors in pupil's beliefs about programming in mathematics. *The International Academy of Technology, Education and Development*.
- Misfeldt, M., & Ejsing-Duun, S. (2016, Mars 16). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. *HAL*, ss. 2520-2530.
- MIT media lab. (2016, August 1). *Professor Emeritus Seymour Papert, pioneer of constructionist learning, dies at 88*. Hentet Mai 2021 fra news.mit: <https://news.mit.edu/2016/seymour-papert-pioneer-of-constructionist-learning-dies-0801>

- Mozelius, P., Ulfenborg, M., & Persson, N. (2019, Mars). Teacher attitudes towards the integration of programming in middle school mathematics. *International Technology, Education and Development Conference*, ss. 701-706.
- Papert, S. (1980). *Mindstorms: children, computers and powerful ideas*. The United States of America: Basic Books, Inc.
- Postholm, M. B. (2005). *Kvalitativ metode*. Oslo: Universitetsforlaget.
- Postholm, M., & Jacobsen, D. (2018). *Forskningsmetode for masterstudenter i lærerutdanning*. Oslo: Cappelen Damm AS.
- Raaheim, A. (2019). *Sosialpsykologi*. Bergen: Fagbokforlaget.
- Regjeringen. (2021, Februar 08). *Regjeringen avlyser skriftlig eksamen, men gjennomfører muntlig eksamen for avgangselevne*. Hentet Mars 2021 fra regjeringen: <https://www.regjeringen.no/no/aktuelt/regjeringen-avlyser-skriftlig-eksamen-men-gjennomforer-muntlig-eksamen-for-avgangselevne/id2832039/>
- Sevik, K. (2016). *Programmering i skolen*. Hentet fra Utdanningsdirektoratet: https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Språkrådet. (2021). *Bokmålsordboka*. Hentet fra ordbok.uib: https://ordbok.uib.no/perl/ordbok.cgi?OPP=frustrasjon&ant_bokmaal=5&ant_nynorsk=5&bokmaal=+&ordbok=bokmaal
- Stephens, M., & Kadjevich, D. M. (2020). Computational/Algorithmic Thinking. I S. Lerman, *Encyclopedia of Mathematics Education* (ss. 117-123). London: Springer.
- Svartdal, F. (2019, Mai 20). *Problemløsning*. Hentet November 11, 2020 fra Store Norske Leksikon: <https://snl.no/probleml%C3%B8sning>
- Svartdal, F. (2021, Mars 24). *frustrasjon*. Hentet Mai 2021 fra snl: <https://snl.no/frustrasjon>
- Tellefsen, C. W. (2021, Mars 10). *Realfaglig programmering*. Hentet Mai 2021 fra Det utdanningsvitenskapelige fakultet, Universitetet i Oslo: <https://www.uv.uio.no/forskning/satsinger/fiks/kunnskapsbase/realfaglig-programmering/realfaglig-programmering---en-innledning/>
- Tranøy, K., & Tjønneland, E. (2020, Juli 18). *analyse*. Hentet 02 2021 fra Store Norske Leksikon: <https://snl.no/analyse>
- Utdanningsdirektoratet. (2017, September 1). *Overordnet del – verdier og prinsipper for grunnopplæringen*. Hentet Mai 2021 fra udir: <https://www.udir.no/lk20/overordnet-del-samlet/>
- Utdanningsdirektoratet. (2020a, August 21). *Læreplan i teoretisk matematikk (MATH09-01)*. Hentet Januar 2021 fra udir: <https://www.udir.no/lk20/mat09-01/kompetansemaal-og-vurdering/kv42?lang=nob>
- Utdanningsdirektoratet. (2020b, November 10). *Hva er fagfornyelsen?* Hentet fra Udir: <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/nye-lareplaner-i-skolen/>
- Utdanningsdirektoratet. (2020c, Desember 8). *Hvordan ta i bruk nye læreplaner?* Hentet Mars 2021 fra udir: <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvordan-ta-i-bruk-lareplanen/#>
- Utdanningsdirektoratet. (2021, Mars 1). *Kompetansepakke for programmering og algoritmisk tenkning*. Hentet Mars 2021 fra udir: <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/kompetansepakke-for-programmering-og-algoritmisk-tenkning/>
- Utdanningsforbundet. (2020, November 20). *Lærerne bekymret for nye eksamensoppgaver i videregående*. Hentet fra utdanningsforbundet: <https://www.utdanningsforbundet.no/nyheter/2020/larerne-bekymret-for-nye-eksamensoppgaver-i-videregaende/>

- Vinnervik, P. (2020, Juni 6). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *Springer*.
- Wang, H., & Cheng, L. (2009). Factors Affecting Teacher's Curriculum Implementation. *The Linguistics Journal*, ss. 135-157.
- Wikimedia Commons. (2016, Januar 24). *File:Seymour Papert.jpg*. Hentet Mai 2021 fra commons.wikimedia: https://commons.wikimedia.org/wiki/File:Seymour_Papert.jpg
- Wilson, T. D. (2000, Februar). A Model of Dual Attitudes. *Psychological Review*, ss. 101-126.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM (Association for Computing Machinery)*, 33-35.
- Wæhle, E., Dahlum, S., & Grønmo, S. (2020, 05 14). *case-studie*. Hentet 02 2021 fra Store Norske Leksikon: <https://snl.no/case-studie>
- Yin, R. K. (2013, Juli 10). Validity and generalization in future case study evaluations. *Sage*, ss. 321-332.
- Zeigler-Hill, V., & Shackelford, T. K. (2020). *Encyclopedia of Personality and Individual Differences*. Cham: Springer.

Vedlegg

Vedlegg 1: Informasjonsskriv

Vil du delta i forskningsprosjektet

«*Matematikklærere og programmering: holdninger*»?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke holdninger hos matematikklærere rundt programmering i matematikk. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Jeg skal skrive en kvalitativ master om tanker matematikklærere har på videregående skoler rundt programmering, hvor ønsket er å komme fram til muligheter og utfordringer rundt dette. Jeg lurer på hvordan lærerne har hatt det med denne pågående integreringen av fagfornyelsen.

Vi skal hovedsakelig snakke om hvilke følelser og tanker dere har rundt temaet, og hvordan dere selv syntes at det har gått dette skoleåret. Opplysningene skal kun brukes til min masteroppgave.

Hvem er ansvarlig for forskningsprosjektet?

Fakultetet for Realfag og Teknologi ved Norges Miljø- og Biovitenskapelige Universitet er ansvarlig for prosjektet, og masterstudent Hana Zukanovic skal skrive oppgaven. Min hovedveileder heter Marte Bråtalen, og er universitetslektor ved seksjon for Læring og Lærerutdanning. I tillegg har jeg en biveileder ved navn Morten Munthe, som skriver en doktorgrad i læring av matematikk ved hjelp av programmering.

Hvorfor får du spørsmål om å delta?

Du jobber som matematikklærer på videregående skole. Det er ønskelig at det blir mest mulig spredning, og derfor spør jeg ulike skoler fra ulike områder. Jeg har først sendt ut mail til avdelingslederen din, og fått tillatelse om å spørre videre.

Hva innebærer det for deg å delta?

Hvis du velger å delta i prosjektet, innebærer det et semistrukturert intervju på ca. 30 min. Det vil si at jeg har noen spørsmål på forhånd som vi kan ta utgangspunkt i, men at jeg er åpen for temaer som ikke har blitt introdusert tidligere. Jeg kommer til å bruke lydopptak under intervjuet, og intervjuet vil foregå på din arbeidsplass/på Zoom.

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. Send meg en epost; hana.zukanovic@nmbu.no om det skulle være noe.

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. Opptaket fra intervjuet vil bli lagret på en sikker server, hvor kun jeg, hovedveilederen og biveilederen har tilgang. Transkripsjonen av intervjuet kommer til å bli anonymisert, og jeg ser ikke på kjønn, skole, kommune etc. Den eneste opplysningen jeg lagrer er alder, men den vil jeg erstatte med en kode som lagres på en egen liste adskilt fra øvrige data. I tillegg kommer ikke eksakt alder til å bli nevnt, men det erstattes med tiåret ditt (f.eks. at du er i 30-årene). Du vil ikke kunne bli gjenkjent i publikasjonen.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene slettes når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er 01.07.2022. Da vil intervjuopptakene slettes fra skyen, og listen med alder destrueres.

Dine rettigheter

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

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

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

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

Hvor kan jeg finne ut mer?

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

- Norges Miljø- og Biovitenskapelige Universitet ved hovedveileder Marte Bråtalen (telefonnummer: +47 67 23 15 71 eller mail: marte.bratalien@nmbu.no)
- Norges Miljø- og Biovitenskapelige Universitet ved masterstudent Hana Zukanovic (telefonnummer: +47 90 23 75 48 eller mail: hana.zukanovic@nmbu.no)
- Vårt personvernombud: Hanne Pernille Gulbrandsen (telefonnummer: +47 40 28 15 58 eller mail: personvernombud@nmbu.no)

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

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

Med vennlig hilsen

Marte Bråtalen
(Hovedveileder)

Hana Zukanovic
(Masterstudent)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «*Matematikklærere og programmering: holdninger*» og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i intervju
- at det kan tas lydopptak av intervjuet

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

(Signert av prosjektdeltaker, dato)

Vedlegg 2: Intervjuguide

Intervjuguide

Dette blir et semistrukturert intervju, som betyr at ikke alle spørsmål blir fastsatt på forhånd.

Likevel er det viktig å ha noen spørsmål som utgangspunkt.

Tidsramme: 30 min

Fase 1: Rammesetting (5 min)

- uformell prat: hvor lenge har du vært lærer og hvilke fag har du i år?
- signering av informasjonsskrivet som har blitt sendt ut på mail til de på forhånd (om det blir på Zoom signerer skjemaet digitalt før intervjuet)
- Informere om lydopptak og starte opptak

Fase 2: Semistrukturert intervjusamtale (20 min)

En kan dele inn spørsmålene i tre ulike typer; spørsmål i intervjuguiden, oppfølgingsspørsmål og inngående spørsmål (Postholm & Jacobsen, 2018). Spørsmålene i intervjuguiden er utarbeidet på forhånd, og i dette tilfellet trenger ikke alle å stilles, og rekkefølgen kan være tilfeldig. Oppfølgingsspørsmål er knyttet direkte opp mot det deltakeren forteller, og hensikten er å kunne gå i dybden eller få frem nyanser av det som har blitt sagt. Inngående spørsmål kan kun stilles der og da i intervjuet, i motsetning til oppfølgingsspørsmål, som kan stilles i et eventuelt oppfølgingsintervju. Formålet med inngående spørsmål er i hovedsak å få deltakeren til å snakke mer om noe spesifikt som de har fortalt.

Spørsmål som er mulige å stille:

Spørsmål	Kommentar
----------	-----------

Hva slags erfaringer hadde du med programmering i matematikk før fagfornyelsen?	Sjekk utgangspunktet deres, om de f.eks. har fått et kurs gjennom jobben.
Kan du fortelle om hvordan du har introdusert programmering inn i 1T?	Få fram atferdsmessig respons (Raaheim, 2019).
Hvordan oppleves det for deg å undervise?	Affektiv respons (Raaheim, 2019).
Hva tenker du rundt eksamensoppgavene i 1T?	Få fram tanker rundt læreplan -> eksamensoppgaver -> undervisning som skal passe inn med en eventuell eksamen (Wang & Cheng, 2009, Vinnervik 2020?)
Hva slags tanker har du rundt å undervise programmering i matematikk?	Få fram kognitiv respons (Raaheim, 2019)
Hvordan oppfatter du at lærerne på skolen forholder seg til å undervise programmering?	Få fram sosiale aspekter, individer påvirkes ofte av gruppen de tilhører i (Raaheim, 2019)
Har du hatt muligheten til å få hjelp av jobben eller lignende med hvordan du skal lære bort programmering i matematikk? / hva har arbeidsplassen din gjort for at du skal kunne lære bort programmering i matematikk?	Finne ut av hvor mye jobben har tilrettelagt for at de skal bli trygge i programmering (programmering utgikk på 90-tallet på grunn av mangelfull kompetanse blant lærere (Kafai & Burke, 2013) en stor andel lærere har mangelfull kompetanse i programmering (Utdanningsforbundet, 2020).
Hva er dine egne forventinger/mål når du underviser programmering i matematikk?	Hvordan er selvtilliten til læreren når det kommer til å mestre programmering i matematikk? Individets selvtillit kan ha en påvirkning på hvordan de opplever situasjoner, hvilke evner og muligheter de selv føler at de har, og dette kan igjen ha en innvirkning på holdningene til personen (Raaheim, 2019)
Kan du nevne positive og negative sider ved å undervise matematikk gjennom programmering?	Få fram positive og negative holdninger (Raaheim, 2019).

Hvordan forbereder du deg til undervisning? Hvordan er det å forberede en matematikk-time med programmering vs. en uten?	Få fram atferdsmessig respons (Raaheim, 2019)
Har lærerrollen din endret seg når du lærer bort programmering?	Lar de elevene samarbeide mer, feilsøking, mer som en støttespiller enn en foreleser? Lærerens rolle kan være avgjørende når man skal lære bort matematikk gjennom programmering (Misfeldt & Ejsing-Duun, 2016).
Hvordan opplever du endringer i skolen generelt?	Affektiv respons til endringer (Hargreaves, 2005).
I hvor stor grad har programmering noe å gjøre i matematikkfaget? / tanker rundt programmeringens plass i matematikk 1T	Har læreren åpen eller lukket holdning til programmering? (Ernst, 1989)

I tillegg kan jeg stille oppfølgingsspørsmål:

- Kunne du sammenliknet ___ med ___?
- Skjer det alltid på denne måten? (dersom deltakeren kommer med generaliseringer)
- Hvorfor tror du at du tenker dette?
- Hva får det deg til å føle? (få tak i følelser)

Jeg kan også stille følgende inngående spørsmål:

- Kan du gi meg et eksempel på dette?
- Har du flere eksempler på dette?
- Kan du fortelle mer om dette?
- Hvordan reagerte du da ...? (få tak i følelser)
- Hva følte du da ...? (få tak i følelser)

Fase 3: Oppsummering (5 min)

- Oppsummere samtalen
- Spørre deltakeren om jeg har forstått de riktig og om det er noe de ønsker å legge til

- Spørre om alderen til deltakeren
- Dersom deltakeren har mer å komme med er det mulig å avtale et oppfølgingsintervju

Vedlegg 3: Godkjenning fra NSD til behandling av personopplysninger

21.1.2021

Meldeskjema for behandling av personopplysninger



NSD sin vurdering

Prosjekttittel

Matematikkfag og programmering: holdninger

Referansenummer

484688

Registrert

22.09.2020 av Hana Zukanovic - hana.zukanovic@nmbu.no

Behandlingsansvarlig institusjon

Norges miljø- og biovitenskapelige universitet - NMBU / Fakultet for realfag og teknologi

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Marte Bråtalen, marte.bratalien@nmbu.no, tlf: 67231571

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Hana Zukanovic

Prosjektperiode

05.10.2020 - 01.07.2022

Status

28.09.2020 - Vurdert

Vurdering (2)

28.09.2020 - Vurdert

Vi viser til mottatt revidert informasjonsskriv. Dato for prosjektslutt er korrigert, og vi legger til grunn for vår vurdering at prosjektet avsluttes og datamaterialet anonymiseres innen 01.07.2022.

Lykke til med prosjektet!

25.09.2020 - Vurdert med vilkår

<https://meldeskjema.nsd.no/vurdering/5f63410b-310b-47a9-9c0b-8708f6183ah5>

1/3

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 25.09.2020, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

DEL PROSJEKTET MED PROSJEKTANSVARLIG (la stå ved studentprosjekt).

Det er obligatorisk for studenter å dele meldeskjemaet med prosjektansvarlig (veileder). Det gjøres ved å trykke på "Del prosjekt" i meldeskjemaet.

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:

https://nsd.no/personvernombud/meld_prosjekt/meld_endringer.html

Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET

Ifølge informasjonsskrivet til utvalget i studien vil prosjektet behandle alminnelige kategorier av personopplysninger frem til juli 2021. Dette er imidlertid ikke i samsvar med informasjonen som er gitt i fanen 'varighet' i meldeskjema, hvor 01.07.2022 er oppført som dato for prosjektslutt. Vi legger til grunn for vår vurdering den datoen utvalget får opplyst, og ber om at informasjonen meldeskjema korrigeres. Dersom 01.07.2022 er den korrekte datoen for prosjektslutt, må dette korrigeres i informasjonsskrivet og revidert informasjonsskriv må lastes opp i meldeskjema.

Når de nødvendige korrigeringsene er gjort, returneres informasjonen til oss ved at du går til fanen 'send inn' og trykker 'fullfør'.

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: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), 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).

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

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

Tlf. Personverntjenester: 55 58 21 17 (tast 1)



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway