Høgskulen på Vestlandet

Matematikk 3, emne 4 - Masteroppgave

MGUMA550-O-2023-VÅR2-FLOWassign

Predefinert informasjon

Startdato:	02-05-2023 09:00 CEST	**Termin:**	2023 VÅR2
Sluttdato:	15-05-2023 14:00 CEST	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	Masteroppgave - Bergen		
Flowkode:	203 MGUMA550 1 O 2023 VÅR2		
Intern sensor:	(Anonymisert)		

Deltaker

Kandidatnr.:	213

Informasjon fra deltaker

Antall ord *:	25023

Egenerklæring *: Ja

Jeg bekrefter at jeg har registrert oppgavetittelen på norsk og engelsk i StudentWeb og vet at denne vil stå på vitnemålet mitt *: Ja

Jeg godkjenner avtalen om publisering av masteroppgaven min *

Ja

Er masteroppgaven skrevet som del av et større forskningsprosjekt ved HVL? *

Ja, LATACME

Er masteroppgaven skrevet ved bedrift/virksomhet i næringsliv eller offentlig sektor? *

Nei

Høgskulen på Vestlandet

MASTEROPPGAVE

Matematikklærere og programmering på mellomtrinnet

Mathematics teachers and programming in upper elementary

Kasper Løvlie

Master i undervisningsvitenskap med fordypning i matematikk

Fakultetet for lærerutdanning, kultur og idrett (FLKI)

15. mai 2023

Veileder: Inge Olav Hauge

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, *jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.*

Sammendrag

I 2020 ble den nye læreplanen iverksatt. En av de store endringene som kom med denne læreplanen er innføringen av algoritmisk tenkning og programmering som en del av kjerneelementene i matematikk. Ettersom dette er et nytt fenomen i denne delen av skolen, er det gjort lite forskning på lærernes tanker om dette. Denne studien tar derfor for seg hva et utvalg matematikklærere på mellomtrinnet sier om programmering i matematikkfaget. Det er utviklet to forskningsspørsmål for å hjelpe og besvare dette:

1)*Hvilke matematiske kompetanser beskriver lærerne i elevenes arbeid med programmering?*
2)Hvilke aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering?

Grunnlaget for datamaterialet i mitt forskningsprosjekt er kvalitative intervju med fire matematikklærere på mellomtrinnet. Intervjuene består av spørsmål rundt programmering i matematikkundervisningen og matematisk kompetanse, og skal være til hjelp for å besvare studiens forskningsspørsmål. Oppgavens rammeverk er basert på et utvalg av Niss og Højgaard Jensens (2002) matematiske kompetanser og en modell hentet av den algoritmiske tenkeren fra Utdanningsdirektoratet (2019a) sine sider, som er basert på BareFoot Computing (2016) sin modell med nøkkelbegrep og arbeidsmåter hos den algoritmiske tenkeren.

Gjennom analyse og diskusjon kommer det frem i studien at lærerne er bevisste på hvordan de arbeider med programmering i undervisningen. Man kan i stor grad se at alle lærerne beskriver de utvalgte kompetansene gjennom intervjuene, med få unntak. Når det kommer til begrepet algoritmisk tenkning, er det større variasjoner hos lærerne. Noen av lærerne framstår sikre når de nevner bortimot alle aspektene ved algoritmisk tenkning, mens andre virker mer usikre. Oppgaven konkluderer derfor med at lærerne i stor grad virker å ha kontroll på programmering i sin undervisning, men samtidig vil det være hensiktsmessig at lærerne setter seg inn i algoritmisk tenkning for å bedre kunne tilrettelegge en undervisning som tillater elevene å arbeide med dette.

Abstract

In 2020, the new curriculum was implemented. One of the major changes introduced in this curriculum was the inclusion of computational thinking and programming as a part of the core elements in mathematics. As this is a new phenomenon in this part of schooling, there has been limited research on teachers' perspectives on this topic. Therefore, this study focuses on what a selected group of primary school mathematics teachers say about programming in the mathematics subject. Two research questions have been developed to help address this:

1) What mathematical competencies do describe in students work with programming?
2) What aspects of computational thinking do teachers highlight in their discussions about programming?

The basis for the data in my research project consists of qualitative interviews with four primary school mathematics teachers. The interviews include questions about programming in mathematics teaching and mathematical competency and are intended to help answer the research question of the study. The framework of the study is based on a selection of mathematical competencies by Niss & Højgaard Jensen (2002) and a model derived from the computational thinker provided by the Norwegian Directorate for Education and Training (2019a), which is based on the BareFoot Computing (2016) model of key concepts and approaches of the computational thinker.

Through analysis and discussion, the study reveals that teachers are conscious of how the work with programming in their teaching. It is evident that the selected competencies are largely described by all of the teachers in the interviews, with few exceptions. However, when it comes to the concept of computational thinking, there are greater variations among the teachers. Some teachers appear more confident as they mention nearly all aspects of computational thinking, while others seem more uncertain. Therefore, the study concludes that teachers largely seem to have control over programming in their teaching, but it would be beneficial for them to familiarize themselves with computational thinking to better facilitate instruction that allows students to engage with it.

Forord

Denne oppgaven markerer at min tid som student nå er forbi. Jeg har gjennom fem år på Høgskulen på Vestlandet lært mye av både dyktige forelesere og ivrige medstudenter. Jeg har gjennom studiet blitt bedre kjent med meg selv som person og som lærer, og gjennom studiet har jeg fått flere vennskap som jeg er sikker på kommer til å vare livet ut. Det å kunne skrive en masteroppgave om noe som jeg både interesserer meg for, og ønsker å lære mer om, er noe jeg setter stor pris på. I perioder var det mye arbeid å få unnagjort, men det var også perioder der jeg fikk møtt alle de bra menneskene på studiet. Uten disse menneskene hadde ikke studietiden vært den samme.

Jeg har gjennom arbeidet med masteroppgaven fått hjelp av flere personer som jeg ønsker å tildele en takk. Jeg vil først takke de fire lærerne som var villige til å bli intervjuet. Jeg setter stor pris på kjapp tilbakemelding og at de satt av en liten time til meg. Jeg vil også takke min veileder Inge Olav Hauge for veiledning og tips når jeg satt fast, og ikke minst korrekturlesning av hele oppgaven.

Videre vil jeg også takke familien for støtte og oppmuntrende ord i perioder der oppgaven har sett uoverkommelig ut. Jeg setter også stor pris på korrekturlesningen som både mamma, pappa, bror og søster har bidratt med. Sist, men ikke minst, vil jeg takke alle mine venner gjennom studiet. Takk for tiden både på og utenfor skolen. Dere har gjort disse årene uforglemmelige.

Kasper Løvlie

Bergen, 15. mai 2023

Innhold

Sammendrag 2

Abstract 3

Forord 4

1. Innledning 9

1.1 Bakgrunn for valg av tema 9

1.1.1 Reform 97 10

1.1.2 Læreplanen, 2006 11

1.1.3 Ludvigsens-utvalget 11

1.1.4 Læreplanen, 2020 12

1.1.5 Utforsking og problemløsning 13

1.2 Oppgavens mål og problemstilling 14

1.3 Programmering i matematikk 16

1.4 Forskningsprosjektet LATACME 17

1.5 Oppgavens disposisjon 18

2 Teori 19

2.1 Matematisk kompetanse 19

2.1.1 Niss & Højgaard Jensen (2002) – Matematiske kompetanser 20

2.2 Algoritmisk tenkning 23

2.2.1. Utdanningsdirektoratet (2020) – Den algoritmiske tenkeren 24

2.3 Tidligere forskning på programmering i matematikkundervisningen 28

2.4 Sammenheng mellom Niss & Højgaard Jensens (2002) utvalgte matematiske kompetanser og Utdanningsdirektoratets modell av Den algoritmiske tenkeren (2019a) . 30

3 Metode 32

3.1 Valg av metode 32

3.2 Intervju 33

3.3 Datainnsamlingen 33
3.3.1 Zoom 34
3.3.2 Pilotintervju 34
3.3.3 Intervjuguide 35
3.3.4 Transkribering 36
3.4 Analyse av datamaterialet 36
3.4.1 Meningsfortetting 36
3.4.2 Meningsfortolkning 38
3.4.3 Analysen 39
3.5 Validitet og Reliabilitet 41
3.6 Etiske hensyn 42
4 Analyse og resultater 43
4.1 Lærer 1 43
4.1.1 Analyse av matematisk kompetanse i intervjuet med Lærer 1 44
4.1.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 1 45
4.2 Lærer 2 48
4.2.1 Beskrivelser av matematisk kompetanse i intervjuet med Lærer 2 48
4.2.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 2 49
4.3 Lærer 3 52
4.3.1 Beskrivelser av matematisk kompetanse i intervjuet med Lærer 3 52
4.3.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 3 55
4.4 Lærer 4 60
4.4.1 Beskrivelser av matematisk kompetanse i intervjuet med Lærer 4 60
4.4.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 4 63
4.5 Oppsummering av analysen 67
5 Diskusjon 68

5.1 Matematiske kompetanser som lærerne beskriver i samtale om programmering .. 69
5.1.1 Problembehandlingskompetansen 69
5.1.2 Resonnementskompetansen 70
5.1.3 Kommunikasjonskompetansen 71
5.2 Aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering 72
5.2.1 Nøkkelbegrep hos den algoritmiske tenkeren 73
5.2.2 Arbeidsmåtene til den algoritmiske tenkeren 75
6 Avslutning 77
6.1 Konklusjon 77
6.2 Begrensninger 78
6.3 Videre forskning 79
Litteraturliste 81
VEDLEGG 84
Vedlegg 1: Informasjonsskriv til deltakerne 84
Vedlegg 2: Samtykkeskjema 86
Vedlegg 3 - Intervjuguide 87

Figuroversikt

Figur 1 – Modell av Niss og Højgaard Jensens Matematiske kompetanser (2002, s. 45)..... 21
Figur 2 Den algoritmiske tenkeren, Utdanningsdirektoratet (2020) 25
Figur 3 Oppgavens forskningsdesign..... 32
Figur 4 Aspekter ved algoritmisk tenkning hos lærerne 40
Figur 5 Aspekter ved algoritmisk tenkning hos Lærer 1..... 48
Figur 6 Aspekter ved algoritmisk tenkning hos Lærer 2..... 52
Figur 7 Aspekter ved algoritmisk tenkning hos Lærer 3..... 60
Figur 8 Aspekter ved algoritmisk tenkning hos Lærer 4..... 67
Figur 9 En oppsummering av de nevnte aspektene hos den algoritmiske tenkere..... 68

1. Innledning

I det følgende kapittelet presenteres bakgrunnen for å skrive om programmering i matematikkundervisningen på mellomtrinnet. Utvikling i samfunnet og læreplaner er et viktig element her. Deretter kommer oppgavens mål og problemstilling frem, samt en avklaring av begrepet *programmering*. Det gis også en kort beskrivelse av forskningsprosjektet oppgaven har vært en del av. Innledningen avsluttes med en disposisjon for resten av oppgaven.

1.1 Bakgrunn for valg av tema

Gjennom min skolegang er det få ting som har fått meg like engasjert som når vi fikk ta i bruk datamaskin i undervisningen. Dette var et sjeldent fenomen, spesielt på de lavere trinnene, og når det endelig var tid for datamaskin i matematikkundervisningen, så var det repetering av gangetabellen som var på agendaen. Jeg har ikke før i de siste årene helt forstått hvor mye matematikk man kan lære ved bruk av digitale virkemidler, da spesielt ved bruk av datamaskinen.

I den største delen av min tid på skolen var det Kunnskapsløftet (LK06) som var gjeldende. I LK06 var det ingen direkte kompetansemål som var knyttet til programmering. Man kunne selvfølgelig bruke programmering, men man var ikke som lærer pålagt å ta dette i bruk. I det nye læreplanverket er det derimot et mål å lære elevene om programmering i matematikkundervisningen fra en ung alder. Allerede i 3. klasse skal elevene bli kjent med tankegangen i programmering ved hjelp av lek og spill. I 5. klasse skal elevene begynne å bli kjent med hvordan man kan bruke variabler, vilkår og løkker på en datamaskin for at noe skal skje på skjermen (Utdanningsdirektoratet, 2019b).

Samfunnet vårt endrer seg fort, og skolens læreplaner skal sørge for at elevene lærer relevant og framtidsrettet kunnskap. I Opplæringsloven kan man lese at «Opplæringen i skole og lærebedrift skal, i samarbeid og forståing med heimen, opne dører mot verda og framtida og gi elevane og lærlingane historisk og kulturell innsikt og forankring.» (1998, § 1-1). Det er for å åpne disse dørene mot verden og fremtiden at utviklingen i samfunnet har ført til at begreper som algoritmisk tenkning og programmering har kommet inn som en sentral del i undervisningen, og allerede på tidlige trinn.

Fra 1997, det året jeg ble født, og frem til 2020 har informasjonsteknologiens plass i matematikkundervisningen gått fra enkel bruk av kalkulator, til at alle elever nå skal få opplæring i programmering. I de følgende underoverskriftene vil jeg kort trekke frem noen sentrale tanker og mål fra de ulike læreplanene, fra reform 97 og frem til den læreplanen som gjelder i dag. Dette for å forsøke å få frem hovedtrekk i utviklingen og vise hvordan skolens læreplaner speiler samfunnsutviklingen og søker å møte samfunnets krav til kompetanse. Samtidig vil dette vise hvordan matematikklærerens oppgave har endret seg, og hvordan det stilles helt nye krav til hvordan dagens matematikklærere skal drive undervisning. Fenomenet programmering er også noe helt nytt for mange av lærerne som underviser i matematikk, og mange har lite eller ingen erfaring i emnet fra egen skolegang eller fra videreutdanning.

1.1.1 Reform 97

Nytt i Reform 97 var at bruk av lommeregner og datamaskin nå for første gang blir nevnt og tatt med i beskrivelsen av fagets *arbeidsmåter* og i *læringsmål* i matematikk. Når *arbeidsmåter* i faget beskrives i innledningen til matematikkplanen, leser vi at «I arbeid med oppgaver og problemer der eksperimentering og undersøkelser vektlegges, gir bruk av lommeregner og informasjonsteknologi muligheter for nye innfallsvinkler» (L97, s. 155). Ved å se på mål for mellomtrinnet, matematikk i dagliglivet, kan man lese at elevene skal "videreutvikle sine begreper om forskjellige størrelser og enheter, vurdere og regne med disse, med penger og tid, *og bli fortrolige med å bruke hensiktsmessige hjelpemidler, spesielt lommeregner og datamaskin.*" (L97, s.162).

Der hovedmomentene for 5. – 7. klasse beskrives ser vi, under hovedmoment *tall,* at sjetteklassinger skal «Undersøke tall og utforske tallmønstre, f.eks. ved hjelp av lommeregner og datamaskin, oppdage og beskrive egenskaper» (L97, s. 164). Videre skal elevene i syvende klasse "trene på å velge og bruke regnearter, metoder, hjelpemidler og redskaper, f.eks. informasjonsteknologi, til å løse problemer og utforske situasjoner." (L97, s. 165). Bruk av datamaskin foreslås som vi ser her som hjelpemiddel og redskap til problemløsning og utforsking, men det står altså ikke at dette er noe alle elevene *må* bruke. Datamaskin som hjelpemiddel begynte på denne tiden å bli tatt i bruk, men det var likevel ikke allment eie, og dataparken på de ulike skolene var nok varierende. Det er derfor ingen overraskelse at L97 ikke har et stort fokus på det digitale når det kommer til matematikk, eller noe fag i det hele

tatt. L97 er likevel det første læreplanverket som inkluderer datamaskin som en mulighet på mellomtrinnet.

1.1.2 Læreplanen, 2006

Fra at elevene i reform97 skulle bli *fortrolige med å bruke* lommeregner og datamaskin, skal elevene nå *lære å bruke* digitale verktøy. Digitale ferdigheter tas her inn i generell del av læreplanen som en av 5 grunnleggende ferdigheter i alle fag. I fagplanen for matematikk leser vi: "Digitale ferdigheiter i matematikk inneber å bruke digitale verktøy til læring gjennom spel, utforsking, visualisering og presentasjon. Det handlar òg om å kjenne til, bruke og vurdere digitale verktøy til berekningar, problemløysing, simulering og modellering."

LK06 har flere kompetansemål i matematikk som beskriver hvordan elevene skal kunne bruke digitale verktøy. For eksempel er det et mål i matematikk at elevene etter 7. trinn skal kunne «beskrive referansesystemet og notasjonen som blir benyttet for formler i et regneark, og bruke regneark til å utføre og presentere beregninger» (LK06). Bruk av regneark er også en form for programmering. Så det at eleven skal kunne bruke regneark til å utføre beregninger, kan man se på som innføring av programmering i læreplanen. Likevel er det først i neste læreplan, LK20, at begrepet programmering kommer inn i kompetansemål i matematikk.

1.1.3 Ludvigsens-utvalget

Samfunnet utvikles raskere enn tidligere. Tanker og ideer som ble sett på som viktige når LK06 ble innført, endres derfor over årene. Etter 7 år med LK06, oppnevnte derfor regjeringen et offentlig utvalg i 2013 som skulle vurdere innholdet i grunnopplæringen (NOU 2015: 8, 2015). Innholdet ble vurdert opp mot kravene til kompetanse i fremtidens samfunns- og arbeidsliv, og gitt videre til Kunnskapsdepartementet 15. juni 2015. Hovedoppgaven til utvalget var å finne ut hva elevene ville trenge å lære på skolen de neste 20-30 årene. De viktigste spørsmålene som da kom frem var hvilke kompetanser som ville være viktige for elevene i både skole, videre utdanning og yrkesliv, og hvilke endringer som måtte implementeres i de ulike fagene for å utvikle disse kompetansene.

Utvalget kom med fire nye kompetanseområder som de anbefalte skulle danne grunnlaget for fornyelsen av innholdet i skolen (NOU 2015: 8, 2015, s.8). Disse var: fagspesifikk kompetanse, kompetanse i å lære, kompetanse i å kommunisere, samhandle og delta, og kompetanse i å utforske og skape. Særlig dette siste; å utforske og skape, inkluderer kritisk tenkning og problemløsning. Som oppgaven vil komme grundigere inn på senere, er dette viktige komponenter i arbeid med programmering, og tett knyttet opp mot matematisk kompetanse og begrepet algoritmisk tenkning, og sentralt i læreplanens kjerneelement utforskning og problemløsning.

Utvalget påpekte flere ganger at samfunnet ble mer og mer digitalt (NOU 2015: 8, 2015, s.44). Denne utviklingen krevde at innholdet i skolefagene måtte fornyes, slik at innholdet i fagene var bedre tilpasset det digitale samfunnet barna vokser opp i. Begrep som framtidens skole og 21st century skills, eller nødvendige ferdigheter for det 21 århundre, var sentrale. Når det gjaldt matematikk beskrev Ludvigsens-utvalget matematisk kompetanse ved hjelp av fem komponenter (NOU 2015: 8, 2015, s. 57). Disse komponentene er: forståelse, beregning, anvendelse (strategisk tankegang), resonnering og engasjement. I oppgavens teoridel vil matematisk kompetanse bli beskrevet nærmere.

1.1.4 Læreplanen, 2020

Høsten 2020 er tidspunktet det nye læreplanverket ble gjeldende for alle norske skoler. I fagplanene ble begrepet kjerneelement innført. Kjerneelementene beskriver fagets bærende ideer og representerer det viktigste elevene skal lære (Utdanningsdirektoratet, 2019c). I matematikk er *utforsking og problemløysing* ett av seks kjerneelement. Nytt var også algoritmisk tenkning, som sammen med programmering, kom inn som et flerfaglig tema, ikke bare i matematikk.

Følgende er eksempler på nye kompetansemål i matematikk i LK20, som alle direkte eller indirekte handler om algoritmisk tenkning eller programmering:

- *Lage og programmere algoritmer med bruk av variabler, vilkår og løkker (5. klasse).*

- *Bruke variabler, løkker, vilkår og funksjoner i programmering til å utforske geometriske figurer og mønstre (6. klasse).*

- *Bruke programmering til å utforske data i tabeller og datasett (7. klasse).*

Som man kan se fra kompetansemålene, er det spor av programmering gjennom hele mellomtrinnet i den nye læreplanen. Dette er første gang programmering nevnes i læreplanen.

1.1.5 Utforsking og problemløsning

Ved implementeringen av LK20 (Utdanningsdirektoratet, 2019c), fikk matematikkfaget som nevnt nye kjerneelement. Utdanningsdirektoratet definerer kjerneelementer som det viktigste innholdet i faget, og «*det elevene må lære for å kunne mestre og bruke faget»* (2019c). De første fem kjerneelementene i matematikk handler om hvilke arbeidsmåter, metoder og tenkemåter som skal brukes i arbeidet med det sjette kjerneelementet; matematiske kunnskapsområder (Regjeringen, 2018, s. 15). Et av de nye kjerneelementene i matematikkfaget er utforsking og problemløsing, og beskrives slik:

> «Utforsking i matematikk handler om at elevene leter etter mønstre, finner sammenhenger og diskuterer seg fram til en felles forståelse. Elevene skal legge mer vekt på strategiene og framgangsmåtene enn på løsningene. Problemløsing i matematikk handler om at elevene utvikler en metode for å løse et problem de ikke kjenner fra før. Algoritmisk tenkning er viktig i prosessen med å utvikle strategier og framgangsmåter for å løse problemer og innebærer å bryte ned et problem i delproblemer som kan løses systematisk. Videre innebærer det å vurdere om delproblemene best kan løses med eller uten digitale verktøy. Problemløsing handler også om å analysere og omforme kjente og ukjente problemer, løse dem og vurdere om løsningene er gyldige.» (LK20)

Dette kjerneelementet er sentralt siden det først og fremst er her programmering kommer inn, og som vi kan se av denne gjennomgangen, har det skjedd store endringer i

matematikkfaget. Samfunnet har endret seg betydelig over de siste årene, og med det har også innholdet i matematikkfaget blitt endret. Endringene har naturlig nok betydning for jobben til matematikklæreren, og dette er det interessant å se litt nærmere på.

1.2 Oppgavens mål og problemstilling

En av de store forskjellene på LK20 og tidligere læreplaner er som vi har sett at programmering nå har blitt en del av matematikkfaget. De aller fleste av dagens matematikklærere hadde da LK20 ble innført lite eller ingen erfaring med programmering fra sin egen grunnutdannelse. Nå når de nye læreplanene har vært i bruk siden 2020, og lærerne begynner å få ulike erfaringer i klasserommet, kommer det et behov for å finne ut hvordan de som faktisk skal undervise i disse nye emnene har håndtert endringen. Hvordan er lærernes kunnskap og forståelse når det kommer til programmering i matematikkfaget? Dette vil ha konsekvenser for hvordan de gjennomfører undervisningen sin. Hvordan programmering tas i bruk som et redskap og en arbeidsmåte i arbeidet med kompetansemålene vil være interessant å se nærmere på. Jeg har i denne oppgaven derfor tenkt å prøve å finne ut av hva noen matematikklærere vektlegger i arbeidet sitt med programmering. Problemstillingen til oppgaven vil derfor være:

> *Hva sier et utvalg matematikklærere på mellomtrinnet om programmering i matematikkfaget?*

Dette vil være med å belyse hvordan noen av lærerne har håndtert innføringen, og hvordan de arbeider med programmering. Videre er dette ikke nødvendigvis en refleksjon av alle matematikklærerne. Selve problemstillingen er overordnet. For å spisse oppgaven inn har jeg utformet to forskningsspørsmål:

1. *Hvilke matematiske kompetanser beskriver lærerne i elevenes arbeid med programmering?*

2. *Hvilke aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering?*

Matematikk og programmering, eller informasjonsteknologi, er to egne og selvstendige fag og utdannelser. På denne bakgrunnen blir det første forskningsspørsmålet, hvor lærerne blir spurt om å beskrive matematiske kompetanser, relevant. Det er interessant å se om, og hvordan, innføringen av programmering i matematikkfaget bidrar til økt matematisk kompetanse hos elevene, og ikke bare er en ekstra ting elevene skal lære. Da er det relevant å se på hva lærerne selv trekker fram, vektlegger og fokuserer på. Det vil si noe om lærernes kunnskap og forståelse for hvordan matematikk og programmering kan integreres på en måte som blir en støtte i arbeidet med ulike emner i matematikkundervisningen.

I den nye læreplanen ble også begrepet *algoritmisk tenkning* innført og integrert, både i kompetansemål og i kjerneelementet *utforskning og problemløsing*. Algoritmisk tenkning er en omfattende måte å løse problemer på, og Utdanningsdirektoratet har laget en modell av den algoritmiske tenkeren, som beskriver og forklarer metoden (Utdanningsdirektoratet, 2019a). Denne vil bli videre forklart i teorikapittelet, for å dekke alt denne problemløsningsmetoden omfatter.

Algoritmisk tenkning er et viktig fellestrekk hos matematikk og programmering. En sammenligning av beskrivelsen av algoritmisk tenkning under kjerneelementet utforskning og problemløsning i matematikk: «Algoritmisk tenking er viktig i prosessen med å utvikle strategier og framgangsmåtar for å løyse problem og inneber å bryte ned eit problem i delproblem som kan løysast systematisk» (Utdanningsdirektoratet, 2019c), med følgende definisjon av programmering: «Programmering handler om å bryte komplekse problemer ned til mindre problemer og deretter gi en fullstendig og nøyaktig beskrivelse av fremgangsmåten for løsningen av problemet» (Statped, 2021), får dette tydelig fram. Det andre forskningsspørsmålet i oppgaven vil kunne si noe om lærernes kunnskap og bevissthet om algoritmisk tenking, både nøkkelbegrep og arbeidsmåter, i arbeidet med programmering i matematikkfaget.

Forskningsspørsmålene henger sammen ved at begrepet algoritmisk tenking, representert ved den algoritmiske tenkeren, innehar flere av de matematiske kompetansene, og da kanskje særlig problemløsningskompetanse. Dette vil bli utdypet og tydeliggjort i teorikapittelet.

1.3 Programmering i matematikk

Selv om man ikke har hørt mye om programmering i den norske skolen før i de siste årene, har programmering fantes lenge, også i skolen til en viss grad. Uten å gå for mye inn i detaljer på historien til programmering før det forrige tiåret, ble det allerede på 60-tallet utviklet et programmeringsspråk, LOGO, som hadde som hensikt å bli brukt i undervisningen (Sevik et al., 2016, s. 8). Videre på 80-tallet ble den norske skole introdusert til programmering gjennom valgfaget EDB. Selv om programmering kom som en mulighet på 80-tallet, ble det derimot ikke noe som alle valgte. Flertallet hadde derfor ikke programmering. Det finnes nok flere grunner til hvorfor programmering ikke tok av på 80-tallet. Tilgangen til datamaskiner var dårligere enn den har blitt de siste årene, og datidens programmeringsspråk og utstyr til å gjennomføre noen form for programmering var dårligere, og ikke minst kompetansen til lærerne var lav.

På 2010-tallet var programmering et fokusområde for EU (Sevik et al., 2016, s. 6). Programmering ble blant annet satt på EU sin *Digital Agenda for Europe*, hvor utdanningsministre i EU-landene blir oppfordret til å fremme nettopp programmering i skolen. Begrunnelsen til denne satsingen er blant annet at programmering er en ferdighet som blir viktigere og viktigere. Den fremmer kreativitet og kan gjøre folk bedre til å samarbeide. Programmering vil også bidra med kommunisering gjennom et felles språk.

Året etter Ludvigsens-utvalget, som ble nevnt i delkapittel *1.1.3*, la en ekspertgruppe oppnevnt av Utdanningsdirektoratet ut en rapport om programmering, *Teknologi og programmering - et nytt fag i grunnskolen?* (Sevik et al., 2016, s. 25). De kom i denne rapporten med en anbefaling om å opprette et eget fag til teknologi og programmering. Man kan se også i Ludvigsen-utvalget at en skole for fremtiden var i fokus. Å innføre programmering ville da være en måte å møte behovet for slik kompetanse i fremtiden

Spol frem noen år og programmering blir en del av matematikken, og dette er bakgrunnen til at denne masteroppgaven skal finne ut hva et utvalg matematikklærere på mellomtrinnet sier om programmering i matematikkfaget, men hva er egentlig programmering? Det er en rekke definisjoner med varierende fokus som beskriver hva det innebærer å programmere. Senter

for IKT i utdanningen har gitt ut en artikkel om programmering i skolen, hvor de definerer programmering slik:

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

Videre har også Statlig Pedagogisk Tjeneste(2021), som er et direktorat under Kunnskapsdepartementet, definert programmering:

> «Programmering handler om å bryte komplekse problemer ned til mindre problemer og deretter gi en fullstendig og nøyaktig beskrivelse av fremgangsmåten for løsningen av problemet. Definisjonen er lett overførbar til situasjoner i dagliglivet. Vi møter ofte komplekse problemer, og vi løser dem ved å bryte dem ned i mindre og mer håndterlige deler. Finner vi ikke løsningen ved første forsøk, prøver vi en gang til.»

Vi kan se fra begge definisjonene at de vektlegger litt forskjellige ting i forklaringen. Det som derimot er felles for begge forklaringene er at de begge snakker om programmering i sammenheng med å løse et problem. Statped sin definisjon nevner ikke at dette har noe med datamaskin å gjøre, mens det gjennom Utdanningsdirektoratets forklaring kommer frem at programmering er hele prosessen fra å identifisere et problem, til å feilsøke koden man har skrevet på datamaskinen.

1.4 Forskningsprosjektet LATACME

Masteren er en del av et forskningsprosjekt ved Høgskulen på Vestlandet kalt LATACME (Learning about teaching argumentation for critical mathematics education in multilingual classrooms). Prosjektet ser på hvordan lærerstudenter og lærere underviser i argumentasjon og kritisk matematikkdidaktikk på barnetrinnet. Prosjektet er tredelt, hvor min masteroppgave går inn under delprosjektet Argumentasjon og IKT.

1.5 Oppgavens disposisjon

Oppgaven er delt inn i 5 kapitler. I dette første kapittelet blir det gjort rede for bakgrunnen for valg av tema, en gjennomgang av fremgangen til digitale hjelpemidler i matematikk i de tre siste læreplanverkene og oppklaring av studiens problemstilling og forskningsspørsmål. Begrepet programmering blir også forklart i denne delen av oppgaven.

Oppgavens 2. kapittel består er en teoridel som skildrer et rammeverk bestående av et utvalg av Niss og Højgaard Jensens (2002) matematiske kompetanser og modellen av den algoritmiske tenkeren på Utdanningsdirektoratet (2019a) sine sider, som er basert på BareFoot Computing (2016) sin modell av Computational thinking. Det vil så komme en gjennomgang av relevant forskning om programmering i matematikkundervisningen. Avslutningsvis i kapittelet vil det bli forklart hvorfor det gir mening å snakke om algoritmisk tenkning og de utvalgte matematiske kompetansene sammen.

Kapittel 3 i denne oppgaven er metodekapittelet, og starter med en presentasjon av valgt metode for å svare på problemstillingen, og en beskrivelse av oppgavens forskningsdesign. I dette kapittelet vil det komme en beskrivelse av hvordan datainnsamlingen gikk for seg. Det vil også forklares hvordan intervjuene blir analysert. Til slutt vil det komme en gjennomgang av oppgavens validitet og reliabilitet og deretter etiske hensyn som ble gjort i oppgaven.

I kapittel 4 vil hvert av intervjuene bli analysert som forklart i metodekapittelet. Oppbygningen av dette kapittelet vil være slik at intervjuet av hver lærer vil analyseres ferdig før analyse av neste intervju påbegynnes. Analysen vil først se på de matematiske kompetansene problembehandlingskompetanse, resonnementskompetanse og kommunikasjons-kompetanse. Deretter vil det være en systematisk gjennomgang av aspektene ved algoritmisk tenkning, hvor de nevnte nøkkelbegrepene vil trekkes frem etterfulgt av arbeidsmåtene. I slutten av analysekapittelet vil det komme en liten oppsummering over det som er kommet frem, og dette vil bli hentet frem i diskusjonskapittelet.

Diskusjon rundt studiens analyse vil skje i kapittel 5. Her vil forskningsspørsmålene bli forsøkt besvart, gjennom diskusjon og drøfting av tidligere nevnt teori. Kapittelet tar først for seg det første forskningsspørsmålet om de matematiske kompetansene som beskrives, og det vil forekomme en diskusjon rundt de tre utvalgte kompetansene til Niss & Højgaard Jensen (2002). Deretter vil diskusjonskapittelet ta for seg det andre forskningsspørsmålet om aspekter ved algoritmisk tenkning. Her vil først nøkkelbegrepene tas hånd om, og deretter arbeidsmåtene.

Det siste kapittelet, kapittel 6, vil være en avslutning på oppgaven hvor studiens funn vil sees på opp mot problemstillingen og forskningsspørsmålene. Her vil det også trekkes frem refleksjoner rundt forskningsprosjektet, som begrensninger knyttet til studien og tanker rundt videre forskning.

2 Teori

Når man skal snakke om programmering, uansett sammenheng, er det noen begreper man ikke kommer foruten. I dette kapittelet vil jeg først se på begrepet matematisk kompetanse og komme med en gjennomgang av Niss og Højgaard Jensen (2002) sine matematiske kompetanser. Deretter vil jeg gi en redegjørelse av begrepet algoritmisk tenkning, eller computational thinking, og algoritmisk tenknings-modellen til Utdanningsdirektoratet (2019a) vil bli forklart. Algoritmisk tenkning er et av de nye sentrale begrepene som i det siste tiåret har fått et stort internasjonalt fokus, også i Norge. Tidligere forskning som har sett på bruk av programmering knyttet til matematikkundervisning vil så bli presentert. Avslutningsvis i kapittelet vil det komme en forklaring på hvorfor det gir mening å snakke om de matematiske kompetansene som er valgt og algoritmisk tenkning i en programmeringssammenheng.

2.1 Matematisk kompetanse

Kompetanse er et begrep som blir brukt mye i dagens samfunn, ikke bare i skolesammenheng. De fleste er kjent med hvordan kompetansemålene i de forskjellige fagene avgjør hva man gjør i timene, men hva vil det egentlig si å ha kompetanse? Utdanningsdirektoratet definerer kompetanse slik:

> 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. (2019d)

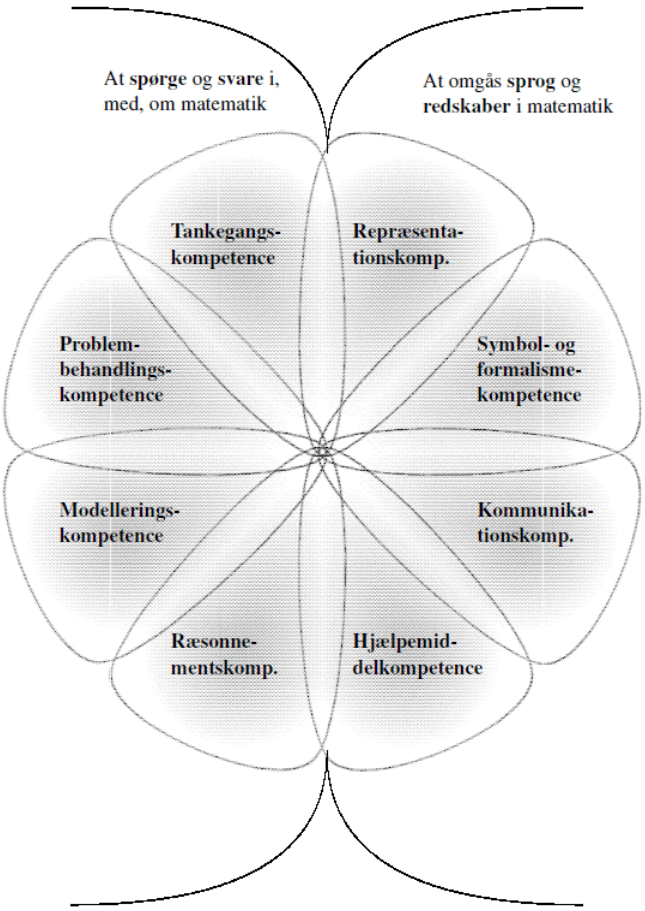
Når det gjelder kompetanse i matematikk, mener forsker og lærebokforfatter Røsseland at en «kompetansebeskrivelse av faget går langt mer direkte på selve undervisningen, for da vil en også sette fokus på ferdigheter som vanskelig lar seg teste i en skriftlig prøve» (2005, s. 14). Hun snakker her om at flere av egenskapene man får bruk for i matematikk vil være svært vanskelig for elevene å vise med papir og penn. Men hva er disse kompetansene som hun snakker om? Det er gjort flere forsøk på å beskrive og sammenfatte hva kompetanse i matematikk innebærer, bl.a. beskrev Ludvigsen-utvalget (NOU 2015:8, 2015), som nevnt i innledningen, matematisk kompetanse ved følgende fem komponenter: forståelse, beregning, anvendelse (strategisk tankegang), resonnering og engasjement. En av dem som har jobbet mye med å formulere hva det vil si å ha matematisk kompetanse, er Mogens Niss. Hans tidligere arbeid ble blant annet tatt i bruk da Niss fikk i oppgave å anvende matematisk kompetanse som en måte å forklare matematikk på som fag (Niss & Højgaard Jensen, 2002).

2.1.1 Niss & Højgaard Jensen (2002) – Matematiske kompetanser

Niss og Højgaard Jensen beskriver en person som besitter kompetanse på et bestemt område slik: «… han eller hun faktisk er i stand til at begå sig med gennemslagskraft, overblik, sikkerhed og dømmekraft inden det pågældende område.» (2002, s. 43). Dette oversetter de videre til matematikk, hvor matematisk kompetanse består av blant annet å ha kunnskap om, å forstå, å utøve, og å bruke og ha meninger om matematikk og matematisk aktivitet i flere sammenhenger, der matematikk spiller stor eller liten rolle. Med andre ord kan man si at en person med matematisk kompetanse har en bred forståelse i faget og er i stand til å bruke denne kunnskapen effektivt i ulike situasjoner.

I sitt arbeid med matematisk kompetanse har Niss og Højgaard Jensen identifisert åtte sentrale kompetanser innenfor matematikk (2002, s. 44), og de forklarer en kompetanse som en «…selvsændig, rimeligt afgrænset hovedkomponent i matematisk kompetence…» (Niss &

Højgaard Jensen, 2002, s. 43). Disse åtte kompetansene blir videre delt inn i to hovedgrupper. Den første gruppen består av fire kompetanser som kan kort oppsummeres med «at kunne spørge og svare i og med matematik» (Niss & Højgaard Jensen, 2002, s. 44). Her er det snakk om evnen til å stille slike spørsmål og være klar over hvilke svar man kan få, evnen til å svare på slike spørsmål på en matematisk måte og evnen til å forstå, bedømme og produsere argumenter til å svare på matematiske spørsmål. I den andre gruppen finner vi de resterende fire kompetansene som kan dekkes av «at kunne håndtere matematikkens sprog og redskaber» (Niss & Højgaard Jensen, 2002, s. 44). Her er det snakk om å kunne håndtere forskjellige representasjoner, kommunisere om matematikk og om bruken av tekniske hjelpemidler. Dette innebærer også å forstå de matematiske symbolene.



Figur 1 – Modell av Niss og Højgaard Jensens Matematiske kompetanser (2002, s. 45)

Det vil nå komme en gjennomgang av et utvalg av kompetansene og hva de innebærer. Det er viktig å påpeke at selv om disse kompetansene er delt inn i det som i utgangspunktet er åtte individuelle matematiske kompetanser, har alle kompetansene noe med hverandre å gjøre,

og man kan ikke være kompetent innenfor bare én av kompetansene. Matematisk kompetanse kan sees på som et knutepunkt i en gruppe av ferdigheter som er tettere samlet nær midten og som blir gradvis tynnere ut mot kanten. Dette betyr i praksis at man ikke kan besitte en kompetanse isolert.

I denne oppgaven har jeg valgt å ta utgangspunkt i de matematiske kompetansene som kommer tydeligst frem i forskningen som presenteres i *2.3 Tidligere forskning på programmering i matematikkundervisningen*. Disse er *problembehandlingskompetansen* (Bintoro et al., 2021; Popat & Starkey, 2019;), *resonnementskompetansen* (Psycharis & Kallia, 2017; Scherer et al., 2021) og *kommunikasjonskompetansen* (Popat & Starkey, 2019; Forsström & Kaufmann, 2018)

2.1.1.1 PROBLEMBEHANDLINGSKOMPETANSEN

Å kunne gjenkjenne, formulere, avgrense og spesifisere forskjellige typer matematiske problemer er en del av *problembehandlingskompetansen* (Niss & Højgaard Jensen, 2002, s. 49) . Den innebærer også å kunne løse slike matematiske problemer på måten de er formulert på, eller om nødvendig, løse det på en annen måte. Slike formulerte spørsmål krever en form for matematisk undersøkelse for å løses. Problembehandlingskompetansen er varierende i den grad at et spørsmål som bare krever en bestemt rutine ikke går inn under denne kompetansen. Det samme spørsmålet kan derimot være en del av problembehandlingskompetansen for en person som ikke har lært selve rutinen som kreves for å løse det. Det er også en forskjell på å kunne formulere og gjenkjenne et problem og det å kunne løse problemet. Det er mulig å ha ferdigheter innenfor det å identifisere et problem, selv om man ikke presterer å løse det, og motsatt kan man være dårlig til å identifisere det matematiske problemet, men dersom noen forklarer hva problemet er kan man ha gode ferdigheter når det kommer til løsningsprosessen.

2.1.1.2 RESONNEMENTSKOMPETANSEN

Den neste kompetansen er en del av hovedgruppen *å kunne spørre og svare i og med matematikk*, og er *resonnementkompetansen.* Denne kompetansen handler i korte trekk om å vurdere resonnement, skriftlig og muntlig, og forstå hva et matematisk bevis innebærer (Niss

& Højgaard Jensen, 2002, s. 54). Dette innebærer også å kunne avgjøre om et matematisk argument er gyldig, slik at man kan bruke det som bevis, eller om det ikke kan brukes som bevis. Å forstå logikken i ulike eksempler er også sentralt innenfor denne kompetansen. Den andre delen av kompetansen består av å kunne komme med argumenter basert på intuisjon. Det er denne resonnementkompetansen som spiller en rolle når man velger hvilke regneoperasjoner man skal benytte seg av. De fleste kompetansene har en form for relasjon, og denne kompetansen er tett forbundet med modelleringskompetansen og problembehandlingskompetansen (Niss & Højgaard Jensen, 2002, s. 210).

2.1.1.3 KOMMUNIKASJONSKOMPETANSEN

All kommunikasjon som skjer gjennom matematikk, tilhører *kommunikasjonskompetansen* (Niss & Højgaard Jensen, 2002, s. 60). Dette handler i all hovedsak om å kunne kommunisere i, med og om matematikk. Dette innebærer å tolke og sette seg inn i andres skriftlige, muntlige eller visuelle matematiske utsagn. Det handler også om å kunne uttrykke seg på forskjellige måter om matematikk både skriftlig, muntlig og visuelt. Som tidligere nevnt har alle kompetansene noe med hverandre å gjøre, og denne er ingen unntak. Denne kompetansen er tett forbundet med blant annet representasjonskompetansen, ettersom all kommunikasjon som skjer gjennom matematikk må representeres på et vis.

2.2 Algoritmisk tenkning

Det nye læreplanverket bærer med seg en hel del nye temaer som tidligere ikke var en del av opplæringen. Et av de nye temaene som skal bidra med å forberede elevene på en ny fremtid er, som nevnt over, algoritmisk tenkning, som blir introdusert gjennom kjerneelementet *Utforskning og Problemløsning*. Algoritmisk tenkning kommer fra det engelske uttrykket «Computational thinking», som Wing introduserte allerede i 2006 (Wing, 2006). Hun mente allerede da at algoritmisk tenkning var en grunnleggende ferdighet for alle, og ikke bare for informatikere, og tenkte at dette burde læres bort til elever på samme måte som lesing og skriving. Hun forklarer at algoritmisk tenkning ikke bare involverer det å løse problemer eller å lage ulike systemer, men sier også at en del av algoritmisk tenkning er å forstå menneskelig oppførsel.

Shute et al. (2017) har i en studie sett gjennom flere artikler som omtaler computational thinking, eller på norsk algoritmisk tenkning. Basert på disse studienes definisjoner av algoritmisk tenkning, har de kommer med følgende definisjon av algoritmisk tenkning:

> The conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts (Shute et al., 2017, s. 10).

Fra denne definisjonen kan vi se at algoritmisk tenkning handler om å løse problemer på en effektiv måte. Det nevnes at det gjøres algoritmisk og at det kan gjøres både ved hjelp av datamaskin, og uten. Denne definisjonen legger også vekt på at selve måten å løse problemet på, ofte vil være mulig å bruke i andre kontekster.

Computational thinking, eller algoritmisk tenkning, har i løpet av de siste 5 årene blitt en av dette århundrets viktigste ferdigheter som en elev trenger for å kunne lykkes i et digitalt samfunn (European Schoolnet, 2017). Flere og flere land velger å innføre programmering og algoritmisk tenkning inn i deres læreplaner, og ifølge European Schoolnet har minst 20 land gjort dette allerede i skrivende stund. Siden denne artikkelen ble gitt ut har også blant annet Norge gitt plass til programmering i den nye læreplanen.

2.2.1. Utdanningsdirektoratet (2020) – Den algoritmiske tenkeren

Som nevnt tidligere er et av de nye kjerneelementene i matematikk utforsking og problemløsning. Problemløsning handler om å analysere og omforme kjente og ukjente problemer, og deretter løse dem og vurdere om løsningene er gyldige (Utdanningsdirektoratet, 2020). Algoritmisk tenkning, som også er inkludert i den nye læreplanen som et sentralt emne, er en viktig del i prosessen med å utvikle strategier og fremgangsmåter for å løse problemer. I forbindelse med algoritmisk tenkning har derfor Utdanningsdirektoratet lagt frem en modell som viser hva det innebærer å være en «algoritmisk tenker». Modellen er oversatt fra den engelske siden Barefoot Computing (2016), som også har lagt til en forklaring til hver av nøkkelbegrepene og arbeidsmåtene, slik at man enkelt kan vite hva de innebærer.

Figur 2 Den algoritmiske tenkeren, Utdanningsdirektoratet (2020)

I *Figur 2 – Den algoritmiske tenkeren* kan vi se at figuren beskriver den algoritmiske tenkeren under to forskjellige overskrifter, *Nøkkelbegrep* og *Arbeidsmåter*. Under nøkkelbegrep kan vi se fra figuren at den trekker frem seks forskjellige begreper, som sammen skal dekke hva begrepet algoritmisk tenkning handler om, etterfulgt av en kort forklaring. Videre kan vi se at det er fem arbeidsmåter som er viktige for den algoritmiske tenkeren. Forklaringene nedenfor er oversatt fra Barefoot Computing sine forklaringer av de forskjellige *nøkkelbegrepene* og *arbeidsmetodene* (Barefoot Computing, 2016). I tillegg har jeg kommentert kort med praktiske eksempler for å knytte begrepene konkret til elevers arbeid med programmering. Det vil være hensiktsmessig å ta med eksempler til hvert av aspektene, som kan minne om det som kommer i analysen. Eksemplene til hvert nøkkelbegrep er selvprodusert:

Logikk hjelper oss når man skal etablere og faktasjekke, ikke bare i matematikk. Det handler også om å gjøre antagelser eller å resonnere seg frem til det riktige svaret, for eksempel i Sudoku. Programmeringsspråk er bygget opp entydig og logisk (if, then, else). Man kan derfor si at når elevene jobber med programmering, er det grunnleggende logikk de holder på med.

En *algoritme* handler om å følge et bestemt sett med steg og regler mens man gjør en oppgave, som en oppskrift man må følge. Når legoroboten rygger og snur dersom den møter på fysiske hindringer, er det fordi elevene har gitt roboten en oppskrift å følge: kjør fram til trykksensoren aktiveres, stopp, kjør så bakover osv.

En viktig del av å kunne tenke algoritmisk, eller programmere, er å kunne *dekomponere* et problem til flere mindre deler. Dette gjør at man enklere kan løse de mindre problemene, og deretter gå løs på hovedproblemet. Et eksempel kan være å få en legorobot til å kjøre fra A til B, samtidig som den kommer seg forbi hinder på veien. Dette kan virke som et komplekst problem, men dersom man klarer å identifisere hvilke delproblemer det er sammensatt av, blir en løsning lettere å finne. En enkel dekomposisjon kan være; Få roboten til å bevege seg, få roboten til å bevege seg mot mål, få roboten til å oppdage hinder, få roboten til å kjøre rundt hinder. Ved å sette sammen løsningene på hvert av disse problemene kan en til slutt få en fullstendig løsning på det opprinnelige problemet.

Ved å legge merke til ulike *mønstre* vil det gjøre det lettere å gjøre antagelse om hva som vil være det neste som skjer, lage ulike regler og løse de neste problemene som oppstår. Et praktisk eksempel kan være femteklassingen som bruker programmene Scratch eller Turtlestitch til å lage kunst av geometriske figurer. Oppmerksomhet på gjentakende handlinger og mønster, og kunnskap om hvordan å bruke løkker i programmet, hjelper eleven til å lage kunst som nesten «skaper seg selv».

Abstraksjon handler om å filtrere vekk informasjonen man ikke trenger, slik at man kan fokusere på den informasjonen som kan hjelpe oss å løse oppgaven. Å få en robot fra A til B kan være vanskelig om man må tenke på alt som skal til for å styre motorene og sensorene . Man kan da bruke abstraksjon og heller arbeide med ferdige instrukser som «kjør frem» og «oppdag hinder», for å sette sammen en løsning. Slik kan man enklere se helheten og løse problemer uten å tenke på alle detaljene som ligger bak.

Det siste nøkkelbegrepet i modellen er *evaluering*. Dette handler om å ta avgjørelser basert på forskjellige faktorer, som innebærer å gjøre forskjellige vurderinger som for eksempel hva

man trenger for å gjøre oppgaven og hva man prøver å oppnå med oppgaven. Evaluering henger også sammen med feilsøking, og vil gjerne være en kontinuerlig prosess i arbeidet med programmering.

Videre tar også Figur 1 for seg arbeidsmåtene som den algoritmiske tenkeren benytter seg av. Disse arbeidsmåtene er å fikle, å skape, å feilsøke, å holde ut og å samarbeide. På samme måte som nøkkelbegrepene, er også forklaringene av det arbeidsmåtene innebærer, oversatt fra Barefoot Computing (2016), men hvor tilføyde eksempler er selvprodusert:

Det *å fikle* kan kort forklares med at man skal utforske og eksperimentere, som betyr at man endrer på ulike ting for å observere resultatet av dette. Det skal en del fikling, prøving og feiling til for å få Spheroballen til å endre retning før den ruller over dørterskelen og gjennom døra, for så å stoppe før den treffer veggen. Hvor mange grader skal den snu for å komme i riktig posisjon, hvor mye fart trenger den for å komme over dørterskelen, og hvor mye fart er for mye dersom man skal unngå krasj med veggen?

Å skape er en kreativ arbeidsmåte som er sentral hos den algoritmiske tenkeren, og er blant annet det å designe eller utvikle nye løsninger. Arbeid med programmering gir uendelige muligheter til å skape og arbeide kreativt i undervisningen. Fra å tegne i Scratch til å skape avanserte programmer i Lego Mindstorms.

En veldig viktig egenskap, og arbeidsmåte, hos den algoritmiske tenkeren er det *å holde ut* når man møter utfordringer, og det er viktig å ha utholdenhet i arbeid med oppgaver som krever litt ekstra. Motivasjonen for å få programmet de har jobbet med over lang tid til å virke til slutt, tross hindringer på veien, kan være en sterk drivkraft til å holde ut. Sånn sett er arbeid med programmering en god aktivitet til å trene utholdenhet.

Samarbeid er også en sentral del av det å være en algoritmisk tenker. Å jobbe sammen i forskjellige «teams» der man kan hjelpe hverandre med blant annet ulike perspektiver og ferdigheter. Programmeringsprosessen består av både planlegging, problemløsing og evaluering. Gjennom hele prosessen trenes og styrkes ferdigheter i kommunikasjon og

samarbeid. Programmering er sånn sett en god aktivitet for elever som har behov for å styrke samarbeidsevnen.

Den siste arbeidsmåten i modellen er å *feilsøke*, som innebærer å systematisk se gjennom det man har gjort for å rette opp i eventuelle feil. I arbeidet med programmering er det å finne ut hva som gjør at koden ikke fungerer som tenkt, å rette opp feilen, en vesentlig del av jobben.

2.3 Tidligere forskning på programmering i matematikkundervisningen

I nyere tid er det gjennomført flere studier som har som mål å finne ut i hvilken grad programmering kan være en ressurs, delvis om programmering i matematikkundervisningen kan øke den matematiske kompetansen til elever i bestemte områder. I dette delkapittelet vil det bli presentert forskning som er gjort de siste årene rundt programmeringens bruk, og potensielle resultater av bruken i matematikkfaget. Forskningen vil bli presentert i rekkefølgen den er gjennomført.

I en studie av Psycharis & Kallia undersøkes det om dataprogrammering har en innvirkning på elevers resonneringsevne og problemløsningsevne (2017). I den ene delen av studiet ble det gjennomført tester for å finne ut om matematikkundervisning i forbindelse med programmering vil påvirke elevenes resonneringsevne, og hvis dette var tilfelle, i hvilken grad? For å finne ut av dette ble 66 forskjellige skoler delt inn i en kontrollgruppe og en eksperimentgruppe. Etter at eksperimentgruppen hadde hatt det de kaller for et «programmeringskurs» var det signifikant forskjell på gruppene, som i utgangspunktet var nesten identiske. Studiens funn indikerer at det er en signifikant forskjell i elevenes resonneringsevne hos elevene som deltok i «programmeringskurset», sammenlignet med de som ikke deltok.

Forsström & Kaufmann presenterte i 2018 en litteraturreview der de hadde sett på 15 relevante artikler, med relevant menes artikler som sa noe om bruk programmering i undervisningen, med varierende innfallsvinkel. Ved å se på de utvalgte studiene, kom det frem tre temaer; motivasjon til å lære matematikk, elevenes prestasjon i matematikk, samarbeid mellom elevene (Forsström & Kaufmann, 2018, s. 18). Det kommer også frem at gjennom

arbeid med programmering og aktiviteter med roboter, vil samarbeid være mye brukt. De trekker her frem at lærerens rolle i klasserommet er avgjørende, hvor en støttende og veiledende lærer vil bidra til at elevene får løse problemer i grupper (Forsström & Kaufmann, 2018, s. 27). Basert på artiklene de har analysert, kommer det også frem at den tilbakemeldingen elever får når de jobber i grupper ofte er mer verdifull enn den de får individuelt.

Popat og Starkey (2019) har skrevet en oversiktsartikkel hvor fokuset er på hvilke læringsutbytter programmering i matematikkundervisningen kan medføre. Ved å se på de ulike artiklene, fant de blant annet ut at problemløsning er noe elevene kan bli bedre på når de lærer programmering, men at det ikke nødvendigvis er noe bedre å lære problemløsning gjennom programmering enn direkte læring. Studiene Popat og Starkey så på viste også at programmering kan gjøre at elevene får bedre sosiale ferdigheter, deriblant samarbeidsevne, og evne til å utvikle kritisk tenkning (2019).

Scherer et al. har i en studie fra 2019 undersøkt om kunnskap og ferdigheter som læres gjennom å jobbe med programmering, kan overføres til andre emner i matematikken (s. 4). Studien viser at det å lære dataprogrammering kan assosieres med visse kognitive fordeler. De trekker frem fordeler som økte ferdigheter i programmering, og peker på muligheten til å trene seg opp på algoritmisk tenkning gjennom programmering (Scherer et. al, 2018, s. 49). De viser også til fordeler i andre kognitive ferdigheter, deriblant resonnement, og viser til at programmeringsferdigheter kan overføres til andre områder. Når det kommer til hvilke områder disse ferdighetene best kan overføres til, nevnes det at det er mer sannsynlig at en overføring vil skje til situasjoner som krever noen av de samme kognitive ferdighetene som programmering.

I Sverige har programmering vært en del av læreplanen i matematikk siden 2018, og en gruppe forskere ved Universitetet i Stockholm (Nouri et al.) publiserte i 2019 sine svar på forskningsspørsmålet: Hva er lærernes perspektiv, hvilke ferdigheter oppnår elevene når de arbeider med programmeringsaktiviteter? I tillegg til å se på ferdigheter innenfor algoritmisk tenkning, identifiserte studiet det de omtaler som generelle ferdigheter (og holdninger):

kognitive ferdigheter, språkferdigheter, samarbeidsferdigheter og kreative problemløsings ferdigheter (Nouri et al., 2019, s. 6). Forskerne trekker fram at samtlige av de intervjuede lærerne la vekt på og understreket kreative problemløsingsferdigheter som et sentralt resultat av undervisning i programmering. Videre blir det framhevet at selv om problemløsning ikke er noe nytt i matematikkundervisningen, ser det ut til at arbeid med programmering kan bidra til å skape flere muligheter for at elevene lar seg engasjere og motivere (Nouri et al., 2019, s. 11). Og når lærere også etter hvert får mer erfaring og blir enda mer klar over de kreative problemløsingsprosessene som er involvert i arbeid med programmering, vil de være i enda bedre stand til å fremme og forsterke disse ferdighetene.

Bintoro et al. (2021) har gjort en studie med mål om å teste om det er en endring i elevers problemløsningsevne før og etter implementering av IKT. Gjennom studien sin har de gjort en rekke tester for å måle elevenes problemløsningsevner. Det kommer i studien frem at elevene som hadde fått implementert IKT-basert undervisning er blitt bedre på problemløsning enn de var før innføringen av dette. Samtidig som dette gir en økt evne innenfor problemløsning, viser også studien at elevene blir mer interesserte og aktive, og undervisningen blir enklere akseptert av elevene når det tas i bruk teknologi.

2.4 Sammenheng mellom Niss & Højgaard Jensens (2002) utvalgte matematiske kompetanser og Utdanningsdirektoratets modell av Den algoritmiske tenkeren (2019a)

Dersom man går gjennom og ser nøye på alle de åtte matematiske kompetansene (Niss & Højgaard Jensen, 2002), vil man kunne knytte dem til den algoritmiske tenkerens nøkkelbegrep og arbeidsmåter (Utdanningsdirektoratet, 2019a). Og når vi tar utgangspunkt i arbeidsmåtene, ser vi at alle disse er beskrivelser av viktige momenter ved matematisk kompetanse.

Problembehandlingskompetansen handler kort om å løse matematiske problemer (Niss & Højgaard Jensen, 2002, s. 49), og algoritmisk tenkning er en problemløsningsmetode. Ved å se på hvordan Forsström & Kaufmann (2018) forklarer programmering « ... process related to the development and implementation of instructions for computer programs so the computer can perform specific tasks, solve problems, and support human interactions», kommer det også

frem at programmering i all hovedsak handler om å løse problemer gjennom å gi instrukser, eller kode, til en datamaskin. Alt dette henger sammen på den måten at algoritmisk tenkning er en problemløsningsmetode, som kan benyttes gjennom programmering. At det er en problemløsningsmetode betyr i hovedsak at alle nøkkelordene ved algoritmisk tenkning vil være sentrale når det kommer til problembehandlingskompetanse.

Resonnementskompetansen vil i stor grad henge sammen med noen av nøkkelbegrepene hos den algoritmiske tenkeren. Denne kompetansen handler i hovedsak om å forstå matematiske bevis (Niss & Højgaard Jensen, 2002, s. 54), altså å forstå hvorfor svaret blir slik det blir. Nøkkelbegrepet logikk er en sentral del av resonnementskompetansen, ettersom logikk handler om å forstå og gjøre antagelser. Ved å se på Barefoot Computing(2016) sin forklaring av logikk, kan vi også se at den trekker frem resonering som et eksempel på logikk er. Resonnementskompetansen kan også ses på i sammen med nøkkelbegrepet mønster. Mønster handler om å gjenkjenne mønster, og gjøre antakelser på hva som vil skje videre og det handler på mange måter om å resonnere.

For eksempel vil evnen til å samarbeide henge nøye sammen med kommunikasjonskompetanse. Hvor godt man klarer å samarbeide om et prosjekt, eller nå et felles mål, avhenger i svært stor grad av både hvor godt man kan fremme sine synspunkt, men også i hvor stor grad man klarer å tolke og sette seg inn i det samarbeidspartneren prøver å formidle (Røsseland, 2015, s. 17.) Kommunikasjonskompetansen handler ifølge Niss om å kommunisere ved hjelp av matte, inkludert å tolke andres matematiske utsagn (Niss & Højgaard Jensen, 2002, s.60). Arbeidsmåten feilsøke kan også knyttes til denne kompetansen, da matematiske utsagn kan være både muntlige, skriftlige og visuelle. Feilsøking kan man si er å kommunisere med programmet, ved å tolke hva programmet sier, og hva responsen bør være for å løse problemet.

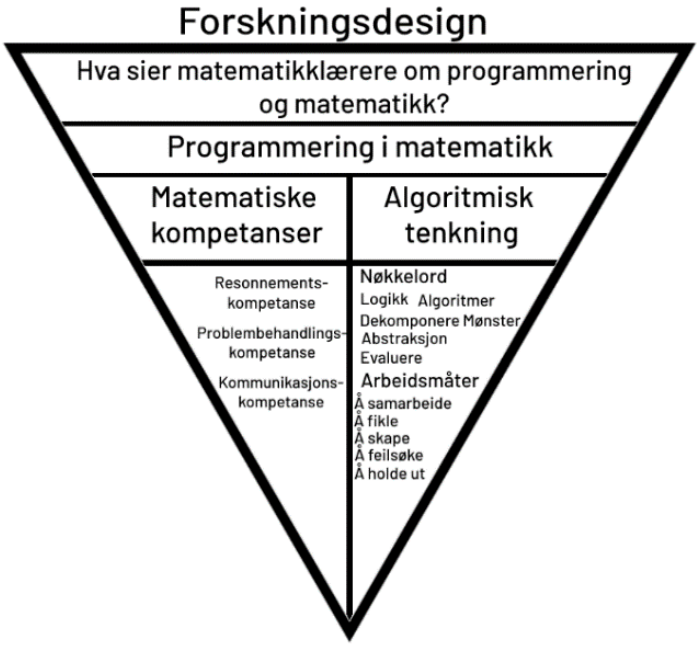
3 Metode

Det vil i dette kapittelet først komme en gjennomgang av valget av forskningsmetode i denne studien. Deretter vil kapittelet ta for seg hvordan datamaterialet har blitt samlet inn, en forklaring over hvem som er intervjuet, hvordan intervjuguiden er utformet og hvordan intervjuene er transkribert. En sentral del av dette kapittelet er også å forklare hvordan datamaterialet vil bli analysert i neste kapittel. Til slutt vil oppgavens reliabilitet, validitet og etiske hensyn gjennomgås

3.1 Valg av metode

Grunnlaget for datamaterialet i mitt forskningsprosjekt er kvalitative intervju med 4 matematikklærere fra Rogaland og Vestland, som i nyere tid har undervist i programmering på mellomtrinnet. Programmering i matematikkundervisningen er et nokså nytt fenomen i norsk skole, selv om det i noen andre land har foregått en stund allerede. Dette gjør at det er lite forskning på nettopp dette området. Cresswell (2014) sier at kvalitativ forskning er utforskende, og det kan være nødvendig med en slik tilnærming når emnet ikke har vært forsket mye på tidligere.

Den overordnede planen og strukturen med forskningen kan illustreres på denne måten:



Figur 3 Oppgavens forskningsdesign

Hensikten med denne masteroppgaven er å finne ut hva et utvalg av matematikklærere som underviser på mellomtrinnet sier om programmering, med et fokus på hvilke av Niss og Højgaard Jensens (2002) matematiske kompetanser de beskriver når de snakker om programmering, og hvilke aspekter ved algoritmisk tenkning de snakker om når de blir intervjuet om programmering. Det vil derfor være hensiktsmessig å samle den dataen som trengs gjennom en rekke intervjuer. Et intervju vil i motsetning til et spørreskjema, gi informantene mulighet til å uttrykke seg med stor frihet (Christoffersen & Johannesen, 2012, s. 78). Dette vil gjøre det mye lettere for dem å uttrykke sine tanker og erfaringer rundt temaet, og intervju «Gjør det mulig å få fyldige og detaljerte beskrivelser» (Christoffersen & Johannesen, 2012, s. 77)

3.2 Intervju

Målet med intervjuet er å hente nok informasjon til å kunne besvare forskningsspørsmålet mitt: «Hva sier et utvalg matematikklærere på mellomtrinnet om programmering i matematikkundervisningen?». Det vil i begynnelsen av intervjuet bli stilt spørsmål som er relativt enkle å besvare, og deretter når de har fått delt litt av det de tenker så kommer de mer krevende spørsmålene. Jeg prøver også å ikke stille ledende spørsmål slik at informantene svarer det de selv mener.

Dette intervjuet vil være et semistrukturert intervju, som betyr at rekkefølgen på spørsmålene ikke nødvendigvis er i den rekkefølgen de står skrevet i med unntak av åpningen og avslutningen av intervjuet. Dette vil gjøre intervjuet mer til en god og naturlig samtale enn en slavisk gjennomgang av spørsmålene.

3.3 Datainnsamlingen

Det vil være noen forutsetninger til informantene i denne studien. Informantene må ha matematikk som undervisningsfag på mellomtrinnet. De må også ha undervist i noen form for programmering i matematikkundervisningen siden den nye læreplanen ble innført. Dette gjør at helt ferske lærere blir utelukket, ettersom de ikke har nok erfaring i emnet. Det er til sammen 4 informanter fordelt på 3 forskjellige barneskoler. De har alle undervist ved hjelp av

programmering i matematikken, noen i større grad enn andre. Det var på forhånd uvisst hvor mye videreutdanning hver av informantene hadde hatt, men det ble antatt at de hadde noen form for kursing i programmering.

For å komme i kontakt med informantene, sendte jeg mail til tidligere lærere jeg hadde vært i praksis hos for å høre om noen på mellomtrinnet var interessert i å bli intervjuet. To av informantene fikk jeg kontaktet gjennom en lærer jeg kjente fra før med oversikt over de ansatte på en barneskole. For å få tak i den siste informanten hørte jeg med rektoren på en skole om det var noen som han tenkte kunne være interessert i et intervju om programmering. Responsen fra alle lærerne som ble spurt var varierende, og flere var ikke interesserte ettersom de selv ikke hadde noe erfaring med programmering i matematikkundervisningen de drev med.

3.3.1 Zoom

Gjennom prosessen med å finne informanter i Bergen, som ville gjort det mulig å intervjue dem fysisk, fant jeg ut at det ville være lettere for meg å spørre lærere jeg kjente fra hjemstedet mitt. Det ble gjort et forsøk på å intervjue disse informantene fysisk da jeg var hjemme en uke, men det lot seg ikke gjennomføre den uken, slik at da ble intervjuene nødt til å foregå over nettet. Gjennom en samtale med ansvarlig for LATACME-prosjektet om hvordan dette kunne bli gjort på en sikker måte for å ivareta anonymiteten til respondentene, ble vi enige om at det ville være mulig å ta opp intervjuene på zoom, og deretter raskt føre filene over på en kryptert minnepinne. Selve prosessen med å intervjue respondentene over Zoom gikk bra. E-post ble sendt ut til hver av dem med en lenke til Zoom-møte rett før planlagt tid. For å ha en tilnærmet lik vanlig intervjusituasjon hadde både jeg og respondenten på kamera, slik at vi kunne se hverandres reaksjoner til enhver tid.

3.3.2 Pilotintervju

Det vil ifølge Bryman (2012, s. 263) alltid være ønskelig å gjennomføre en pilot før et eventuelt intervju, spesielt dersom man har laget spørsmålene selv. Det er flere grunner til at man bør gjennomføre en pilot før man tenker å utføre de faktiske intervjuene. Når man foretar seg en pilot på intervjuguiden får man som intervjuer mer erfaring, som igjen kan gi en følelse av

selvtillit når man skal intervjue. For meg, som har minimal erfaring med intervju, var et pilotintervju skummelt, men på samme måte betryggende. Det gjorde også at jeg ble kjent med selve intervjusituasjonen. Bryman skriver videre at spørsmål som ser ut til å ikke bli forstått, vil bli synlig gjennom piloten, og at en pilot vil gjøre det mulig å identifisere de spørsmålene som kan få respondentene til å føle seg ukomfortable. Dette fikk jeg selv erfare da intervjuobjektet i piloten var kritisk til spørsmålene som det var vanskelig å svare på. Det kom altså frem gjennom pilotintervjuet jeg gjennomførte at noen av spørsmålene trengte å bli formulert på en annen måte for å gjøre det mindre uklart for respondentene hva de egentlig skulle svare på. Noe annet som ble synliggjort i løpet av piloten var at noen av spørsmålene ga bortimot samme svar. Her ble to spørsmål slått sammen og et annet spørsmål ble fjernet.

For å få selve intervjusituasjonen til å bli så autentisk som mulig, gjennomførte jeg et pilotintervju med en person som hadde omtrent like forutsetninger som respondentene. Personen er utdannet matematikklærer og underviser på flere trinn, deriblant mellomtrinnet. Personen har også videreutdanning i matematikk og programmering.

3.3.3 Intervjuguide

Kvale og Brinkmann (2021) foreslår å starte intervjuet med en konkret situasjon som vil gjøre det enklere for informanten å sette seg inn i tankegangen uten å sette seg fast. Deretter bygger man videre på svarene man får fra det første spørsmålet. Det første spørsmålet angående programmering i intervjuet vil derfor være om spesifikke erfaringer respondenten har til programmering.

Første delen av intervjuet vil bli viet til generelle spørsmål om matematikk, og hva matematikk er for respondenten. Videre vil de neste spørsmålene ha som hensikt å få informasjon om lærernes erfaringer, oppfatninger og meninger om programmering i matematikkundervisningen. Avsluttende vil spørsmålene være mer sentrert rundt begrepene matematisk kompetanse og algoritmisk tenkning. Intervjuet vil foregå sammenhengende, men dersom en av respondentene nevner noe som kan minne om et av de andre spørsmålene, vil det være naturlig å hoppe litt fram og heller komme tilbake til det spørsmålet som i

utgangspunktet var neste. Informantene vet på forhånd hva de har gått med på å bli intervjuet om, slik at de kan tenke seg til hvilken type spørsmål som kan komme, men de har derimot ikke fått utdelt spørsmålene. Noen av spørsmålene kan være litt krevende, og da blir de hjulpet i gang dersom det trengs.

3.3.4 Transkribering

Ved alle intervjuene var det en lydopptaker til stede. For å forsikre at lydopptakeren virket slik den skulle, ble det gjennomført testopptak før selve intervjuene begynte. Dette gjaldt også opptakene som ble gjort over Zoom, da en privat Zoom-samtale ble opprettet på forhånd for å teste hvordan opptak i Zoom fungerte. Dette gjorde det mulig å senere spille intervjuene av om igjen slik at jeg fikk transkribert intervjuene. Å transkribere noe innebærer det at man transformerer noe fra en form til en annen (Kvale & Brinkmann, 2021) I dette tilfellet vil det være fra muntlig til skriftlig form. Ved å skrive ned en så nøyaktig transkripsjon som mulig, vil analyseprosessen av intervjuene gå enklere for seg, ettersom det da ikke vil være nødvendig å åpne lydopptakene igjen.

Intervjuene inneholdt alle de samme spørsmålene, men informantene har svart med ulik mengde på dem. I løpet av noen av intervjuene med lærerne var det også til tider snakk utenom intervjutema. Dette ble sett på som uviktig, og er utelatt fra transkriberingen, men med tydelig markering hvor avsporingen skjedde. Lengden på opptakene varierer fra 25 minutter opptil 55 minutter.

3.4 Analyse av datamaterialet

3.4.1 Meningsfortetting

Under analysen i en kvalitativ studie der data man har samlet inn er gjennom intervju, vil man forsøke å gjøre noen fortolkninger av det informantene har sagt i intervjuene. Her ser man også etter ulike mønstre som kan forekomme i datamaterialet. Som intervjuer vil man danne seg ulike forestillinger av informantens meninger gjennom hele intervjuet. Videre vil man gjøre en ytterligere fortolkning av det informanten har sagt når man skal transkribere intervjuet i etterkant.

Analysen i denne studien baserer seg på meningsfortetting og meningsfortolkning. Meningsfortetting innebærer å korte ned informantenes uttalelser til kortere formuleringer (Kvale & Brinkmann, 2021). Av og til kan informantene si veldig mye om et bestemt emne. Det vil da være hensiktsmessig å gjengi deler av dette med få ord, uten å gi en annen mening til det som er sagt. Under er et utdrag fra transkripsjonen av intervjuet sammen med meningsfortettingen.

Transkripsjon av intervju	Meningsfortetting
S: Holdningen din til programmering, har den endret seg fra 2020 til nå? M3: Ja, det tror jeg. For i begynnelsen følte jeg det var viktig at elevene skulle få en oppskrift, gjennomføre den og gå videre. Men nå tenker jeg at fokuset er mye mer på det med at en får teste mye forskjellige, få bruke god tid på å putle, gjøre det på det nivået de er. Bruke tid på å bli kjent med de forskjellige tingene, og det at elevene får jobbe der de er, så noen jobber med å lage et blinkende hjerte på microbitten, og noen lager avanserte microbitter som styrer andre microbitter som igjen styrer en tredje ting. Så det er greit. Men kompetansen min er jo slik at da er de forbi meg, de som er flinkest, de som jeg hadde i 7. trinn i fjor som var flinkest, de programmerte jo i javascript og kjørte på, og jeg hadde ikke sjans til å henge med. **S: Åjaa, de skrev kode selv?** M3: De skrev koden istedenfor å blokkprogrammere det, slik som jeg må gjøre, og da er det på en måte ... min jobb å veilede de i feilsøking og det der med å bryte ned til små enheter og så se hvor er det feilen kan være, eller hvor er det det stopper opp. Så jeg tenker at jeg har inntatt en mer sånn type gå rundt og veileder på de enkle tingene, slik som at 2 like blokker da fungere ikke programmer, eller bare å klare å	Har endret holdning fra å slavisk følge oppskrift til mer utforskende for elevene. Elevene jobber med varierende vanskelighetsgrad De flinkeste i 7. klasse er forbi hans nivå. Hjelper med feilsøking

laste opp tingene. Det er mer den jobben jeg tar, og så er elevene selvgående på selve programmeringen eller kodingen.	Inntar en veiledende rolle

3.4.2 Meningsfortolkning

Meningsfortolkning er å fortolke meningsinnholdet i intervjuene man har gjennomført på en dypere og mer kritisk måte. En slik fortolkning vil gå utover det som direkte blir sagt av intervjuobjektene og «… finne frem til meningsstrukturer og betydningsrelasjoner som ikke fremtrer umiddelbart i transkripsjonen» (Kvale & Brinkmann, 2021). Det betyr altså at man prøver å tolke hva informanten egentlig mener med det som blir sagt. Her er et eksempel av min tolkning av det en av informantene svarte på hva han tenker programmering kan bidra med når det kommer til å øke den matematiske kompetansen:

> Jeg tror jo det at en kan nok få til ganske mye, men en må ikke glemme basisferdighetene heller. Det med å skjønne at når en da putter noe inn i en kalkulator eller inn i en kode så har det en betydning. Den gjør ikke alt for deg. Hvis de klarer å se at den kodingen ... En kalkulator er jo en slags kode det og, det er jo en algoritme som gjør at du kommer frem til svar. Og da må du ikke tro blindt på at hvis den spyr ut 742, så er dette rett. Du må ha en viss basiskunnskap i bakgrunnen «det var ikke det, det var 742000», men jeg tror nok det, ja.

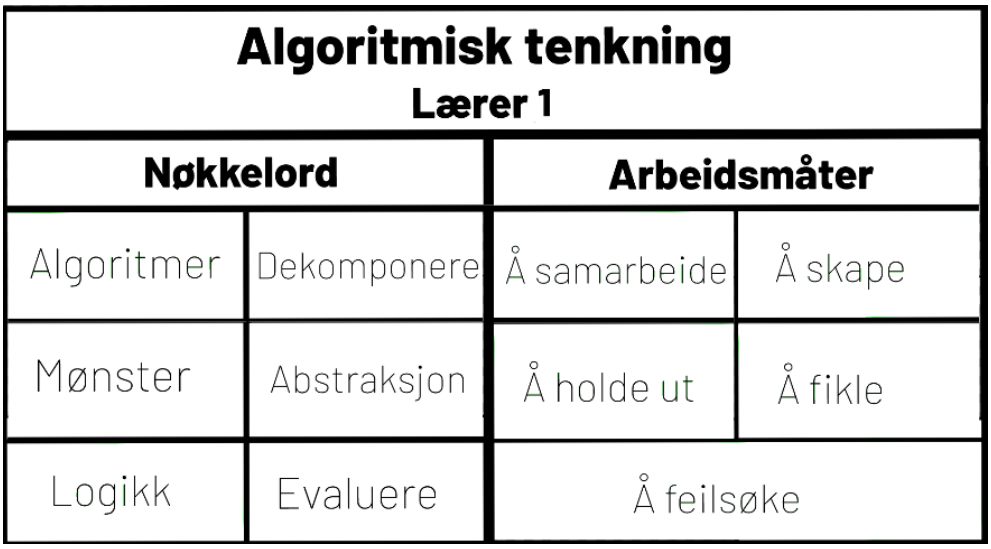
Jeg tolker dette som at informanten har tiltro til at programmering kan være med på å øke den matematiske kompetansen hos elevene. Selv om han mener at programmering kan bidra til dette, mener han at det er minst like viktig, og kanskje viktigere, at elevene har basiskunnskapene på plass før man går til verks, slik at man kritisk kan se på svaret om det gir mening. Slike fortolkninger vil variere i noen grad basert på hvem som gjør tolkningene. Det betyr at det ikke er sikkert at en annen person ville tolket transkripsjonen over på samme måte som det blir gjort i denne oppgaven.

3.4.3 Analysen

Oppgaven er sentrert rundt det et utvalg matematikklærere på mellomtrinnet har å si om programmering i matematikkundervisningen. Den har to forskningsspørsmål, hvor det ene er «Hvilke matematiske kompetanser beskriver lærerne i elevenes arbeid med programmering?», og det andre er «Hvilke aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering?». For å besvare forskningsspørsmålene, vil analysen i neste kapittel ha en todelt fremstilling for hver enkelt lærer, men før selve analysen, vil det først komme en kort presentasjon av hver enkelt lærer. Lærerne har forskjellig bakgrunn og holdninger til programmering, slik at for å forstå svarene til lærerne bedre, vil dette være hensiktsmessig.

I den første delen av analysen vil de matematiske kompetansene som lærerne beskriver, trekkes frem i analysen, sammen med eksempler fra intervjuet som underbygger kompetansene. Rekkefølgen på kompetansene er som følger; problembehandlings-, resonnerings- og kommunikasjonskompetansen.

Den andre delen av analysen vil handle om det andre forskningsspørsmålet, som går på aspektene ved algoritmisk tenkning. Her vil de forskjellige nøkkelbegrepene og arbeidsmåtene som hver lærer har snakket om bli trukket frem. Nøkkelbegrepene vil forekomme i denne rekkefølgen; logikk, algoritmer, dekomponering, mønstre, abstraksjon og evaluering, og arbeidsmåtene i denne rekkefølgen; å fikle, å skape, å feilsøke, å holde ut, å samarbeide. Når analysen av de nevnte aspektene er ferdig, vil det for hver lærer følge med en tabell, for å gi en oversikt over hvilke av aspektene som ble omtalt. Tabellen ser slik ut:



Figur 4 Aspekter ved algoritmisk tenkning hos lærerne

Gjennom hele analysen vil det tilstrebes å først presentere det aktuelle spørsmålet læreren svarer på, deretter vil det komme et utdrag av hva læreren sier, og til slutt vil dette gjenfortelles eller tolkes til en av kompetansene eller aspektene ved algoritmisk tenkning. Det kan forekomme at dette mønsteret brytes for å noen steder få bedre flyt. Det vil også være en mulighet at hverken de matematiske kompetansene eller aspektene ved algoritmisk tenkning ikke blir nevnt med navn i intervjuene. Dette fører til at store deler av analysen vil være basert på en meningsfortolkning, som fortalt i *3.5.2 Meningsfortolkning.*

I analysen kan det oppstå at det samme utdraget, eller deler av det samme utdraget blir brukt både når det kommer til de matematiske kompetansene, og når det kommer til noen av aspektene ved algoritmisk tenkning. Dette vil skje ettersom de har med hverandre å gjøre. Et eksempel her kan være et utdrag der læreren snakker om noen elever som jobber sammen. Dette utdraget kan bli tatt i bruk for å vise at elevenes kommunikasjonskompetanse heves, og det samme utdraget kan tas frem for å vise at samarbeid er en av arbeidsmåtene elevene jobber med når de programmerer i matematikk. Der hvor det ikke er mulig å gjenkjenne et bestemt aspekt ved algoritmisk tenkning, vil disse ramses opp i slutten analysen.

3.5 Validitet og Reliabilitet

Ifølge Cohen, Morrison og Manion (2007) er den mest praktiske måten å oppnå økt validitet på å være så lite partisk som mulig. Det er flere kilder som kan gjøre at intervjuet ikke blir helt upartisk. Dette går både på intervjueren og respondenten. Cohen et al. nevner at holdningene, meningene og forventningene til intervjueren kan bidra til dette. De skriver også som en kilde til et partisk intervju at intervjueren med vilje stiller spørsmålene slik at svarene blir akkurat det han er ute etter. Til slutt er det viktig at det ikke oppstår noen feiloppfatninger fra intervjuerens side om hva som faktisk blir sagt, og at spørsmålene blir stilt slik at det ikke kan oppstå misforståelser.

Agar (Agar, referert i Cohen et al,. 2007) kommer med en påstand at ved bruk av en kvalitativ datainnsamling vil den personlige involveringen og dybdesvarene til individene gi et tilstrekkelig nivå av både validitet og reliabilitet. Datamaterialet i denne masteroppgaven er utelukkende hentet gjennom kvalitative intervjuer. Ifølge Agar vil dette gi et visst nivå av validitet og reliabilitet. Det er derimot en rekke ting jeg ville gjort annerledes dersom datamaterialet skulle blitt samlet inn på nytt, og som ville gjort dette til en potensielt bedre oppgave.

Alle lærerne sa ja til å være med på selve intervjuet, og de var klar over at det var programmering som var på agendaen. Det ble allikevel klart for meg at noen av spørsmålene som ble stilt, kunne for informantene oppleves som krevende å svare på. Det ville vært hensiktsmessig å aktivere informantene på en annen måte en slik det ble gjort. På spørsmål som «Hva legger du i begrepet algoritmisk tenkning?», vil det for lærerne som ikke nettopp har vært inne og lest på dette, være vanskelig å komme på et bra svar på kort betenkningstid. Her burde jeg gjort et bedre arbeid med å peile lærerne inn på disse litt krevende spørsmålene. Potensielt kunne de også fått beskjed på forhånd at dette var noe som ville bli snakket om i intervjuet, slik at de kunne forberedt seg litt.

En annen ting som må påpekes er at jeg i ettertid ser at intervjuene tok plass for tidlig i forskningsprosessen. Dette har ført til en ufullstendig intervjuguide med tanke på hva forskningsspørsmålene innebærer. Da intervjuene ble gjennomført, var det ikke avklart hvilke

av de matematiske kompetansene jeg skulle fokusere på, som igjen forklarer hvorfor det er en mangel på spørsmål rundt de spesifikke kompetansene. Som en konsekvens av disse tidlige intervjuene, var spørsmålene som ble stilt derfor relativt generelle. Dersom jeg hadde hatt dette klart på forhånd, kunne jeg for eksempel spurt spørsmål som gikk direkte på de bestemte kompetansene, slik som: «Hva tenker du om samarbeid som arbeidsmåte når elevene jobber med programmering i undervisningen?». Dette spørsmålet vil gi informasjon som kan brukes til kommunikasjonskompetansen, og samtidig vil det også avklare om dette aspektet ved algoritmisk tenkning er noe de tenker over.

3.6 Etiske hensyn

Alle lærerne ble tildelt et informasjonsskriv (Vedlegg 1) med informasjon om mitt masterprosjekt, samt informasjon om selve prosjektet LATACME, som min oppgave er en del av. I dette informasjonsskrivet står det blant om deres rettigheter som informant i dette prosjektet, og det står blant annet at de har rett på å få se hva som skrives om dem, og eventuelt få slettet sin personinfo. I skrivet står det også hva de blir med på, hvordan data blir tatt vare på for å øke sikkerheten og kontaktinfo til de sentrale personene i prosjektet. Dette var noe de alle var nødt til å skrive under på for å delta i intervjuet. Samtidig som de fikk tildelt dette informasjonsskrivet, fikk de også tildelt et samtykkeskjema (Vedlegg 2), for å være sikker på at de synes det var greit at lyden ble tatt opp.

En av de etiske utfordringene jeg sto overfor var hvordan data skulle oppbevares. Data i en slik innsamling skal ikke være på en enhet som er tilkoblet internett. Ettersom noen av mine intervju er gjort over en samtale på zoom, vil det være umulig å ikke være tilkoblet internett. Løsningen på dette ble at så fort intervjuet var ferdig, ble internettet slått av, og data ble flyttet over på en kryptert minnepinne, som senere ble brukt når intervjuene skulle transkriberes.

4 Analyse og resultater

Denne oppgavens overordnede problemstilling er å se på hva et utvalg matematikklærere på mellomtrinnet sier om programmering, med et fokusområde på matematiske kompetanser og algoritmisk tenkning. Oppgaven har blant annet et mål om å finne ut noe om lærernes kunnskap og forståelse når det kommer til å ta i bruk programmering i matematikkundervisningen. I dette kapittelet vil intervjuene til en og en lærer bli analysert. Lærerne har forskjellige utgangspunkt med tanke på bakgrunn innenfor programmering, så før selve analysen av hvert intervju begynner, vil det være en kort beskrivelse av hvor mange studiepoeng de har i matematikk, eventuelle studiepoeng i IT og hvilken holdning de har til programmering. Analysen av intervjuene vil først ta for seg problembehandlingskompetansen, deretter resonneringskompetansen, og til slutt kommunikasjonskompetansen. Deretter vil analysen først ta for seg nøkkelbegrepene til algoritmisk tenkning; logikk, algoritmer, dekomposisjon, mønstre, abstraksjon og evaluering, etterfulgt av arbeidsmåtene til den algoritmiske tenkeren; å fikle, å skape, å feilsøke, å holde ut og å samarbeide. For ordens skyld vil de nevnte aspektene legges frem i rekkefølgen som nettopp ble presentert, og dersom noen aspekter ikke forekommer i intervjuet, vil de nevnes i slutten av hvert delkapittel. For hvert aspekt vil det først forklares hvilket spørsmål lærerne svarer på, deretter presenteres hva de har sagt, etterfulgt av en gjenforklaring eller tolkning rundt dette for å koble det til et bestemt aspekt. Avslutningsvis i kapittelet vil komme en felles oppsummering som tar for seg både de matematiske kompetansene og aspektene ved algoritmisk tenkning.

4.1 Lærer 1

Lærer 1 er den første av informantene som ble intervjuet. Han har 15 studiepoeng i matematikk fra lærerskolen da han først ble lærer, og har i senere tid tatt 30 nye poeng i matematikk via Høyskolen. Han har også noen studiepoeng i IT fra lærerutdanningen, som ifølge han selv gjorde det lettere å sette seg inn i de nye kravene. Siden den nye læreplanen trådte i kraft for tre år siden (2020), har han undervist i først 5. klasse og 6. klasse, og programmering er noe han ser positivt på.

4.1.1 Analyse av matematisk kompetanse i intervjuet med Lærer 1

4.1.1.1 Problembehandlingskompetanse

Den første matematiske kompetansen som kommer frem i intervjuet er problembehandlingskompetansen. Gjennom intervjuet gjentar han flere ganger at elevene ikke gidder når de får noe som kan minne om problemløsningsoppgaver. Dette kommer blant annet frem når han snakker om hvilke tanker han har om at programmering har blitt en del av matematikkfaget, hvor han sier «… for det blir litt sånn der, «klikketyklikk-generasjonen», sant. At de bare klikker mange nok ganger så får de det til. Det at du klarer å holde det til den problemløsningen…». Her trekker han frem problemløsning, og det kommer fram at han tenker det er vanskelig for mange av elevene. I et annet eksempel fra klasserommet, hentet fra samme spørsmål, nevner han at elevene skal programmere noe som kan minne om telefonspillet Angry Birds, hvor han forklarer at «…de jo mer opptatt av å klare og fly 1000 meter, sant. Da skal det ikke være noe feil. Hvis de da støter på en hindring skal de ikke falle ned, de skal bare få utrolig mye poeng». På en måte kan man si at elevene ikke gjør problemløsningsoppgaven de har blitt tildelt, men på en annen måte løser de jo en problemløsningsoppgave selv om det ikke er den tildelte, nemlig å få den fuglen til å fly langt, uten å stoppe. Så deres problemløsningskompetanse er i bruk under oppgaven.

4.1.1.2 Resonnementskompetanse

Ved å se gjennom intervjuet med Lærer 1, kommer resonnementskompetansen tydelig frem. Dette blir først synlig når han skal svare på hva han tenker den viktigste jobben til læreren i matematikkfaget er. Han nevner at det er en stor utfordring å få alle elevene til å skjønne det de holder på med, og «Få de til å være med på tankegangen, ikke nødvendigvis at de skal kunne den og den formelen, men skjønne hva de holder på med». Her snakker han først om at han tenker det er viktig at elevene er med på tankegangen, før han sier videre «At de klarer å knytte det opp til den praktiske biten. … Men hvis en klarer da å knytte det inn i vanlige situasjoner». Det er dette resonnementskompetansen handler om. For det første klare å forstå matematikken, og på bakgrunn av noe de har lært i matematikk, resonnere seg frem til hvordan matematikken kan bli anvendt på et annet området.

Senere blir han spurt om hva han tenker programmering kan bidra med når det kommer til å øke den matematiske kompetansen til elevene. Her er det første han trekker frem noe som kan minne om det han svarte på hva han tenker er den viktigste jobben til matematikklæreren. Han sier at «Hvis du kan få til at elevene ser en vridning, og at det elevene gjør, det de øver seg på, er noe de selv kan få bruk for i ulike ting». Her snakker han om hvordan programmering kan bidra til at elevene blir bedre til å gjenkjenne det de lærer av programmering i andre områder. På bakgrunn av disse utsagnene er det liten tvil at resonnementskompetanse er noe han ønsker at elevene hans skal bli bedre på.

4.1.1.3 Kommunikasjonskompetanse

Den siste matematiske kompetansen er kommunikasjonskompetansen. Dette er en kompetanse som blir tidlig omtalt i intervjuet med Lærer 1. Da han ble spurt om hva han tenker er den viktigste jobben til læreren svarer han blant annet følgende «Haha, ja, da skal jeg jo si slik som utdanningen sier, man skal snakke matte», men allikevel er det bortimot ingen eksempler fra intervjuet der han snakker om noe som kan minne om kommunikasjonskompetanse.

4.1.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 1

4.1.2.1 Nøkkelbegrep

Gjennom analyse av intervjuet kommer det frem flere nøkkelbegreper. På spørsmålet om hva han mener programmering kan bidra med i matematikkfaget når det kommer til å øke den matematiske kompetansen, svarer han følgende:

> En kalkulator er jo en slags kode det og, det er jo en algoritme som gjør at du kommer frem til svar. Og da må du ikke tro blindt på at hvis den spyr ut 742, så er dette rett. Du må ha en viss basiskunnskap i bakgrunnen «det var ikke det, det var 742000»

Her sier han først at en kalkulator også krever en algoritme for å komme frem til svaret, og deretter sier han at det svaret man får på kalkulatoren ikke nødvendigvis er det man er ute etter. Dette kan tolkes som at **logisk** tenkning er nødvendig dersom man skal operere en kalkulator, eller programmering generelt. Videre sier han «Få de til å være med på

tankegangen, ikke nødvendigvis at de skal kunne den og den formelen, men skjønne hva de holder på med». Man kan se at det er viktig for han at elevene forstår hva de gjør og hvorfor. Det kommer frem at logikk er noe han tenker er viktig.

Når han blir spurt direkte om han er kjent med begrepet algoritmisk tenkning svarer han: «ehh, det datt litt ut for det var en som prøvde å ringe meg, jo... Men det er jo den biten der da med å klare og knytte det til å se de algoritmene. Hva er det som skal skje?». Det kan virke som han ikke kom på dette i farten ettersom han mistet litt fokus da telefonen ringte mens vi snakket over Zoom. Han nevner dette med **algoritmer**, men utdyper ikke hva det betyr å tenke algoritmisk da han blir spurt. Dette betyr derimot ikke at læreren ikke forklarer noen av nøkkelbegrepene når det kommer til den algoritmiske tenkeren. Et av disse kan man se når han snakker om at han tenker at programmering kan øke elevenes matematiske kompetanse «Hvis du kan få til at de ser en vridning og at det de gjør, det de øver seg på, er noe de selv kan få bruk for i ulike ting». Han snakker her om at programmering kan bidra til en økt kompetanse i matematikk dersom elevene klarer å se at det de gjør i programmeringen kan anvendes i andre deler av matematikken som ikke har med programmering å gjøre. Her er det snakk om at elevene klarer å se forskjellige **mønstre** og likheter mellom programmering og andre ting.

Det er ingenting som kan minne om evaluering, abstraksjon eller dekomponering i intervjuet.

4.1.2.2 Arbeidsmåter

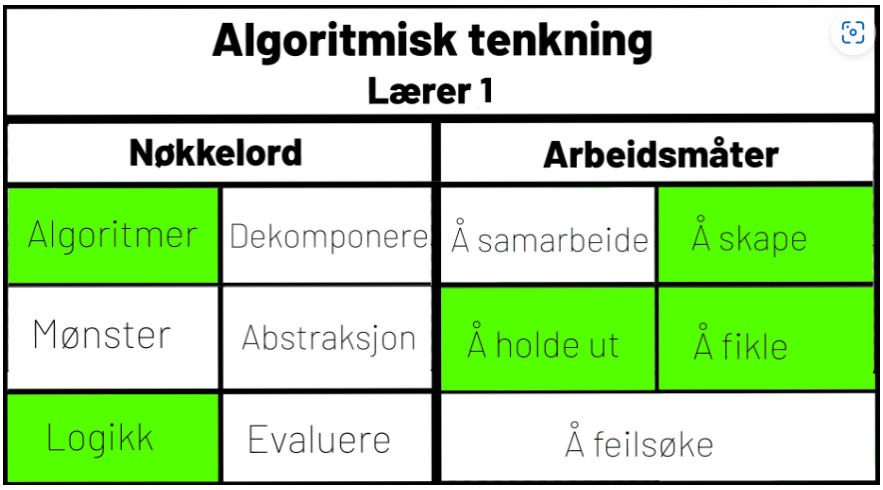
Gjennom analyse av intervjuet med Lærer 1 kan man se at flere av arbeidsmåtene har blitt nevnt på ulike steder i intervjuet. Mens han snakker om hvordan de bruker noen av læremidlene de har tilgjengelig på skolen sier han «Men det er jo det med å prøve seg, det med å teste ut ting, det med å leke seg litt med dette og å finne ut av det». Her kommer det frem at han tenker at flere av læremidlene de har tilgjengelig i arbeid med programmering kan brukes til å prøve seg frem og teste ut ting, eller **å fikle**, som er det samme. Senere blir han spurt om hvordan programmering kan bidra til at elevene får jobbe med blant annet algoritmisk tenkning, og da trekker han igjen frem «For da er det mer at de skal utforske, og

de skal prøve seg. Okei, da må de jo teste». Han trekker også her frem det med å utforske og prøve seg frem, som viser at å fikle er en arbeidsmåte han tenker er viktig.

Mens han snakker om erfaringer fra klasserommet, trekker han fram et prosjekt elevene skulle ha, der programmering var en sentral del. Prosjektet var i Minecraft og her skulle de «…bygge opp ei blokk der de hadde hver sin leilighet og de skulle skrive litt om dette». Dette prosjektet handlet om å designe og lage noe eget. Det skulle **skapes** noe, som er enda en av arbeidsmåtene. Dette kommer også frem senere i intervjuet når han snakker om et annet læremiddel de har benyttet, Lego WeDo, som er et sett med legoklosser og et programmeringsverktøy for PC. Her sier han «Ja… vi har brukt Lego WeDo. Da klarer du å få den kreativiteten med å bygge, sant. Å se noe som man skal få til». Her trekker han igjen fram dette med å bygge, eller skape, noe eget.

Noen av arbeidsmåtene nevner han ikke med ord, men man kan allikevel skimte dem om man ser gjennom intervjuet. Når han snakker om positive erfaringer fra undervisningen, sier han følgende «Men noen har jo prøvd og prøvd. Det var noen jenter som prøvde og prøvde, så klarte de det til slutt». Det fortelles om noen jenter i klassen som ikke ga opp. En arbeidsmåte som er viktig for den algoritmiske tenkeren er det **å holde ut**, og det tolkes her som at elevene holder ut når de prøver igjen og igjen.

Gjennom en analyse av intervjuet hvor det blir sett etter om han nevner noen av nøkkelbegrepene eller arbeidsmåtene til den algoritmiske tenkeren, eller noe som kan minne om disse, kommer det frem at noen av nøkkelbegrepene uteblir. I hans forklaring av algoritmisk tenkning kommer det ikke frem en utdypende forklaring av hva dette begrepet innebærer. Han nevner noen av nøkkelbegrepene på ulike steder i intervjuet, mens nøkkelbegrep som dekomponering, abstraksjon og evaluering ikke blir snakket om. Når det kommer til arbeidsmåter er han innom flere, men det kommer ikke frem noe som kan tolkes som å feilsøke og å samarbeide. Under er en tabell med oversikt over hvilke aspekter som er å finne i intervjuet.



Figur 5 Aspekter ved algoritmisk tenkning hos Lærer 1

4.2 Lærer 2

I utdannelsen sin har han det som han selv kaller «kvartårseining» i matematikk, som er 15 studiepoeng. Han har ikke tatt noen flere studiepoeng i matematikk de senere årene, og har heller ingen formell kompetanse når det kommer til IT. De siste to årene har han undervist i 6. og 7. klasse, men bare en klasse om gangen. Skolen han jobber på var tidlig ute på dette med programmering, så han har benyttet dette i noen år allerede, og er veldig positiv til programmering i undervisningen generelt.

4.2.1 Beskrivelser av matematisk kompetanse i intervjuet med Lærer 2

4.2.1.1 Problembehandlingskompetanse

Når det kommer til om han beskriver noen av kompetansene, er dette til tider vanskelig å legge merke til ettersom svarene ofte er litt korte, og det ofte er vanskelig å finne samsvar med spørsmålene. Det virker også til tider som om programmering bare blir brukt fordi både han og elevene synes det er gøy. Dette betyr derimot ikke at han ikke er innom noen av de matematiske kompetansene som det sees etter. Når han skal svare på hvilke vurderinger han gjør før han velger programmeringsoppgaver til elevene sier han «… jeg elsker de oppgavene «prøv, test dette her ut», klarer de faktisk å få denne kula over den broa. Slik at de ikke er så matematisk.». I dette utdraget skildrer han en oppgave hvor de får tildelt et problem, som elevene blir nødt til å finne en løsning på. En slik oppgave vil ta utgangspunkt i elevenes **problembehandlingskompetanse**. Videre når han skal snakke om programmering kan bidra

til at elevene får jobbe med utforskning og problemløsning, svarer han «Ja, det blir jo litt i samme gate som jeg svarte på det andre. Så definitivt mener jeg det.». Her tolkes det som at han viser til det forrige spørsmålet, der han snakket om vurdering rundt programmeringsoppgaver, hvor han skildret en problemløsningsoppgave. Man kan derfor si at problembehandlingskompetansen er til stede i intervjuet, men er noe han snakker lite om.

4.2.1.2 Resonnementskompetanse

Når det kommer til **resonnementskompetansen**, er det vanskelig å finne spesifikke eksempler fra intervjuet der han snakker om noe som kan minne om dette.

4.2.1.3 Kommunikasjonskompetanse

Et av spørsmålene spør om hva elevene hans gjør i en «vanlig» matematikktime. Han nevner her at elevene ofte må «… snakke litt og reflektere både høyt eller med læringspartner». Kommunikasjonskompetansen handler om nettopp dette, kommunisere gjennom å snakke i, med og om matematikk. Videre når han snakker om positive erfaringer i programmeringen, sier han blant annet at «… gjennom samarbeid har de løst ganske avanserte oppgaver uten at de selv har tenkt at de har jobbet med så mye matematikk». Igjen viser han til at elevene jobber sammen, både når de jobber til vanlig og når de har programmering på agendaen. Programmering i grupper handler nettopp om å kommunisere både muntlig, skriftlig og visuelt, og det kan tolkes som at kommunikasjonskompetansen er noe som han tenker er sentralt i arbeid med programmering i matematikkundervisningen.

4.2.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 2

4.2.2.1 Nøkkelbegreper

På spørsmålet om han er kjent med begrepet "algoritmisk tenkning", svarer han fort og kort ja, men videre når blir han bedt om å forklare begrepet, fremstår han som mer usikker. Han gjentar spørsmålet for seg selv før han blir spurt om hvordan han ville forklart begrepet.

> Nei, at de skal tenke ... Nei, jeg vet faktisk ikke. Jeg vet ikke helt om jeg klarer å finne noe ord på å oversette algoritmisk tenkning. At det er ... nei, jeg vet ikke. Det er for tidlig på morgenen *Ler*.

Det er en mulighet at han ikke kom på det i farten. Han nevner heller ikke noen av nøkkelordene ved den algoritmiske tenkeren på selve spørsmålet om han vet hva begrepet innebærer, men man kan derimot finne noen av disse nøkkelbegrepene i løpet av intervjuet.

På spørsmålet om hvordan han tenker at programmering kan bidra til at elevene får jobbe med algoritmisk tenkning sier han blant annet at «… de kanskje ikke ser logikken av det med en gang de starter…». Dette kan tolkes som at logikk er noe han tenker er en del av algoritmisk tenkning og at programmering kan bidra med den **logiske** delen av algoritmisk tenkning. Dette kan tolkes som at han har fått med seg hva begrepet til en viss grad handler om. På spørsmålet om han tenker at programmering kan bidra til økt matematikkforståelse, svarer han blant annet «…av og til er det greit å følge de oppskriftene for at de skal skjønne de prinsippene og hva det går ut på…». Her forklarer han det **algoritmer** går ut på, nettopp å følge en oppskrift.

4.2.2.2 Arbeidsmåter

En ting er å forklare hva algoritmisk tenkning innebærer, og en annen ting er hvordan man arbeider med det. Gjennom intervjuet snakker han mye om hva elevene gjør i timen, og hvordan de arbeider mye med arbeidsmåter som karakteriserer den algoritmiske tenkeren. Utdraget under er hentet fra det han svarer på om han tenker om programmering kan bidra til økt matematikkforståelse for noen:

> … av og til er det greit å følge de oppskriftene for at de skal skjønne de prinsippene og hva det går ut på, men jeg elsker de oppgavene «prøv, test dette her ut», klarer de faktisk å få denne kula over den broa.

Dette med å prøve seg frem, kan man også kalle for **å fikle**, som er en av arbeidsmåtene. Ved å gi elevene oppgaver der de ikke umiddelbart kan se hva som trengs, og uten at de har fått en oppskrift, vil elevene begynne å fikle for å se om det de ønsker skal skje skjer. Å få elevene til å prøve seg frem er ikke eneste arbeidsmåten som kommer frem i intervjuet. Selv om han ikke bruker ordet direkte, så kan utdraget under, som er hentet fra når han snakker om

hvordan programmering kan bidra til algoritmisk tenkning, tolkes som at han vil at eleven/elevene skal feilsøke koden for å se hva eller hvor feilen kan være:

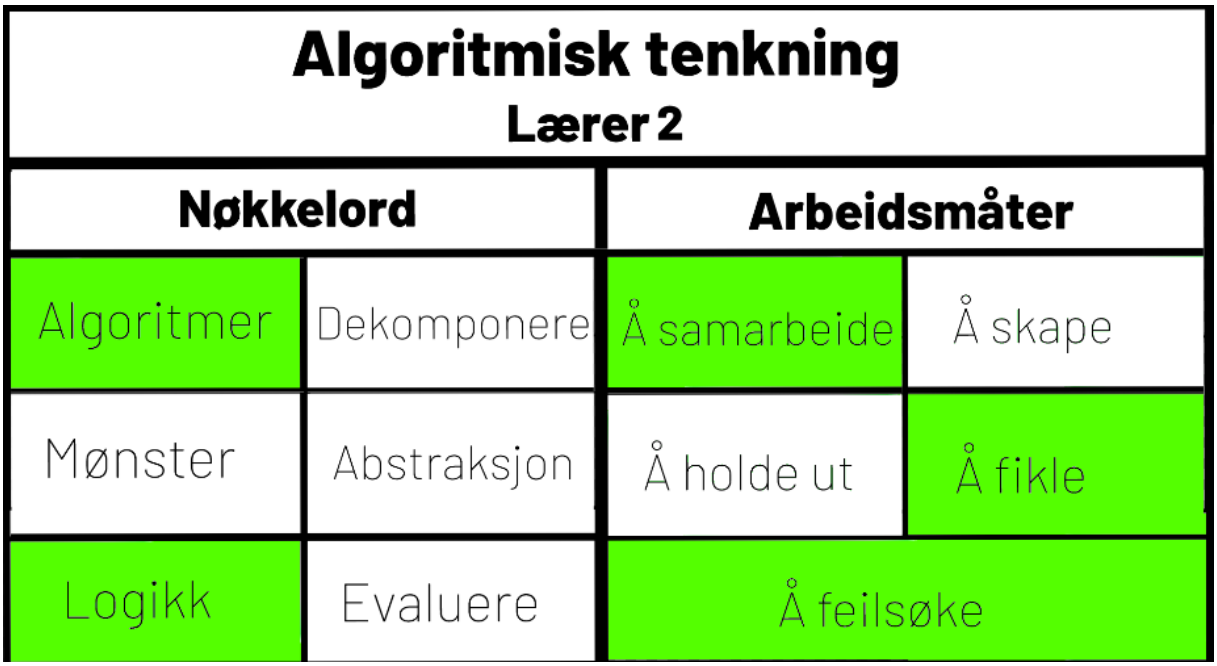
> «men se her, her har du skrevet i koden din at det skal skje, og så skjedde noe annet, hvorfor det?», at de på en måte kan gå inn å både arrestere seg selv, og ja. At de skjønner ting litt etter at de faktisk har gjennomført operasjonene.

Ved å spørre elevene hvorfor noe annet enn det de ønsket skjedde, får han elevene til å se gjennom koden sin for å se etter hva som kan være feil. Dette er enda en av arbeidsmåtene, altså det **å feilsøke**. Videre på spørsmålet om hvilke positive erfaringer han har hatt med programmering i undervisningen, svarer han:

> Og samarbeid, gjennom samarbeid har de løst ganske avanserte oppgaver uten at de selv har tenkt at de har jobbet med så mye matte. Så det er kanskje det gøyeste med programmeringen slikt sett.

Vi kan se i utdraget at han spesifikt snakker om at elevene hans **samarbeider** når de skal løse oppgaver i programmeringen. Dette er også en av arbeidsmåtene til den algoritmiske tenkeren.

Gjennom analyse av intervjuet fremstår det som om programmering er noe både han og klassen synes er gøy. Selve begrepet «algoritmisk tenkning» sier han selv at han ikke helt vet hvordan han ville forklart. Det er derfor noen mangler når det kommer til nøkkelbegrepene i modellen, som det å dekomponere, evaluere, finne mønstre og abstraksjon. Selv om han ikke forklarer alle begrepene ordrett, kommer det frem at elevene hans jobber med flere av arbeidsmåtene som kjennetegner en algoritmisk tenker. Under er en tabell som viser hvilke av nøkkelbegrepene og arbeidsmåtene som dekkes i intervjuet med Lærer 2.



Figur 6 Aspekter ved algoritmisk tenkning hos Lærer 2

4.3 Lærer 3

Lærer 3 er den tredje informanten som ble intervjuet. I matematikk har han 60 studiepoeng. Med unntak av ulike kurs, har han ingen formell kompetanse når det kommer til programmering. I de siste årene har han undervist programmering i 5., 6. og 7. klasse, men har bare en klasse om gangen. Han hadde drevet litt med programmering i skolesammenheng før innføringen i 2020, men mesteparten av hans kompetanse stammer fra at dette er en personlig interesse. Han tenker at det er en selvfølge at programmering skal inn i skolen, og mener at det er naturlig å plassere det i matematikkfaget.

4.3.1 Beskrivelser av matematisk kompetanse i intervjuet med Lærer 3

4.3.1.1 Problembehandlingskompetanse

Etter å ha sett på hvilke matematiske kompetanser som Lærer 3 beskriver i intervjuet, kommer alle kompetansene tydelig frem. Han har gjort det enkelt å gjenkjenne de ulike matematiske kompetansene med å svare utdypende på det spørsmålene faktisk spør om. Tidlig i intervjuet blir han spurt om hva han tenker er den viktigste jobben til læreren i matematikkfaget, og svarer følgende:

> Jeg tenker at det med interesse og gode strategier, det er kanskje det aller viktigste. Når de møter en oppgave eller en utfordring så vet de litt hvordan de skal løse det, eller har noen ting som de kan hente fram.

Vi kan se i utdraget at han nevner at det er viktig at elevene har gode strategier når de møter utfordringer. Så fort en oppgave er utfordrende, og elevene ikke har en kjent bestemt måte å løse den på, er det snakk om **problembehandlingskompetanse.** I utdraget kan vi se at han tenker det er viktig at elevene blir bedre på problembehandlingskompetanse, og han sier at dette kan gjøres ved å gi dem gode strategier til å løse problemer.

Problemløsning er et emne som blir nevnt flere ganger i løpet av intervjuet. På spørsmålet om han tenker programmering kan bidra til det nye kjerneelementet utforskning og problemløsning, sier han blant annet at «Så jeg tenker det er masse muligheter når det gjelder problemløsning og utforskning i kodingen…», og «Altså, mange av de oppgavene som ligger ferdig om programmering er jo slike problemløsningsoppgaver». Det kommer tydelig frem at han tenker at problemløsning er en sentral del av programmering, og når han blir spurt om hvordan han tenker programmering kan bidra til at elevene får jobbe med problemløsning og tenke algoritmisk, svarer han:

> Det er jo det med at når de koder et program, så koder de det for å komme frem til en løsning. Det er jo på en måte problemløsningen, eller vi har og hatt noen oppgaver «Hvordan kan vi bruke Micro:Bit eller teknologi til å for eksempel løse miljøutfordringer».

I utdraget forklarer han hvordan det å kode, eller programmere, i seg selv er en form for problemløsning. Han trekker deretter frem en av oppgavene de har jobbet med i klasserommet. Denne oppgaven er uten tvil en oppgave som vil styrke elevenes problembehandlingskompetanse, ettersom elevene ikke har noen «rutine» for å komme frem til en løsning, og vil kreve en form for matematisk undersøkelse for å løses. Det er også flere eksempler av problemløsningskompetanse i intervjuet, men dette er de mest åpenbare.

4.3.1.2 Resonnementskompetanse

Neste kompetanse som også blir beskrevet i intervjuet når han snakker om diverse programmeringssituasjoner er **resonnementskompetansen**. Dette snakker han blant annet om når han skal snakke om hvordan han tenker programmering kan bidra til å øke den matematiske kompetansen til elevene. Han nevner følgende i sammenheng med det å feilsøke et feil svar:

> Og få de til å forstå det at hvis jeg ser på det, og det ikke stemmer overhodet med oppgaven, hvor er det feilen har skjedd, eller hvis du får svaret 0,1cm og du startet med 2km har du gjerne regnet feil en plass.

I eksempelet snakker han om at programmering kan gjøre elevene flinkere til å forstå hva som kan være feil når de ikke har riktig svar. Når svaret man har fått ikke stemmer med det man i utgangspunktet startet med, kan altså programmering bidra til at elevene ser en logisk brist. Dette kan tolkes som at han mener at elevene vil enklere kunne resonnere seg frem til at svaret de har fått må være feil. Resonnementskompetansen handler blant annet om å avgjøre om et matematisk bevis er gyldig, eller som i dette eksempelet, om svaret kan stemme.

4.3.1.3 Kommunikasjonskompetanse

Når det kommer til **kommunikasjonskompetansen**, er denne også noe han beskriver når han snakker om programmering. På spørsmålet om han har noen positive erfaringer fra klasserommet når det kommer til programmering i matematikkundervisningen, snakker han først om en oppgave elevene hadde gjort der de samarbeidet, og deretter sier han «Altså hver gang jeg har programmering så er alle elevene med. Så du får veldig deltagende og aktive elever, og så blir de utrolig flinke til å hjelpe hverandre.». Fra utdraget kan vi se at han snakker om hvordan elevene er flinke til å hjelpe hverandre. For at elevene skal kunne hjelpe hverandre med programmeringsoppgaver, er de nødt til å kommunisere matematisk, og kommunikasjonskompetansen kommer tydelig frem. Samtidig som de kommuniserer verbalt, vil en slik samhandling mellom elevene i en programmeringssammenheng, også innebære en form for visuell eller skriftlig form for kommunikasjon, som forsterker kommunikasjonskompetansen i eksempelet.

4.3.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 3

4.3.2.1 Nøkkelbegrep

Lærer 3 har ved flere anledninger gjennom intervjuet nevnt både nøkkelbegreper innenfor algoritmisk tenkning, uten å bli direkte spurt om det. Vi kan se i det følgende utsagnet, hentet fra spørsmålet om programmering kan øke den matematiske forståelsen, at han tenker at programmering kan gjøre elevene flinkere til å se logiske løsninger:

> Og få de til å forstå det at hvis jeg ser på det, og det ikke stemmer overhodet med oppgaven, hvor er det feilen har skjedd, eller hvis du får svaret 0,1cm og du startet med 2km har du gjerne regnet feil en plass.

I eksempelet er det snakk om å kunne se at svaret man har fått er feil, basert på oppgaven man skulle regne i utgangspunktet. Dette kan tolkes som at han mener at programmering vil gjøre elevene bedre på blant annet å resonnere seg frem til om svaret kan gi mening. Det er dette **logikk** i den algoritmiske tenkeren handler om.

Når han blir spurt direkte om han er kjent med selve begrepet algoritmisk tenkning svarer han først «Ja, litt». Han blir forsikret om at dette er noe han egentlig har snakket litt rundt tidligere. Han svarer da følgende:

> Når jeg tenker «å tenke algoritmisk» så er det å bryte ned litt større enheter til en sånn steg for steg måte å komme frem til et svar på. Så man går på en måte fra en helhet til deler, men man kan jo også gå andre vei og, å bygge på del til del og komme frem til en ferdig algoritme.

Han snakker her om å bryte ned, eller **dekomponere**, noe som i utgangspunktet er stort og potensielt vanskelig, til noe som er mindre og håndterlig. Han nevner også dette med steg for steg, og hvordan dette kan gjøres andre vei, at man setter sammen flere mindre deler for å deretter sitte igjen med en ferdig **algoritme**, som også er et av nøkkelbegrepene. Dekomponering kommer også tydelig frem når han svarer på spørsmålet om hvordan han

tenker programmering i matematikken kan bidra til at elevene får jobbe med problemløsning og det å tenke algoritmisk:

> Gjerne lage fire koder, det holder vi på med nå, de skulle lage en lang kode med sikkert 50 blokker. Da lagde vi en 7-8 blokker om gangen og så satt vi de sammen etter hvert. Da var det veldig lett for elevene å gå tilbake å se «Hvor var feilen, hvor gikk det galt, hva må jeg endre på for å endre programmet?», at de klarer å bryte det ned.

Et av spørsmålene er på hvilken måte han tenker at programmering kan øke elevenes matematiske kompetanse. Her trekker han frem flere punkter som han mener programmering kan bidra med. Først kommer det frem at han tenker at dette er tilfelle, og når han skal forklare hvorfor programmering kan bidra sier han følgende:

> For det er så steg for steg. Så de lærer seg sånn steg for steg i kodingen som jeg tror at de kan ta med seg videre inn i matematikken og forstå at det er steg for steg i matematikken.

Dette kan tolkes som at han mener at programmering kan bidra til at elevene tar det de lærer gjennom programmering, for så å benytte seg av dette i andre emner i matematikk. De gjenkjenner altså likheter på tvers av emnene, som er det nøkkelbegrepet **mønster** handler om.

På spørsmålet om han har noen eksempler på positive erfaringer i klasserommet, trekker han frem et eksempel hvor elevene programmerte biler:

> Da prøvde de koden og bilen gikk bare rundt og rundt og rundt, men den skulle egentlig gå frem, svinge og deretter tilbake og svinge. Det var veldig enkelt. Jeg så det med en gang. Da tok elevene og begynte å trykke litt, se litt, og plutselig fant de ut av det.

I sammenheng med dette utdraget nevner han også at dette er noe han tror «… de lærer utrolig mye av. Det å gå tilbake å se «hva det er jeg har gjort feil»». Elevene må da se på hvilke

detaljer i koden som er viktig for at bilen skal gå fremover, altså bruke **abstraksjon** for å luke vekk detaljene i koden som ikke er viktige.

4.3.2.2 Arbeidsmåter

Når han snakker om hvordan programmering har utviklet seg i hans undervisning, snakker han om hvordan det i begynnelsen var viktig for han at elevene fikk en slags oppskrift som de skulle komme seg gjennom, og deretter gå videre til neste oppgave. Men etter hvert har dette endret seg, og han sier følgende:

> Men nå tenker jeg at fokuset er mye mer på det med at en får teste mye forskjellige, få bruke god tid på å putle, gjøre det på det nivået de er. Bruke tid på å bli kjent med de forskjellige tingene, og det at elevene får jobbe der de er, så noen jobber med å lage et blinkende hjerte på Micro:Bit, og noen lager avanserte Micro:Bits som styrer andre Micro:Bits som igjen styrer en tredje ting.

I utdraget snakker han om at han tenker det er viktig at elevene ikke lenger får en slavisk fremgangsmåte, slik som tidligere, men heller at de får tid til «å putle», som betyr akkurat det samme som **å fikle**. Videre i samme utdrag forklarer han at noen av elevene jobber med å få et blinkende hjerte til å vise på Micro:Bit, som kort fortalt er en liten datamaskin som kan programmeres til å gjøre enkle oppgaver. Dette handler om å designe og lage egne programmer, altså **å skape**. Dette snakker han også om i andre anledninger, blant annet når han blir spurt hvilke vurderinger han gjør når han skal velge programmeringsoppgaver til elevene. I utdraget snakker han om hvordan han tenker at elevene skal komme frem til et resultat, som i denne sammenhengen kan være å lage eller skape et spill:

> Så liker jeg at de på en måte klarer å komme frem til et resultat, om det er et lite spill eller bare en melding eller om det er en bil som kjører en hinderløype, men at de føler på en måte mestring.

På mellomtrinnet er det som regel blokkprogrammering elevene holder på med. Læreren har heller ikke mye erfaring når det kommer til mer avansert programmering enn dette. Når han

skal snakke om holdningen sin til programmering, snakker han om hvordan noen av de ivrigste elevene har passert hans kompetanse:

> Men kompetansen min er jo slik at da er de forbi meg, de som er flinkest, de som jeg hadde i 7. trinn i fjor som var flinkest, de programmerte jo i JavaScript og kjørte på, og jeg hadde ikke sjans til å henge med. De skrev koden istedenfor å blokkprogrammere det, slik som jeg må gjøre, og da er det på en måte ... min jobb å veilede de i feilsøking og det der med å bryte ned til små enheter og så se hvor det er feilen kan være, eller hvor er det det stopper opp.

Læreren lar her elevene jobbe på det nivået de selv ligger på, selv om dette er over hans eget nivå. Han hjelper da elevene med det han kan bidra med, **å feilsøke**. Dette er en sentral del av det å programmere og en viktig arbeidsmåte. Feilsøking er selvfølgelig ikke noe han bare hjelper de elevene som skriver kode med, men også elevene som driver med enklere programmering. Han forteller videre på samme spørsmål at han bistår med feilsøkingen ved å se på ting som «… slik som at to like blokker, da fungerer ikke programmet…». Læreren kommer med flere eksempler på feilsøking ved flere anledninger gjennom intervjuet. Mens han snakker om positive erfaringer fra klasserommet nevner han også følgende:

> «Hva i din kode er det som kan gjøre at dette programmet ikke virker?». Da prøvde de koden og bilen gikk bare rundt og rundt og rundt, men den skulle egentlig gå frem, svinge og deretter tilbake og svinge. Det var veldig enkelt. Jeg så det med en gang. Da tok elevene og begynte å trykke litt, se litt, og plutselig fant de ut av det.

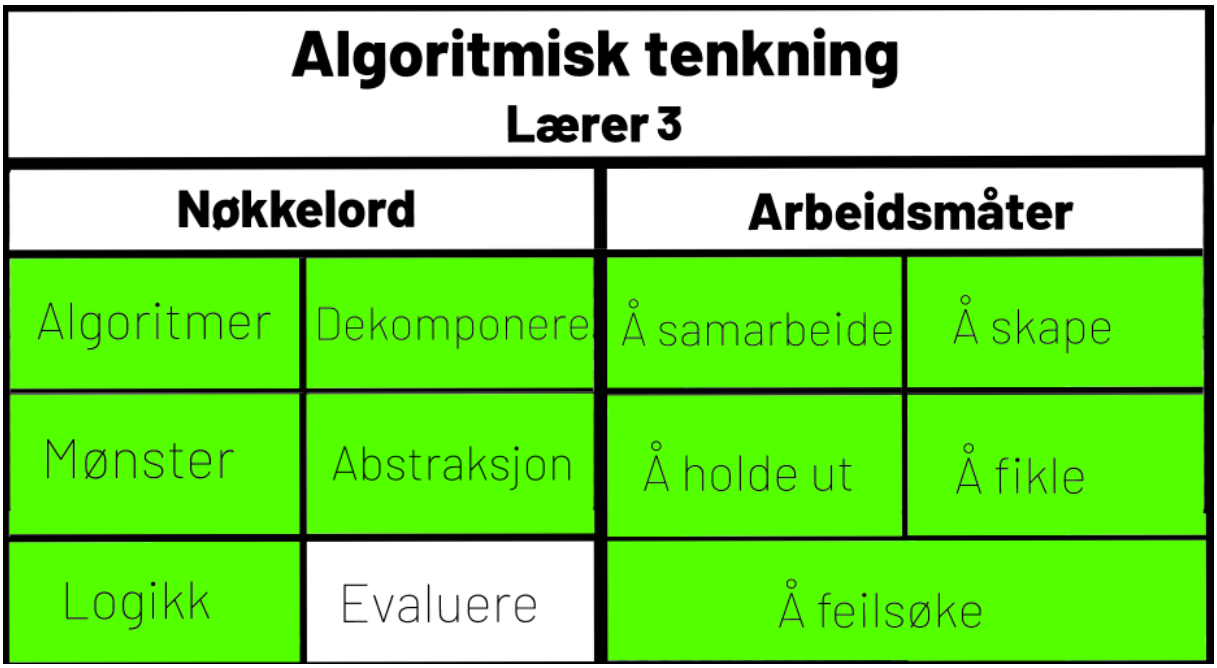
Vi kan se at feilsøking er noe han er veldig bevisst på når han spør elevene spesifikt hva i koden som ikke fungerer. I samme utdrag kan vi igjen se hvordan han forteller at elevene ikke ga opp, men fortsatte å prøve seg frem, som også går på den tidligere nevnte arbeidsmåten å fikle. Dette samme eksempelet viser også hvordan elevene får arbeidet med enda en av arbeidsmåtene, altså **å holde ut**. Det er fort gjort å gi opp når man arbeider med programmering og det ikke går helt som man vil. En essensiell del av programmeringen er

derfor å ikke gi opp når man møter motgang, men holde ut til man får det resultatet man er ute etter.

Gjennom intervjuet nevnes det at elevene **samarbeider** flere ganger. På spørsmålet om hvordan en vanlig matematikktime pleier å se ut, svarer han at han alltid prøver «… å ha noen oppgaver som elevene løser i par på tavlen, så får de tenke litt selv og jobbe litt sammen». Dette kom frem før programmering ble et tema i samtalen, men også etter programmering ble tatt opp kom det frem at samarbeid er noe elevene gjør både når de jobber i grupper og utenom gruppearbeid. Når han skal komme med noen eksempler på positive erfaringer med programmering i undervisningen, sier han «Da prøvde de koden og bilen gikk bare rundt og rundt…», som viser til at det var en gruppe elever som jobbet sammen. Også i arbeid med programmering uten bruk av datamaskin snakker han om hvordan «… elevene skal programmere hverandre i løkker…». Det kommer ekstra tydelig frem at samarbeid er noe elevene hans har som vane når han forklarer om positive opplevelser med programmering i undervisningen:

> Altså hver gang jeg har programmering så er alle elevene med. Så du får veldig deltagende og aktive elever, og så blir de utrolig flinke til å hjelpe hverandre. De som får det til er veldig villige til å forklare og vise til de som strever litt og står litt fast. Så det blir veldig inkluderende og bra læringsmiljø av å jobbe med noe så praktisk.

Etter å ha analysert intervjuet med Lærer 3, kommer det frem fra hans forklaringer av algoritmisk tenkning at dette er noe han vet hva innebærer. Ved å se på det han sier om algoritmisk tenkning, kan vi se at dette dekker flere av nøkkelbegrepene. Også gjennom selve intervjuet kan man se at han er innom alle nøkkelbegrepene til den algoritmiske tenkeren utenom det å evaluere. Når det kommer til arbeidsmåtene kan vi se at alle disse er snakket om i løpet av intervjuet, noen mer enn andre. Under er en tabell med oversikt over de forskjellige aspektene i algoritmisk tenkning som blir nevnt i løpet av intervjuet med Lærer 3.



Figur 7 Aspekter ved algoritmisk tenkning hos Lærer 3

4.4 Lærer 4

Lærer 4 er den siste informanten som ble intervjuet. Han har 90 studiepoeng i matematikk, som noe er fra lærerskolen og noe fra ingeniørfag. Han har tidligere undervist i matematikk på ungdomsskolen, men underviser for øyeblikket i 7. klasse. På skolen der han for øyeblikket jobber har han drevet med programmering i en del år nå. Han mener selv han er veldig kompetent til å drive med programmering i matematikkundervisningen med bakgrunnen han har, i tillegg til at det er en personlig interesse.

4.4.1 Beskrivelser av matematisk kompetanse i intervjuet med Lærer 4

4.4.1.1 Problembehandlingskompetanse

Når det kommer til de forskjellige matematiske kompetansene, som Lærer 4 legger vekt på og beskriver i arbeid med programmering, er det mye å ta av. Det kommer tidlig frem i intervjuet at problembehandling er noe han har fokus på. Han sier blant annet følgende når han skal kort fortelle hva han tenker er lærerens viktigste oppgave i matematikkfaget: «Å gi et verktøy til problemløsing, vil jeg tenke. Det er så mange hjelpeverktøy nå, matematikk er jo et problemløsningsfag». Vi kan se fra utsagnet at han omtaler matematikkfaget som et problemløsningsfag, hvor han ønsker å gi elevene et verktøy til å løse disse problemene.

Problembehandlingskompetansen handler om evnen til å identifisere og løse matematiske problemer. Å gi elevene et slikt hjelpeverktøy som han snakker om vil styrke deres problembehandlingskompetanse.

Senere i intervjuet kommer det også frem at han er veldig bevisst over mulighetene til læremidlene de har tilgjengelig til bruk i matematikkundervisningen. Når han blir spurt om hvilke læremidler han tenker egner seg best til å jobbe med kompetansemålene i matematikk, trekker han frem Micro:Bit, som er en liten programmerbar datamaskin, som det beste verktøyet de har på skolen. Dette fordi «… det dekker begge deler. Det dekker både det rent programmeringstekniske og problemløsningen». Videre når han senere svarer på om han tenker programmering kan bidra til å forbedre elevenes matematikkforståelse, sier han følgende:

> Så er det, om ikke annet, en annen måte å presentere de samme problemstillingene på som vi har presentert for hundrevis av forskjellige måter i alle år. Om ikke annet, så er dette en ny en, og det er en veldig relaterbar en for veldig mange.

Her trekker han frem at programmering kan være en ny, og kanskje mer relaterbar, metode å arbeide med ulike problemstillinger. Programmering er altså en ny måte elevene kan utvikle kompetanse når det kommer til problembehandling. Også når han blir spurt om hva programmering kan bidra med i matematikken når det kommer til å øke den matematiske kompetansen til elevene, svarer han som tidligere, at «Det handler jo om problemløsning, helt tilbake til det første jeg sa, for meg så handler det om problemløsning». På bakgrunn av dette kan man si at han tenker at programmering kan bidra til å forsterke elevenes problembehandlingskompetanse.

4.4.1.2 Resonnementskompetanse

Etter at intervjuet egentlig var over, sendte han en mail med noe han følte han ikke hadde fått skikkelig frem. Dette handlet hovedsakelig om spørsmålet hvor han skulle snakke om hvordan man går frem når man skal jobbe med programmering. I intervjuet snakket han om å ha tydelige mål, og vite hvor man skal hen, mens i mailen han sendte, skriver han følgende:

> I mitt hode er programmering et meget bra verktøy for å sjekke det hvis man skal ta et utgangspunkt i mål om dybdelæring. Kan det du lærte anvendes i en annen sammenheng? Hvis vi tar for oss et av punktene som ble nevnt i vår samtale, variabler, så er det lett å se for seg at kunnskap om variabler kan anvendes i et bredt spekter av oppgaver innenfor programmering, eller eventuelt overføres til en diskusjon om ukjente i likninger.

I mailen snakker han om programmering som et verktøy for dybdelæring, og hvordan noe elevene lærer gjennom programmering kan benyttes både i andre programmeringssituasjoner og i helt andre matematiske situasjoner. **Resonnementskompetansen** handler kort fortalt om å forstå matematikken og logikken bak, men handler også om hvilke operasjoner man skal bruke til enhver tid. I utdraget kommer det frem at han mener programmering kan gjøre det lettere å anvende kunnskapen i andre områder. Dette krever at elevene resonnerer seg frem til at det de har lært kan benyttes i andre matematiske emner.

Også på spørsmålet om hvordan programmering kan øke den matematiske kompetansen, trekker han inn noe som kan minne om resonnementskompetanse:

> Så er det en veldig tydelig overgang fra teori til praksis. Dette jeg gjør, jeg bruker den matematiske kunnskapen min til noe, og det får et resultat, om de laget et spill, om de lage ... det at de får en forståelse for hvordan matematikken påvirker alt du har med deg.

Her snakker han i utgangspunktet om at programmering gir en tydelig overgang fra teori til praksis, og fortsetter med at de gjennom programmering «får en forståelse for hvordan matematikken påvirker alt du har med deg». Han mener at programmering vil gi en bedre

forståelse for matematikk i alle settinger. Altså mener han at programmering vil øke elevenes resonnementskompetanse.

4.4.1.3 Kommunikasjonskompetanse

Når det er snakk om **kommunikasjonskompetansen**, er det fort gjort å tenke at denne handler om kommunikasjon mellom to personer, men en minst like viktig del av kommunikasjonskompetansen i en programmeringssammenheng, er hvordan man kommuniserer med datamaskinen. I intervjuet, når han blir spurt om programmering kan bidra til utforsking og problemløsning, snakker han ikke om hvordan elevene samhandler seg imellom, men påpeker derimot viktigheten med å kommunisere nøyaktig med dataprogrammet man benytter seg av i utdraget under:

> Jeg har jo laget et fantastisk program her, men det fungerer ikke, «Hva er feil!?». Så å leite, og kanskje finner du problemet, kanskje lager du et nytt problem når du løser problem nummer en. Du forandrer en ting, men plutselig så trengte den variabelen det, og så har du plutselig forandra den, da fungerer ikke det.

Her skildrer han en situasjon som kan oppstå i klasserommet, der en elev ikke helt forstår hvor feilen ligger. Eleven har kommunisert skriftlig til datamaskinen hva hen ønsker at skal skje, men ønsket resultat skjer ikke. I slike situasjoner som dette er kommunikasjonskompetansen høyst nødvendig. Programmering er et eget språk å kommunisere på, på samme måte som norsk når man snakker muntlig. Dette kommer også Lærer 4 inn på og sier at «Det som du på en måte får da, at de skjønner dette språket, mange skjønner dette språket», når han snakker om matematikkforståelsen knyttet til programmering. Kommunikasjonskompetansen til elevene vil da på mange måter bli styrket ved at de lærer nye måter å kommunisere på.

4.4.2 Aspekter ved algoritmisk tenkning i intervjuet med Lærer 4

4.4.2.1 Nøkkelbegrep

Han nevner algoritmisk tenkning ganske tidlig i intervjuet når han svarer på hva han tenker er nyttig med programmering når det kommer til matematisk kompetanse, uten å bli spurt

direkte om det. Han forteller at med «Alt som handler om algoritmisk tenkning er det jo nyttig å ha programmering». Han fortsetter med å kort fortelle hva algoritmisk tenkning handler om:

> Hvis du skal trekke det helt, helt ut, så vil jeg si det at algoritmisk tenking handler jo om, altså om problemløsning og bryte ned problemet i mindre komponenter, analysere og løse problemet, håndtere biter, planlegging. Så, altså det er åpenbart at det at noe å si i forhold til algoritmer.

Allerede her trekker han frem noen av nøkkelbegrepene til den algoritmiske tenkeren.
Å analysere er en sentral del av algoritmisk tenkning, og faller under nøkkelbegrepet **logikk**, som også hjelper den algoritmiske tenkeren med å gjøre antakelser. Videre i utdraget sier han at han åpenbart tenker at programmering kan øke den matematiske kompetansen i forhold til **algoritmer.** I utdraget snakker han også om at algoritmisk tenkning handler å bryte ned et problem i mindre komponenter, også kalt **å dekomponere**.

Når han senere i intervjuet blir bedt om å si hva han legger i begrepet algoritmisk tenkning svarer han igjen «… kunne bryte ned et problem…», «… å kunne analysere biter og løse de på en måte som gjør at man jobber systematisk med å gjøre ting håndterbare», og igjen er det dekomponering og logikk som kommer frem i forklaringen.

Ved å se over hans forklaring på algoritmisk tenkning, kan vi se at han vet godt hva dette innebærer. Han nevner de mest sentrale poengene med den algoritmiske tankegangen, som det å bryte et problem til mindre håndterbare problemer, og deretter det å jobbe systematisk for å prøve å løse problemet.

> … prøve og feile, og løse problemer, bryte ned problemer, håndtere små biter, ta det over fra et gangestykke, altså hvis du bryter ned 384x17, hvordan bryter du ned til et håndterbart spørsmål. Det er kanskje det du får mest ut av.

Dette tolkes som at ved å arbeide med programmering og ha fokus på algoritmisk tenkning, kan elevene se at denne algoritmiske tankegangen også kan benyttes utenom

programmeringen. Nøkkelbegrepet **mønster** handler blant annet om dette med å være oppmerksom på gjentakende handlinger eller mønster. Å se at den algoritmiske tankegangen kan tas i bruk i andre områder enn programmering vil derfor komme inn under dette nøkkelbegrepet.

Det siste nøkkelbegrepet til den algoritmiske tenker-modellen er **abstraksjon**. Under et av spørsmålene der han snakker om hvordan programmering kan bidra til at elevene blir bedre på problemløsning, sier han følgende:

> Hvordan går jeg frem da for å løse dette problemet? Jeg må bryte det ned, hvilke funksjoner trenger jeg for å få dette til. Skal den bevege seg på en bestemt måte? Skal den gå fort, skal den gå sent, skal den høyre, venstre, opp eller ned, skal den bli større, mindre, hva er det jeg skal få til for noe.

I dette utdraget stiller han først spørsmål til hvordan man går frem for å løse et problem man er blitt tildelt. Abstraksjon handler om å identifisere hva man trenger av informasjon, og hva man ikke trenger. Vi kan se fra utdraget at han snakker om hvilke funksjoner som trengs for å løse problemet, og deretter finne ut hva man vil skal skje.

4.4.2.2 Arbeidsmåter

De fleste arbeidsmåtene kommer tydelig frem i intervjuet med Lærer 4. På spørsmålet om hva han mener programmering kan bidra med i matematikkfaget når det kommer til å øke den matematiske kompetansen, svarer han blant annet:

> Det å planlegge, identifisere utfordringer, prøve og feile, det å stå i noe som er vanskelig, og liksom evne å se «Det fungerte ikke, hva er det som er galt? Hvorfor fikk jeg det ikke til?», og prøve igjen.

Her trekker han frem flere av aspektene ved algoritmisk tenkning. Det første han nevner er prøving og feiling. Dette er en arbeidsmåte som er identisk til det **å fikle**, og handler om å prøve seg frem ved å gjøre små endringer og se hva som skjer. Fra utdraget kan man også se

han mener programmering kan bidra til at elevene ikke gir opp, ved å stå i noe som er vanskelig, og det kan tolkes som at han mener programmering kan føre til at elevene blir bedre på **å holde ut**. Det siste poenget man kan ta fra utdraget er at han tenker elevene får en økt evne til å se over noe de har gjort feil, og deretter prøve å finne ut hvor feilen ligger. Dette er det **feilsøking** handler om. Feilsøking blir også nevnt når han svarer på om han tenker at programmering kan bidra til utforsking og problemløsning:

> Hvor er problemet? Hvordan går jeg fram for å finne ut av det? Hva kan jeg gjøre for å finne det problemet? OG når jeg finner det, hvordan løser jeg det?». Hva er min tilnærming til dette her da. Det å lære den systematiske feilsøkingen rett og slett…

I eksempelet under svarer han på spørsmålet om han har hatt noen utfordringer med å implementere programmering i matematikkundervisningen sin. Her snakker han om 10. klasse, som han har undervist tidligere, om hvordan elevene får se et resultat av det de selv har gjort i arbeid med programmering:

> Det at du ser et resultat som du har laget selv. Det er veldig sårbart for stort spenn i ferdighet. I 10. klasse for eksempel, bare for å ha nevnt det, der programmerte vi egne apper.

I dette utdraget kan vi se at elevene må lage ulike programmer selv. Dette innebærer også at elevene må designe sitt program slik de vil, og både det å lage og designe noe går under en av arbeidsmåtene hos den algoritmiske tenkeren, nemlig **å skape**.

En av arbeidsmåtene som ikke kommer så tydelig frem i løpet av intervjuet er samarbeid. Dette kan ha en sammenheng med at det er ingen direkte spørsmål som spesifikt skal undersøke om elevene samarbeider mye i undervisning med programmering. Når han snakker om utfordringer han har hatt med programmering i klasserommet, kommer han inn på nivåforskjeller. Her drar han frem at de som har mest erfaring får jobbe sammen og «sitter og lager apper og laster opp på telefonene deres og så sitter og spiller snake sammen». Her kommer det frem at de elevene som kan litt ekstra **samarbeider** for å lage litt mer avanserte

ting enn resten av klassen. Det nevnes ikke at også resterende i klassen som er på samme nivå jobber i grupper, men dette tas som en selvfølge.

Etter en analyse av intervjuet med Lærer 4, kan man se at han snakker om både nøkkelbegrepene og arbeidsmåtene hos den algoritmiske tenkeren. Når det kommer til nøkkelbegrepene er det noen av dem som kommer klarere frem enn andre, ettersom han forklarer dem flere ganger når han snakker om algoritmisk tenkning gjennom intervjuet. De nøkkelbegrepene han ikke nevner med ord kan man også finne i løpet av intervjuet med å tolke noen av utsagnene hans. Arbeidsmåtene er også noe som kommer tydelig frem, selv om det også her kreves noe tolkning av utsagn for å dekke alle. Under er en tabell med oversikt over de ulike aspektene ved algoritmisk tenkning som kommer frem etter å ha analysert intervjuet med Lærer 4.

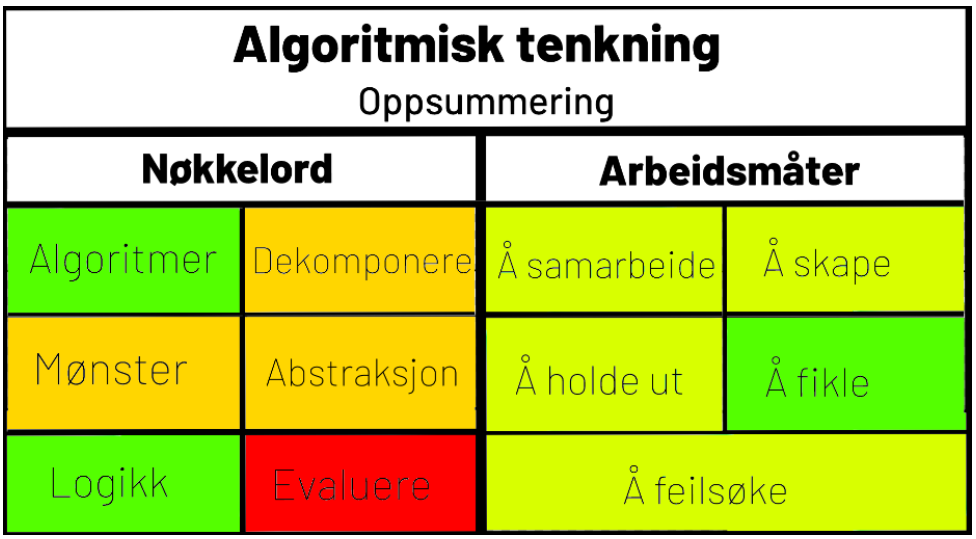
Algoritmisk tenkning **Lærer 4**			
Nøkkelord		**Arbeidsmåter**	
Algoritmer	Dekomponere	Å samarbeide	Å skape
Mønster	Abstraksjon	Å holde ut	Å fikle
Logikk	Evaluere	Å feilsøke	

Figur 8 Aspekter ved algoritmisk tenkning hos Lærer 4

4.5 Oppsummering av analysen

Etter å ha sett gjennom intervjuene er det flere ting som kommer frem. Av de matematiske kompetansene som analysen skulle ta utgangspunkt i, blir de beskrevet i ulik grad. Kompetansen som kommer tydeligst frem er problembehandlingskompetansen, hvor alle lærerne kommer med flere beskrivelser gjennom intervjuet om ting som kan minne om problembehandlingskompetansen. De andre kompetansene, resonnementskompetansen og kommunikasjonskompetansen, kommer også frem i intervjuet, men i mindre grad.

Analysen som er gjennomført i de foregående delkapitlene har sett etter både nøkkelbegrepene og arbeidsmåtene hos den algoritmiske tenkeren som lærerne direkte har sagt, eller som kan tolkes som en av dem. Når det kommer til nøkkelbegrepene ved algoritmisk tenkning, kommer det frem at de eneste nøkkelbegrepene som alle lærerne snakker om er algoritmer og logikk. Mønster, abstraksjon og dekomponering blir bare snakket om av Lærer 3 og 4, mens det å evaluere er fraværende hos alle. Hvis man skal se på hvilke arbeidsmåter lærerne skildrer på ulike tidspunkt i intervjuene, kan man se at alle lærerne har i løpet av intervjuet snakket om å fikle. Det er selvfølgelig store variasjoner i hvor mye de forskjellige arbeidsmåtene blir lagt vekt på, men denne variasjonen blir sett vekk fra. De resterende arbeidsmåtene blir alle nevnt av 3 av lærerne. Tabellen under er en representasjon over hvor mange som har snakket om de bestemte nøkkelbegrepene og arbeidsmåtene, der grønn betyr at alle har vært inne på det, og rødt tilsier at ingen har snakket om det.



Figur 9 En oppsummering av de nevnte aspektene hos den algoritmiske tenkere

5 Diskusjon

Denne masteroppgaven har hatt som mål å finne ut hva et utvalg matematikklærer på mellomtrinnet sier om programmering i matematikkundervisningen. Dette diskusjonskapittelet har som formål å utforske funnene som ble presentert i analysen i lys av teori, for å prøve å svare på studiens to forskningsspørsmål:

1) *Hvilke matematiske kompetanser beskriver lærerne i elevenes arbeid med programmering?*
2) *Hvilke aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering?*

I diskusjonen vil jeg trekke frem de sentrale funnene fra intervjuene, etter å ha analysert dem ved bruk av rammeverket bestående av et utvalg av Niss og Højgaard Jensens (2002) matematiske kompetanser og Utdanningsdirektoratets modell for den algoritmiske tenkeren (2020).

5.1 Matematiske kompetanser som lærerne beskriver i samtale om programmering

Intervjuene er analysert ved hjelp av Niss og Højgaard Jensens (2002) matematiske kompetanser, der det er deres beskrivelser av kompetansene problembehandling, resonnering og kommunikasjon, som sammenlignes med det lærerne sier for å kunne påstå at en lærer beskriver en gitt kompetanse. Under hvert av de neste delkapitlene vil det først komme en oppsummering av hvor mange som har omtalt kompetansen, og deretter i hvilken sammenheng de har omtalt den matematiske kompetansen. Etter dette vil kompetansen knyttes opp mot tidligere teori, som kan forklare hvorfor lærerne har beskrevet kompetansene. Diskusjonen vil først ta for seg problembehandlingskompetansen, deretter resonnementskompetansen og til slutt kommunikasjonskompetansen.

5.1.1 Problembehandlingskompetansen

Gjennom analyse av intervjuene kommer det frem at av de utvalgte matematiske kompetansene, er *problembehandlingskompetansen* den som blir beskrevet flest ganger. Ifølge Niss og Højgaard Jensen (2002) innebærer denne kompetansen å løse matematiske problemer på måten de er formulert på. Et slikt matematisk spørsmål vil også kreve en form for matematisk undersøkelse for å løses (Niss & Højgaard Jensen, 2002, s. 49). Alle lærerne skildrer en eller flere problemløsningsoppgaver elevene deres har jobbet med, hvor det kreves å utføre en slik matematisk undersøkelse som Niss & Højgaard Jensen forklarer. Videre snakker også Lærer 3 om hvordan programmering i seg selv er problemløsning, og Lærer 4 trekker frem programmering som en ny måte å presentere problemstillinger.

Forsström & Kaufmann (2018, s. 19) forklarer I sin definisjon av programmering at det handler om å gi instruksjoner til et dataprogram, slik at dataprogrammet blant annet kan løse problemer. Denne definisjonen legger vekt på at programmering brukes når man skal løse et problem. Dette kan forklare hvorfor problembehandlingskompetansen vil være til stede i de fleste omstendigheter, dersom det snakkes om programmering. Ettersom dette er et intervju som hovedsakelig spør om programmering, vil nok dette være hovedgrunnen til at problembehandlingskompetansen kommer så tydelig fram.

Ved å se på hva tidligere forskning har konkludert med når det kommer til problemløsning matematikkundervisningen, kan vi se at problemløsning er en sentral del i flere studier, for eksempel Bintoro et al. (2021). I nevnte studie undersøkte de i hvilken grad en IKT-basert undervisning ville øke elevenes ferdigheter i problemløsning. Det kom her frem at en slik form for undervisning kan gjøre elevene bedre i problemløsning. Denne studien tok for seg IKT, og nevner ikke programmering, men siden programmering er en del av IKT, så vil dette allikevel være aktuelt.

Noen andre som har undersøkt på hvilken måte programmering kan bidra til spesifikke ferdigheter er Popat & Starkey (2019). De analyserte en rekke artikler som på forskjellige måter tok for seg programmering i en matematikkundervisningssammenheng, hvor de konkluderer med at programmering kan gjøre at elevene blir bedre på problemløsning. Når lærerne i denne sammenheng beskriver problembehandlingskompetansen, betyr det at denne kompetansen er noe elevene får jobbe med og derfor får en økt kompetanse, som den tidligere forskningen viser til.

5.1.2 Resonnementskompetansen

Fra analysen kommer det frem at *resonnementskompetansen* er noe de fleste av lærerne beskriver i løpet av intervjuet. For å forklare hva denne kompetansen er, trekker Niss & Højgaard Jensen (2002, s. 54) frem at den handler om å vurdere resonnement, skriftlig og muntlig. Den innebærer å forstå matematiske bevis, vite om matematiske argumenter er gyldige, og ikke minst forstå logikken i eksempler. Lærerne har beskrevet ulike deler av

resonnementskompetansen, der et eksempel er Lærer 3 som påpekte at elevene måtte se logikken i et eksempel. Lærer 1 og 4 snakker derimot om dette med å forstå matematikken, hvor de også påpeker at dette vil hjelpe elevene å knytte det de lærer gjennom programmering til andre situasjoner. Gjennom intervjuet med Lærer 4 er det vanskelig å finne noe spesifikt som kan minne om resonnementskompetansen.

Resonnementskompetansen er noe lærerne beskriver gjennom arbeid med programmering. Dette er ikke tilfeldig, og det er flere studier som har tatt for seg i hvilken grad programmering i matematikkundervisningen kan påvirke elevers resonneringsevne. Psycharis & Kallia (2017) gjennomførte en studie på 66 skoler, med både eksperimentgruppe og kontrollgruppe. Studiens funn indikerer at det er en signifikant forskjell i elevenes resonneringsevne hos elevene som deltok i «programmeringskurset», sammenlignet med de som ikke deltok.

Videre har også Scherer et al. (2019) gjort en studie hvor de undersøker om kunnskap og ferdigheter elevene opparbeider seg gjennom å jobbe med programmering, vil kunne overføres til andre emner i matematikken, slik som både Lærer 1 og Lærer 4 snakker om. Resultatet fra studien viser at ferdigheter som resonnement, kan overføres til andre områder, hvor dette spesielt gjelder områder som krever flere av de samme kognitive ferdighetene som man får bruk for i programmering. En gjennomgang av tidligere forskning viser altså at elevers resonneringsevne vil påvirkes i en positiv retning, og kan forklare hvorfor lærerne beskriver resonneringskompetansen når de snakker om programmering i matematikkundervisningen.

5.1.3 Kommunikasjonskompetansen

Kommunikasjonskompetansen er den siste kompetansen som ble analysert. Også denne kompetansen er godt representert i flere av intervjuene. Ved å se på hva Niss & Højgaard Jensen (2002, s. 60), kan vi se at det i all hovedsak handler om å snakke i, med og om matematikk. Inn her kommer også evnen til å uttrykke seg matematisk både gjennom å skrive, snakke og visualisere. I intervjuene snakker både lærer 2 og 3 spesifikt om at elevene samarbeider i grupper når de arbeider med programmering i matematikkundervisningen. Når elever jobber i grupper, vil en muntlig kommunikasjon være til stede. Når de i tillegg jobber i grupper i en programmeringssammenheng, vil også den skriftlige kommunikasjonen være til

stede, og ofte også den visuelle kommunikasjonen. Lærer 4 snakker ikke om programmering i en samarbeidssammenheng, men snakker heller om tydelig skriftlig kommunikasjon med programmet.

Etter å ha analysert 15 artikler som alle omhandlet programmering, konkluderer Forsström & Kaufmann (2018) med flere ting. Det kommer blant annet frem at arbeid med programmering og aktiviteter med roboter som regel vil resultere i elevsamarbeid, men dette samarbeide er også avhengig av en veiledende lærer for at dette skal gi best resultater. Videre har også Popat og Starkey (2019) sett over flere artikler hvor programmering i undervisningen var temaet. Studien deres framhever at elever, gjennom å programmere i matematikkundervisningen, kan få bedre sosiale ferdigheter, der samarbeid blir nevnt.

Hos alle lærerne som er intervjuet, arbeider elevene med programmering. Ved å se på hva de nevnte artiklene konkluderer med, som at programmering i stor grad vil oppfordre til samarbeid, og at programmering i matematikken kan gjøre elevene bedre i samarbeid, kan vi se at kommunikasjonskompetansen kan økes gjennom programmering. Vi kan også se at som Forsström & Kaufmann (2018) sin artikkel viser, snakker nesten alle lærerne om hvordan elevene samarbeider når de jobber med programmering. Det er også sannsynlig at Lærer 4 sine elever samarbeider når de programmerer, men han ikke tenkte dette var relevant å si i løpet av intervjuet.

5.2 Aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering

Ved å se over selve analysen av aspektene ved algoritmisk tenkning, kan det være at noen av tolkningene som er blitt gjort kan virke litt fjerne. Det er viktig å påpeke at i analysen er ALT som kan minne om et av aspektene blitt inkludert. I noen tilfeller kan det være at noen av lærerne trekker frem et av aspektene, uten å uttale seg presist. Så lenge de likevel er innom innholdet i aspektene, har jeg valgt å ta det med for å ikke ekskludere disse tilfellene. Samtidig vil en slik vid tolkning føre til at noen av lærerne får noen av aspektene «gratis», uten å bevisst snakke om dem. Det er en garanti at en eller flere av aspektene vil bli omtalt gjennom en samtale om nettopp programmering.

Når det kommer til hvordan aspektene vil diskuteres i dette delkapittelet, vil de først bli nevnt en etter en, hvor nøkkelbegrepets innhold gjentas, om lærerne har nevnt det gitte begrepet, og deretter en forklaring på hvorfor det kan være slik. Når alle nøkkelbegrepene er lagt frem vil algoritmisk tenkning heller ses på som en helhet, ettersom jeg i denne oppgaven ikke har fokusert spesifikt på teori som sier noe om de enkelte nøkkelbegrepene.

5.2.1 Nøkkelbegrep hos den algoritmiske tenkeren

Fra analysen av intervjuene er *algoritmer* et av de nøkkelbegrepene som kommer best ut. Alle lærerne er inne på dette med algoritmer, men noen bare så vidt. Lærer 1 nevner kun algoritmer når han skal forklare hva algoritmisk tenkning er, og dette er da det eneste han sier i forbindelse med algoritmisk tenkning. Lærer 2, 3 og 4 snakker alle om algoritmer utenom spørsmålet om algoritmisk tenkning, der Lærer 2 og 3 snakker om dette i forbindelse med en oppskrift. Lærer 4 snakker om algoritmer i forbindelse med at han mener at programmering har noe med det å gjøre. En forklaring på at alle lærerne har rørt ved nøkkelbegrepet algoritmer, kan være så enkelt som at algoritmisk tenkning er noe alle ble spurt om, og derfor ligger ordet langt framme i bevisstheten. Dette virker å være tilfelle for Lærer 1, som bare trekker frem at algoritmisk tenkning har med algoritmer å gjøre.

Det andre nøkkelbegrepet er *dekomponering*. Sammenlignet med noen av de andre begrepene i den algoritmiske tenkeren, er dekomponering en av de mer konkrete. Dersom man har satt seg inn i hva algoritmisk tenkning innebærer, er det å bryte ned et krevende problem til mindre delproblemer, enkelt å forklare. Etter å ha analysert intervjuene har hverken Lærer 1 eller 2 vært inne på dekomponering. Ikke når de snakker om programmering generelt eller når de blir spurt om hva de legger i algoritmisk tenkning. Det å bryte ned problemet til mindre komponenter er derimot det første Lærer 1 og Lærer 2 snakker om når det kommer til å forklare algoritmisk tenkning.

Nøkkelbegrepet *mønster* kan være litt krevende å identifisere. Fra analysen kan vi se at hverken Lærer 1 eller 2 snakker om aspektet mønster. Mønster handler, litt selvforklarende, om å se ulike mønstre i matematikken. Dette innebærer å klare og se at noe man lærer kan anvendes andre steder i matematikken. Lærer 3 og 4 snakker om at det de lærer gjennom

programmering, også kan benyttes i matematikk. Ingen av lærerne nevner mønster med ord, og det er vanskelig å finne spesifikke eksempler på noe som kan minne om dette.

Abstraksjon er også et av nøkkelbegrepene som ikke er helt opplagt å identifisere i intervjuene. Dette kan være en av grunnene til at bare Lærer 3 og 4 har omtalt dette aspektet i løpet av intervjuet. Abstraksjon handler om å se vekk fra informasjonen som egentlig ikke er relevant for det oppgaven spør om, og kun identifisere den informasjonen som trengs (Barefoot Computing, 2016). Lærer 3 kommer inn på abstraksjon når han snakker om å identifisere et problem i koden, mens Lærer 4 snakker om hva man trenger å vite for å få til noe.

Logikk er det andre nøkkelbegrepet vi kan se fra analysen at alle har snakket om. Logikk handler i korte trekk om å etablere, faktasjekke, gjøre antakelser og resonnere seg frem (Barefoot Computing, 2016). At alle har snakket om noe som kan minne om logikk kan forklares med at alt av matematikk er logisk. Når de kommer med eksempler på hva de har gjort i programmeringen er det sannsynlig at de kommer inn på noe som har med logikk å gjøre.

Gjennom alle intervjuene var det ingen av lærerne som nevnte nøkkelbegrepet *evaluering*. Barefoot Computing skriver at evaluering handler om å ta valg basert på flere forskjellige faktorer(2016). At ikke dette aspektet kommer frem i noen av intervjuene kan det være flere grunner til. Dette nøkkelbegrepet, som de andre, blir ikke etterspurt direkte, og dersom man skal snakke om noe som har med algoritmisk tenkning å gjøre, er det ikke åpenbart å ta opp evaluering.

Fra analysen av intervjuene kommer det fram at flere av aspektene er fraværende hos lærerne. Disse nøkkelbegrepene, som analysen var ute etter å gjenkjenne i intervjuene, er kort fortalt ord som i stikkord-form forklarer hva algoritmisk tenkning innebærer. En mangel på disse aspektene i intervjuene kan tyde på at begrepet algoritmisk tenkning ikke er helt klart

for de lærerne som har størst frafall med nøkkelbegrep. Ved å se på definisjoner av algoritmisk tenkning:

> The conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts (Shute et al., 2017, s. 10).

vil man se at disse ikke tar for seg noen av aspektene ved den algoritmiske tenkeren. Algoritmisk tenkning blir heller omtalt som en problemløsende metode man kan bruke i flere situasjoner. Dette betyr at den eneste plassen lærerne kan ha blitt kjent med disse aspektene er gjennom Utdanningsdirektoratets artikkel om algoritmisk tenkning (2019a).

5.2.2 Arbeidsmåtene til den algoritmiske tenkeren

Fra analysen av intervjuene kommer det nokså tydelig frem at alle lærerne er innom nesten alle arbeidsmåtene. Alle disse arbeidsmåtene som er typiske for den algoritmiske tenkeren er viktige dersom man skal lykkes med denne problemløsningsmetoden. Dersom man ufrivillig ekskluderer en av dem, kan føre til at til en svekkelse i selve problemløsningsmetoden, algoritmisk tenkning.

Hvis vi først ser på det *å samarbeide*, kan man lese fra analysen at alle lærerne utenom Lærer 1 er innom arbeidsmetoden. Fra analysen kan vi også lese at lærerne kommer inn på samarbeid på ulike måter. Både Lærer 2 og 3 forteller tydelig om elevene som samarbeider mens de jobber med programmering i undervisningen. Lærer 1 og 4 derimot, nevner ikke spesifikt at samarbeid er noe elevene holder på med, men lærer 4 har i en eller flere forklaringer inkludert noe som kan tolkes til at elevene samarbeidet. Dette kan ha flere forklaringer. Ingen av dem ble spurt direkte om samarbeid er viktig i undervisningen deres. Dette ville nok gitt en garanti på at de snakket om det.

Når det kommer til at det *å skape* er også dette noe alle lærerne snakker om. Det er viktig å spesifisere at det gjennom alle intervjuene kun er Lærer 4 som ordrett har snakket om å skape, men likevel har alle lærerne snakket om noe som kan tolkes som å skape. Det å skape er en

kreativ arbeidsmåte, og handler hovedsakelig om å designe eller utvikle egne løsninger, og alle lærerne har til en viss grad fortalt om en situasjon eller oppgave der elevene blir nødt til å lage eller designe noe eget. At alle lærerne har nevnt noe som kan minne om å skape kan også forklares med at når lærerne skal svare på noen av de litt krevende spørsmålene, er det fort å ta eksempler fra egen undervisning. Dette har alle lærerne gjort.

Neste arbeidsmåte er *å fikle*. Det kommer frem i analysen at alle lærerne nevner dette aspektet. Ordet «fikle» er et ord få folk bruker, og kan med andre ord forklares som en måte å utforske eller eksperimentere. Under analysen av intervjuene kommer det også frem andre ord og uttrykk som tolkes som det samme. Lærer 3 bruker ordet «putle», mens Lærer 1, 2 og 4 snakker alle om en variant av å prøve seg frem.

Arbeidsmåten *å holde ut* kan man se fra analysen at alle lærerne har i noen grad snakket om, utenom Lærer 2. Det å holde ut hand ler om å ikke gi seg dersom man møter utfordringer. Ingen av lærerne har ordrett snakket om å holde ut som en viktig del av programmeringen, men Lærer 1 og 3 kommer begge med eksempler på elever som ikke gir opp. Dette tolkes til å holde ut, ettersom det betyr det samme. Også lærer 4 sin forklaring kan tolkes til å holde ut når han sier at programmering handler om å stå i noe som er vanskelig.

Å feilsøke er den siste arbeidsmåten i den algoritmiske tenker-modellen. Hvis jeg i analysen bare hadde sett etter lærerne som ordrett snakket om feilsøking, hadde det bare vært Lærer 3 som snakket om dette. Han snakket bevisst om at han hjalp elevene i feilsøkingsprosessen. Feilsøking handler kort fortalt om å se systematisk over det man har gjort for å lete etter eventuelle feil. Det tolkes derfor som at både Lærer 2 og 4 snakker om feilsøking når de snakker om å se over hva som er galt, og stille spørsmål til hvorfor det ikke gikk. Lærer 1 nevner ikke noe som kan minnes om feilsøking.

Lærerne berører i stor grad de ulike arbeidsmåtene til den algoritmiske tenkeren. Selv om det er noen forskjeller i hvordan lærerne snakker om disse arbeidsmåtene, er det tydelig at de er bevisste på deres betydning i programmeringsundervisningen. Dette indikerer at lærerne har god kontroll på programmering i sin undervisning, men det kan være nyttig for dem å sette

seg inn i algoritmisk tenkning for å bedre tilrettelegge undervisningen og styrke elevenes evne til problemløsning.

6 Avslutning

Det er i denne studien undersøkt hva et utvalg matematikklærere på mellomtrinnet har å si om programmering i matematikkfaget. For å undersøke dette har det blitt brukt relevant teori og innsamlede data i form av intervju. I dette kapittelet vil funn bli oppsummert og studiens problemstilling vil bli besvart. Deretter vil det også redegjøres hvilke begrensninger denne studien står overfor, og hva som vil være hensiktsmessig å forske videre på når det kommer til temaet programmering i skolen.

6.1 Konklusjon

I denne masteroppgaven er det undersøkt hva et utvalg lærere som underviser i matematikk på mellomtrinnet sier om programmering som en del av matematikkfaget. I utgangspunktet er studiens problemstilling veldig vid, og det ble derfor utviklet to forskningsspørsmål til å spisse oppgaven og hjelpe til med å besvare problemstillingen. Forskningsspørsmålene er som følger:

1. *Hvilke matematiske kompetanser beskriver lærerne i elevenes arbeid med programmering?*
2. *Hvilke aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering?*

For å svare på studiens første forskningsspørsmål har Niss og Højgaard Jensens (2002) matematiske kompetanser vært essensielle. Flere av de matematiske kompetansene de nevner er sentrale når det kommer til arbeid med programmering. Etter å ha analysert og diskutert de matematiske kompetansene som lærerne beskriver i elevenes arbeid med programmering, kan man si at både problembehandlings-, resonnerings- og kommunikasjonskompetansen kommer tydelig frem i elevenes arbeid med programmering. Det kan derfor konkluderes at de er bevisste på hvilke kompetanser som kan økes hos elevene gjennom programmering.

Denne studiens andre forskningsspørsmål «Hvilke aspekter ved algoritmisk tenkning trekker lærerne frem i samtale om programmering?» skal fortelle hvilke deler av den algoritmiske tenker-modellen (BareFoot Computing, 2016) som lærerne uttaler seg om under et intervju der temaet er programmering. Det kommer frem gjennom analyse av intervjuene og diskusjon rundt resultatene av analysen at det trengs at noen lærere blir bedre kjent med hva det innebærer å tenke algoritmisk. Intervjuobjektene er av de mer interesserte i programmering, og dersom det er mangler i deres forståelse av begrepet, vil dette sannsynligvis også gjelde andre matematikklærere som underviser i programmering. Når det derimot kommer til analysen og diskusjonen rundt arbeidsmåtene i den nevnte modellen, kommer det frem at alle lærerne utnytter seg av en rekke arbeidsmåter som samsvarer med det modellen inneholder.

6.2 Begrensninger

En begrensning ved denne studien er, som omtalt i 3.6 *Validitet og Reliabilitet,* oppbygningen av intervjuguiden. Intervjuguiden virket å være grei da intervjuene skulle utføres, men etter at oppgaven har utviklet seg, har intervjuguiden vist seg å være problematisk. Intervjuguiden er alt for generell, noe som gjør at det kan være tilfeldigheter som gjør at en lærer velger å snakke om for eksempel samarbeid, og at en annen lærer ikke snakker om det. Dette har igjen ført til at det i denne oppgaven er gjort mange tolkninger, der noen kanskje er feilaktig, for å finne ut om det lærerne sier kan minne om noen av aspektene ved algoritmisk tenkning. Dette gjelder også de matematiske kompetansene.

Utvalget av informantene som er blitt intervjuet i denne studien er delvis tilfeldig. Flere lærere ble spurt, men valgte å ikke delta av den grunn at de selv mente de ikke var kompetente i programmering, eller at programmering ikke var noe de prioriterte i sin matematikkundervisning. Dersom noen av disse lærerne hadde sagt ja, kunne man endt opp med et annerledes resultat. Dette er en kvalitativ studie hvor man ikke tar hensyn til informantenes bakgrunn, så dersom informantene hadde vært tilfeldige matematikklærere, kunne konklusjonen sett annerledes ut.

Rekkefølgen på intervjuene kan ha spilt en stor rolle i forhold til hvordan jeg stilte oppfølgingsspørsmål og inviterte til utdyping. Hvis jeg hadde hatt siste intervjuobjektet først, ville det vært lettere å vite hvilke svar jeg kunne få om jeg spurte bedre oppfølgingsspørsmål. Jeg hadde også lite erfaring når det kom til selve intervjusituasjonen, men ble sikrere med hvert intervju. Resultatmessig vil jeg si at det mest sannsynlig ville vært en nok tilsvarende konklusjon selv om jeg hadde endret rekkefølgen eller hadde hatt mer erfaring.

6.3 Videre forskning

Programmering er fremdeles relativt nytt for alle parter. Elevene, lærerne og de som underviser lærerne har alle en del å sette seg inn i før programmering kan brukes på best mulig måte, i læringssammenheng. Nå i første omgang er programmering hovedsakelig tenkt at skal tilhøre matematikkfaget, men dette betyr ikke at det er eneste plassen det kan benyttes. Det vil også være interessant å undersøke hvordan man kan bruke programmering for å styrke ferdighetene til elevene i andre fag, spesielt i naturfag.

Alle informantene i denne studien var uten unntak de lærerne som på hver sin skole var over gjennomsnittet interessert i programmering. De har alle drevet med programmering før det ble innført et krav om dette, som gjør at konklusjonen i studien ikke vil være dekkende for alle landets matematikklærere. Programmering i skolen er kommet for å bli, og det vil være nødvendig med forskning på en større skala, som sier noe om lærerne er kompetente nok til å drive en matematikkundervisning med programmering på en måte som fører til at elevene blir bedre i problembehandling, resonnering og kommunikasjon, slik som mye av litteraturen konkluderer med.

Etter å ha intervjuet disse lærerne som alle aktivt bruker programmering i sin undervisning, kommer det frem at alle ønsker å lære mer om programmering i undervisningssammenheng. Det ligger ifølge dem selv en del på nettet, men nevner at kursing i ulike læremidler ville vært ønskelig, slik at de på en bedre måte kan drive med programmering. Flere av dem føler at de mangler både kompetansen som kreves for at elevene skal oppnå kompetansemålene, og kompetanse til å ta i bruk forskjellige læremidler. Dette vil nok gjelde flere lærere her i landet,

og det vil være ønskelig med forskning undersøker lignende problemstillinger som denne masteroppgaven, men i en mye større skala.

Litteraturliste

Barefoot Computing. (2016). *Quick guide to computational thinking*. Barefoot Computing https://www.barefootcomputing.org/docs/default-source/at-home/quick--guide-to-computational-thinking.pdf

Bintoro, H. S., Zaenuri, K. & Wardono, D. (2021). Application of information technology and communication-based lesson study on mathematics problem-solving ability. *Journal of Physics: Conference Series,* 1918(4), 1-6 https://dx.doi.org/10.1088/1742-6596/1918/4/042105

Bryman, A. (2012). *Social Research Methods* (4. utg.). Oxford University Press.

Cohen, L., Morrison, K., & Manion, L. (2007). *Research methods in education* (6. utg.). Routledge.

Cresswell, J. W. (2014). *Research design: Qualitative, quantitative and mixed methods approaches* (4. utg.). SAGE.

European Schoolnet. (2017). The integration of Computational Thinking (CT) across school curricula in Europe. *European Schoolnet, 2.* http://www.eun.org/documents/411753/665824/Perspective2_april2017_onepage_def.pdf/70b9a30e-73aa-4573-bb38-6dd0c2d15995

Forsström, S. & Kaufmann, O. T. (2018). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research, 17*(12), 18-32. https://doi:10.26803/ijlter.17.12.2

Gjøvik, Ø. & Torkildsen, H. A. (2019) Algoritmisk tekning. *Tangenten – tidsskrift for matematikkundervisning, 30*(3). s. 31-37. http://tangenten.no/wp-content/uploads/2021/12/Tangenten-3-2019-Gjovik-Torkildsen.pdf

Kunnskapsdepartementet. (2021, 24. juni). *Hvorfor nye læreplaner*. https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvorfor-nye-lareplaner/

Kvale, S. & Brinkmann, S. (2021). *Det kvalitative forskningsintervju* (3. utg.). Gyldendal Akademisk.

Det kongelige kirke-, utdannings- og forskningsdepartementet. (1996). Læreplanverket for den 10-årige grunnskolen. https://www.nb.no/items/f4ce6bf9eadeb389172d939275c038bb

Niss, M. & Højgaard, J. T. (2002). *Kompetencer og matematiklæring, Ideer og inspiration til udvikling af matematikundervising i Danmark.* Roskilde Universitetscenter.

NOU 2015:8. (2015, 6. 15.). *Fremtidens skole – fornyelse av fag og kompetanser*. Kunnskapsdepartementet. https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001/

Nouri, J., Zhang, L., Mannila, L. & Norén E. (2019). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry, 11*(1), 1-17. https://doi.org/10.1080/20004508.2019.1627844

Opplæringsloven. (1998). *Lov om grunnskolen og den vidaregåande opplæringa* (LOV-1998-07-17-61). Lovdata. https://lovdata.no/dokument/NL/lov/1998-07-17-61

Popat, S. & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers and Education, 128*, 365-376, https://doi.org/10.1016/j.compedu.2018.10.005

Psycharis, S. & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science, 45*(5), 583-602. http://dx.doi.org/10.1007/s11251-017-9421-5

Regjeringen. (2018). Kjerneelementer i fag. Fastsatt av Kunnskapsdepartementet som føringer for utforming av læreplaner for fag til LK20. Regjeringen. https://www.regjeringen.no/contentassets/3d659278ae55449f9d8373fff5de4f65/kjerneelementer-i-fag-for-utforming-av-lareplaner-for-fag-i-lk20-og-lk20s-fastsatt-av-kd.pdf

Røsseland, M. (2005). Hva er en matematisk kompetanse. *Tangenten – tidsskrift for matematikkundervisning, 2005*(1), s. 12-18. https://beta.matematikksenteret.no/sites/default/files/attachments/page/rosseland_1_2005.pdf

Scherer, R., Siddiq, F. & Sánchez, V. B. (2019). The Cognitive Benefits of Learning Computer Programming: A Meta-Analysis of Transfer Effects. *Journal of Educational Psychology, 111*(5), 764-792. https://doi.org/10.1037/edu0000314

Sevik, K. (2016). Programmering i skolen. Notat fra Senter for IKT i utdanningen. Utdanningsdirektoratet. https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Shute, V. J., Sun, C. & Asbell-Clarke, J. (2017). Demytifying computational thinking. *Educational Research Review*. https://doi.org/10.1016/j.edurev.2017.09.003

Statlig Pedagogisk Tjeneste. (2021, 1. mars). Programmering. https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/?depth=0#1

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

Utdanningsdirektoratet. (2019b). *Læreplan i matematikk 1. -10. trinn.* Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. https://www.udir.no/lk20/mat01-05?lang=nob

Utdanningsdirektoratet. (2019c). *Kjerneelementer*. Utdanningsdirektoratet. https://www.udir.no/lk20/mat07-02/om-faget/kjerneelementer?TilknyttedeKompetansemaal=true&anchorId=KE21&curriculum-resources=true

Utdanningsdirektoratet. (2019d). Kompetanse i fagene. Utdanningsdirektoratet. https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/kompetanse-i-fagene/?lang=nob&curriculum-resources=true

Utdanningsdirektoratet. (2020). *Hva er nytt i matematikk?* Utdanningsdirektoratet. https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/

Wing, J. (2019). Computational thinking. *Communications of the ACM, 49*(3), 33-35, https://doi.org/10.1145/1118178.1118215

VEDLEGG

Vedlegg 1: Informasjonsskriv til deltakerne

Forespørsel til lærere om deltakelse i forskningsprosjektet
Programmering i matematikkundervisningen

Jeg skal gjennom Høgskolen på Vestlandet (HVL) gjennomføre en studie for forskningsprosjektet *Learning about Teaching Argumentation for Critical Mathematics Education* (LATACME). Formålet med studien min er å undersøke hvilke erfaringer lærere som underviser i matematikk på mellomtrinnet har med programmering i undervisningen, og hva de sier om dette. I dette skrivet informerer jeg om innholdet i prosjektet og hva det innebærer å delta.

Formålet med prosjektet
Målet med prosjektet er å undersøke hva lærere på mellomtrinnet som underviser i det nylig innførte emnet programmering, sier om dette. Prosjektet varer i fire år og baserer seg på samarbeid mellom lærerutdannere, lærerstudenter, lærere og elever. Prosjekt-gruppen består av masterstudenter, PhD-studenter og tilsatte ved HVL som arbeider med matematikkundervisning, og prosjektet har nasjonale og internasjonale samarbeidspartnere. Det er medlemmer fra prosjektgruppen som samler inn data.

Hva innebærer det å delta?
Deltagelse innebærer at du kommer til å delta i et intervju hvor det vil bli gjort lydopptak.

Personvern – Hva skjer med opplysningene?
Alle personopplysninger blir behandlet konfidensielt og materiale med personopplysninger lagres på HVL. Kun forskere i prosjektgruppen vil ha tilgang til datamaterialet. Deltakere vil ikke kunne bli identifisert i publikasjoner. Prosjektet skal etter planen avsluttes 31.12.2023, og alle opptak vil bli slettet når prosjektet avsluttes.

Hvorfor får du spørsmål om å delta?
Du er en lærer som underviser i matematikk på mellomtrinnet, og du har litt erfaring med programmering i undervisningen.

Frivillig deltagelse
Det er frivillig å delta i studien og man kan trekke seg uten å oppgi grunn, så lenge studien pågår.

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

- innsyn i hvilke personopplysninger som er registrert om deg,
- å få rettet personopplysninger om deg,
- få slettet personopplysninger om deg,

- få utlevert en kopi av personopplysninger om deg
- å sende klage til personvernombudet eller Datatilsynet om behandlingen av personopplysninger om deg.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.

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

Hvem er ansvarlig for forskningsprosjektet?

HVL er ansvarlig for prosjektet, og det er ledet av Professor Tamsin Meaney. Prosjektet gjennomføres i samarbeid med Bergen Kommune, og det er støttet av Norges forskningsråd.

Hvor kan jeg finne ut mer?
Dersom du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:

- Ansvarlig for prosjektet ved HVL: Tamsin Meaney, Epost: Tamsin.Jillian.Meaney@hvl.no
- Kasper Løvlie, Epost: Repsak97@hotmail.com
- HVLs personvernombud, Advokat Halfdan Mellbye, personvernombud@hvl.no
- NSD – Norsk senter for forskningsdata AS, på epost (personverntjenester@nsd.no) eller telefon: 55 58 21 17.

Med vennlig hilsen

……

Vedlegg 2: Samtykkeskjema

Samtykke til deltakelse i studien

Jeg har mottatt og forstått informasjon om studien *Argumentasjon og kritisk matematikkundervisning i flerspråklige klasserom* og har fått anledning til å stille spørsmål.

Jeg samtykker til at jeg, …….. (navn) kan

- ☐ delta i lydopptak
- ☐ delta i intervju

Jeg samtykker til at opplysninger behandles frem til prosjektet er avsluttet, 31.12.2023.

(Signert av lærer, dato)

Vedlegg 3 - Intervjuguide

- Før vi begynner, har du noe du lurer på?
- Har du noen formell utdanning i matematikk?
- Hvilke klasser underviser du i nå, og har undervist i siden lk20 kom?
- Hva tenker du er den viktigste jobben til læreren i matematikkfaget?
- Når du planlegger undervisningen din, kan du beskrive hva du legger vekt på at en vanlig matematikktime skal inneholde? «Vanlig».
- Med LK20 nå, så kom det et krav om at programmering skal være en del av matematikkfaget. Hva tenker du om at det er blitt en del av mellomtrinnet?
- Opplevde du å ha nok kompetanse i programmering når det kom i 2020?
- Holdningen din til programmering, har den endret seg fra 2020 til nå?
- Har du et inntrykk av hvordan innføringen av programmering har blitt mottatt av matematikklærere generelt? Om du har noen venner eller kollegaer eller om du har lest noe på nettet.
- Har du noen formell kompetanse i programmering?
    - Eventuelt har du hatt noen videreutdanning, noen kurs?
- Skulle du ønske at du hadde et bedre tilbud om videreutdanning eller kursing?
    - Sånn spesifikt, hva skulle du ønske du kunne fått kursing i?
- Nå nevnte jeg kursing, men en samling sammen med andre lærere der det kunne vært erfaringsdeling og undervisningsopplegg, har dere hatt noe slikt?
- Hvilke læremidler har dere tilgjengelig på skolen her. Til programmering.
- Er det noen læremidler som du tenker egner seg bedre til bestemte kompetansemål i arbeid med programmering? *leser opp kompetansemål fra 5. til 7. klasse* Hvordan jobber du med disse?

- Er det noen spesielle matematiske kunnskapsområder du tenker det er nyttig å bruke programmering? Med matematiske kunnskapsområder så tenker jeg på tall og tallforståelse, algebra, geometri, statistikk og funksjoner.
- Har du hatt noen utfordringer med å implementere programmering i undervisningen din? Hvilke utfordringer?
- Har du noen positive erfaringer?
    - Har du noen eksempler fra klasserommet?
- Hva mener du programmering kan bidra med i matematikkfaget når det kommer til å øke den matematiske kompetansen?
- Tenker du at programmering kan bidra til økt matematikkforståelse for noen?
- Hvilke vurderinger gjør du når du velger hvilke programmeringsoppgaver du skal gi til elevene dine?
- Er du kjent med begrepet å tenke algoritmisk?
    - Kunne du prøvd å forklare begrepet?
- Utforskning og problemløsning er jo et av de nye kjerneelementene nå. Hvordan tenker du at programmering kan bidra med dette?
- Hvordan tenker du programmering kan bidra til at elevene får jobbe med problemløsning og det å tenke algoritmisk?
- Har du noe på hjerte som ikke er blitt nevnt om programmering?