



Høgskulen på Vestlandet

Masteroppgave

MASIKT-OPG-OM-1-2021-VÅR-FLOWassign

Predefinert informasjon

Startdato:	18-05-2021 09:00	Termin:	2021 VÅR
Sluttdato:	01-06-2021 14:00	Vurderingsform:	Norsk 6-trinns skala (A-F)
Eksamensform:	Masteroppgave		
Flowkode:	203 MASIKT-OPG 1 OM-1 2021 VÅR		
Intern sensor:	(Anonymisert)		

Deltaker

Navn:	Torodd Flatråker Ottestad
Kandidatnr.:	202
HVL-id:	245987@hvl.no

Informasjon fra deltaker

Antall ord *:	26246
----------------------	-------

Egenerklæring *: Ja

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

Jeg godkjenner avtalen om publisering av masteroppgaven min *

Ja

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

Nei

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

Nei



Høgskulen
på Vestlandet

MASTEROPPGÅVE

Programmering som didaktisk verktøy i
matematikkfaget

Computer programming as a didactical
tool in mathematics education

Torodd Flatråker Ottestad

Master i IKT i læring

Institutt for pedagogikk, religion og samfunnsfag

Rettleiar: Anders Grov Nilsen

Innleveringsdato: 01.06.2021

Eg stadfestar at arbeidet er sjølvstendig utarbeida, og at referansar/kjeldetilvisingar til alle kjelder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.

Samandrag

Programmering og algoritmisk tenking har skuleåret 2020/2021 gjort sitt inntog i den vidaregåande skulen i Noreg. Tidlegare forskning har vist at tilnærminga og kompetansen til læraren har ei viktig rolle for elevane sitt utbyte, når det kjem til programmering i matematikkfaget (Misfeldt & Ejsing-Duun, 2015, s. 10).

Gjennom oppgåva har eg freista å svara på problemstillinga *Korleis kan programmering nyttast som didaktisk verktøy i matematikkfaget?* For å svara på problemstillinga har eg undersøkt kva *refleksjonar lærarane gjer seg kring kompetanseutviklinga si knytt til algoritmisk tenking*. I tillegg har eg sett på korleis dei *legg til rette for å løysa problem gjennom å nytta algoritmisk tenking*.

Eg har tilnærma meg forskingsfeltet gjennom ein holistisk kollektiv kasusstudie. I datainnsamlinga eg har har eg nytta kvalitative forskingsintervju av lærarar som hovudmetode. I tillegg har eg brukt innsamling av undervisningsmaterieell frå lærarane som supplerande metode.

Oppgåva legg eit sosiokulturelt syn på læring til grunn. I tillegg er definering av omgrepet *algoritmisk tenking*, og koplinga algoritmisk tenking har til matematikkfaget ein sentral del av det teoretiske bakteppet. Teori knytt til utvikling av kompetanse, vert òg presentert i teorikapittelet.

Funna mine viser at programmering vert nytta på ulike måtar i matematikkfaget. På den eine sida som eit sjølvstendig tema eller emne, og på den andre sida som ein integrert del av matematikkundervisinga. Korleis lærarane nyttar programmering ser ut til å vera avhengig av den didaktiske kompetansen deira, framfor matematisk kompetanse eller programmeringkompetanse.

Oppgåva har vist korleis den didaktiske kompetansen til lærarane kan utvikla seg, gjennom ulike refleksjonar knytt til både praksis og læringsutbyte, samt kurs og vidareutdanning. Samarbeid og utvikling i eit profesjonelt fellesskap ser òg ut til å leggja til rette for refleksjonar som kan vera med på å utvikla den didaktiske kompetansen til lærarane.

Abstract

Computer programming and Computational Thinking (CT) has made its way into the mathematics in upper secondary school in Norway, as a part of the ongoing transformation of the K-12 curriculum. Research has shown that the approach and competence of the teacher, plays an important role concerning student outcome when using computer programming in mathematics education (Misfeldt & Ejsing-Duun, 2015, s. 10). Through this paper I have tried to answer the research question «How can computer programming be used as a didactical tool in mathematics education?». To answer this question i have studied what *reflections teachers make, regarding development of their competence, related to CT*. I have also seen how the teachers *facilitate problem solving, through the use of CT*.

The approach to the research field has been through a *holistic multiple case-study*. The data collection has mainly been through qualitative research interviews with teachers. In addition, I have collected educational material from the teachers as a supplying source of data.

The paper is based on a socio-cultural view on learning. The theoretical backdrop consists of definition(s) of CT, and connections between CT and mathematics. In addition, some theory connected to teacher development is presented.

Findings I have made, shows that computer programming is used in different ways, in the mathematics education. On one side, it is treated as a separate theme by some teachers. On the other hand, it has been used as an integrated part of the teaching by others. How the teachers use programming, seems to depend on their didactical competence, not their competence in mathematics or programming.

The paper shows how the teacher's didactical competence can develop, through reflections, regarding both practice and learning outcomes, in addition to further courses and education. Collaboration and development in a professional community seems to facilitate reflections that can contribute the development of the teacher's didactical competence.

Føreord

Det kjennast både rart og godt å levera frå seg ei så stor oppgåve som denne. Det har vore to travle, interessante og nyttige år på masterstudiet i IKT i læring. Kombinasjonen jobb og fulltidsstudie har vorte utfordra av at pangolinar, flaggermus eller andre eksotiske råvarer frå Kina, har ført til både nettundervising, raude og gule lys i skulen samt minimalt med fysiske samlingar på studiet.

Å arbeida med eit prosjekt som dette, over så lang tid, har vore spennande. Det har vore rikt å få intervjuva fem ulike lærarar om kva dei tenkjer om eigen kompetanse og praksis. Trass i at det har vore bratte motbakkar i prosessen, kjenner eg no at eg gler meg til å sjølv få ta i bruk programmering og algoritmisk tenking i mi eiga undervising, frå neste skuleår.

Tusen takk til de fem som har stilt opp og delt av tida dykkar, i tillegg til tankar og praksis! Ein stor takk til rettleiaren min, Anders, for gode råd og konstruktive tilbakemeldingar gjennom prosessen. Takk til alle medstudentar for kjekke samlingar, sjølv om det vart litt færre enn planlagt. Særskild takk til Hanne for godt samarbeid i dei ulike emna, og for gode råd på vegen!

Sist, men ikkje minst, stor takk til Lillian for tolmod og god støtte når eg har vore langt inne i bobla mi!

God lesing!

Torodd F. Ottestad
Bergen, våren 2021

Innhald

Samandrag	ii
Abstract	iii
Føreord	iv
1 Innleiing	4
1.1 Mål med oppgåva.....	4
1.2 Bakgrunn for tema.....	5
1.2.1 Problemløysing og matematisk kompetanse	5
1.2.2 Stofftrensle.....	6
1.2.3 Programmering og matematikk.....	7
1.3 Problemstilling og forskingsspørsmål.....	7
1.4 Tidlegare forsking	11
1.4.1 Utvikling av algoritmisk tenking	11
1.4.2 Algoritmisk tenking og matematikk	12
1.4.3 Utfordringar og pedagogiske strategiar.....	13
1.5 Strukturen til oppgåva	14
2 Teori	15
2.1 Eit sosiokulturelt syn på læring.....	15
2.2 Algoritmisk tenking.....	15
2.2.1 Utgangspunkt i programmering.....	16
2.2.2 Utfordrande å definera	17
2.2.3 Algoritmisk tenking i skulen	18
2.2.4 Språklege utfordringar	19
2.2.5 Abstraksjon sentralt.....	19
2.2.6 Rammeverk.....	20
2.3 Å løyse problem	22
2.3.1 Kva er eit problem og kva er problemløysing?	22
2.3.2 Strategi for problemløysing	24
2.3.3 Problemløysing i matematikken og algoritmisk tenking	25
2.3.4 Klasseromskultur for problemløysing.....	26

2.4	<i>Utvikling av kompetanse</i>	27
2.4.1	Lærarprofesjonen og profesjonell utvikling.....	27
2.4.2	Kompetanseutvikling og drift.....	29
3	Metode	31
3.1	<i>Vitskapsteoretisk perspektiv</i>	31
3.2	<i>Forskningsdesign</i>	32
3.3	<i>Utval</i>	33
3.4	<i>Intervju</i>	34
3.4.1	Intervjuguide.....	34
3.4.2	Pilotintervju.....	35
3.4.3	Gjennomføring av intervju.....	36
3.5	<i>Innsamling av oppgaver</i>	37
3.6	<i>Analyse av data</i>	38
3.6.1	Transkribering av intervju.....	38
3.6.2	Koding og tolking av datamaterialet.....	39
3.7	<i>Kvalitet</i>	41
3.7.1	Reliabilitet	41
3.7.2	Validitet.....	41
3.7.3	Etikk.....	42
4	Presentasjon av funn	43
4.1	<i>Utvikling av kompetanse</i>	43
4.1.1	Kompetansen og oppfatningane til lærarane (<i>Personal domain</i>)	43
4.1.2	Klasserommet (<i>Domain of Practice</i>)	46
4.1.3	Kva skal me med dette? (<i>Domain of Consequence</i>)	48
4.1.4	Samarbeid og kollegaar (<i>External Domain</i>)	49
4.1.5	Kompetanse i åra som kjem.....	51
4.2	<i>Leggja til rette for å løysa problem</i>	54
4.2.1	Samtalen og spørsmål i klasserommet	54
4.2.2	Leggje til rette for læring	55
4.2.3	Feilsøking og uthald	58
4.3	<i>Innsamla undervisningsmateriale</i>	59
4.3.1	Tangram	60
4.3.2	Lese kode	60

4.3.3	Likningssett	61
5	Diskusjon	62
5.1	<i>Utvikling av kompetanse</i>	62
5.1.1	Kurs og videreutdanning	62
5.1.2	Klasserommet som testarena	65
5.1.3	Læring saman med andre	67
5.1.4	Kompetansen til lærarane når elevane sin kompetanse aukar	69
5.1.5	Kompetanseutvikling knytt til algoritmisk tenking gjennom algoritmisk tenking?	70
5.2	<i>Løse problem med algoritmisk tenking?.....</i>	71
5.2.1	Utforsking.....	71
5.2.2	Programmering og problemløysing?	74
5.3	<i>Kompetanse og praksis.....</i>	75
6	Avslutning.....	77
6.1	<i>Konklusjon.....</i>	77
6.1.1	Kompetanseutvikling knytt til algoritmisk tenking	77
6.1.2	Løse problem ved hjelp av algoritmisk tenking	79
6.2	<i>Implikasjonar og avgrensingar</i>	80
6.3	<i>Vegen vidare</i>	81
	Referansar	83
	Oversikt over vedlegg.....	88
	<i>Vedlegg 1: Intervjuguide</i>	89
	<i>Vedlegg 2: Kvittering NSD.....</i>	91
	<i>Vedlegg 3: Invitasjonsskriv.....</i>	94
	<i>Vedlegg 4: Tangram-oppgåve.....</i>	97

1 Innleiing

I inneverande skuleår (20/21) har startskotet gått for innføring av nye læreplanar i den norske skulen. På vidaregåande er første trinn i gang med nye læreplanar i alle fag. Det har i fleire år vore arbeidd med ei utvikling og fornying av skulen. I den offentlege utgreiinga «Fremtidens skole - Fornyelse av fag og kompetanser» frå 2015 vert det trekt fram at elevane har trong for opplæring i naturfag, matematikk og teknologi for m.a. å «forstå og engasjere seg i samfunnsspørsmål» (NOU 2015:8, s. 24). I regjeringa sin strategi «Framtid, fornyelse og digitalisering: Digitaliseringsstrategi for grunnsopplæring 2017–2021» vert dette igjen teke opp, med eit enno større fokus på teknologi-aspektet:

Det er viktig at elever forstår både samfunnsmessige konsekvenser av digitaliseringen, og hvordan teknologiene fungerer. Å kunne vurdere et problem på en slik måte at en datamaskin kan hjelpe oss til å løse det, forutsetter evnen til å bryte problemet ned i logiske steg – i en algoritme. (Kunnskapsdepartementet, 2017, s. 18)

Dei offentlege dokumenta nemnt over er med på å danna bakteppet for endringane som er gjort med matematikkfaget i Fagfornyinga. I tillegg til endringar i kompetansemåla i dei ulike læreplanane, har Fagfornyinga innført *kjerneelement* i dei ulike faga. I matematikk er eit av desse kjerneelementa «Utforsking og problemløysing» der programmering og algoritmisk tenking står sentralt (Kunnskapsdepartementet, 2018). Utforsking og problemløysing finn ein att i dei ulike kompetansemåla til dei einskilde matematikkfaga i skulen. I kompetansemåla til IT (teoretisk matematikk) på finn me døme på dette. Eit av dei nye kompetansemåla der seier at eleven skal «...formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering» (Utdanningsdirektoratet, 2019b).

1.1 Mål med oppgåva

For å gje eit innblikk i temaet og måla mine med oppgåva, vil eg nytta inndelinga til Maxwell (2008, s. 219) i *personlege, intellektuelle og praktiske mål*.

Dei **personlege måla** mine for oppgåva er knytt til jobben min som lærar. Eg arbeider som matematikklærar i den vidaregåande skulen. Gjennom arbeidet mitt har eg utvikla interesse for bruken av digitale verktøy i matematikkfaget. Noko av det eg er oppteken av er å finna digitale verktøy som kan vera med på å auka interessa og læringsutbyttet til elevane. Døme på dette kan vera at elevane brukar GeoGebra til å sjå på samanhengar mellom ulike matematiske omgrep, eller at dei nyttar video for å trenast seg i å *snakka matte* og får trening i å forklara matematiske omgrep. I inneverande skuleår, 2020/2021, er arbeidet med å innføra Læreplanverket for Kunnskapsløftet 2020 (LK20) i gang på første trinn i den vidaregåande skulen. Med LK20 kjem programmering og algoritmisk tenking inn i læreplanen og matematikkfaget. Med utdanning innan matematikk for industri og teknologi (anvendt matematikk) er dette omgrep som interesserer meg, og som det var naturleg å fordjupa seg i når eg skulle i gang med ei masteroppgåve innan IKT i læring.

Dei **intellektuelle måla** handlar om kva rolle prosjektet mitt har i forskingsfeltet. Ein har dei siste åra vurdert å implementera programmering i den norske skulen. Når dette no kjem inn i skulen gjennom Fagfornyinga vil det vera trong for forskning som ser på korleis denne fornyinga av matematikkfaget går føre seg. Det intellektuelle målet med prosjektet mitt å sjå nærare på korleis lærarane handterer denne Fagfornyinga.

Ein del av motivasjonen med prosjektet er òg knytt til det **praktiske målet**. Det er eit mål at matematikklærarar skal kunna få inspirasjon og kunnskap knytt til bruk av programmering og algoritmisk tenking i skulen, og ta med dette inn i eiga utvikling og undervising.

1.2 Bakgrunn for tema

1.2.1 Problemløysing og matematisk kompetanse

Utdanningsdirektoratet (2019a) omtalar algoritmisk tenking som ein problemløysingsstrategi. Problemløysing er eit omgrep som er mykje nytta i matematikdidaktikken. I eit forsøk på å definera matematisk kompetanse trakk Niss og Høygård Jensen (2002, s. 45) fram «problembehandlingskompetence» som ein av åtte delkompetansar. Dei omtalte eit «problem» i matematikk som matematiske oppgåver som ikkje kunne løysast gjennom rutineferdigheiter, og at det med dette var eit *relativt*

omgrep avhengig av kompetansen til den som løyser problemet (s. 49-50).

Utdanningsdirektoratet (2019b) definerer det som problemløysing når «...elevane utviklar ein metode for å løyse eit problem dei ikkje kjenner frå før» som er tett knytt opp mot definisjonen nemnt over. I matematikdidaktikken vert ofte «god» undervising mellom anna kopla til bruken av problemløysing. National Council of Teachers of Mathematics (2014, s. 7-8) omtalar «god undervising» som undervising som legg til rette for at elevane skal utvikla (djup) forståing. Eit av kjenneteikna på denne type undervising er bruk av problemløysing (s. 17).

Sjølv om programmering i læreplanen er nytt i skulen i Noreg, er programmering i skulen noko som har vore omtalt i fleire tiår. For meir enn førti år sidan uttrykte Seymour Papert at ved å «...teaching the computer how to think, children embark on an exploration about how they themselves think» (Papert, 1980, s. 19). Denne tenkemåten knytte han opp mot omgrepet han kalla for «computational thinking» (Papert, 1980). På norsk vert ofte omsettinga «algoritmisk tenking» som nemnt tidlegare nytta om dette omgrepet. I kap. 2.2 vil eg gå nærare inn på definisjonen og omsettinga av dette omgrepet.

1.2.2 Stofftrensle

Digitaliseringsstrategien for 2017 - 2021 nemnt innleiingsvis trekk òg fram at det i Fagfornyinga skal arbeidast med korleis ein kan få plass til algoritmisk tenking og programmering, og kva rolle det skal få, i faga matematikk og naturfag (Kunnskapsdepartementet, 2017, s. 18). I 2016 vart det gjort ein litteraturgjennomgang knytt til teknologi og programmering i grunnsopplæringa i Noreg (Sanne et al., 2016). Der vert nytta av programmering i matematikk trekt fram, og algoritmisk tankegang og evne til problemløysing er nokre av føremonane dei nemner. I rapporten kjem det fram at skulen vil kunne nytta godt av ei fornying, der ein brukar programmering og teknologi (Sanne et al., 2016, s. 20). Samstundes legg dei til grunn at elevane har eit «teknologifag» der dei kan læra sjølve programmeringa (Sanne et al., 2016, s. 61-62). På den måten vil ein kunne ha fokus på å bruka programmering som eit verktøy for å løysa matematiske problem, framfor å ha fokus på opplæring i sjølve programmeringa. I Fagfornyinga er ikkje eit slikt teknologifag på plass, og opplæring i programmering og algoritmisk tenking er i hovudsak lagt til matematikkfaget. Intervju gjort med lærarar dei to siste åra viser at ein del lærarar er uroa for om dette kjem til å føra til

«stofftrengsle» i matematikkfaget (Finstad, 2020, s. 110; Jakobsen, 2019, s. 54). Med dette meiner ein frykt for at programmering og algoritmisk tenking berre vert enno eit tema som ein må gjennom, og at tida og fagstoffet ikkje samsvarar. Det kan tenkjast at dette vil vera knytt til korleis programmeringa vert innført og nytta i undervisinga.

1.2.3 Programmering og matematikk

Det er gjort nokre interessante studiar knytt til programmering og matematikk. Her vil eg trekkja fram nokre av dei. I kap. 1.4 vil eg gå grundigare inn på funna frå ein litteraturgjennomgang eg har gjort, som ein del av førebuinga til denne oppgåva.

Ein nyare norsk studie som har sett på programmering og matematikk trekkjer fram eit dilemma ein som lærar må ta stilling til: nyttar ein programmering for å gje ei betre forståing for matematikken, eller nyttar ein matematikk for å betra evner i problemløysing knytt til programmering (Kaufmann & Stenseth, 2020, s. 17)? Den same studien trekkjer fram **trongen til å sjå meir på korleis læraren påverkar problemløysingsprosessen til eleven** (s. 18). Ein litteraturgjennomgang som tek føre seg programmering i matematikkopplæringa peikar på **trongen til å sjå meir på kva rolle læraren har når det kjem til den pedagogiske praksisen** (Forsström & Kaufmann, 2018, s. 30). Dette heng saman med at det er *vanlege* matematikklærarar som vert sett til undervising i programmering (s. 30). Det er naturleg å tenkja at programmeringskompetansen hjå norske matematikklærarar er låg, då programmering ikkje har vore ein obligatorisk del av skulefaget tidlegare. Ein artikkel som har sett nærare på matematikklæring gjennom programmering trekkjer fram ulike føremonar og ulemper dette kan ha. Her vert det konkludert med at det kan sjå ut som utbyttet av matematikklæring gjennom programmering «...depends largely on the teacher's approach and didactical principles» (Misfeldt & Ejsing-Duun, 2015, s. 10).

1.3 Problemstilling og forskingsspørsmål

Desse forskingsresultata kombinert med Fagfornyinga og eigen interesse for teknologibruk i matematikkundervisinga gjer at eg finn det interessant å sjå på korleis algoritmisk tenking og programmering kan nyttast i matematikkfaget, samt sjå nærare på læraren si rolle. Rolla til læraren og føresetnadene hans er ein sentral del av fagområdet matematikkdiraktikk.

Som følge av at inneverande skuleår er det første året der programmering er ein del av læreplan og kompetansemål i matematikkfaget, vil det vera av interesse å få eit innblikk i korleis denne implementasjonen vert gjennomført på lærarnivå. Det er òg trong for forskning knytt til norske klasserom med dei rammevikåra som gjeld her, då mykje av forskinga knytt til programmering og matematikk er frå utlandet. Fokuset i arbeidet mitt vil vera i den vidaregåande skulen, og matematikkfaget 1T på vg1, sidan det per no berre er dette trinnet på vgs som har tatt i bruk dei fornya læreplanane. Den overordna problemstillinga mi for arbeidet vidare er:

P: Korleis kan programmering nyttast som didaktisk verktøy i matematikkundervisinga?

Det didaktiske aspektet i problemstillinga omhandlar måten lærarane nyttar programmering i undervisinga si. Programmering heng tett saman med tenkjemåten og omgrepet *algoritmisk tenking*. I læreplanmålet for 1T er det eit tydeleg skilje mellom algoritmisk tenking og programmering då det vert sagt at eleven skal kunne «...formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering» (Utdanningsdirektoratet, 2019b). I denne oppgåva nyttar eg det nynorske «tenking» som «tenkning» på bokmål, det vil seie at det er snakk om ein algoritmisk tenkjemåte, meir enn berre algoritmiske tankar. Omgrepet algoritmisk tenking er omsett frå det engelske Computational Thinking (CT). Gjøvik og Torkildsen (2019, s. 32) trekkjer fram algoritmisk tankegang som ei meir forklarande omsetting. Omgrepa algoritmisk tenking og programmering vert ofte knytt tett saman, men det er ein vanleg oppfatning at algoritmisk tenking er eit vidare omgrep enn programmering: «A broad understanding that frames CT as something more than programming, encompassing key 21st century skills like problem solving, logical thinking and creativity» (Balanskat & Engelhardt, 2015). I rapporten «Teknologi og programmering for alle» vert denne definisjonen nytta på programmering: «Programmering vil si å bryte et gitt problem ned i et sett av kommandoer og så få en datamaskin til å utføre disse kommandoene» (Sanne et al., 2016, s. 18). Ein annan definisjon finn me hjå Balanskat og Engelhardt (2015, s. 7) som seier at «Computer programming is the process of developing and implementing various sets of instructions to enable a computer to perform a certain task, solve

problems, and provide human interactivity». I begge desse definisjonane ligg det meir enn berre det å få datamaskina til å følgja instruksjonar, sjølve skrivinga av programkoden. Prosessen med å utforma kommandoane er òg inkludert. I oppgåva vil eg nytta programmering som eit utvida omgrep, der både nedbryting av problemet i handterlege kommandoar samt skriving av programkode er inkludert.

I datainnsamlinga vil eg i hovudsak spørja om algoritmisk tenking, og ikkje programmering. På den måten vil det vera mogleg å sjå kva for, om nokre, koplingar lærarane trekkjer mellom algoritmisk tenking og programmering.

For å konkretisera det vidare arbeidet mitt har eg utforma to forskingsspørsmål:

F1: Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?

F2: Korleis legg lærarar til rette for å løyse problem ved hjelp av algoritmisk tenking?

Gjennom det første forskingsspørsmålet er målet å sjå nærare på lærarane sine refleksjonar kring eigen kompetanse. Trass i at algoritmisk tenking og programmering no vert ein sentral del av matematikkfaget er det nok ein stor del av lærarane som ikkje har den erfaringa eller kompetansen dei skulle hatt. Det er nærliggjande å tenkja at læraren sin kompetanse vil ha verknad på *korleis læraren nyttar programmering som didaktisk verktøy*. Særleg interessant er det å sjå på korleis lærarane tenkjer og reflekterer rundt eigen kompetanse og utvikling av denne kompetansen for å kunna nytta algoritmisk tenking i undervisinga si. Gjennom F1 vil ein få tilgang til denne type refleksjonar hjå lærarane.

Desse refleksjonane vil det vera interessant å sjå opp mot korleis lærarane legg til rette for at elevane skal nytta algoritmisk tenking. F2 er utforma for å gjera denne vinklinga handterleg. I kompetansemålet der algoritmisk tenking vert nytta i den nye læreplanen står det at algoritmisk tenking skal nyttast for å løyse problem. Det er med andre ord ein tydeleg samanheng mellom forskingsspørsmålet og det aktuelle kompetansemålet.

Ved å sjå nærare på desse to forskingsspørsmåla vil ein få eit innblikk i noko av det som ligg bak måten lærarar nyttar programmering på i faget, samt døme på korleis

programmering vert brukt som verktøy. I tillegg vil det vera interessant å sjå om det er nokon samanheng mellom korleis lærarane reflekterer kring utviklinga av kompetansen sin, og måten dei underviser på sjølv.

1.4 Tidlegare forskning

Som ein del av eit tidlegare emne ved masterstudiet (MASIKT-TEK02 Teknologisk fagprofil 2) har eg gjennomført ein litteraturgjennomgang knytt til algoritmisk tenking og programmering og matematikkfaget. I litteratursøket nytta eg meg av søkematrissa vist i Tabell 1. I tillegg har eg gjort ulike strategiske og tilfeldige søk i etterkant av litteraturgjennomgangen som har vore av interesse for temaet. Desse vil eg òg inkludera i kapittelet. I det følgjande vil eg presentera nokre av funna eg gjorde i dette arbeidet.

Tabell 1 - Søkematrise

Tema	Inkludert	Ekkludert
Database	ERIC, Academic Search Elite, Google Scholar	PubMed, Science Direct, Web of Science
Tid	2014 - 2020	2013 og eldre
Fokus	Studiar og rapportar som omhandlar programmering og algoritmisk tenking i vidaregåande skule og ungdomsskule.	Litteratur knytt til barnehage og barnskule og mellomtrinn (1-7).
Type aktivitet	Litteratur som omhandlar programmering og algoritmisk tenking i K7 - K12 i, eller i samband med, matematikkfag.	Litteratur utan tilknytning til algoritmisk tenking eller matematikk.
Språk	Norsk, engelsk	Andre språk
Søkeord	Computational thinking, programming, high school, secondary school, mathematics, learning, teaching Algoritmisk tenking, programmering, vidaregåande, matematikk, læring, undervisning	
Metode	Kvalitative, kvantitative, mixed method	Ingen

1.4.1 Utvikling av algoritmisk tenking

Ein nyare svensk studie utført av Nouri et al. (2020) freista å sjå om elevane utviklar nokre ferdigheiter gjennom programmering. Forskingsspørsmålet til studien var «From

the teachers' perspective, what skills do pupils obtain when engaging in programming activities?» (Nouri et al., 2020, s. 3). Studien har sett på ferdigheiter knytt til algoritmisk tenking, samt andre digitale ferdigheiter. Når ferdigheitene knytt til algoritmisk tenking vart analysert, nytta forskarane eit rammeverk utvikla av Brennan og Resnick (2012, s. 3-10) som deler inn algoritmisk tenking i underkategoriene *computational thinking concepts, practices* og *perspectives*. Dette rammeverket skriv eg meir om i teorikapittelet (kap. 2.2.6). Dei mest grunnleggjande ferdigheitene knytt til programmering er det ein finn att i den første underkategorien i rammeverket, *concepts*. I den svenske studien kjem det fram at fleirtalet av ferdigheitene som vart utvikla hjå elevane kom inn under nettopp *computational thinking concepts* (Nouri et al., 2020, s. 8). Det er ikkje godt å seie kva som gjer at elevane ikkje utviklar ferdigheiter på eit «høgare» nivå. Eit viktig moment som kjem fram i studien er at *dei svenske lærarane ikkje har god kompetanse eller erfaring med programmering eller det å undervisa i programmering* (Nouri et al., 2020, s. 12). Dette gjer at ein risikerer at lærarane ikkje kjenner til dei ulike føremonane programmering kan ha, og kva ferdigheiter ein kan utvikla gjennom programmering og algoritmisk tenking. Kanskje kan den manglande kompetansen hjå lærarane farga oppfatninga dei har av elevane sitt læringsutbytte. Desse funna gjer det *interessant å sjå nærare på utvikling av kompetanse hjå norske lærarar* når dei no vert «tvungen» til å nytta algoritmisk tenking og programmering i undervisinga si.

1.4.2 Algoritmisk tenking og matematikk

Nokre av resultatata (Costa et al., 2017; Kaufmann & Stenseth, 2020) frå litteraturgjennomgangen min såg nærare på koplinga mellom matematikk og algoritmisk tenking, især verknaden algoritmisk tenking har på elevane sine problemløysingsferdigheiter. I ei brasiliansk studie utført av Costa et al. (2017) har dei sett på om det å tilføra algoritmisk tenking, til forklaringar og spørsmål i arbeid med matematikk, har innverknad på korleis elevane gjer det når dei held på med problemløysing. Studien er utført på to elevgrupper (ei testgruppe og ei kontrollgruppe) frå ein skule som var med i PISA i 2012 (Costa et al., 2017, s. 3). Testgruppa gjekk gjennom eit undervisningsopplegg med øvingsoppgåver og testar bygd opp etter ein 9-stegsmodell basert på ulike ferdigheiter knytt til algoritmisk tenking, medan kontrollgruppa fekk «vanlege» spørsmål og oppgåver (Costa et al., 2017, s. 4). I etterkant av opplegget fekk begge gruppene ein test basert på PISA-oppgåver. Elevane i

testgruppa fekk ifølgje studien betre konsentrasjon, og brukte med det kortare tid på testen, i tillegg til at svara deira hadde høgare kvalitet enn kontrollgruppa (Costa et al., 2017, s. 5). Noko av det som *er interessant med denne studien er at det ikkje er nytta programmering*, og resultatane tyder på at bruk av algoritmisk tenking «analogt» kan ha ei positiv effekt.

Ein systematisk litteraturgjennomgang gjort av Popat og Starkey (2019, s. 367) har sett på læringsutbytte knytt til programmering i skulen «...beyond computer science and computational thinking». I alle dei ti forskingsartiklane dei har gjennomgått har ein sett at eit læringsutbytte når ein nyttar programmering kan vera problemløysing gjennom matematiske omgrep (Popat & Starkey, 2019, s. 367). Likevel vert det peika på at det å lære matematisk problemløysing gjennom programmering ikkje nødvendigvis er betre enn om ein lærer det direkte (Popat & Starkey, 2019, s. 368). Studien peikar vidare på at dei pedagogiske intensjonane til læraren er ein viktig faktor (Popat & Starkey, 2019, s. 368).

1.4.3 Utfordringar og pedagogiske strategiar

Sentance og Csizmadia (2017) har sett på lærarar i Storbritannia sitt syn på innføring av programmering som ein del av pensum. I undersøkinga er over 300 lærarar, som underviser i programmering, spurd om deira syn på pedagogiske strategiar og utfordringar knytt til denne undervisinga (Sentance & Csizmadia, 2017, s. 471). Nokre av utfordringane som kjem fram i studien er læraren sin kompetanse, manglande forståing hjå elevane, tekniske problem og elevane sin mangel på vilje og evne til problemløysing (Sentance & Csizmadia, 2017, s. 479). 75 % av lærarane i studien underviste elevar mellom 14 og 16 år, og 57 % underviste elevar mellom 16 og 18 år (Sentance & Csizmadia, 2017, s. 478). Lærarane i Storbritannia rapporterte om god effekt av praktiske øvingar og demonstrasjonar for å undervisa elevane i ferdigheiter knytt til algoritmisk tenking (Sentance & Csizmadia, 2017, s. 485). Studien peikar på at *det er stor skilnad i det å undervisa i programvare, som mange lærarar har god erfaring med, og det å undervisa i programmering* (Sentance & Csizmadia, 2017, s. 490).

I artikkelen «Learning Mathematics Through Programming: An Instrumental Approach to Potentials and Pitfalls» av Misfeldt og Ejsing-Duun (2015) vert det presentert ein

modell med tre føremonar læring med programmering kan ha: elevane vert produsentar, programmering støttar abstrakt tenking og utvikling av algoritmisk tenking (Misfeldt & Ejsing-Duun, 2015, s. 3). Artikkelen til Misfeldt og Ejsing-Duun (2015, s. 10) konkluderer, basert på presentert teori og døme frå klasserom, med at potensialet til læring av matematikk gjennom programmering «...depends largely on the teacher's approach and didactical principles». Kva tilnærming og didaktiske grep ein lærar gjer i undervisning av algoritmisk tenking og programmering vil nok vera avhengig av i kva grad læraren sjølv har av kompetanse, erfaringar og kunnskap.

Noko av det som går igjen i fleirtalet av studiane i litteraturgjennomgangen min, er rolla læraren sin kompetanse har for nytta av programmering og algoritmisk tenking i matematikkfaget. Ein annan studie utført av Voogt et al. (2015) har sett på litteratur som omhandlar algoritmisk tenking og definisjonar av dette omgrepet. Dei konkluderer med at det er trong for forskning på implementering av algoritmisk tenking i skulen, mellom anna utvikling av algoritmisk tenking i andre fag enn «Computer Science» (Voogt et al., 2015, s. 726).

1.5 Strukturen til oppgåva

I dette kapitlet har eg gjort greie for måla og bakgrunnen for oppgåva, samt presentert problemstillinga og forskningsspørsmåla eg har utvikla. I kapittel 2 presenterer eg teori knytt til algoritmisk tenking og rundt det å leggja til rette for å løyse problem. Eg vil òg sjå på aktuell teori knytt til utvikling av kompetanse hjå lærarar. Kapittel 3 tek føre seg dei metodiske vala eg har gjort, og prosesser rundt innsamling og analyse av data. I kapittel 4 presenterer eg funna eg har gjort i datamaterialet. Vidare, i kapittel 5, drøftar eg dei presenterte funna og knyt dei opp mot aktuell teori. Kapittel 6 inneheld konklusjonen for oppgåva, og mine tankar om vegen vidare.

2 Teori

I dette kapitlet vil eg gjera greie for aktuell teori knytt til oppgåva. Først vil eg presentera læringssynet eg tek utgangspunkt i, før eg vidare trekkjer fram teori knytt til algoritmisk tenking, løysing av problem og utvikling av kompetanse.

2.1 Eit sosiokulturelt syn på læring

Problemstillinga eg arbeider ut frå handlar om *bruk av programmering og algoritmisk tenking i undervisinga*. Forskingsspørsmåla ser nærare på *korleis lærarar utviklar kompetansen sin og korleis lærarane legg til rette for at elevane skal læra*. Dette gjer at oppgåva mi ser på læring i fleire nivå: læring hjå lærarane og læring hjå elevane. Av erfaringa mi frå skulen veit eg at samarbeid gjerne er ein viktig del av profesjonsfellesskapet og utvikling av kompetanse. Samarbeid er òg eit ord som er tett knytt til algoritmisk tenking og til matematikdidaktikk. Dette gjer det naturleg å ta utgangspunkt i eit sosiokulturelt læringssyn i denne oppgåva. Ein mykje brukt metafor for læring i matematikdidaktikken er «læring som deltagelse» (Skott, 2018, s. 91-136). Metaforen er tett knytt opp til sosiokulturell læringsteori. I den sosiokulturelle læringsteorien er utgangspunktet at læring skjer gjennom deltaking i fellesskap, samt gjennom å nytta medierande artefakt (Säljö, 2013, s. 72-73). Medierande artefakt kan vera psykiske reiskaper som språk, eller fysiske reiskaper som datamaskiner eller annan teknologi (Säljö, 2013, s. 73). I programmering nyttar ein språk, både som ein del av samarbeidet, men òg som ein del av sjølve kodinga (programmeringsspråk). Dialog og samarbeid er, i sosiokulturell læringsteori, tett knytt opp til omgrepet om den proksimale utviklingssona som er området mellom det ein klarar åleine og det ein klarar med hjelp frå ein meir kapabel læringspartner (Vygotskiï & Cole, 1978, s. 86). Teknologi vert gjerne sett på som eit hjelpemiddel som gjev tilgang til ei rekkje meir kapable læringspartnarar, som ekspertar og kunnskapsressursar (Selwyn, 2016, s. 86). Alt dette er med på å gjera at eg vel å leggja det sosiokulturelle synet på læring til grunn for denne oppgåva.

2.2 Algoritmisk tenking

Algoritmisk tenking (Computational Thinking) er eit omgrep med ei rekkje ulike definisjonar og meiningar. Dette gjer det naudsynt å gå litt i djupna, slik at skilnadane

på dei ulike definisjonane kjem fram. I det følgjande vil eg gje ei oversikt over ulike definisjonar og skildra ulike fasettar av omgrepet algoritmisk tenking. Sidan problemstillinga og forskingsspørsmåla i oppgåva omhandlar det norske klasserommet, vil eg òg tydeleggjera korleis algoritmisk tenking vert nytta som omgrep i Fagfornyninga, samt kva perspektiv på algoritmisk tenking eg nyttar i oppgåva.

2.2.1 Utgangspunkt i programmering

Som nemnt i innleiinga vart omgrepet introdusert av Seymour Papert i starten av 80-talet. I boka *Mindstorms* trakk han samanhengar mellom det å «læra datamaskina å tenkja» og å læra noko om eiga tenking (Papert, 1980, s. 19). Arbeidet til Papert omhandla i hovudsak *programmering* i programmeringsspråket LOGO (Papert, 1980, s. 8). Papert meinte at ein gjennom bruk av datamaskiner ville kunna skapa effektive og suksessrike læringsmiljø (i kontrast til den eksisterande skulen) slik at «...schools as we know them today will have no place in the future» (Papert, 1980, s. 9). Omgrepet algoritmisk tenking har seinare utvikla seg til å famna om meir enn det å «læra datamaskina å tenkja», sjølve programmeringa, til å vera eit vidare omgrep. Voogt et al. (2015, s. 716) trekkjer fram at det i litteraturen ofte verkar som om algoritmisk tenking og programmering er det same, og at ein *må* nytta programmering ved algoritmisk tenking. Basert på tidlegare forskning trekk forfattarane fram at:

To focus on programming as the primary instantiation of CT, and to expect programming alone to result in more refined thinking, similar to the early research on students' use of LOGO, would be a mistake both conceptually and pedagogically. (Voogt et al., 2015, s. 718)

Det er med andre ord sentralt å sjå på algoritmisk tenking som *meir enn programmering*, og at det ikkje er nokon automatikk mellom bruk av programmering åleine og eit endra eller betra tankesett. For å få eit innblikk i kva algoritmisk tenking kan vera, meir enn programmering, vil eg ta føre meg nokre definisjonar på omgrepet. Barr og Stephenson (2011, s. 51) seier at algoritmisk tenking «...is an approach to solving problems in a way that can be implemented with a computer». Her står bruken av datamaskin sentralt, samstundes som omgrepet vert definert som ein måte å tilnærma seg løysing av problem. Jeanette M. Wing definerer på si side algoritmisk tenking på denne måten: «Computational thinking is the thought processes involved in formulating

a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out» (Wing, 2017, s. 8). I denne definisjonen av algoritmisk tenking står formulering og løysing av problem òg fram som ein sentral del. Samstundes vert det, i motsetnad til Barr og Stephenson sin definisjon, framheva at ein «computer» både kan vera menneske og maskinar. Omgrepet er med andre ord ikkje uløyselig knytt til datamaskiner og programmering, men dreiar seg om eit *tankesett brukt i problemløysing*.

2.2.2 Utfordrande å definera

Å definera eit omgrep som algoritmisk tenking er ikkje utan utfordringar. Voogt et al. (2015, s. 719) trekkjer fram ein problematikk som oppstår når ein fokuserer for mykje på kva som skal inngå som «core competencies» og «peripheral competencies» i defineringa av omgrepet. Ei av desse utfordringane er at når denne type opplisting av kompetansar vert for brei, vert algoritmisk tenking vanskeleg å skilja frå andre «21st century skills» (Voogt et al., 2015, s. 719). Ei anna utfordring kan vera at fokuset vert for knytt til den logiske definisjonen, framfor eit fokus på korleis ein nyttar algoritmisk tenking (Voogt et al., 2015, s. 719). Samstundes kan ein argumentera for det motsette, at ein ved å definera kva som skal inngå som kjernekompetansar får ein meir nyttig definisjon. Dette vert trekt fram av Barr og Stephenson (2011, s. 50) som hevdar at «To be useful, a definition must ultimately be coupled with examples that demonstrate how computational thinking can be incorporated in the classroom». I deira arbeid med å definera omgrepet utvikla dei ei oversikt over «core concepts and capabilities» med døme på korleis dei ulike elementa kunne knytast til ulike fag i K12 (Barr & Stephenson, 2011, s. 51-52).

Ein kan gjerne tenka at dei ulike definisjonane og måtane å definera algoritmisk tenking på, har ulike bruksområde. Når ein skal nytta definisjonen som analytisk utgangspunkt i forskning kan ein tenkja at ein definisjon som den til Wing (2017, s. 8) er fin å nytta. På den andre sida kan det tenkjast at denne definisjonen ikkje er like nyttig for å syna lærarar korleis dei kan implementera algoritmisk tenking i undervisinga si. Då vil gjerne ein definisjon med ulike «core concepts and capabilities», som til dømes den presentert av Barr og Stephenson (2011, s. 51-52), vera betre.

2.2.3 Algoritmisk tenking i skulen

Computing At School (CAS) presenterer òg algoritmisk tenking gjennom ulike omgrep (*concepts*) og arbeidsmetodar (*approaches*) (Csizmadia et al., 2015, s. 6). Deira definisjon av omgrepet er presentert i ein guide til lærarar for å forstå omgrepet algoritmisk tenking. I presentasjonen av omgrepet er dei tydelege på at algoritmisk tenking handlar om å utvikla eit tankesett (thinking skills), samt at det er noko som støttar læring og forståing (Csizmadia et al., 2015, s. 6). I arbeidet med Fagfornyinga er det denne definisjonen som er nytta av Udir, og dei har laga ein norsk versjon av CAS sin illustrasjon, vist i Figur 1 (Utdanningsdirektoratet, 2019a). I denne oppgåva nyttar eg denne forståinga av omgrepet, mellom anna av di det er denne operasjonaliserte versjonen som ligg til grunn frå Utdanningsdirektoratet si side. Samstundes er inndelinga i arbeidsmåtar og nøkkelomgrep nyttig, og forståeleg, for lærarane i skulen.



Figur 1 – Den algoritmiske tenkaren. Henta frå <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>

2.2.4 Språklege utfordringar

Det engelske omgrepet *computational* vert i den norske definisjonen omsett til *algoritmisk*. Dette er to ord med noko ulik mening. Det er naturleg å knyta ordet *computational* opp mot både *computer* (datamaskin) og *computation* (rekning). I CAS sin definisjon vert det som den norske utgåva presenterer som «Algoritmar» kalla «Algorithmic thinking» (Csizmadia et al., 2015, s. 7). Direkte omsett vil dette vera *algoritmisk tenking*. Problemet er at dette er knytt til den bokstavlege tolkinga av omgrepet, at det i hovudsak handlar om at ein nyttar og lagar reglar og stegvise framgangsmåtar (Csizmadia et al., 2015, s. 7). Denne forståinga kan ein knyta opp mot det å laga algoritmar som ein seinare skal programmera. Kluge (2020, s. 123) nyttar *digital tenking* som den norske omsettinga av *computational thinking*. Vidare omtalar Kluge (2020, s. 123) den bokstavlege forståinga som «*digital tenking som programmeringskompetanse*». Det kan tenkjast at det er denne type smal forståing av *algoritmisk tenking* (algorithmic thinking) som vil råda hjå lærarar utan kjennskap til det vidare omgrepet *algoritmisk tenking* (computational thinking). Kluge (2020, s. 123) omtalar ei litt vidare forståing av omgrepet som «*digital tenking som forståelse av verden rundt oss og aktiviteten i den*». Her vert mellom anna programmering, algoritmeutvikling og samarbeid trekt fram som sentrale element (s. 123). Denne doble meninga eller forståinga som kan ligga i det norske omgrepet *algoritmisk tenking* er noko eg tenkjer er viktig å ha med i bakhovudet vidare i oppgåva.

2.2.5 Abstraksjon sentralt

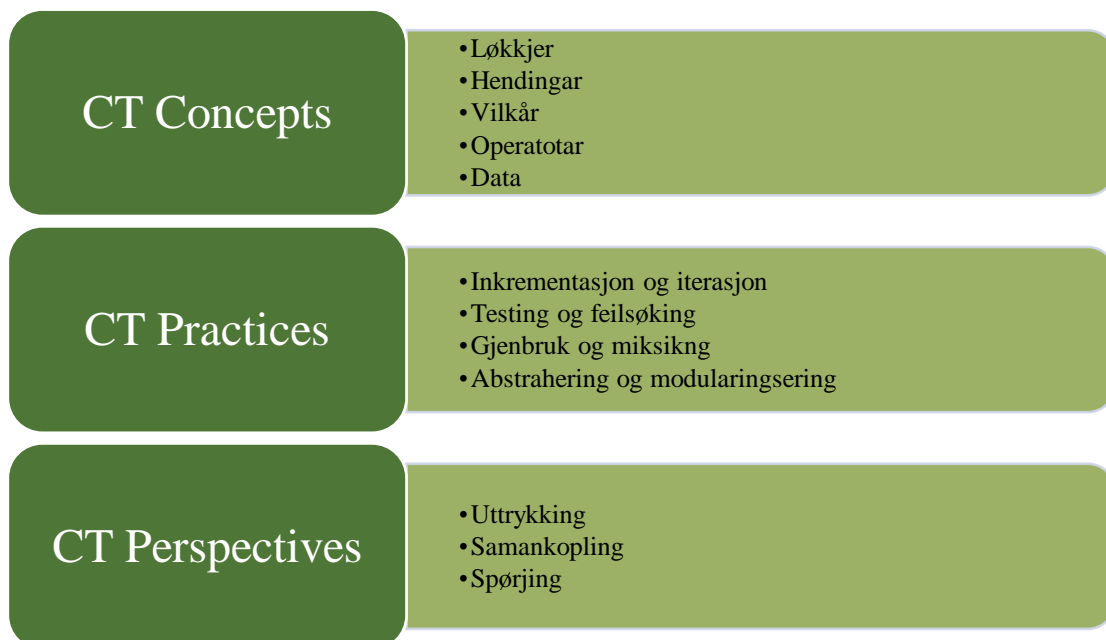
Noko som går att i mange definisjonar er ordet «abstraction» (García Peñalvo et al., 2016, s. 19). I definisjonen, presentert av CAS, handlar abstraksjon om å gjera problem lettare ved å sjå vekk frå unødvendige detaljar (Csizmadia et al., 2015, s. 7). Dei trekkjer fram det verdskjende kartet over undergrunnsbana i London som eit døme på dette: distanse, geografisk plassering, gater er forstyrrende element som ikkje treng vera med (Csizmadia et al., 2015, s. 7). Wing (2017, s. 8) omtalar abstraksjon som heilt sentralt når det kjem til algoritmisk tenking. Abstraksjonsprosessen kan skje rekursivt¹, og med det kan ein gradvis laga store system sett saman av små og handterlege komponentar (Wing, 2017, s. 8). Både mellom forskarar og utdannarar i datateknologi er det konsensus om at abstraksjon er det mest sentrale, når ein snakkar om algoritmisk

¹ Gjentakande

tenking. Grover og Pea (2013, s. 39), som har gjort ein større litteraturgjennomgang knytt til algoritmisk tenking i skulen, seier det slik: «The value of abstraction as CT's keystone (distinguishing it from other types of thinking) is undisputed».

2.2.6 Rammeverk

For å kunna vurdera barn og unge si utvikling av algoritmisk tenking, gjennom programmering i det blokkbaserte programmeringsspråket Scratch², utvikla Karen Brennan og Mitchel Resnick eit nytt rammeverk (Brennan & Resnick, 2012). Dette rammeverket er bygd opp kring tre hovuddimensjonar. Desse er «computational thinking concepts», «computational thinking practices» og «computational thinking perspectives» (Brennan & Resnick, 2012, s. 3). Trass i at rammeverket er utvikla med blokkbasert programmering som fokus, og utvikling av «interaktive media» vil det nok kunna nyttast på andre programmeringsspråk og aldersgrupper. Dette støttar Lye og Koh (2014, s. 52-53) som peikar på at rammeverket godt kan brukast for å studera utvikling av algoritmisk tenking, ikkje berre gjennom Scratch, men gjennom programmering generelt. Figur 2 viser ei oversikt over dimensjonane i rammeverket med tilhøyrande nøkkelord.



Figur 2 - Rammeverk for algoritmisk tenking

² <https://scratch.mit.edu>

Computational thinking concepts

Den første dimensjonen i rammeverket er «concepts». Dette handlar om dei ulike omgrepa ein møter når ein arbeider med programmering. Døme på denne type omgrep, eller konsept, kan vera løkkjer, hendingar, vilkår, operatorar og data (Brennan & Resnick, 2012, s. 3). Desse omgrepa er grunnleggjande for effektiv og god programmering, og for å forstå korleis programma er laga. Til dømes bruka ein løkkjer rundt små delar av koden for å repetera han, i staden for å skriva ut alle repetisjonane i koden. Eit anna døme på viktige konsept i programmering, er å ta avgjerder basert på vilkår (*visss ... så ...*) (Brennan & Resnick, 2012, s. 5).

Computational thinking practices

Neste nivå eller dimensjon i rammeverket er «practices» (Brennan & Resnick, 2012, s. 6). Denne dimensjonen omhandlar dei praksisane som vart nytta i design og konstruksjonsarbeidet. Forfattarane trekkjer fram at dimensjonen fokuserer på «...the process of thinking and learning, moving beyond what you are learning to how you are learning» (s. 7). Det er med andre ord snakk om metalæring og det å lære å lære, som er ein viktig del av det som inngår i 21st century skills (Trilling & Fadel, 2009, s. 49). Under praksisar trekk forfattarane fram inkrementasjon og iterasjon, testing og feilsøking, gjenbruk og miksing samt abstrahering og modularisering som døme (Brennan & Resnick, 2012, s. 7). Inkrementasjon og iterasjon handlar om programmering som ein ikkje-lineær prosess, men ei veksling mellom m.a. planlegging, utvikling og utprøving - inkludert testing og feilsøking undervegs i prosessen (Brennan & Resnick, 2012, s. 7). Ein annan sentral praksis er den som omhandlar abstraksjon og modularisering. Dette vert definert av Brennan og Resnick (2012, s. 9) som å byggja noko stort ved å setje saman mindre delar.

Computational thinking perspectives

Den siste dimensjonen i rammeverket er «perspectives», som omhandlar eit skifte i perspektiv hjå programmerarane (Brennan & Resnick, 2012, s. 10). Som døme vert uttrykking (*expression*), samankopling (*connecting*) og spørjing (*questioning*) trekt fram (s. 10-11). Under utviklinga av rammeverket såg forfattarane at dei som programmerte såg på programmering som eit verktøy for å skapa, og å uttrykkja seg sjølv (s. 10). Eit anna perspektivdøme er det som går på spørjing og det å tenkja som ein programmerar. Her vert både det å stille spørsmål gjennom programmering for å forstå verda, og det å

undra seg over korleis «alt» er programmert trekt fram (Brennan & Resnick, 2012, s. 11).

Gjennom å nytta dette rammeverket, vil ein få eit innblikk i kva *nivå* den algoritmiske tenkinga som vert nytta eller utvikla ligg på. Medan *concepts* omhandlar læring av grunnleggjande ferdigheiter innan programmering generelt, er *practices* og *perspectives* dimensjonar på eit høgare kognitivt nivå. Dermed kan det tenkjast at dimensjonane *practices* og *perspectives* er moglege å knyta til det å løysa problem, og det å bruka programmering for å læra og forstå andre fag som t.d. matematikk. Med dette vil rammeverket vera nyttig, når forskingsspørsmålet om *korleis lærarar legg til rette for å løysa problem ved hjelp av algoritmisk tenking* skal svarast på.

2.3 Å løyse problem

Å løyse problem er sentralt i matematikkfaget og i forskingsspørsmålet til oppgåva som er nemnt over. Difor vil det vera naudsynt å sjå på kva som ligg i omgrepa problem og problemløysing i matematikken.

2.3.1 Kva er eit problem og kva er problemløysing?

Niss og Høygård Jensen (2002, s. 45) nemner «problembehandlingskompetence» som ein av åtte delkompetansar i det dei definerer som matematisk kompetanse. Omgrepet *problemløysing* har historisk hatt mange ulike definisjonar og tolkingar i matematikdidaktikken (Schoenfeld, 2016, s. 4). Tolkingane av omgrepet går frå at problemløysing er når ein løyser oppgåver i matematikken, til at problemløysing er ein ferdighet eller kompetanse (Schoenfeld, 2016, s. 4). Når ein tolkar problemløysing som berre det å rekna oppgåver, ser ein gjerne på *problem* som alle slags matematiske oppgåver, inkludert rutineoppgåver og innlæringsoppgåver. Dette skil seg frå Niss og Høygård Jensen (2002, s. 49) som presiserer at problemomgrepet er relativt:

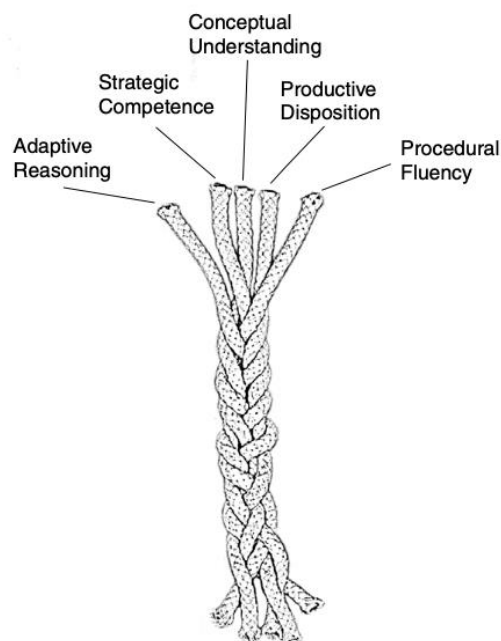
Et (formuleret) matematisk problem er et særlig type matematisk spørgsmål, nemlig ét hvor en matematisk undersøgelse er nødvendig for besvarelsen. Sådan set kunne også spørgsmål, som kan besvares alene ved hjælp af (få) specifikke rutineoperationer, falde ind under begrebet “problem”. Sådanne spørgsmål, som for den der skal løse dem, kan besvares ved aktivering af rutinefærdigheder,

henregner vi imidlertid ikke under matematiske problemer i denne forbindelse.
(Niss & Højpgård Jensen, 2002, s. 49)

Kva som er eit matematisk problem er med andre ord avhengig av nivået til den som løyser det, sidan kva som vert oppfatta som rutineferdigheiter vil variera etter nivå. Definisjonen over legg til grunn at det skal inngå ei form for undersøking i løysingsprosessen.

I ein større rapport om matematisk kompetanse og matematikklæring, vert problemløysing løfta fram som noko heilt sentralt (Kilpatrick et al., 2001, s. 421). Dei trekkjer fram matematisk kompetanse (*proficiency*) som eit tau som er sett saman av fem trådar (*strands*), som vist i Figur 3 (Kilpatrick et al., 2001, s. 5). Mot slutten av rapporten trekkjer dei fram viktigheita av problemløysing i matematikklæringa:

Problem solving should be the site in which all of the strands of mathematics proficiency converge. It should provide opportunities for students to weave together the strands of proficiency and for teachers to assess students' performance on all of the strands. (Kilpatrick et al., 2001, s. 421)

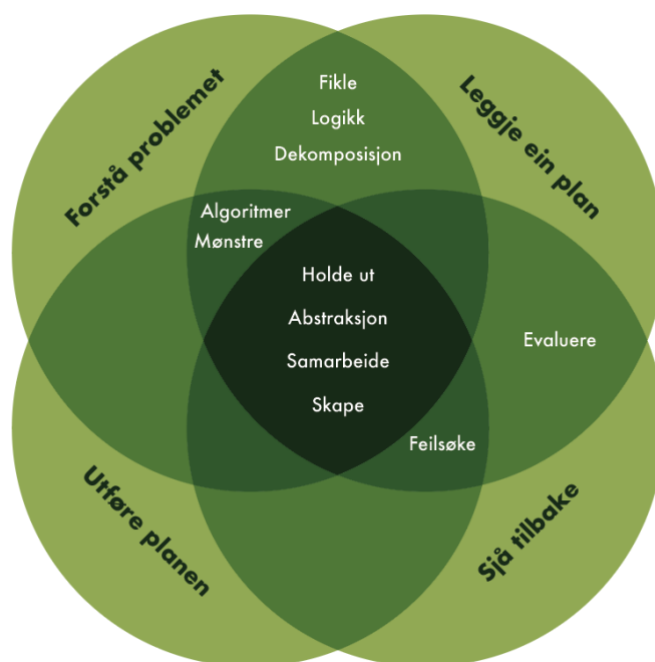


Figur 3 - Mathematical proficiency (Kilpatrick et al., 2001)

2.3.2 Strategi for problemløysing

Algoritmisk tenking vert gjerne trekt fram som ein problemløysingsstrategi (Csizmadia et al., 2015, s. 5; Wing, 2017, s. 8). Ein av dei første som omtala (matematisk) problemløysing var George Polya. I boka *How To Solve It* frå 1945 presenterer han ein problemløysingsstrategi med fire fasar for å løysa matematiske problem (Polya, 1973, s. 5). Dei fire fasane er *å forstå problemet*, *å leggja ein plan*, *å utføra (planen)*, og *å sjå attende* (Polya, 1973, s. 5-6).

Å forstå problemet handlar om å forstå kva som er målet med problemet, kva det er ein skal finna ut av, og korleis kan ein gå fram. I skildringa av denne fasen brukar Polya ord og omgrep som er tett knytt opp til algoritmisk tenking: «The student should consider the principal parts of the problem attentively, repeatedly, and from various sides» (Polya, 1973, s. 7). Her kan ein sjå parallellar til både fikle og dekomponere som me kjenner att frå Figur 1. Når ein vidare skal **leggja ein plan** handlar det om å få ei oversikt over kva utrekningar og steg ein må gjennom for å svara på oppgåva (Polya, 1973, s. 8). I forklaringa av planleggingsfasen vert det trekt fram at prosessen kan vera langvarig, ein kan gjera mange *feil*, og at det er viktig å *halde ut* (Polya, 1973, s. 8). Eit viktig spørsmål ein kan stilla seg i denne fasen er «*Do you know a related problem?*» (Polya, 1973, s. 9, kursiv i originalen). Her handlar det om å sjå etter fellestrekk med tidlegare problem, og her vil *abstraksjon* vera eit heilt sentralt hjelpemiddel. Om ein fjernar detaljar som ikkje er naudsynte kan det tenkjast at det er lettare å sjå koplingar til tidlegare løyste problem. Vidare er den tredje fasen å **utføra planen**, der ein gjerne bør kontrollere dei ulike stega ein utfører (Polya, 1973, s. 12). Den siste fasen i problemløysinga handlar om å **sjå attende** og her kan ein igjen sjå att omgrep frå algoritmisk tenking. Polya (1973, s. 15) trekkjer fram *evaluering* som sentralt, både når det kjem til *feilsøking* og det å gjere *vurderingar* av løysinga. Det er med andre ord mange omgrep frå dei fire fasane av problemløysing som har tette band til definisjonen av algoritmisk tenking. I Venn-diagrammet vist i Figur 4, har eg freista å visualisera samanhengen mellom dei ulike nøkkelomgrepa og arbeidsmetodane frå definisjonen av algoritmisk tenking, som vert nytta i Fagfornyninga (Figur 1), og dei fire elementa i problemløysingsstrategien til Polya.



Figur 4 - Polya og algoritmisk tenking

2.3.3 Problemløysing i matematikken og algoritmisk tenking

Som nemnt over vert gjerne algoritmisk tenking omtala som ein strategi for å løysa problem. I *Teknologi og programmering for alle* vert det trekt fram at «Algoritmisk problemløysing handler om å omforme et generelt problem til en form som kan løses ved hjelp av programmering. Prosessen fra idé til produkt er en kreativ problemløysingsprosess» (Sanne et al., 2016, s. 19). Gjennom å nytta algoritmisk tenking er det, med andre ord, eit mål å gje elevane ein måte å takla problem, både matematiske og generelle problem, gjerne ved hjelp av programmering. Weintrop et al. (2016, s. 139) trekkjer fram at det ikkje er slik at elevane må verta veldig gode i programmering for at ein skal ha nytte av det:

While it is not reasonable to expect all students to be programming experts, basic programming proficiency is an important component of twenty-first century scientific inquiry. Students who have mastered this practice will be able to understand, modify, and create computer programs and use these skills to advance their own scientific and mathematical pursuits. Weintrop et al. (2016, s. 139)

Sfard og Leron (1996, s. 190) peikar på eit interessant aspekt knytt til skilnadane mellom det å løysa problem i matematikklasserommet kontra på «the computer lab».

Her kjem det fram at det å løysa problem i matematikk er noko ein gjer *for* læraren mot ros eller karakter og vurdering, medan problemløysing gjennom programmering gjev fridom til å utforska og gjera feil (s. 193).

2.3.4 Klasseromskultur for problemløysing

Korleis ein *legg opp til å løysa problem* er sentralt i oppgåva. Det same er det sosiokulturelle læringssynet der ein lærer i fellesskap. Difor er det naturleg å sjå på kva teorien seier om kjenneteikn for undervising som legg til rette for problemløysing.

Pennant (2013) presenterer åtte ulike kjenneteikn på ein klasseromskultur som er med på å leggja til rette for problemløysing. Kjenneteikna er presentert i Tabell 2.

Utgangspunktet for kjenneteikna er yngre elevar i tidleg del av grunnskulen (Pennant, 2013). Likevel tenkjer eg at dei også vil vera relevante for høgare trinn i skulen. Sjølv om det ut frå eiga erfaring gjerne vert nytta litt mindre heilklasseundervising i vidaregåande skule, er aspekta relevante for det å leggja til rette for å løysa problem. Aspekta handlar om å gje elevane støtte til å utvikla problemløysingsferdigheiter:

This means that your students require thinking and playing-with-the-problem time. They need to test out ideas, to make conjectures, to go up ‘dead ends’ and adjust their thinking in the light of what they learn from this, discuss ideas with others and be comfortable to take risks. (Pennant, 2013)

Her kan ein dra parallellar til omgrep som *fikle* (playing-with-the-problem), *feilsøke* (test out ideas) og *mønstre* (make conjectures) for å nemna nokre. Det er med andre ord tette band mellom klasseromskulturen Pennant (2013) løftar fram og algoritmisk tenking.

Tabell 2 - *Developing a Classroom culture that supports a Problem-solving Approach to Mathematics* (Pennant, 2013)

Aspects to consider
1) Who does most of the talking in whole-class parts of the lesson?
2) What questions do I ask?
3) Who answers the questions?

4) How well do I listen to the students' answers and seek to understand what they are saying?
5) What do I do with the students' answers?
6) How do I facilitate the learning?
7) How confident are the students to take a risk, to try out ideas, to make mistakes?
8) What does my body language communicate?

2.4 Utvikling av kompetanse

Læreren sine refleksjonar kring eiga utvikling av kompetanse står sentralt i denne oppgåva. Forskingsspørsmål 1 tek føre seg dette: *Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?* Tidlegare forskning har vist læreren sin kompetanse som sentral når algoritmisk tenking skal inn matematikkfaget. I det sosiokulturelle læringssynet er lærarrolla tett kopla til rolla som ein meir kapabel læringspartner, som nemnt i kap. 2.1. Dette gjer det hensiktsmessig å sjå nærare på nokre teoretiske aspekt knytt til utvikling hjå lærarar.

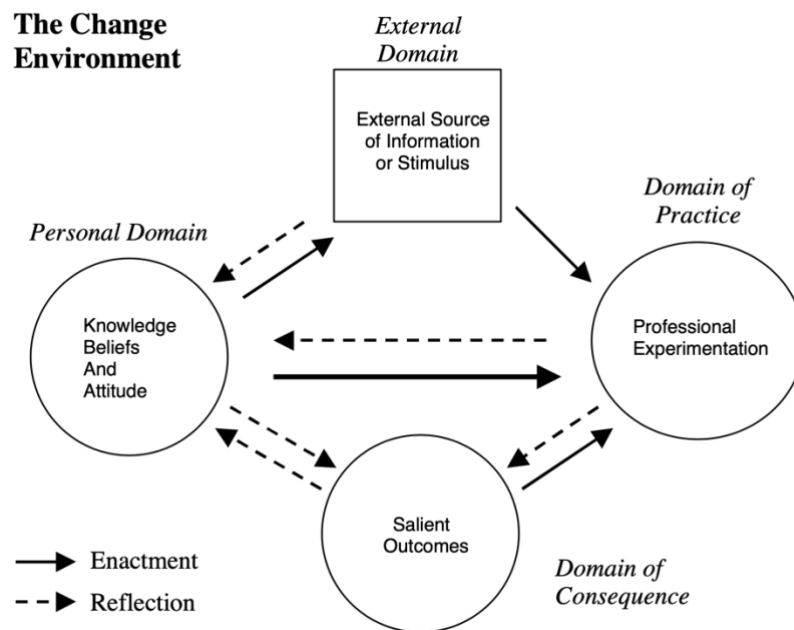
2.4.1 Lærarprofesjonen og profesjonell utvikling

Læreryrket vert ofte omtala som ein profesjon. Profesjonsomgrepet er gjerne knytt meir opp mot spesialisering og teori enn «yrke». Tolo (2017, s. 137) trekkjer fram at det er det venta at lærarar som gruppe utviklar undervisinga si i takt med tida, og at ein støttar kvarandre som kollektiv, både gjennom utvikling og deling av kunnskap. Vidare vert det presisert at «En av de viktigste funksjonene til en profesjon er å forvalte kunnskap som profesjonsutøverne skal bygge sin praksis på» (Tolo, 2017, s. 137). Den profesjonelle læreren har, med andre ord, individuell kunnskap, samt tilgang til kollektiv kunnskap gjennom andre i profesjonen. Fokuset på profesjonsfellesskapet finn me òg at i kapittel 3.5 i den overordna delen av det nye læreplanverket:

Lærerprofesjonen bygger sin profesjonsutøvelse på felles verdier og et felles forsknings- og erfaringsbasert kunnskapsgrunnlag. Profesjonen og den enkelte lærer forvalter et ansvar for å utøve skjønn i komplekse spørsmål. Lærere og ledere utvikler faglig, pedagogisk, didaktisk og fagdidaktisk dømmekraft i

dialog og samhandling med kolleger. Utøvelse og utvikling av det profesjonelle skjønnet skjer både individuelt og sammen med andre. Faglig dømmekraft forutsetter også jevnlig oppdatering. (Utdanningsdirektoratet, 2017, s. 17)

Clarke og Hollingsworth (2002, s. 947) har utvikla ein modell som omhandlar profesjonell utvikling av lærarar, hjå dei omtala som «teacher professional growth». Her vert omgrepet definert som «...professional growth is represented as an inevitable and continuing process of learning» (Clarke & Hollingsworth, 2002, s. 947). I denne *endringsmiljømodellen* vert den stadige utviklinga og læringa ein som lærar går gjennom ikkje berre peika på, men omtala som uunngåeleg. Profesjonell utvikling hjå lærarar handlar, i følgje modellen, om moglege koplingar mellom fire ulike domene i lærarverda (Clarke & Hollingsworth, 2002, s. 950). Modellen er visualisert i Figur 5.



Figur 5 - Endringsmiljømodellen (Clarke & Hollingsworth, 2002, s. 951).

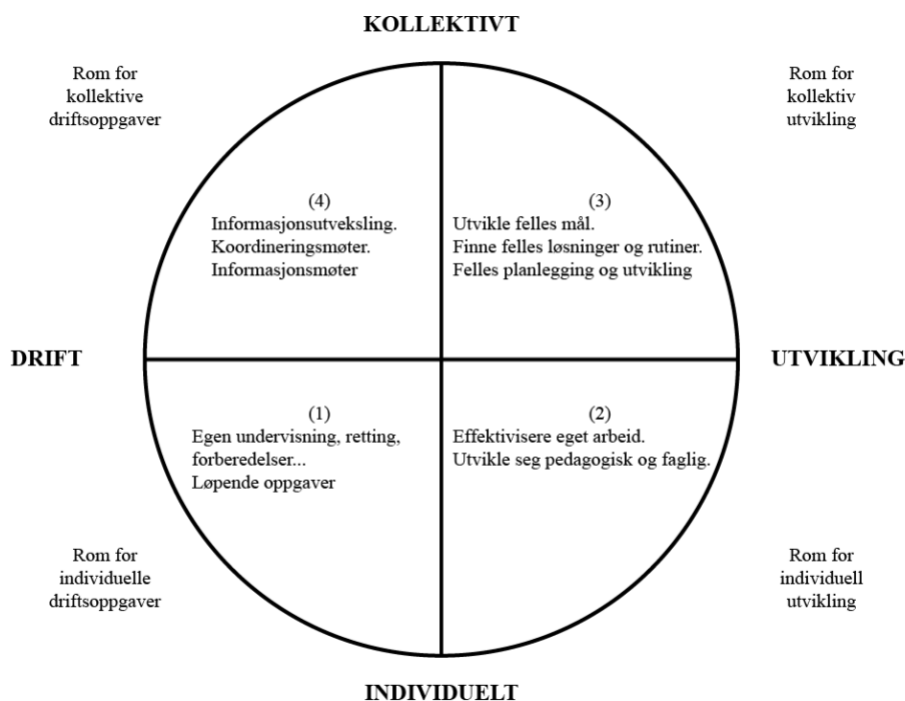
Dei fire domena er knytt til ulike sider av profesjonsutøvinga. Det personlege domenet (*Personal Domain*) handlar mellom anna om læraren sine haldningar, kompetanse og pedagogiske credo eller overtyding (fritt omsett frå *beliefs*). I praksisdomenet (*Domain of Practice*) finn me profesjonell eksperimentering, gjerne i klasserommet.

Konsekvensdomenet (*Domain of Consequence*) er knytt til utbytte av den nye praksisen. Det siste domenet er det eksterne domenet (*External Domain*), knytt til informasjon og påverknad frå eksterne kjelder. I denne oppgåva er det naturleg å tenkja at endring i

læreplanen (til å inkludera programmering og algoritmisk tenking) vil falla inn under dette domenet, saman med kurs, etterutdanning og annan ytre påverknad. Forfattarane trekkjer fram at endring i eit av desse domena kan føra til endring i dei andre domena (Clarke & Hollingsworth, 2002, s. 950). Dette kan skje gjennom «enactment», gjerne omsett til (endra) handling, og refleksjon (s. 951). Sjølv om *enactment* og *handling* ikkje inneheld den heilt same meininga vil eg nytta *handling* som norsk omsetting i mangel av betre ord.

2.4.2 Kompetanseutvikling og drift

Modellen over fokuserer på den einkilde lærar si profesjonelle utvikling. Sidan samarbeid og fellesskapet står sterkt, både i den sosiokulturelle læringstradisjonen og i algoritmisk tenking, vil det vera hensiktsmessig å ta med ein modell for utvikling som fokuserer på dette perspektivet. Utviklingshjulet til Irgens (2010, s. 136) presenterer ein forenkla modell av ein «...skole i bevegelse». Hjulet, som er illustrert i Figur 6, ser på ulike rom som ein finn i spenninga mellom drift og utvikling, og den individuelle og den kollektive utøvinga av profesjonen (Irgens, 2010, s. 128).



Figur 6 - Et «utviklingshjul» for en skole i bevegelse (Irgens, 2010, s. 136)

Dei ulike romma i figuren over inneheld oppgåver og arbeid etter korleis dei ligg mellom skjæringa av dei to aksane. Rom 1 inneheld dei individuelle driftsoppgåvene,

dei daglege oppgåvene ein har som lærar. I den andre rommet finn ein det som kan knytast til individuell utvikling, som til dømes fagleg eller pedagogisk oppdatering, og effektivisering av eigne oppgåver. I den kollektive delen av hjulet finn ein rom 3, knytt til kollektiv utvikling, og rom 4 knytt til kollektiv drift. Dette hjulet vil vera eit interessant teoretisk bakteppe for analysen av datamaterialet, spesielt når eg skal sjå nærare på korleis dei ulike informantane samarbeider med kollegaar i møte med den nye læreplanen. Irgens (2010, s. 137) trekkjer fram at det er sentralt at skulen må vera i alle dei fire romma for at skulen skal vera i bevegelse. Her er det naturleg å forstå «bevegelse» som utvikling (framover). Vidare trekk forfattaren inn viktigheita av samarbeid og det kollektive:

Gode enkeltlærere er en nødvendig, men ikke tilstrekkelig forutsetning for å skape en god skole. Gode skoler krever derfor lærere som har kompetanse i alle de fire rommene, og skoleledere som evner å tilrettelegge for og følge opp lærernes arbeid også utenom deres løpende undervisningsorienterte arbeid i det individuelle rommet. (Irgens, 2010, s. 140)

3 Metode

I dette kapitlet vil eg presentera dei metodiske vala eg har gjort i oppgåva. Eg vil først gjera greie for det vitenskapsteoretiske perspektivet som ligg til grunn for oppgåva, før eg presenterer forskingsdesignet eg har nytta. Deretter følgjer ein del som omhandlar datainnsamlinga mi, med grunngjeving og presentasjon av metodar, utval osv. Til slutt vil eg gjera greie for kvaliteten til oppgåva og etiske omsyn.

3.1 Vitenskapsteoretisk perspektiv

I humanvitenskapar, som utdanningsvitenskapen fell inn under, handlar det mykje om å gje mening til ulike typar tekst, gjennom fortolking. Hermeneutikken kan omtalast som «læren om fortolkingen av tekster» (Kvale & Brinkmann, 2015, s. 73). I denne studien vil eg nytta eit utvida tekstomgrep, der tekst er noko som har mening. På den måten kan både transkriberte intervju og innsamla undervisningsmateriale falla inn under tekstomgrepet.

Som menneske er me påverka av kulturen vår, livssituasjonen, språket, opplevingar m.m. Dette gjer at me ikkje er nøytrale når me skal fortolka, men at me har det, som i hermeneutikken, vert kalla ei førforståing (Zimmermann, 2015, s. 37). Desse føresetnadane me har, førforståinga vår, vert gjerne kalla forståingshorisonten vår (Kvarv, 2014, s. 73). Me har gjerne ei oppfatning om kva som er rett og galt eller bra og dårleg i ulike situasjonar, og me har fordommar som påverkar oss i tolkinga vår. Som Kvarv (2014, s. 82) trekkjer fram så talar teksten «...til leseren ut fra sin horisont, fordømmer og målestokker». Gadamer (2004, s. 305) meiner forståing handlar om «the fusion of horizons». For å forstå ein tekst må det altså skje ei samansmelting av horisontar. Når eg skal forstå ein tekst, må eg, som fortolkande menneske, prøva å frigjera meg frå mi førforståing og min forståingshorisont, og prøva å setja meg inn i teksten sin forståingshorisont.

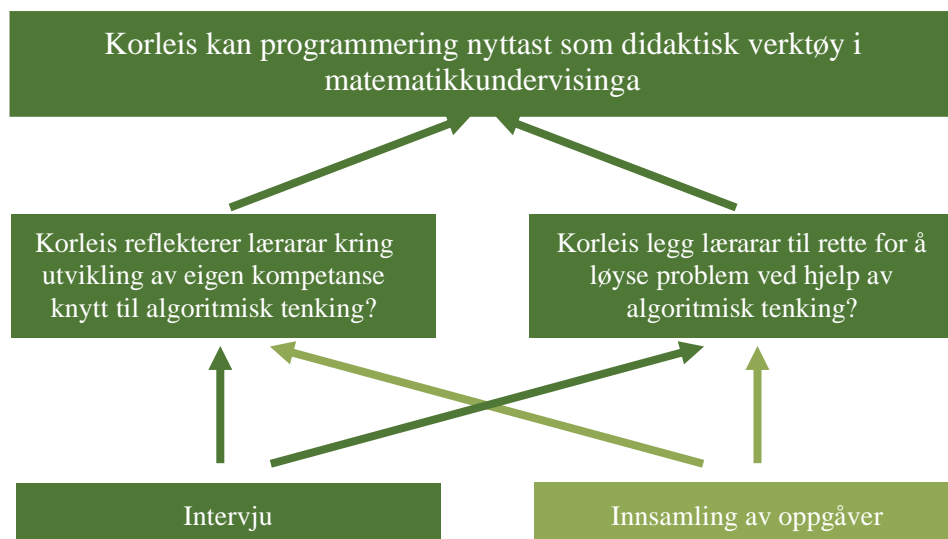
Til dagleg er eg matematikklærer og ikkje forskar. Dermed er det viktig for meg å prøva å frigjera meg frå lærerarrolla, for å tre inn i ei rolle der eg ser på matematikklærarar frå utsida. Likevel er det naturleg å tenkja at det vil vera umogleg for meg å frigjera meg heilt frå dei fordommane og den førforståinga eg har som lærar. Dette kan, utan at eg er

klar over det sjølv, vera med på å farga studien og funna mine. Samstundes vil det vera ein styrke at eg har kjennskap til, og forståing for, livsverda til informantane. Det gjer at eg skjønar omgrepa og situasjonane dei omtalar frå ein matematikklærer-kvardag.

3.2 Forskingsdesign

Forskningsdesignet på masterstudien min er ein kvalitativ kasusstudie. Skogen (2018, s. 82) trekkjer fram at kasusstudiar er føremålstenlege når ein skal arbeida med spørsmål som omhandlar «korleis» eller «kvifor». Tanken er å løfta fram ulike lærarar sine praksisar og refleksjonar, når det kjem til bruk av programmering og algoritmisk tenking som ein del av matematikkundervisinga deira. Dette opnar for at «...andre får del i denne praksisen som et tankeredskap for utvikling av praksis i sine egne klasserom» (Postholm, 2010, s. 52).

Kasusstudien er designa på ein slik måte at dei ulike informantane mine vert sine egne kasus, i sine kontekstar. Dette er det Yin (2018, s. 53) kallar ein *multiple-case study*. I oppgåva vil eg bruka det norske omgrepet kollektiv kasusstudie som vert nytta av Postholm (2010, s. 52). I analysen vil eg sjå på alle kasusa samstundes. Det vil seie at eg ikkje gjer egne heilt separate analyser for dei ulike lærarane. Dette kjem inn under det Yin (2018, s. 51) peikar på som ei *holistic* tilnærming. Forskningsdesignet som er nytta er altså eit holistisk kollektivt kasusstudie.



Figur 7 - Problemstilling, forskingsspørsmål og metodar

For å svara på forskingsspørsmåla og problemstillinga har eg valt å nytta intervju som den sentrale datakjelda. I tillegg har eg nytta meg av innsamling av

oppgåver/undervisningsmateriell som støttande datakjelde. Figur 7 viser samanhengen mellom metodane, forskingsspørsmåla og problemstillinga i oppgåva.

3.3 Utval

Gjennom denne oppgåva er målet å sjå på *korleis programmering kan nyttast som didaktisk verktøy i matematikkundervisinga*. Eg er interessert i den vidaregåande skulen, og matematikkfaget 1T no når det har fått ny læreplan. Utvalet av informantar i studien min er eit *strategisk utval*. I strategisk utval ligg det at eg har valt informantar som eg meiner er i stand til å kunna uttala seg om temaet på ein reflektert måte (Tjora, 2021, s. 145). Det har vore eit krav at informantane mine skal ha matematisk undervisningskompetanse. Dei er med andre ord ein del av lærararprofesjonen med formell kompetanse, både fagleg og pedagogisk, for undervising i matematikk 1T. Vidare har eg hatt som krav at informantane skal undervisa i 1T i inneverande skuleår (20/21), då dette er heilt naudsynt sidan det først er i år den nye læreplanen er sett i verk.

Eg har gjeve informantane som har vore med *fiktive namn*. Som det kjem fram av oversikta i Tabell 3, er utvalet sett saman av lærarar med ulik erfaring frå skulen, frå heilt nyutdanna til godt erfarne lærarar. Utvalet er sett saman av lærarar frå skular med ulik storleik, og dermed ulik storleik på kollegia lærarane er ein del av. Lærarane kjem frå skular i både by og landsbygd. Felles for alle skulane er at det er ein-til-ein dekning av datamaskinar, dvs. alle elevane har kvar sin datamaskin. I studien er det eit poeng å sjå på korleis ulike lærarar, med ulike utgangspunkt, handterer endringa i læreplanverket. Samstundes har dei ulike kasusa (lærarane) så mange likskapar at dei er relevant for det same temaet (Tjora, 2021, s. 148).

Tabell 3 - Oversikt over informantane

Informant	Erfaring i skulen (ca.)	Skule
Maria	15 år	Stor skule, stor by
Lars	4 år	Middels skule, stor by
Helge	16 år	Stor skule, stor by
Torrey	1 år	Middels skule, liten by
Tove	6 år	Liten skule, bygd

3.4 Intervju

Hovudinnsamlinga av data i denne studien er gjennom kvalitative forskingsintervju. Intervju som verktøy for datainnsamling legg til rette for ein type munnleg sjølvrapportering, og er godt eigna for datainnsamling på individnivå (Befring, 2015, s. 74). I denne studien har eg nytta ei semistrukturert intervjuform. Det vil seie at det vert stilt strukturerte spørsmål, medan svara og rekkefølga er av ein open karakter (Befring, 2015, s. 75). I tillegg gjev det semistrukturerte intervjuet rom for digresjonar (frå informanten) og med det nye tema eller poeng som ikkje var planlagt av intervjuar på førehand (Tjora, 2021, s. 128). I denne oppgåva er målet å sjå nærare på eit utval lærarar sine erfaringar og refleksjonar. Tjora (2021, s. 128) trekkjer fram det semistrukturerte intervjuet som eit godt verktøy for å «...studere meninger, holdninger og erfaringer».

3.4.1 Intervjuguide

I forkant av intervjuet utforma eg ein intervjuguide (sjå vedlegg 1). Intervjuguiden vert av Kvale og Brinkmann (2015, s. 162) trekt fram som «...et manuskript som strukturerer intervjuforløpet mer eller mindre stramt». Då eg skulle utforma intervjuguiden tok eg utgangspunkt i teoretiske aspekt, som kunne vera nyttige i den seinare analysen. Mellom anna er ein del av spørsmåla og temaa direkte knytt til aspekta til Pennant (2013) og til omgrepet algoritmisk tenking med underdimensjonar presentert i teorikapittelet (Figur 1 i kap. 2.2.3). Det at eg har ein tanke om kva eg spør om, og kvifor, kan vera ein føremon med tanke på analysen. Kvale og Brinkmann (2015, s. 165) peikar på at dette gjev meg, som forskar, høve til å avklara meiningar i det informantane seier, undervegs i intervjuet.

Strukturen i intervjuguiden er tredelt, med oppvarming, refleksjon og avrunding, som presentert av Tjora (2021, s. 159). I oppvarmingsdelen har eg utforma spørsmål om utdanning, undervisningsfag og erfaring. Spørsmåla i denne delen er av ein type som krev låg grad av refleksjon, og er med på å skapa ein tryggleik hjå informanten i intervjusituasjonen (Tjora, 2021, s. 160). Informanten får praktisk informasjon om lydopptak og anonymisering av personvernopplysingar. Dei vert òg minna om teieplikta dei har som lærarar, slik at dei ikkje bryt ho i intervjusituasjonen ovanfor meg som ekstern intervjuar.

Refleksjonsdelen er todelt, ein del om kompetanse og kompetanseutvikling, og ein del om undervisningspraksis. I refleksjonsdelen har eg skrive ut ein del hovudspørsmål, med tilhøyrande stikkord og småsetningar. Ein del av spørsmåla til dei ulike temaa er introduksjonsspørsmål. Dette er spørsmål som «Kan du seie noko om...» og desse har til hensikt å få fram det informantane sjølv finn som det mest sentrale ved det aktuelle temaet (Kvale & Brinkmann, 2015, s. 166). Vidare er dei tilhøyrande stikkorda ei hjelp til å utforma aktuelle og relevante oppfølgingsspørsmål, knytt til det informanten har snakka om. Om informantane kjem innom tema utanfor intervjuguiden, eller ikkje svarar utdjupande nok, vil eg nytta inngåande spørsmål. Det er spørsmål av typen «Kan du seie litt meir om det?» eller det å be om fleire døme (Kvale & Brinkmann, 2015, s. 166). For å få til gode overgangar mellom dei ulike temaa i intervjuet, har eg lagt opp til å nytta strukturerande spørsmål. Dette gjer eg for å tydeleggjera, og informera om, desse skifta for informanten. Desse er ikkje skriven ut i intervjuguiden. Strukturerande spørsmål kan òg nyttast for å avbryta informanten om svara vert for lange, eller mindre relevante (Kvale & Brinkmann, 2015, s. 167).

I den avsluttande delen av intervjuet legg eg, i intervjuguiden, opp til å la informanten få koma med ytterlegare refleksjonar eller moment, som informanten sjølv meiner burde vore med. Som Tjora (2021, s. 160) trekkjer fram, har avslutninga som mål å normalisera situasjonen mellom intervjuar og intervjuobjekt att, etter intervjuet, og ta ned refleksjonsnivået. Eg vil nytta avslutninga til å takke for deltaking, informere om kontaktinfo ved spørsmål og liknande. Samstundes vil eg fortelja om den vidare gongen i forskingsprosessen, samt handsaminga av lydopptak og personopplysingar.

3.4.2 Pilotintervju

Etter at intervjuguiden var gjort klar, gjennomførte eg eit pilotintervju. Dette gjorde eg for å sjå om spørsmåla i intervjuguiden var forståelege og eigna til å gje gode og utdjupande svar. Som pilotinformant nytta eg ein lærar som tilfredsstilte dei same krava som utvalet for studien. Pilotinformanten hadde erfaring frå undervising i IT, og underviser i faget i inneverande skuleår. Kvale og Brinkmann (2015, s. 194) trekkjer fram ulike kvalitetskriterium for eit intervju, som mellom anna «spontane, innholdsrike, spesifikke og relevante svar», og kortast mogleg spørsmål kombinert med lengst mogleg svar. Erfaringane frå pilotintervjuet var at spørsmåla eigna seg for å oppfylle desse kriteria. Pilotinformanten gav gode refleksjonar rundt eigen kompetanse og eigen

praksis. For min del, som intervjuar, vart eg merksam på at eg må opplysa om at spørsmåla knytt til økter der det vert arbeidd med algoritmisk tenking, primært omhandlar heilklasesituasjonen. Om ikkje informantane arbeider mykje med heilklasseundervising vil det vera naturleg å snakka om det som skjer i mindre grupper, eller på individnivå. Ei anna interessant oppdaging i samband med pilotintervjuet var knytt til omgrepet algoritmisk tenking. Pilotinformanten gav uttrykk for at det ikkje var gjennomført så mange undervisingsøker knytt til algoritmisk tenking. Eg bad han då om å inkludera programmeringsøker. Då fekk eg skildringar av tenkje- og arbeidsmåtar som er tett knytt opp til måten eg i denne oppgåva nyttar omgrepet algoritmisk tenking. Dette vil vera viktig å ha i bakhovudet under dei reelle intervju. Då vil eg freista å få fram kva informantane tenkjer algoritmisk tenking er, dersom det ikkje kjem tydeleg fram av intervjuet. Då kan eg vidare gje nokre stikkord knytt til denne oppgåva si forståing av algoritmisk tenking. Pilotintervjuet var òg nyttig trening i å stilla oppfølgingsspørsmål og i å følgja og nytta intervjuguiden i intervjusituasjonen.

3.4.3 Gjennomføring av intervju

Grunna koronasituasjonen med eit vekslende smittepress og «trafikklysnivå»³ i vidaregåande skular, har det vore litt avgrensingar i kvar det har vore mogleg å gjennomføra fysiske intervju. Difor er nokre av intervju gjennomført som videomøte gjennom Teams i dei tilfella smittepress og geografiske tilhøve har gjort fysiske intervju umoglege å gjennomføra. Intervju vart gjennomført i februar og mars 2021.

Kvaliteten av djupneintervju er, i følgje Tjora (2021, s. 130), avhengig av tillit mellom intervjuar og informant. Gjennom presentasjon av studien og den innleiande delen av intervjuet, freista eg å oppretta denne tilliten, og gjera intervjuobjektet trygg i intervjusituasjonen. Både i dei fysiske (2) og dei digitale (3) intervju la eg opp til ein avslappa stemning, med færrast mogleg avbrytingar frå meg. På denne måten kunne informantane oppleve at dei trygt kunne dela personlege synspunkt og erfaringar, noko som er ein føresetnad for gode djupneintervju (Tjora, 2021, s. 132). I den fortolkande forskingstradisjonen er det eit poeng at intervjuet er som ein samtale mellom «likemenn», der det òg er opning for å ta opp tema eller spørsmål som informanten

³ <https://www.helsedirektoratet.no/veiledere/smittevern-for-skoletrinn-1-7-covid-19/smitteforebyggende-tiltak/nivainndeling-av-smitteverntiltak-trafikklysmode>ll

nemnar (Postholm, 2010, s. 75). Mellom anna sidan intervjuar hadde ein avslappa stemning, opplevde eg at informantane uttalte seg fritt og at dei kom med spontane skildringar og refleksjonar. Tjora (2021, s. 135) trekkjer fram at ein gjerne bør intervjuar informantane i ei setting der dei kjenner seg trygge t.d. heime eller på arbeidsplassen. Dei fysiske intervjuar eg har gjennomført, har vore på informantane sine arbeidsplassar. I dei digitale intervjuar er det større fleksibilitet til kvar informant er, og eg har opplevd at informantane både har vore heime og på arbeidsstaden sin.

Hadde denne studien vore gjennomført før koronapandemien, ville det vore nærliggjande å tenkja at informantane ikkje kjende seg trygge i videomøte-konteksten. Sidan lærarane har hatt ein god del digital undervising og møteverksemd det siste året, opplevde eg ikkje dette som eit problem. Eg valde å gje informantane tilbod om intervjuar anten i Teams eller Zoom etter kva dei kjende seg mest trygge i. Det er nokre forseinkingar i lyden i digitale intervju kontra den augneblinklege kontakten ein har i fysiske intervju, men dette opplevde eg som positivt. Dette av di denne forseinkinga var ei hjelp til å nytta «taushet» som oppfølgingsspørsmål (Kvale & Brinkmann, 2015, s. 171). Det var heller ikkje problem å gjera lydopptak av datamaskin-lyden i dei digitale intervjuar. Stemma vart gjerne litt endra, sett opp mot dei fysiske intervjuar, men kvaliteten på opptaka var meir enn god nok til å kunna brukast vidare i arbeidet. Alle lydopptaka vart gjort med eigen mobiltelefon som lydopptakar. I etterkant av intervjuar transkriberte eg dei, og la dei inn i Nvivo. Transkriberinga kjem eg attende til i kap. 3.6.1.

3.5 Innsamling av oppgåver

I følgje Yin (2018, s. 126) er ein av styrkane til kasusstudiar det høve ein har til å nytta fleire ulike datakjelder i innsamlinga av data. På denne måten får ein enno betre mogleik til å gå i djupna, som er noko av målet med ein kasusstudie (Yin, 2018, s. 126). I denne studien har eg nytta innsamling av oppgåver som informantane har brukt i klasserommet som ein supplerande metode for datainnsamling. Dette har vore med på å gje data som i hovudsak kan nyttast for å svara på det andre forskingsspørsmålet mitt: *Korleis legg lærarar til rette for å løyse problem ved hjelp av algoritmisk tenking?* Oppgåver som vert nytta av lærarane i undervisinga vil eg studera i lys av rammeverket til Brennan og Resnick (2012) for algoritmisk tenking. Dermed vil eg kunna analysera

kva av dei tre nivåa (concepts, practices og perspectives) oppgåvene siktar seg inn på, og dermed om oppgåvene er eigna for å løyse problem. Eg har samla inn oppgåver og undervisningsmaterieil frå fire av dei fem lærarane. Døme på ei av desse oppgåve ligg ved denne masteroppgåva (Vedlegg 4).

3.6 Analyse av data

Etter at dataa var innsamla, starta arbeidet med å gjera dei klare for analyse. Eg vil i det følgjande gjera greie for transkriberinga og analyseprosessen.

3.6.1 Transkribering av intervju

Etter at intervjuet var gjennomført, transkriberte eg dei frå munnleg til skriftleg form. På denne måten får dei ein struktur som gjer at dei enklare kan analyserast (Kvale & Brinkmann, 2015, s. 206). Eg valde å transkribere intervjuet ordrett, og inkludera dei lengste og tydelege «eh»-ane der informantane hadde vanskar med å finna orda. Det medfører at gjentakningar og ufullstendige setningar òg er med. Når eg i kapittel 4 presenterer funna, vil eg i størst mogleg grad skriva informantane sine utsegner på ein meir samanhengande måte. Dette gjer eg, både med omsyn på informantane og lesarane av denne oppgåva (Kvale & Brinkmann, 2015, s. 213). For å gje eit inntrykk av måten informantane har sagt ting på, har eg vald å markera latter i dei tilfella der dette har skjedd. Eg har vald å normalisera transkripsjonane, og har nytta nynorsk som målform (Tjora, 2021, s. 186). Dette har eg gjort for å gjera det enklare for meg sjølv i analysearbeidet. Intervjutranskripsjonar på eit normert skriftspråk vil gjera det mogleg for meg å søka etter ulike nøkkelord og frasar, i dei ulike tekstane.

Arbeidet med å transkribere intervjuet var tidkrevjande, men samstundes nyttig. Ved at eg sjølv transkriberte intervjuet fekk eg friska opp i intervjusituasjonen, og i tillegg gått gjennom intervjuet fleire gongar. Dette gjorde at eg fekk høve til å notera ned spontane tankar og refleksjonar som ein del av analyseprosessen.

I etterkant av datainnsamlinga såg eg at eg ikkje fekk lærarane til å reflektera like mykje rundt kompetansen sin som tenkt. Refleksjonane vart fort retta mot eigen praksis, eller elevane sin kompetanse. Fleire av lærarane fortalde ved ulike høve om den generelle matematikkundervisninga deira, ikkje berre når det var algoritmisk tenking og

programmering på planen. Likevel sit eg att med mange gode refleksjonar, som eg tenkjer kan vera med på å svara på spørsmåla i oppgåva.

3.6.2 Koding og tolking av datamaterialet

For å få ei oversikt over datamaterialet har eg koda dei ulike datakjeldene. I utviklinga av desse kodane har eg valt å ta utgangspunkt i dei teoretiske perspektiva som ligg til grunn for oppgåva. Som Yin (2018, s. 168) trekkjer fram, er det tette band mellom teorien, forskingsspørsmåla og datainnsamlinga, noko som gjer det naturleg å nytta teorien som utgangspunkt i analysen. Etter å ha teke utgangspunkt i teorien sat eg att med kategoriane og kodane vist i Tabell 4. Kategoriane er *algoritmisk tenking*, *problemløysing* og *kompetanse*. I kolonnane i tabellen står kodane (to nivå). Gjennom analyseprosessen såg eg at det var trong for fleire kodar, under det eksterne domenet, då dette var tema som gjekk att hjå fleire av informantane. Kodane som er lagt til undervegs er markert i kursiv i tabellen.

Tabell 4 - Oversikt over dei tre kategoriane samt dei to nivåa kodar.

Algoritmisk tenking	Problemløysing	Kompetanse
Fikle	Kven snakkar mest	Personleg domene
- Utforske og eksperimentere	- Samtalen	- Eigne oppfatningar
Skape	Spørsmål	- Kunnskap
- Designe	- Kva type spørsmål	Konsekvensdomene
- Lage	- Kven svarar	- Læringsutbytte
Feilsøke	- Kva gjer eg med svara	Eksternt domene
- Oppdage feil	Læring	- Kurs og etterutdanning
- Rette feil	- Legge til rette for læring	- Kollegaar
Holde ut	Elevane	- Samarbeid
- Fortsette	- Korleis er elevane	- <i>Eksamen</i>
- Prøve igjen	Kroppsspråk	- <i>Forlag/Lærebok</i>
Samarbeid	- Kva kommuniserer kroppsspråket mitt	- <i>Leiing</i>
- Dele		Praksisdomene
- Jobbe saman		- Klasserommet
Logikk		
- Analysere og forutsjå		
Algoritmer		
- Reglar		
- Steg for steg		
Dekomposisjon		
- Bryte ned i mindre delar		

Mønster <ul style="list-style-type: none"> - Finne og bruke likskapar Abstraksjon <ul style="list-style-type: none"> - Fjerne unødvendige detaljar Evaluering <ul style="list-style-type: none"> - Gjere vurderingar 		
---	--	--

Når eg har analysert dataa mine, har eg nytta dataverktøyet Nvivo⁴. Dette har gjort det lettare å strukturera datamaterialet og gjera systematiske analyser av materialet. Denne prosessen har vore iterativ, og som nemnt over har eg lagt til kodar etterkvart i prosessen. Slik eg ser det, har digitale verktøy utvida funksjonalitet, samanlikna med analyse med penn og papir, som er nyttige for meg. Dette inkluderer mogleiken til å enkelt ha oversikt over kor mange referansar dei ulike kodane er knytt til, og at ein kan søka etter hyppigheita på ord som kjem fram i datamaterialet, for å nemna nokre. Intervjutranskripsjonane eg sit at med er omfattande, med omlag 40 000 ord fordelt over 80 sider. Det gjer denne type digitalt verktøy særst nyttig, for å raskt kunna bla mellom ulike informantar og ulike kodar i analysen, og ha oversikt i materialet.

I hermeneutikken er tolking eit sentralt omgrep, og ein viktig del av analyseprosessen. Når det kjem til det å tolka er den hermeneutiske sirkelen trekt fram. Det handlar om å veksla mellom å studera heilskapen og dei ulike delane i teksten (Kvarv, 2014, s. 73). Med andre ord må ein sjå på kva ein setning kan ha å seia for meininga med heile teksten, eller kva heilskapen i teksten har å seia for meininga i ein setning. Som Zimmermann (2015, s. 26) trekkjer fram er det slik at vår «...better understanding of particular lines will reshape our grasp of the whole». Dermed vil det vera nyttig å sjå på kva som vert sagt i ein setning opp mot heilskapen i informanten sine utsegner, og andre vegen, for å forstå meininga i det som vert sagt. Dei innsamla undervisingsopplegga og oppgåvene vil då vera ein del av denne *heilskapen*.

⁴ <https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home>

3.7 Kvalitet

For å snakka om kvalitet i forskning nyttar ein ofte omgrepa *reliabilitet*, *validitet*, og *generaliserbarhet* (Tjora, 2021, s. 259). Eg vil i det følgjande gjera greie for kvaliteten i studien min gjennom å sjå på dei ulike kriteria. I tillegg vil eg sjå på etiske omsyn knytt til studien.

3.7.1 Reliabilitet

Reliabilitet, eller pålitelegheit, handlar om intern logikk og samanheng i forskingsprosjektet (Tjora, 2021, s. 259). For å auka reliabiliteten til studien min har eg skrive utfyllande om det eg har gjort, metoden, i kapittel 3 (Postholm, 2010, s. 131). Ved å vera så transparent som eg klarar, vil lesaren ha høve til å sjå samanhengen mellom dei ulike vala eg har gjort i prosessen. På denne måten vil lesaren kunna sjå gjennom forskingsprosessen min og ta stilling til den (s. 131).

I alle intervjuar har eg nytta den same intervjuguiden, med dei same spørsmåla og temaa. For å auka reliabiliteten til transkripsjonane samanlikna eg lydopptaka og dei transkriberte intervjuar fleire gongar, for å sikra at det var samsvar mellom dei. Presentasjonen av funna mine har eg gjort så objektivt som mogleg, utan å diskutera eller kommentera undervegs.

3.7.2 Validitet

I kvalitative studiar er spørsmålet om validitet, eller gyldigheit, ofte knytt opp mot omfanget av «*researcher bias*» (Befring, 2015, s. 54). Dette heng saman med at eit av dei viktigaste instrumenta som vert brukt i forskinga, er meg sjølv som utfører forskinga (Postholm, 2010, s. 127). Som tidlegare gjort greie for er dette tett knytt opp mot tolkingsvitskapen, hermeneutikken, og førforståinga eg har i møte med datamaterialet. For å auka validiteten, har eg i oppgåva freista å posisjonera meg i forskingsfeltet på ein tydeleg måte. Frå første kapittel har eg vore tydeleg på at eg til dagleg arbeider som matematikklærer i den vidaregåande skulen, noko som gjer det utfordrande å frigjera meg heilt frå førforståinga mi og fordommane eg har.

I presentasjonen av analyse og funn har eg valt å ha med fyldige utklipp av utsegner frå transkripsjonane, før eg i eit eige kapittel tolkar og gjer greie for mitt syn på det som har

kome fram (Befring, 2015, s. 55). På denne måten vil det koma tydelegare fram kva som er mine tolkingar, og kva som er intervjuobjekta sine tolkingar. Gjennom å samla inn data frå ulike metodar, som både undervisningsmateriale og intervju, har eg oppnådd ein form for triangulering som òg vil vera med på å styrkja validiteten (Krumsvik, 2019, s. 192). Denne type validitet som er presentert over omhandlar den indre validiteten, gjerne kalla kredibiliteten til studien (Krumsvik, 2019, s. 192).

I studien vil eg skildra, analysa og reflektera kring nokre praksisar, og det vil ikkje vera eit poeng at funna kan generaliserast til å gjelda alle matematikklærarar i Noreg. Dermed vil ikkje ytre validitet, generaliserbarheit, vera eit sentralt omgrep for studien (Krumsvik, 2019, s. 192). Eg tek sikte på ein studie som legg til rette for naturalistisk generalisering, som vil seie at lesaren kan kjenna att sin eigen situasjon, og sjå overføringsverdi til eigen praksis (Postholm, 2010, s. 131).

3.7.3 Etikk

I arbeidet med dette forskingsprosjektet har eg gjort det eg kan for å følgja retningslinjene til Den nasjonale forskningsetiske komité for samfunnsvitenskap og humaniora (NESH, 2016). Gjennom å melda prosjektet til NSD har eg fått ei kvalitetssikring på grepa eg har gjort for å sikra personvernet til informantane. Godkjenning frå NSD ligg ved oppgåva (vedlegg 2). I oppgåva har eg anonymisert informantane med å gje dei fiktive namn, og eg tek ikkje med utsegner som kan knytast opp til dei eller kollegaar av dei. Informantane fekk eit skriv med informasjon om prosjektet, databehandling og samtykkeskjema før datainnsamlinga. I skrivet var det òg informasjon om kontaktinfo for spørsmål og opning for å trekkja seg i etterkant for dei som ønska det. Fuglseth (2018, s. 261) peikar på at forskingsrapportar kan nyttast politisk. Forskinga mi kan verta eit «innlegg i debatten» for eller mot innføring av programmering i den norske skulen. Dette er ikkje mi hensikt med oppgåva, då eg er ute etter å sjå på *korleis* denne innføringa vert møtt av lærarar som har teke i bruk verktøyet i undervisinga si som ein del av Fagfornyninga.

Gjennom prosessen har eg arbeidd sjølvstendig. Eg har gjennom oppgåva referert til alle kjelder som er sitert og parafasert, slik at det kjem tydeleg fram av oppgåva kva som er mine og andre sine tankar.

4 Presentasjon av funn

I dette kapitlet vil eg presentera funna eg har gjort i studien. Kapitlet er delt inn i to delkapittel. Det første handlar om *utvikling av kompetanse* som er tema i F1. Det andre delkapitlet handlar om korleis lærarane legg til rette for å *løysa problem med algoritmisk tenking* som er tema i F2.

4.1 Utvikling av kompetanse

Det første forskingsspørsmålet i oppgåva mi omhandlar utvikling av kompetanse hjå lærarar: *Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?* I intervju med lærarane har eg fått fram ulike refleksjonar knytt til kompetansen deira og utvikling av han. Funna her er presentert i delkapittel tett knytt opp til dei ulike domena i endringsmiljømodellen presentert i kap. 2.4.1. I tillegg kjem det eit delkapittel som handlar om framtida og dei komande åra.

4.1.1 Kompetansen og oppfatningane til lærarane (*Personal domain*)

Tre av dei fem lærarane eg har snakka med har teke formell utdanning (har studiepoeng) i algoritmisk tenking eller programmering. Maria og Helge har begge teke eit vidareutdanningskurs på 15 studiepoeng ved NTNU. Maria fortel: «Så tok eg dette her som heiter programmering for lærarar 8-13 trinn på NTNU, i fjor (...) med fokus på blokkprogrammering og tekstbasert programmering med Python, og det å laga undervisningsopplegg ved bruk av programmering». Tove har òg teke vidareutdanning, men på Universitetet i Bergen. Kurset ho tok vart kalla «Algoritmisk tenkning og programmering i matematikkfaget» og var på 15 studiepoeng. Tove trekkjer fram utbyttet etter kurset slik:

Poenget var at det var retta mot mattelærarar som kom til å måtta undervisa i programmering (...) og det var eit studie der me lærte grunnleggjande om programmering. (...) så skreiv me sjølv oppgåver som var knytt inn mot undervisinga. Så eg syns ikkje det var så mykje knytt opp mot mi undervising i desse samlingane, det var meir opp til meg sjølv når eg skreiv oppgåver. (Tove)

Som følgje av at fokuset på samlingane var på grunnleggjande programmering, og ikkje så didaktisk vinkla som ho hadde håpa, tenkjer Tove å ta enno eit kurs:

Så difor har eg søkt om å få ta enno ein sånn type kurs der du må ha litt programmeringsforkunnskapar, men det er meir fokus på korleis ein kan nytta det i undervisinga. Fordi eg opplever at når eg no underviser i år, så føler eg meg litt sånn... Eg skulle gjerne kunna meir då, og spesielt retta inn mot undervisinga. (Tove)

Lærarane har gjort seg refleksjonar kring bruken av algoritmisk tenking og programmering i matematikkfaget, i studiesamanheng. Eit døme på denne type refleksjonar vert trekt fram av Maria:

Kva er det dette her med bruk av programmering, bruk av datamaskin, den er dum, den gjer berre det du ber han om, sant, altså korleis kan me utnytta det i ein pedagogisk fornuftig måte? (Maria)

Litt det same finn me hos Tove: «Eg trur nok eg brukte mest energi på å lure på *kvifor er dette smart, kvifor er dette bra for elevane våre*». Trass i at spørsmåla Tove stiller seg sjølv kan sjå negative ut (*kva skal me med dette?*), så kom det fram av tonefall og kontekst i intervjuet at det handla om interesse og refleksjon kring nytteverdien.

På spørsmål om kva som låg bak avgjerda om å ta vidareutdanning, trekk Helge fram den nye læreplanen.

Nei, det var eigentleg ny læreplan. Prøve å ha eit lite forsprang. Ikkje måtte bruke alt for mykje tid på dei nye elementa, då. At eg var eit hestehovud framfor elevane. Slik at eg kunne hjelpa dei med det som dei måtte lure på. (Helge)

Lærarane som har teke vidareutdanning har gjort seg refleksjonar kring eigen undervising, gjennom studieløpet. Det vert trekt fram refleksjonar rundt korleis det som vert undervist på studiet kan brukast vidare av lærarane, i deira eigne klasserom. Tove peikar på at det som vart gjennomgått i kurset var gjenstand for refleksjon knytt til eigen bruk. På spørsmål om korleis ho har førebudd seg på innføringa av dei nye elementa i

læreplanen vert dette trekt fram: «Det er vel mest det med at eg har tatt dette kurset før eg kom til 1T, så eg prøvde jo då å tenkja litt undervegs der, kva kan eg bruka det til» (Tove).

Maria brukte sluttvurderinga i kurset sitt til å laga eit undervisningsopplegg som vart nytta av alle 1T-gruppene ved oppstart av nytt skuleår. Målet var å gjera elevane kjende med algoritme-omgrepet.

Då var alle 1T-klassane gjennom det opplegget som me hadde laga til eksamen. Okay, me viser den videoen med peanøttsmøret, me går gjennom det med tangramma, me får dei til å på ein måte skjønna det, sånn at når dei no seinare møter oppgåver der det står «skriv ein algoritme som...». (Maria)

Videoen med peanøttsmøret det vert referert til her, omhandlar ein far som prøver å få borna sine til å skriva gode nok algoritmar, eller oppskrifter, til at han kan følgja dei når han skal laga ei skive med peanøttsmør⁵. Oppgåva med tangram kjem eg attende til i kap. 4.3.

Av dei to lærarane som ikkje har formell vidareutdanning har ein av dei delteke på kurs som ein del av førebuinga til den nye læreplanen. Lars har vore på kurs i regi av skuleeigar, der tema har vore undervisningsopplegg med programmering. Han har gjort seg fleire tankar om det presenterte undervisningsopplegget. Opplegget han reflekterer rundt handlar om programmeringsoppgåver der elevane vert gjevne ferdige strukturar eller rammer, for t.d. viss-setningar, men innhaldet i dei manglar. Lars ser både fordelar og ulemper med denne framgangsmåten:

Altså det er klart ein kan visa nytteverdien av programmering i enno større grad, få enno meir avanserte program og kulare resultat, kanskje. Men eg tenkjer jo at no er det viktig at dei lærer ting frå grunnen av. At det er ein sentral del av den nye læreplanen, så, eg er jo skeptisk til det som utgangspunkt. (Lars)

⁵ https://youtu.be/cDA3_5982h8

Lars peikar vidare på at programma elevane skal laga, ikkje må vera vanskelegare enn at dei klarar å laga dei sjølv, då gjerne med hjelp av eit kommandokart som hjelpemiddel. Refleksjonane vert vidare knytt opp mot den nye eksamensordninga: «Det blir jo litt spennande å sjå kvar det landar, det er litt den retninga eg tolkar eksamen og, at dei får kanskje ein sånn pseudokode kanskje, men dei får ikkje eit ferdig oppsett...» (Lars). Maria har òg vore på kurs i tillegg til vidareutdanninga. Her er det snakk om kurs som vert arrangert gjennom fagforeininga ho er med i. Kursa er retta mot lærarar og undervising, og Maria fortel om utbyttet:

Då får du nokre ferdiglaga undervisingsopplegg som du då kan «modde» i staden for å byggja dei frå grunnen. Eg er ikkje så god i Python at eg kan absolutt alle finesser og bibliotek som ein kan bruka for å laga glidarar og sånt, det var jo ikkje sånt som eg visste om før eg tok det. (Maria)

4.1.2 Klasserommet (*Domain of Practice*)

Undervisinga i inneverande skuleår har vore avgjerande for å utvikla den didaktiske kompetansen knytt til algoritmisk tenking og programmering for lærarane. Det heng saman med at det er det første året dei gjer dette. Det som går att hjå somme av dei, er at det har vore ein del prøving og feiling for å sjå kva som fungerer og ikkje. Lars fortel om prosessen: «Nei, det har jo vore litt sånn, litt famlande [ler] i starten. Eh, det har jo blitt litt sånn prøving og feiling då». Litt det same finn me i refleksjonane til Torrey, som trur kompetansen vil utvikla seg i takt med erfaring og ei forsøksbasert tilnærming til undervisinga:

Hadde du spurd meg om eit år, om to år, fem år, så hadde eg sikkert hatt nokre fleire svar, kor eg hadde fått prøvd dette her, og prøvd å feila litt. Erfart litt. Men akkurat no så er det litt synsing, rett og slett. Ein må prøve, og så må ein berre sjå litt korleis det går, og å lære litt av det... (Torrey)

Lars og Torrey er dei to lærarane i studien som ikkje har teke formell vidareutdanning innan algoritmisk tenking eller programmering.

Tove reflekterer litt rundt klasseromspraksisen og algoritmisk tenking. Ho trekkjer fram algoritmisk tenking som rike oppgåver (gjernar store), der ein kan verta dradd i ulike

retningar. Dette er ikkje noko ho ser på som nytt: «For det har jo vore sånn før òg, at du skal ha rike oppgåver og få elevane til å reflektera sjølv. Det har eg alltid syntest har vore veldig vanskeleg, å laga gode oppgåver av den typen». Tove trekk vidare fram at ho skulle ønskt ho hadde sett seg meir inn i det som handlar om algoritmisk tenking. Ho trekkjer fram det å setja seg ned, og setja av tid, for å sjå på læreplanen og finna aktuelle oppgåver som passar til dei ulike temaa. Dette vert trekt fram som eit problem i kvardagen: «For eg opplever det at når eg kjem til eit nytt tema, så har eg ikkje tid til å sitja å fundera over kva eg kan gjera her» (Tove).

Noko som går att hjå fleire av lærarane, er at dei ikkje har fokusert på algoritmisk tenking men programmering, i klasserommet. Både Tove og Torrey trekkjer fram dette eksplisitt. Torrey grunngjev fokuset sitt på programmering, i forståinga han har av læreplanen: «Sånn som eg har forstått det, er det med algoritmisk tenking noko dei vil at elevane skal jobba med mellom anna gjennom programmering. Så eg har på ein måte fokusert mest på programmeringsbiten, kan du seie». Helge fokuserer òg på programmering i undervisinga si. Første økt vart det gjennomført eit opplegg med fokus på algoritmar og oppskrifter. Om undervisinga resten av året trekkjer han fram dette: «Og så har me hatt jamne drypp heile vegen, med små program i Python» (Helge).

Lars på si side, har ei anna forståing av algoritmisk tenking. Han trekkjer fram dette om praksisen sin i år:

Me har jobba meir utforskande, og inn der så kjem jo ein del av desse elementa, med å starta enkelt, og gjera det meir og meir avansert. Sånn sett så kan ein bruka algoritmisk tenking sjølv om ein ikkje jobbar med programmering. (Lars)

Han trekkjer vidare fram eit døme på ei oppgåve eller problem han har gjeve elevane, som illustrerer denne tankegangen. Elevane kan få spørsmål som: «Viss du har eit vilkårleg rektangel, og så aukar du lengda med 10 % og så reduserer du breidda med 10 %, kva skjer med arealet då?» (Lars). Her trekkjer han fram det å bryta problemet opp i mindre bitar som ein mogleg strategi elevane kan nytta. Sjå på eit rektangel med bestemte sider, og samanlikna ulike storleikar, for så å sjå etter mønster. Maria trekkjer òg fram denne type oppgåver og opne spørsmål, som noko ho tenkjer kan løysast ved å nytta algoritmisk tenking.

4.1.3 Kva skal me med dette? (*Domain of Consequence*)

Når det kjem til elevane sitt læringsutbytte har lærarane ulike tankar. Helge trekkjer fram at elevane hans ofte viser ein del motstand: «Dei er gode på å setta seg på bakbeina, og spør *kva skal me med dette, kva er vitsen?*». I møte med dette trekkjer han fram at ein må ha fornuftige argument. Helge har nytta oppgåver knytt til m.a. radioaktivitet og halveringstid, slik at elevane kan arbeida med ekte data: «Då kan dei bruka programmering for å ta inn eit stort datasett og bearbeida det. Og ut i frå det dei kan av programmering kan dei då berekna ei halveringstid, så det er jo litt kjekt».

Fleire av lærarane viser til at det er vanskeleg å få til å bruka algoritmisk tenking og programmering i undervisinga, på ein god måte, som følgje av elevane sin kompetanse. Torrey trekkjer fram dette: «Elevane mine er ikkje der enno. Det vert eit for stort spenn mellom matematikken og programmeringa, for dei kan ikkje nok programmering rett og slett. Og det vert litt to ulike verdenar enn så lenge». Tove trekk òg fram elevane sin kompetanse som utfordrande, med tanke på bruken av algoritmisk tenking og programmering i matematikken:

Veldig mange kan ingenting, så eg føler det er meir for å halda hovudet over vatnet, meir enn at eg klarar å gjer dei gode oppgåver, der dei skal utforska og vera kreative. For dei har ikkje verktøya enno, sant? (Tove)

Vidare tenkjer Tove at dette er noko som vil endra seg, etterkvart som åra går: «Det vil jo betra seg om fem år, når dei kjem frå ungdomsskulen og har lært seg ein del der (...). At ein faktisk kan fokusera mykje meir på den algoritmisk tenking-delen». Helge har òg tru på at dette vil endra seg om tid: «Eg trur jo det vert ein forse om dei klarar å utvikle grunnleggjande ferdigheiter i ungdomsskulen».

Som følgje av at både Tove og elevane hennar i år kjenner på manglande kompetanse, har ho lagt opp til eit fokus på det å lese kode. Ho fortel om refleksjonane sine: «Der me er no, er det nok viktigare, og det er nok lettare for dei, å lesa program enn å koma på dei sjølv». Tove har laga eit opplegg som inneber å lesa program for å forstå kva som skjer i koden (Kap. 4.3.2). Vidare reflekterer Tove: «Tanken er at dess meir program dei les, dess meir kan dei få innblikk i kva programmet gjer. Når dei då skal gjera noko seinare, så hugsar dei kanskje at dei har lese eit sånn program før». Likevel kjenner

Tove på dårleg samvit, både som følgje av tidspress og nivået på eigen kompetanse: «Eg kjenner jo litt på det, at elevane mine lir litt av at eg ikkje har hatt kunnskap nok. At det krev meir av meg å laga eit godt opplegg om programmering, og at det er difor dei kan så lite».

4.1.4 Samarbeid og kollegaar (*External Domain*)

Alle lærarane trekkjer fram samarbeid og kollegaar som viktige på ein eller annan måte når det kjem til utvikling av kompetanse. Lærarane gjev uttrykk for at leiinga på skulane legg til rette for samarbeid mellom matematikklærarane, og fleire har timeplanfesta møtetid til samarbeid. I tillegg til kollegaar på eigen arbeidsplass, vert det trekt fram at vidareutdanning kan opna for samarbeid. Maria fortel om sine erfaringar:

På NTNU brukte dei Slack som kommunikasjonsform som betydde at det heile tida skjedde noko. Når det er 200 studentar som kan både stille spørsmål, veldig mange stiller jo spørsmål, men det var ein god del kompetente folk rundt om i Noregs land, både på ungdomsskular og vidaregåande skular som hadde små tips, så kom det ein YouTube-video her, og ein YouTube-video der, og så blir du sitjande å sjå på litt ting då, sant. (Maria)

Det skjer òg mykje samarbeid på skulane mellom kollegaane. Fire av dei fem lærarane har kollegaar som òg underviser i IT. Her kjem det fram at lærarane har ulikt fokus i samarbeidet sitt. Maria fortel om eit samarbeid som fokuserer på framdrift og kor langt i læreboka dei skal vera komen til ei kvar tid.

(D)et blir laga ein felles årsplan, som er veldig detaljert. Ikkje detaljert i forhold til oppgåver og pedagogisk opplegg, men, i den timen så skal du sjå på delkapittel 6.1 som handlar om gjennomsnittleg vekstfart, i den timen skal du sjå på det, og så vidare (...) Veldig fokus på kapittelprogresjon. (Maria)

Ein av årsakene til at framdrift er sentralt i samarbeidet i gruppa Maria er ein del av, følgjer av at dei vanlegvis har nokre av undervisingstimane felles, i eit større rom.

Maria peikar på at det ikkje er fokus på dei nye omgrepa frå ny læreplan og eksamen. Det er, i følgje ho, for lite snakk om dette i lærarkollegiet:

For deg som no tek ein master i dette, og skal inn i eit lærarsamarbeid, kan bringa inn den stemma; «Kan me og vera tydeleg med det språket?». For dette er ord som er tydelege i læreplanen og eksempeloppgåvene frå eksamen. Me må snakka dette språket mellom oss for at me skal snakka det med elevane, slik at dei har ein sjanse til å læra seg kva det går i. (Maria)

Torrey på si side, er del av eit team som har eit anna fokus. Her kjem det fram at ein stor del av samarbeidet er knytt opp mot didaktiske val og utforming av undervisningsopplegg. Samarbeidet vert opplevd som godt:

Spesielt i dei tidene me er inne i no, med nye læreplanar, å ha nokon å dele idear og utfordringar med, og tankar med generelt. Kor me får mange gode diskusjonar på korleis me kan tolka læreplanmåla og korleis me kan løyse dette på ein best mogleg måte. (Torrey)

Vidare trekk Torrey fram døme på kva som kan vera tema i samarbeidet: «Korleis me skal løyse denne problemstillinga med programmering. Kor mykje tid skal me vektlegge på det? I kva tema skal me inkludera det? Og så vidare...». Lars trekk òg fram eit liknande samarbeidsmiljø som Torrey, der faglærarane i IT treffest mykje, gjerne meir enn det er sett av tid til på timeplanen, for å diskutera og samtala om praksisen. Samarbeidet som vert trekt fram her, handlar om å skapa ei felles forståing i lærargruppa om korleis ein skal tolka den nye læreplanen med dei nye elementa som algoritmisk tenking og problemløysing.

Når det kjem nye ting så tek det gjerne eit år for å liksom heilt skjønna kva omfanget av dette her er, så det blir jo litt sånn prøving og feiling, meir enn at me kan ha ein sånn heilskapleg plan på korleis kan me utvikla algoritmisk tenking. (Lars)

Helge trekkjer fram at det er ein del samarbeid om kor langt ein er komen, framdriftsplan, og det å laga vurderingar til elevane. I tillegg er det ein diskusjon kring kva for nivå programmeringsundervisinga skal vera på: «Me har hatt litt diskusjonar no i forhold til programmering. Kor langt skal me gå? Kor fort skal me pusha dei gjennom? Det som er vanskeleg er å definera nivået, kanskje, sant?».

Tove arbeider på ein skule som berre har ei gruppe med 1T, og er med det åleine som 1T-lærer. Likevel vert samarbeid trekt fram som nyttig her òg.

I matten så er det berre eg som har programmeringa, så eg står litt åleine, men der har dei vore veldig fine dei andre mattelærarane. Me har snakka ein del i lag om desse læreplanane sjølv om dei andre ikkje har faget. (Tove)

Tove tok vidareutdanning saman med ein kollega. Kollegaen vert no nytta som sparringspartner i utarbeiding og planlegging av undervising:

Sånn som kollegaen eg studerte med i fjor har eg spurt til råds: «Kva tenkjer du om at eg gjer dette?» når eg laga det der opplegget med at dei skulle lesa ein del program, så endra litt på koden og sånn. Då diskuterte eg det med kollegaen først. (Tove)

Det er ikkje berre samarbeid mellom kollegaar som vert trekt fram av lærarane. Både Tove og Torrey ser på seg sjølv og klassane sine, som team som jobbar saman for å læra. Tove reflekterer òg rundt korleis hennar kompetanse i programmering, som ho meiner ikkje er så høg, gjer at ho og klassen må jobba saman:

«Det prøver eg å vera veldig ærleg på. Her må me finna ut av ting i lag. Det tenkjer eg er positivt, at [elevane] føler det er ein prosess me gjer i lag. At det ikkje berre er eg som står og matar dei, men at me gjer dette saman. (Tove)

4.1.5 Kompetanse i åra som kjem

Ettersom elevane som tek 1T inneverande skuleår kjem frå 10 år i grunnskulen med læreplanar frå Kunnskapsløftet, har dei aller fleste lite, eller ingen, erfaring med programmering og algoritmisk tenking (som omgrep). Dei komande åra vil dette endra seg, og elevane som kjem inn i 1T vil, etter kvart, ha god erfaring med desse elementa når dei har hatt matematikklæreplanar frå Fagfornyinga. Lærarane har gjort seg fleire refleksjonar rundt kompetansen sin, og utviklinga av denne i åra som kjem. Lars er ikkje veldig uroa, trass i at læraren ikkje har hatt programmering eller algoritmisk tenking i lektorutdanninga si i stor grad:

Sånn sett så har eg jo på ein måte måtta friska opp dette her parallelt med at me har hatt det i undervisinga. Og så lærer jo eg raskare enn elevane, så det går greit. Men dei kjem til å kunna meir og meir, så eg trur nok eg skal holda tritt (...) Ein kan jo ikkje heva taket heile tida heller, ein må jo på ein måte seie at dette er så langt ein skal i IT (...), så elevane kjem nok til å nå eit tak på eit eller anna tidspunkt, men det vert sikkert nokre år til då... (Lars)

Maria har litt same tankar om kompetansen sin i åra som kjem: «Eg òg må jo læra meg ting, og det betyr jo at for kvart einaste år det kjem inn nye elevar så må me laga nye opplegg på ein måte som er tilpassa det som dei kan frå før».

Helge er heller ikkje uroa over utviklinga dei komande åra, men finn ro på ein litt anna måte enn Maria og Lars.

Altså eg kan grunnleggjande programmering, og tenkjer at eg er heldig som klarar meg greitt i forhold til det. Men eg ser jo at dei som er racarar dei må berre få lov til å bli mykje flinkare enn kva eg er til programmering. Og eg har ikkje nokre kvaler at eg må kunna svara på eit kvart spørsmål. Dei som er ekstremt flinke får eg prøva å geleida inn til å søka informasjon på nett. (Helge)

Torrey er litt meir uroa for utviklinga dei komande åra, no som det første året som lærar og i ny læreplan nærmar seg slutten:

Nei, det vert jo litt skummelt, at jo fleire år det går, jo flinkare vert desse elevane som kjem, så då må jo eg kunne enno meir. (...) Eg har jo 10 studiepoeng i grunnleggjande programmering i Python, og det er nok til å læra vekk grunnleggjande programmering til elevane, men det er jo ikkje nok til å ta alle desse didaktiske vala me gjer i matematikkfaget generelt, sant? Det å faktisk velje dei temaa der programmering faktisk er mest nyttig, for din klasse. (...) Så det er mange val her, ein må ta, og eg ser tydeleg no at eg har ikkje god nok kompetanse i det, rett og slett. (Torrey)

Samstundes fortel Torrey om korleis han meiner kollegaane med vidareutdanning og meir erfaring har hatt utbyte av studiane:

Eg veit at mange av kollegaane mine og har kjend på manglande kompetanse, og mange av dei har vore litt frustrerte på denne programmeringa, for det er noko heilt nytt i matematikkfaget. Men eg ser no, etter at dei har vore på vidareutdanning og lært seg å programmere skikkeleg, ser dei mange mogleikar med det (...). Dei har jo litt meir erfaring enn meg med å undervise matematikk. Eg trur dei kanskje har det litt lettare enn meg, med å sjå «dette kan me bruka i dette og dette temaet» som eg kanskje ikkje har erfart godt nok enno. (Torrey)

Tove har òg gjort seg nokre tankar om korleis det vert når elevane har meir kompetanse frå før. Ho fortel om ein elev frå årets gruppe med erfaring, og korleis han har eit føremon: «Det er spesielt ein som er veldig flink, og han tenkjer litt annleis enn meg. Eg merkar at han er mykje meir kreativ enn meg» (Tove). Tove trekkjer fram at dette mest truleg vil bli den nye normalen om nokre år, og at kompetansen hennar ikkje er klar for det per no. Eit alternativ som vert skissert er å studera meir, ta noko «rein programmering», medan det andre alternativet Tove peikar på, handlar om å våga å læra av og med elevane:

Eg skal jo framleis læra dei noko, og eg kan jo det matematiske, sant? Og då kan ein kanskje tenkja seg at ein kan få til litt meir samarbeid. Eg kan jo leggja opp til algoritmisk tenking, for eksempel ved hjelp av programmering, sjølv om eg ikkje har løyst det på same måte som dei (...). Det må jo vera målet, at elevane skal kunna vera kreative, og ikkje at dei skal føla at dei må fram til ein fasit som eg har, men at dei kan finna andre løysingar òg. (Tove)

Felles for alle, er at dei tenkjer at kompetansen må utvikla seg dei komande åra. Som Maria seier det: «Så eg er ikkje utlært, og takk og lov for det. Då hadde det jo vore keisamt!».

4.2 Leggja til rette for å løysa problem

Lærarane har gjort seg ulike refleksjonar kring undervisinga deira når algoritmisk tema står på agendaen. I dette kapittelet vil eg trekkja fram funna eg har gjort. Inndelinga mi tek utgangspunkt i dei ulike aspekta knytt til klasseromskultur for problemløysing som nemnt i kapittel 2.3.4. Dei handlar i hovudsak om *samtalen, legge til rette for læring og elevane*.

4.2.1 Samtalen og spørsmål i klasserommet

Det som går att hjå lærarane når dei skildrar klasserommet og samtalen er at dei fokuserer på kva elevane tenkjer. Fleire av lærarane trekkjer fram at dei stiller spørsmål som skal vera med på å få elevane inn i ein utforskande tankegang. Maria reflekterer rundt dette:

«Kva trur de eg kan gjera då, om eg skal finna ut av det?» Altså at dei får lov til å reflektera litt sjølv. Men då er det ikkje så mykje på algoritmisk tenking, men meir på den der utforskande tankemåten. Men eg tenkjer jo det at skal du vera god på utforskande oppgåver, så må du vera god på algoritmisk tenking. Men viss du skal vera god på algoritmisk tenking (...) så må du jobba med meir utforskande oppgåver, for viss ikkje blir det instrumentelt! (Maria)

Lars stiller spørsmål for å få fram ulike tilnærmingar frå elevane. Han fortel om undervisinga (som har gått føre seg på Teams, grunna koronaviruset) der læraren på førehand har sett på dei ulike gruppene sitt arbeid:

Ja, for eksempel: «De løyste det jo litt på ein annan måte, Torodd, kan du fortelja korleis de gjorde det?». Så er gjerne eg sekretær og skriv det dei forklarar at eg skal gjera. Dei må jo øva seg litt på å bruka språket òg. Og så på sett og vis så kan me utvikla program i lag. Så sånn sett får me jo trening på algoritmisk tenking då, med at me startar enkelt. (Lars)

Vidare fortel Lars om korleis ein, med utgangspunkt i enkle program i fellesskap, kan utvida funksjonaliteten. Døme på dette er andregradslikningar der ein ikkje har to, men ei eller ingen løysingar. Torrey fortel òg om eit fokus på å henta inn forslag frå elevane,

og på å forstå det som skjer: «Kva er det som skjer i denne funksjonen? Kva er det dei vil me skal finne? Okay, har me nokre tankar på korleis me kan gjera det?». Vidare kjem det fram at Torrey er oppteken av å få fram elevane sine refleksjonar og ulike måtar å løysa problem på: «Eg prøver å bruka tid på dei samtalan og dei diskusjonane med elevane om *korleis har du tenkt, kvifor har du tenkt som du gjer, er det andre måtar me kan løysa det på?*».

Helge trekkjer fram at det er viktig å ha oppsummerande samtalar undervegs, for å få alle med, med forklaringar av det som er gjort så langt:

Det å samla dei undervegs, det trur eg er avgjerande. Å visa sånn for alle, og forklara grundig. Her har me sett inn ei for-løkkje, kva skjer i for-løkkja? Sant, heilt sånn trinnvis. Kvifor var dette viktig, i forhold til algoritmen vår? Kvifor har me dette elementet på plass? (Helge)

Lars prøver å vera litt på elevane sitt nivå i samtalen når elevane arbeider med utforsking:

Eg prøvar å spela litt naiv på ein måte, at eg ikkje skjønar heilt. Eg stiller kritiske spørsmål, og når eg spelar litt uvitande så må dei forklara meir. «Åja, det er sånn ja... Korleis vert det då? Kva betyr eigentleg det?». (Lars)

Tove er oppteken av å stilla spørsmål på ein sånn måte at elevane ikkje oppfattar det som om det er ein «kva tenkjer læraren på no»-konkurranse: «Eg prøver å ikkje ha eitt svar eg skal fram til når eg stiller spørsmål, eg prøver å stilla opne spørsmål».

4.2.2 Legge til rette for læring

I intervju kjem det fram at det å leggje til rette for utforsking er noko lærarane knytt opp til algoritmisk tenking, ikkje berre gjennom spørsmåla dei stiller og samtalen. Tove trekkjer fram dette:

Eg syns det er aller kjekkast når elevane ikkje kan så mykje. Då driv dei og utforskar litt kvar for seg, og kjem med tankar, og så er det ingen som kan sei til

dei «nei det er feil» men det vert heller «men er det ikkje litt meir sånn?».

(Tove)

Lars trekkjer òg fram ein utforskande arbeidsmåte som arbeidsmetode i undervisinga hans: «Eg ønskjer at elevane skal finna ut mest mogleg på eigenhand, at eg skal forklara minst mogleg, litt sånn i tråd med utforsking, som er eit anna kjenneteikn i denne læreplanen». Han meiner likevel at det er ein skilnad mellom algoritmisk tenking og utforsking:

I algoritmisk tenking så er ein kanskje i litt større grad fokusert mot at ein skal landa på eit bestemt resultat. Ein kan jo laga ulike program som fungerer, med ulike løkker eller sånne ting, men det er kanskje litt meir definert mål då. (Lars)

Likevel er det fleire av lærarane som syns det er utfordrande med utforsking i klasserommet. Helge trekkjer fram at utforsking er noko av det vanskelegaste i pensum, for elevane: «Så vert det jo nokre seansar der dei vert sitjande å kjenna litt på hjelpeløysa og prøver å finne ut av ting på eiga hand».

Torrey trekkjer igjen fram kompetansen til elevane når det kjem til utforsking og programmering:

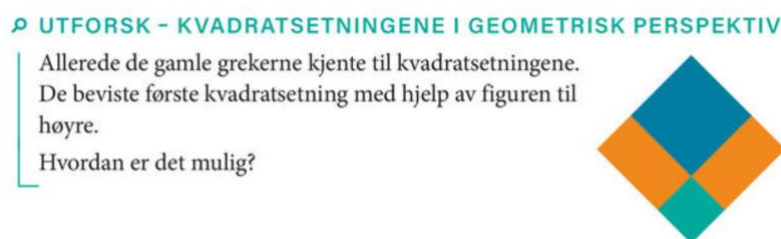
Eg skulle ønske me kunne brukt mykje mindre tid på det grunnleggjande innan programmering, og brukt meir tid på tema kor me kan bruke programmering til å utforske og kanskje sjå samanhengar i ting, og ha litt gøy rett og slett, vera litt kreative i matematikk. (Torrey)

Lars har nytta problemløysing og utforskande oppgåver med elevane. I eit samarbeidsopplegg like før jul arbeidde dei med dette. Lars fortel frå økta:

Då brukte fleire av dei elevane, litt over middels, programmering som verktøy for å løysa dei problema. Det vart meir ei sånn valfri greie. Og då fekk eg inntrykk av at motivasjonen vart litt annleis. Programmering vart litt meir eit verktøy eller hjelpemiddel, ikkje noko som dei var tvungne til. (Lars)

Maria trekkjer fram at det har vore tradisjon i matematikken og gjerne eksamen at det ikkje er rom for utforsking. Oppgåvene er utforma med mange deloppgåver, som stegvis tek elevane mot den *eigentlege* oppgåva, siste deloppgåve. Maria skildrar dette som lært hjelpeløyse, og at ein kan legge til rette for læring, og utforsking, ved å fjerne desse stega. Då kan elevane få meir opne oppgåver: «Korleis du løyser det spelar inga rolle, og då må du dela det opp i delproblem sjølv. Då må du inn med algoritmisk tenking». Maria peikar på at læreboka har tips til utforskande oppgåver. Desse har ho nytta med elevane sine. Døme på ei av desse utforskingsoppgåvene er vist i Figur 8.

I boka er utforskingsoppgåvene typisk berre tre setningar, og følast heilt uoverkommeleg for ein del av dei (...). Eg avsluttar gjerne timen før lunsj med å gje dei dei fire linjene, og det er ganske komplekst. Då pirrar eg litt i dei, og spør om dei har nokon forslag. (...) Så startar eg timen etter lunsj med same oppgåve, men med litt meir informasjon. (Maria)



Figur 8 - Utforskande oppgåve frå læreboka Maria nyttar (Oldervoll, 2020)

Lærarane fortel òg om ein type utforsking i heilklasseundervising. Helge trekk koplingar til programmering:

Sånn som i kodeverda så går det an å testa ting ut. Då berre prøver med det, då. Eg avfeiar ikkje ting som galt i utgangspunktet, men me testar det ut. «Okay, gjev dette noko meining då?» At eg prøver å ta innspela på alvor. (Helge)

Tove gjer det på same måte, og referer til bruken hennar av opne spørsmål i klasseromsamtalen: «Altså om [elevane] kjem med forslag til korleis me skal gjera det, så prøver eg alltid å gjera det som dei seier. For det er jo det som er poenget mitt med å spørja ope».

Fleire av lærarane er opptekne av at elevane skal snakka saman når dei jobbar saman. Tove trekkjer fram at samtale mellom elevar er noko ho legg til rette for: «Eg brukar veldig mykje at dei skal snakka i lag to og to». Lars nyttar òg samarbeid som ein stor del av undervisinga. Han har nytta samarbeid som ein del av vurderingssituasjonar: «Ein slags samarbeidsprøve, der elevane fekk ukjende problem frå før» (Lars).

4.2.3 Feilsøking og uthald

Når det kjem til elevane og korleis dei arbeider, så har lærarane gjort seg ulike tankar om korleis ein skal handtera feil, og korleis elevane taklar det å gjera feil. Maria har ikkje hatt problem med at elevane ikkje prøver seg fram og tek sjansar, med unntak av heilt i starten: «Eg veit ein del av kollegaane mine har klaga litt over det at elevane torde ikkje prøva seg for dei var redd for å gjera feil. Det har eg eigentleg ikkje hatt som eit problem» (Maria).

Torrey prøver å skapa eit klasserom der det å gjera feil er sentralt. Han knyt det opp mot den utforskande arbeidsmetoden:

Eg prøvar å lære dei at det å gjere feil ikkje berre er lov, men naudsynt. Då kan me skapa ein sånn kultur i klasserommet der me er litt utforskande og finn ut av ting i lag, og der me ser at oppgåver kan løysast på fleire måtar. Det er mange interessante diskusjonar å ha der. (Torrey)

Tove er òg oppteken av å gjera feil, men syns at det er utfordrande å handtera feil: «Korleis få dei til å våga å ha feil? Fordi det er jo då du mister dei i samtalen, med ein gong dei ikkje vågar å ha feil». Lars trekkjer fram at han har vore for rask med å fortelja elevane kva som er feil i programkoden og seier gjerne ting som «Ser du det raude krysset der, det betyr at du har skriva eit komma i staden for punktum». Han fortel at dette er utsegner han angrar på i etterkant, men grunngjev det i ei frykt for at elevane skal gje opp. Han seier vidare: «På kort sikt er det sikkert ikkje så dumt, men på lang sikt er det nok ein dårleg ide».

Maria fortel om situasjonar der elevane ikkje er uthaldne. Dette skjer gjerne når dei ikkje veit kvar dei skal byrja og er «heilt blanke». Maria reflekterer: «Eg innser at det er

ein inngangsport her, problemet er at eg ikkje har opna den døra heilt». Tove har elevar som også har lett for å ikkje ta store sjansar, og gjerne kan gje opp:

Eg trur nok matteundervisinga vår er litt tungrodd. [Elevane] er veldig vande med at dei får ting servert, og så skal dei bruka det i oppgåver som er like. Med ein gong du får noko som er litt annleis, så syns dei det er utfordrande å tenkja utanfor boksen eller prøva seg fram sjølv. (Tove)

Helge fortel at han syns at det er nokre typar elevar som gjev opp ganske lett. Han skildrar elevane:

Det verkar ikkje som om dei klarar å byggja stein på stein. Det er ei steinrøys eigentleg. Dei klarar ikkje å få bitane til å passa. Dei hugsar ikkje frå gong til gong. Dei klarar ikkje å skilja metodar frå kvarandre og ser ikkje kva tid det er fornuftig å nytta det eine kontra det andre. (Helge)

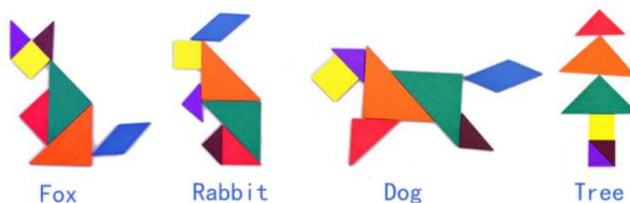
Han skildrar vidare elevane i klassen som ei elevgruppe som ikkje tek nokre sjansar, og ikkje torer å ta eigne val, i arbeid med programmering: «Dei må ha bekrefta, «er dette greitt», «er dette greitt», «er dette greitt», så det er jo å gå frå linje til linje i korleis eit program verkar, og kva tid dei står fast» (Helge). Som Tove, tenkjer Helge at dette kan ha å gjera med at ein har snudd om på det tradisjonelle mønsteret, når ein har teke algoritmisk tenking inn i undervisinga: «Dei er vande med å få algoritmen og ikkje utarbeida han sjølv. Når ein no snur på det, og dei skal forstå og prøva seg så... Somme lærer fort å sykle, medan andre treng støttehjul i lengre tid» (Helge).

4.3 Innsamla undervisningsmateriale

I tillegg til dei små oppgåvene og problema elevane har fått som er kome fram gjennom intervjuet så har eg samla inn oppgåver og undervisningsmateriale lærarane har nytta i undervisinga. Grunna omfanget av denne oppgåva vil eg presentera eit lite utval her, dei opplegga som var mest ulike.

4.3.1 Tangram

Maria trekk, som tidlegare nemnt, fram eit opplegg knytt til tangram som ho og ein kollega har utvikla gjennom vidareutdanningsstudiet (Vedlegg 4). Tangram er eit sett geometriske figurar som til saman kan danne eit kvadrat. Gjennom å setja dei saman på ulike måtar kan ein laga mange typar figurar av dette, nokre døme på dette ser ein i Figur 9.



Figur 9 - Del av tangramoppgåva

Oppgåva Maria har laga handlar om at to elevar som sit med ryggen til kvarandre skal veksla på å laga og forklara figurar for den andre. Elev A skal altså, etter instruks frå elev B, leggja brikkar for å laga sin kopi av elev B sin figur. Tanken til Maria er at elevane skal få trening i å gje så presise og tydelege forklaringar at figurane vert like.

4.3.2 Lese kode

Tove har laga eit opplegg om å lese kode som ho har nytta i arbeidet med algoritmisk tenking. Opplegget handlar om å sjå samanhengar mellom Python-kode og output frå programmet.

d) Kva gjer programmet under? Kva trur de skjer om me bytter ut $t=0$ med $t=1$?

```
1 t=0
2
3 while t<20:
4     t=t+2
5     print(t)
```

Figur 10 - Lese kode

I Figur 10 er ei av oppgåvene frå oppgåvearket. Oppgåvene omhandlar ulike *while*-løkker som i figuren over, samt ulike *for*-løkker. I ein del av oppgåvene er all koden gitt i oppgåva, det vil seie elevane får skjermbilete av ferdige program, samt skjermbilete av dei respektive programma sin output. I andre, som i biletet over, så er det opp til elevane å finna ut kva som vert output når ein køyrer programmet.

4.3.3 Likningssett

Lars har gjeve elevane ei oppgåve som handlar om løysing av likningssett. Oppgåva er utvikla av NDLA⁶. Gjennom oppgåva vert elevane leia gjennom ulike steg, i form av deloppgåver. Først vert elevane gjevne ei diskusjonsoppgåve, der dei skal analysere kva for informasjon som er naudsynt for å løysa problemet. Vidare handlar oppgåvene om å stilla opp problemet generelt, ved å nytta bokstavar, før ein skal utforma ein algoritme og deretter koda denne. Når programmet er koda skal det testast, og ein kan leggja til funksjonalitet for likningssett som ikkje let seg løysa på denne måten.

I tillegg har Lars laga sine egne forslag til utviding av programma.

Utvidelser

- Hva hvis systemet har ingen løsninger?
- Hva hvis systemet har uendelig mange løsninger?

Figur 11 - Lars sine utvidingar til oppgåva om likningssett

Elevane får òg spørsmålet «Hva har jeg lært av dette, og hva vil jeg gjøre annerledes neste gang?».

⁶ ndla.no

5 Diskusjon

Målet med denne oppgåva er å svara på problemstillinga *Korleis kan programmering nyttast som didaktisk verktøy i matematikkundervisinga?* I dette kapittelet vil eg drøfta funna mine i lys av den tidlegare presenterte teorien. Eg vil dela inn kapittelet i to hovuddelar, knytt til dei to forskingsspørsmåla eg har arbeidd ut frå. Grunna omfanget til oppgåva vil eg ikkje ha med lange og utfyllande utsegner frå lærarane i dette kapittelet.

5.1 Utvikling av kompetanse

Det første forskingsspørsmålet eg prøver å svara på i denne oppgåva, handlar om utvikling av kompetanse hjå lærarar: *Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?* I det følgjande vil eg sjå litt på refleksjonane som lærarane gjer seg knytt til kompetansen sin. Alle dei fem lærarane som har vore informantar i studien min, har undervist i den fornya varianten av matematikk 1T i år. Dette har medført at det har vore læreplanfesta at elevane skal arbeida med programmering og algoritmisk tenking. Lærarane har med andre ord måtta ta i bruk algoritmisk tenking og programmering i undervisinga dette året. Som det kom fram av intervjuet har heller ikkje elevane noko særskild erfaring med temaet, med unntak av somme som har hatt engasjerte og interesserte lærarar på ungdomsskulen eller programmering som valfag.

5.1.1 Kurs og vidareutdanning

Tre av lærarane i studien har teke vidareutdanning som ein del av førebuinga til implementering av algoritmisk tenking og programmering i praksisen sin. Tove trekkjer fram at ho syns det var for mykje fokus på grunnleggjande programmering i emnet og ikkje korleis ho kunne nytta det i undervisinga. Samstundes kjem det fram at studiet har fått ho til å reflektera rundt nytteverdien av programmering i undervisinga og fått ho til å spørja seg sjølv om «kvifor er dette bra for elevane våre?». Vidareutdanninga kan tenkjastst å vera ein faktor i det eksterne domenet i endringsmiljømodellen til Clarke og Hollingsworth (2002) (Kap. 2.4). Gjennom stimuli frå kurset gjer Tove seg refleksjonar som kan vera med på å endra overtydinga (*beliefs*) hennar og føra til endringar i det personlege domenet. Tove trekkjer òg fram at ho ikkje er utlært og at ho har meldt seg

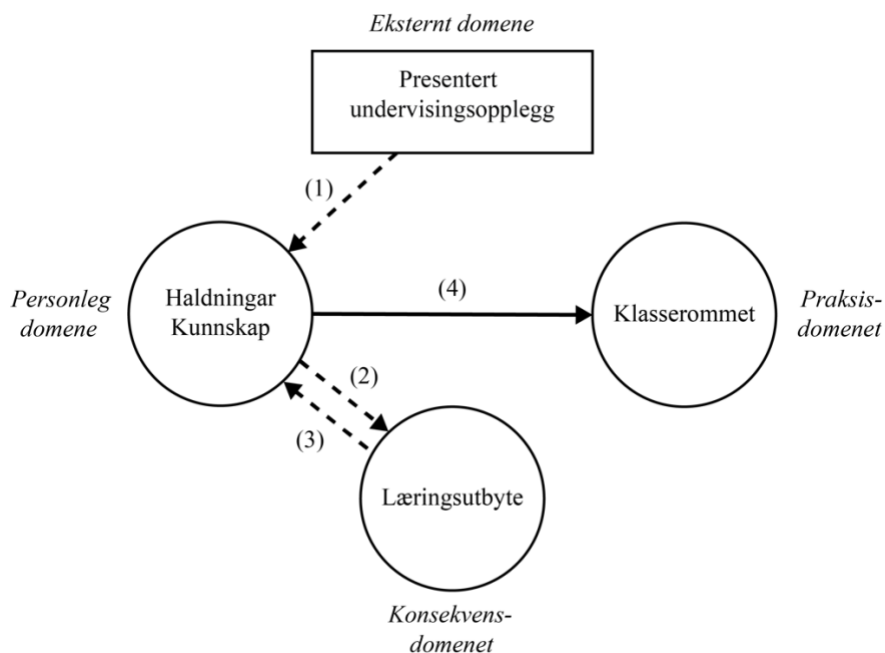
på eit nytt kurs kommande skuleår. Håpet for dette kurset er at det skal fokusera meir på sjølve undervisinga. Det kan sjå ut som kompetansen Tove har utvikla gjennom kurset omhandlar grunnleggjande programmering. Likevel ser ikkje Tove ut til å oppleve at dette gjev ho den didaktiske kompetansen ho treng til undervisinga si. Sett opp mot endringsmiljømodellen, kan det tenkjast at Tove ønskjer at vidareutdanninga skal kunna føra til handling (*enactment*) i praksisdomenet. Samstundes kan ein spørja seg om det som vert undervist i vidareutdanningskursa for lærarar er døme på eksemplarisk undervising. Kanskje kan det vera ein føremon, at kursa fører til refleksjonar som dei Tove trekkjer fram frå kurset ho *har* teke. Maria har teke vidareutdanning ved ein anna institusjon enn Tove, og ho peikar òg på denne type refleksjonar: «*korleis kan me utnytte det i ein pedagogisk fornuftig måte?*». Gjennom denne type refleksjonar kan ein tenkja at lærarane opplever endring i det personlege domenet. Det kan sjå ut som haldningane til programmering og algoritmisk tenking kan verta påverka, noko som igjen kan føra til endring i dei andre domena i modellen.

Maria laga eit undervisingsopplegg (tangram-oppgåva, kap. 4.3.1; vedlegg 4) som ein del av sluttvurderinga i vidareutdanninga hennar. Dette opplegget vart gjennomført i alle 1T-gruppene ved skulen hennar i starten av året. På denne måten har det vore ei direkte kopling mellom det eksterne domenet (studiet) og praksisdomenet (klasserommet). Gjennom å prøva ut dette opplegget i praksisen har Maria kunna gjera seg refleksjonar rundt gjennomføringa. Ho trekkjer fram at ho gjennom året har kunna ta dette opplegget fram når elevane påstår at dei ikkje veit kva algoritmar er for noko.

Noko som er interessant er at oppgåva som Maria laga i samband med vidareutdanninga omhandlar algoritmisk tenking utan bruk av programmering. Som forskinga til Costa et al. (2017, s. 5) viste kan trening i algoritmisk tenking utan bruk av programmering ha god effekt hjå elevane. På same tid er det fleire av lærarane i studien min som trekkjer fram at dei ikkje har fokusert på algoritmisk tenking, men på programmering i undervisinga si. Det kan sjå ut som om påverknaden frå vidareutdanninga har vore med på å gje Maria ei forståing av at algoritmisk tenking handlar om meir enn programmering. Å trena på algoritmisk tenking utan programmering, heng tett saman med det Voogt et al. (2015, s. 718) seier om at det å fokusera på programmering som den einaste måten å læra algoritmisk tenking ikkje har noko for seg. Oppgåva inneheld

òg element av abstraksjon som m.a. Grover og Pea (2013, s. 39) trekkjer fram som hjørnesteinen i utvikling av algoritmisk tenking.

Ein kan og sjå andre koplingar mellom det personlege domenet og det eksterne domenet hjå lærarane, når det kjem til vidareutdanning. Innføringa av den nye læreplanen kan ein òg sjå på som ei stor endring for lærarane. Helge gjorde seg refleksjonar om at han ikkje var budd på ny læreplan (kunnskap i det personlege domenet), og ut frå dette tok han eit val om å ta vidareutdanning (*enactment*, eksternt domene). Dette kan igjen ha ført til refleksjonar kring eigne oppfatningar og kunnskap knytt til algoritmisk tenking og programmering.



Figur 12 - Moglege samanhengar mellom refleksjon og handling hjå Lars

Andre kurs enn formell vidareutdanning kan òg verka som påverknad i det eksterne domenet. Lars har vore på kurs og fått presentert eit undervisningsopplegg som han reflekter rundt. Opplegget handlar om at elevane får relativ avanserte program der dei skal fylla ut det som dei har kompetanse til, og strukturar dei ikkje kan laga på eiga hand er gitt på førehand. Refleksjonane til Lars vert påverka av hans oppfatningar og haldningar frå det personlege domenet. I Figur 12 har eg skissert ein mogleg samanheng mellom refleksjon (stipla piler) og handling, *enactment* (heiltrukken pil). Han ser at læringsutbytet kan vera positivt i den forstand at programma kan bli «kulare» og at ein

kan visa nytteverdien. Samstundes tenkjer han at det er betre om elevane skal laga program som er på eit nivå der dei kan laga alt sjølv. Dette ser me att i det som skjer i klasserommet hans. I ei av oppgåvene Lars har nytta på elevane sine (Kap. 4.3.3), ser me at elevane skal laga heile programmet på eiga hand. I tillegg må elevane utforma algoritmen sjølv, noko ein går glipp av om dei får ferdige løkker og kodestruktur. På denne måten kan det tenkjast at kurset har vore med på å få Lars til å reflektera over eigen praksis på ein måte som gjer at undervisinga hans legg opp til algoritmisk tenking i større grad enn tidlegare. Det kan sjå ut som at kurset har fått Lars til å utvikla den didaktiske kompetansen sin ved at refleksjonane han har gjort seg gjennom kurset har ført til endringar i klasserommet.

Funna her peikar i retning av at kurs og vidareutdanning, i tillegg til å leggja til rette for utvikling av programmeringskompetanse, legg til rette for utvikling av didaktisk kompetanse. Både utsegnene til Lars og Maria viser at dei har gjort seg didaktiske refleksjonar som konsekvens av deltaking i kurs og vidareutdanning.

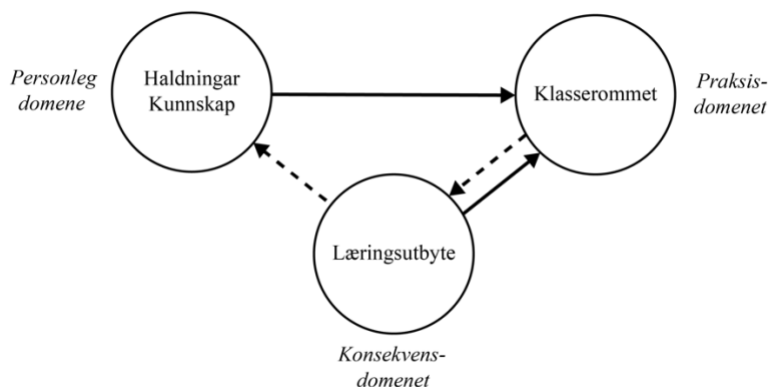
5.1.2 Klasserommet som testarena

Funna i studien viser òg at det som skjer i klasserommet er ein arena som opnar for kompetanseutvikling og refleksjonar rundt dette, både gjennom undervisinga (praksisdomenet) og utbyttet av undervisinga (konsekvensdomenet).

Lærarane trekkjer fram at implementeringa av dei nye elementa i læreplanen har vore, som Lars sa det, «litt famlande». Torrey nytta «prøving og feiling» for skildra eigen undervising med programmering og algoritmisk tenking. Dette er det same me finn at i måten Torrey legg til rette for løysing av problem hjå elevane. Her trekkjer han fram at han «...prøvar å lære dei at det å gjere feil ikkje berre er lov, men naudsynt» (Torrey). Torrey sin praksis modellerer altså ein slik haldning han ønskjer at elevane skal ha, der det å prøva seg fram er høgt verdsett og det er rom for feiling. Det kjem ikkje fram at dette er ei bevisst handling av Torrey.

Ser me denne type framgangsmåte opp mot endringsmiljømodellen kan det tenkjast at ein får ulike verknader mellom dei ulike domena. Ein prøver gjerne ut eit opplegg, reflekterer over læringsutbyttet og konsekvensane av opplegget. Denne refleksjonen kan igjen føra til endringar i undervisinga i klasserommet. Det kan òg tenkjast at

refleksjonar knytt til evaluering av læringsutbyttet av undervisinga kan få innverknad på det personlege domenet. Lærarane kan få endra oppfatning om kva type aktivitetar og oppgåver som kan ha noko føre seg, når ein skal arbeida med algoritmisk tenking, eller berre med programmering. Dei aktuelle sløyfene er freista visualisert i Figur 13.



Figur 13 - Samanheng mellom klasserom, læringsutbyte og egne oppfatningar

Tove sitt opplegg knytt til det å lesa kode, kan vera eit døme på dette. Ho har reflektert rundt kompetansen sin og kompetansen til elevane, samt kva læringsutbyttet til elevane er. Dette har igjen ført til at ho har laga oppgåva med lesing av programkode, som følgje av at elevane ikkje er der at dei kan laga program på eiga hand. Ho reflekterer rundt utbyttet av dette opplegget, at det har positive sider, som kanskje gjer at dette er eit undervisningsopplegg som ho vidareutviklar og tek attende til klasserommet.

Av funna kjem det òg fram at både Maria og Lars er opptekne av å ha ei utforskande tilnærming til matematikkundervisinga. Oppgåvene og spørsmåla dei stiller heng ikkje saman med programmering direkte. Lars nemner likevel at dei jobbar både med å bryta ned problemet i mindre bitar og at dei startar enkelt og gjer det meir og meir avansert. Han tenkjer at dette er ein måte å nytta algoritmisk tenking på. Det han trekkjer fram som døme, er tett knytt til omgrep som *dekomponering* og *abstrahering*, som me kjenner att frå definisjonen av algoritmisk tenking (Csizmadia et al., 2015; Utdanningsdirektoratet, 2019a). Trass i manglande kompetanse hjå elevane og lærarane gjev denne type oppgåver mogleiken til å nytta algoritmisk tenking i undervisinga. Dette er noko ein går glipp av ved å fokusera på programmering som einaste inngangsport til algoritmisk tenking.

Helge trekkjer fram at elevane er gode på å stilla kritiske spørsmål om kva dei «skal med dette». Dette gjer at Helge vert tvungen til å reflektera kring læringsutbyttet til elevane for å kunna gje dei gode svar. Gjennom refleksjonen kjem det fram at Helge ser på det å jobba med ekte datasett som ein nyttig bruk av programmering. Dette fører igjen til at han nyttar eit opplegg knytt til radioaktivitet i klasserommet sitt. Her ser me korleis refleksjonar, mellom konsekvensdomenet og det personlege domenet, fører til handling i praksisdomenet.

Fleire av lærarane trekkjer fram at nivået til elevane, knytt til algoritmisk tenking og programmering, er så lågt at dei finn det utfordrande å knyta programmeringa saman med matematikken. Tove trekkjer fram dette hjå sine elevar. Ser me på undervisningsopplegget hennar om å lesa kode nemnt over så, handla det om å lesa kode sidan elevane ikkje var i stand til å laga eigne program. Det matematiske innhaldet i programma som skulle lesast handlar mykje om heilt grunnleggjande aritmetikk (Sjå døme i Figur 10). Det er gjerne snakk om addisjon av heiltal, men addisjonen skjer gjennom ulike løkker og er gjerne styrt av ulike vilkår. I matematikkfaget 1T er nivået på matematikken litt høgare enn dette. Dette understøttar påstandane til Tove om at ho syns det er vanskeleg med koplingar mellom matematikken og programmeringa. Som Torrey trakk fram: «(D)et vert litt to ulike verdenar enn så lenge».

5.1.3 Læring saman med andre

Ein arena som kan tenkjast å vera god for utvikling av kompetansen er i samarbeid med andre. Lærarane har litt ulike forteljingar om korleis dei nyttar samarbeid i kvardagen. På den eine sida vert det fortalt om kollegasamarbeid der Maria, som lærar, sit att med ei oppleving av eit einseitig fokus på progresjon og synkronisering av framdrift. Sett opp mot utviklingshjulet (Figur 6) til Irgens (2010, s. 136) fell dette inn under *rom for kollektive driftsoppgåver*. På den andre sida fortel både Lars og Torrey om samarbeid der det didaktiske perspektivet har størst plass. Her vert det mellom anna diskutert korleis ein skal tolka den nye læreplanen, og ein diskuterer korleis ein skal implementera programmering og algoritmisk tenking i undervisinga. Gjennom diskusjonar arbeider ein med å skapa ei felles forståing for den rolla algoritmisk tenking og programmering skal ha, i undervisinga. I tillegg kjem samkøyring og koordinering av drifta inn i dette samarbeidet. Her ser me at ein både har samarbeid knytt til *rom for kollektive driftsoppgåver* og *rom for kollektiv utvikling* (Irgens, 2010, s. 136). For at ein

skule skal vera i utvikling må han ifølgje Irgens (2010) vera i alle dei fire romma i utviklingshjulet. I situasjonen Maria skildrar er det ikkje teikn til at samarbeidet er knytt til *rom for kollektiv utvikling*. Dette kan tyda på at utviklinga av kompetansen her, vert opp til den einskilde læraren. På den andre sida vil eit samarbeid som Lars og Torrey skildrar, leggja opp til at utvikling av kompetanse er noko som skjer i fellesskapet.

Sett opp mot dei fire domena i endringsmiljømodellen kan ein tenkja at samarbeid med didaktiske og faglege diskusjonar med kollegaar fell inn under det eksterne domenet som «External Source of Information or Stimulus» (Clarke & Hollingsworth, 2002, s. 951). Av dei to, ganske ulike, samarbeidsmiljøa som har kome til syne hjå lærarane er det naturleg å tenkja at det er varianten med faglege diskusjonar som fører til mest ytre stimuli. Dette kan då vidare føra til eksperimentering i klasserommet (praksisdomenet), eller endring av haldningar og kunnskap (det personlege domenet). Samstundes vil nok eit samarbeid der ein berre fokuserer på framdrift ha lite føre seg med tanke på vidare endring og utvikling. Maria, som skildrar denne typen samarbeid, etterlyser eit fokus på algoritmisk tenking som ein del av språket i samarbeidet, og kollegaar som set lys på dette. Her vert mellom anna rolla mi som både forskar og lærar trekt fram: at det vert sentralt for meg å jobba mot eit fokus på algoritmisk tenking og programmering i mitt profesjonsfellesskap.

Maria trekkjer fram at vidareutdanninga og den digitale kommunikasjonsplattforma Slack, som dei har brukt der, har vore nyttig for hennar utvikling av kompetanse. Som ho seier har kontakten med 200 medstudentar ført til at det har vorte både stilt gode spørsmål og delt gode ressursar. Her har studiet vore med på å gje læraren høve til å læra av og med kollegaar på andre skular og på andre trinn. Dette kan ein sjå på som ein viktig faktor for stimuli i det eksterne domenet i endringsmiljømodellen. Påverknad her kan igjen føra til refleksjonar som er med på utvikling av kompetanse (på same måte som i Figur 12).

Lærarane reflekterer òg rundt korleis elevane, og samarbeid med dei, spelar inn på kompetanseutviklinga. Både Torrey og Tove ser på det heile som eit felles prosjekt mellom lærar og klasse. Ser ein på utsegnene til Tove, ser ein at ho er ærleg med elevane at dette med programmering i undervisinga er noko dei må finna ut av saman. Når Vygotskiï og Cole (1978, s. 86) trekkjer fram den proksimale utviklingssona som

det elevane klarar med hjelp av ein meir kapabel læringspartnar, kan ein stille spørsmål om korleis dette passar med det Tove fortel. Samstundes som ho sjølv meiner ho har låg kompetanse knytt til programmering og algoritmisk tenking gjer ho seg òg refleksjonar om korleis ho er meir kompetent enn dei lærande (elevane): «Eg skal jo framleis læra dei noko, og eg kan jo det matematiske, sant?» Her vil Tove vera meir kompetent når det kjem til matematikken, medan kompetansen med algoritmisk tenking og programmering er noko dei utviklar i fellesskap. Som Tove trekkjer fram: «Eg kan jo leggja opp til algoritmisk tenking, for eksempel ved hjelp av programmering, sjølv om eg ikkje har løyst det på same måte som dei ender opp med å løysa det på».

5.1.4 Kompetansen til lærarane når elevane sin kompetanse aukar

Noko som er spesielt med dei første åra i Fagfornyinga (inneverande og nokre av dei komande), er at dei aller fleste elevane i vg1 har lite eller inga erfaring med algoritmisk tenking og programmering. Lærarane trekkjer fram at få elevar har vore borti desse elementa tidlegare, alt etter om dei har hatt engasjerte lærarar eller valfag på ungdomsskulen. Fagfornyinga gjeld, i tillegg til vgs, heile grunnskulen. Det vil seie at etterkvart som åra går vil elevane ha erfaring og kunnskap knytt til programmering og algoritmisk tenking. Både Helge og Tove trekkjer fram at dei ser føremonar knytt til at elevane har utvikla grunnleggjande kunnskapar til programmering når dei kjem inn i den vidaregåande skulen.

Når det kjem til refleksjonar rundt eigen kompetanse, knytt til denne utviklinga hjå elevane, har lærarane ganske ulike tankar. Helge er ikkje uroa for framtida. Dette heng saman med at han er av den oppfatninga av at han kan grunnleggjande programmering, og at det han kan er nok. At det er ein mogleik at han får elevar som kan meir enn han tek han med ro. Helge er av den oppfatninga at han klarar å rettleia elevane som eventuelt vert flinkare enn han til å finna informasjon på nettet. På den andre sida finn me Tove og Torrey som er litt uroa med tanke på framtida. Tove reflekterer rundt det ho kjenner på som ein trong til å ta fleire studiepoeng, meir vidareutdanning. Ho kjenner at det er i programmeringa skoen trykkjer. Torrey gjer seg og tankar om eigen kompetanse sett opp mot den komande endringa. Han trekkjer fram at den kompetansen han per i dag har i grunnleggjande programmering ikkje er nok til å ta gode didaktiske val. Her ser me igjen at kompetansen er delt i programmeringskompetanse og didaktisk kompetanse. Det kjem ikkje fram at Torrey har planar om å ta kurs eller

vidareutdanning, men i intervjuet kjem han indirekte inn på effekten vidareutdanning kan ha. Gjennom samarbeid med kollegaar ser han at dei som har teke vidareutdanning og som har meir undervisingserfaring har lettare for å ta dei didaktiske vala han sjølv finn vanskeleg.

Både Lars og Maria trur på si side det skal gå fint å utvikla kompetansen sin i takt med at elevane som kjem inn har meir og meir kjennskap til algoritmisk tenking. Maria grunnjev dette i at kompetansen hennar vil utvikla seg, i takt med at ho får nye elevar som ho må laga tilpassa opplegg til. Igjen kan me sjå føre oss ein gjentakande refleksjonsprosess liknande den i Figur 13. Her vert kompetanseutviklinga ein iterativ prosess som gjentek seg gjennom året med klassen, samt frå skuleår til skuleår med nye klassar.

Begge dei to syna på kompetanse og utvikling av denne indikerer at lærarane kjenner dei har trong for betre kompetanse. Dette gjeld både dei med vidareutdanning og dei utan. Sjølv om nokon av lærarane gjer seg refleksjonar om korleis dei skal forbetra kompetansen er det ei felles kjensle av manglande kompetanse for fire av dei fem lærarane. Eit spørsmål ein kan stilla seg er om lærarane nokon gong kjenner at dei *kan nok* og at kompetansen er god nok. Funna over viser at både Tove, Lars og Torrey som er dei lærarane i studien med kortast fartstid i skulen reflekterer rundt manglande kompetanse og trong for meir. Maria som er ein erfaren lærar tek òg sikte på å måtta utvikla kompetansen sin vidare, og trekkjer fram dette som noko av det som er kjekt med læraryrket. Helge gjev uttrykk for at han ikkje er uroa for at elevane skal verta flinke. Samstundes har han teke vidareutdanning grunna i at han ville liggja litt framfor elevane, når programmering skulle inn i matematikkfaget. Det kan peika i retning av at heller ikkje Helge kjenner på at kompetansen er *god nok*. Lærarane sine refleksjonar peikar i retning av at det ikkje er programmeringskompetanse eller matematisk kompetanse som er den største uroen, men den didaktiske kompetansen.

5.1.5 Kompetanseutvikling knytt til algoritmisk tenking gjennom algoritmisk tenking?

I kapittel 2.3 trakk eg fram eit (matematisk) problem som noko som ikkje kunne løysast ved hjelp av rutineoperasjonar og at det var avhengig av kompetansen til problemløysaren basert på Niss og Høygård Jensen (2002, s. 49). Ser ein på innføringa av programmering og algoritmisk tenking i undervisinga til desse lærarane, finn me

mange likskapar med *problem*-omgrepet. Det er nok ikkje ei oppgåve dei kan løysa ved hjelp av rutineoperasjonar i form av innarbeidd praksis og undervisningsopplegg. Etter å ha sett på dei ulike refleksjonane lærarane gjer seg, og kvar desse refleksjonane har sitt utspring, kan ein sjå att mange av elementa frå algoritmisk tenking i kompetanseutviklinga deira. Det kan sjå ut som om lærarane nyttar algoritmisk tenking for å løyse *problemet* med innføring av programmering og algoritmisk tenking i eigen praksis. Både samarbeid, evaluering, prøving og feiling, starte enkelt og eksperimentering er omgrep me finn att både når det kjem til algoritmisk tenking og i lærarane sine refleksjonar. Om dette er bevisst eller ikkje kjem ikkje tydeleg fram i funna, men det å løysa problem gjennom å nytta algoritmisk tenking er noko dei skal leggja til rette for som ein del av den nye læreplanen.

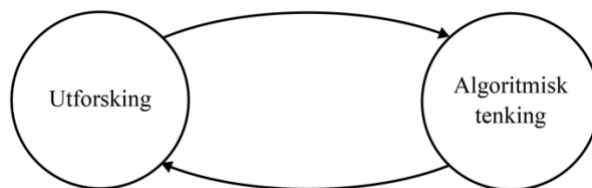
5.2 Løyse problem med algoritmisk tenking?

Det andre forskingsspørsmålet i denne oppgåva er knytt til praksisen til lærarane: *Korleis legg lærarar til rette for å løyse problem ved hjelp av algoritmisk tenking?* I det følgjande vil eg presentera det lærarane fortel om eigen praksis å prøva og setja det i samanheng med problemløysing og algoritmisk tenking.

5.2.1 Utforsking

Som ein ser av kap. 0 står utforsking sentralt for lærarane både når det kjem til *samtalen* og korleis dei *legg til rette for læring* ved hjelp av algoritmisk tenking.

Spørsmåla lærarane bruka i klasserommet har ei vinkling som gjer at elevane må forklara kva dei tenkjer. Maria trekkjer fram at ho legg opp til at elevane får reflektera litt rundt oppgåver dei skal løysa saman, for å finna ut kva oppgåva spør etter. Vidare reflekterer ho rundt at det ikkje er algoritmisk tenking, men utforsking. Samstundes trekkjer ho fram det ho ser som ein samanheng mellom omgrepa. Maria meiner at om ein skal vera god i algoritmisk tenking må ein vera god i utforsking og på same måte må ein jobba med utforsking for å bli god i algoritmisk tenking. Figur 14 viser samanhengen mellom omgrepa som Maria trekkjer fram.



Figur 14 - Samanheng mellom algoritmisk tenking og utforsking?

Tangram-oppgåva (Kap. 4.3.1) eg har samla inn frå Maria, omhandlar algoritmisk tenking, men ikkje direkte utforsking. Ser me tangram-oppgåva opp mot rammeverket til Brennan og Resnick (2012) som vart presentert i kapittel 2.2.6 kan me kjenna att ulike element. Oppgåva kan ein tenkja handlar om det å fjerna detaljar som ikkje er viktige (abstrahering) som ligg under *Computational Thinking Practices*. Samstundes handlar oppgåva i stor grad om å uttrykkja seg, som me finn under *Computational Thinking Perspectives*. Desse to nivåa av algoritmisk tenking er på eit høgare kognitivt nivå enn *Computational Thinking Concepts*. Brennan og Resnick (2012, s. 7) peikar på at *CT Perspectives* handlar om *korleis* ein lærer framfor *kva* ein lærer. På same måte handlar *CT Perspectives* om korleis eit utviklar forståing av seg sjølv samt relasjonane med andre og verda rundt (s. 10). Oppgåva legg altså opp til at elevane skal nytta element frå algoritmisk tenking. Likevel er det ikkje tydeleg at det er eit (matematisk) problem som skal løysast, då elevane veit ein del om løysinga: dei skal kommunisera brikkene dei legg til kvarandre. Det kan tenkjast at dette har samanheng med oppfatninga Maria har av omgrepet algoritmisk tenking. Kanskje ser ho på algoritmisk tenking som *digital tenking som programmeringskompetanse* som Kluge (2020, s. 123) peikar på. På den andre sida trekkjer Maria fram ein bruk av utforskande oppgåver (kap. 4.2.2) der algoritmisk tenking kan vera eit verktøy for å finna ut av problemet. Lars har, på si side, ei anna oppfatning av koplinga mellom algoritmisk tenking og utforsking. Han tenkjer at algoritmisk tenking er noko med eit meir *definert mål* enn utforsking. Ei årsak til dette kan vera ei forståing av algoritmisk tenking som programmeringskompetanse. Då kan ein tenkja at målet med algoritmisk tenking (forstått som programmering) handlar om å koma fram til eitt bestemt program.

Dei åtte aspekta som Pennant (2013) trekkjer fram handlar om klasseromsamtalen, læraren og elevane. Ein klasseromskultur som legg til rette for problemløysing ber preg av at elevane snakkar meir enn læraren, at ein ikkje stiller lukka spørsmål, at det er god spreing i kven som svarar på spørsmål, at ein får tydeleg fram elevane sine refleksjonar

og poeng, og at elevane torer å prøva seg fram og ikkje er redde for å gjera feil, for å nemna nokre (Pennant, 2013). Dette er knytt til problemløysing generelt, og ikkje algoritmisk tenking spesielt. Likevel ber undervisinga til lærarane preg av dette, når dei fortel om korleis dei arbeider med algoritmisk tenking. Spørsmål som «korleis har du tenkt», «kvifor har du tenkt som du gjer», «Åja, det er sånn ja... Korleis vert det då? Kva betyr eigentleg det?» ligg tett opp til den presenterte klasseromskulturen.

Det kjem fram at elevane kanskje ikkje er vande med å jobba så mykje med utforsking. Tove trur dette heng saman med at matematikkundervisinga er tunggrodd, og at elevane er vande med å få oppgåver som er like kvarandre. Då er både løysingsmetode og *målet* med oppgåva tydeleg for elevane utan at dei treng å tenkja seg for mykje om. Maria peikar på at oppgåver i undervising og eksamen med ei rekkje deloppgåver som gradvis tek ein mot svaret er noko av bakgrunnen for at elevane ikkje er så gode på utforsking og nye typar oppgåver. Ho trekkjer fram at når ein tek vekk alle deloppgåvene må elevane sjølv dela opp problemet i handterbare delar. Dette kjenner me att som *dekomponering* som er eit av nøkkelomgrepa når det kjem til algoritmisk tenking (Utdanningsdirektoratet, 2019a). Helge tenkjer utfordringa ligg i at «[elevane] er vande med å få algoritmen og ikkje utarbeida han sjølv».

Det går att hjå dei fleste lærarane at elevane ofte har lett for å gje opp når dei står fast, og mange er redde for å gjera feil. Maria reflekterer rundt dette: «Eg innser at det er ein inngangport her, problemet er at eg ikkje har opna den døra heilt». Både det å gjera feil og det å halda ut er nemnt som arbeidsmåtar som kjem inn under algoritmisk tenking (Utdanningsdirektoratet, 2019a). Sjølv om lærarane trekkjer fram elevar som ikkje torer å gjera feil så er dette noko dei prøver å skapa ein kultur for. Torrey trekkjer *feil* fram som heilt essensielt i sin praksis. Dette kan ein tenkja handlar om å endra kultur frå *matematikk-klasserom* til *data-sal* som Sfard og Leron (1996, s. 190) er inne på, der det er rom for å bruka lang tid på ei oppgåve, og at feil og feilsøking er ein naturleg del av læringsprosessen.

Lærarane er òg oppteken av samarbeid mellom elevar, og at ein får fram fleire ulike refleksjonar rundt same problemet. Lars fortel om at han samlar inn og ser på elevane sine arbeid og har dette som utgangspunkt i den styrte klassesamtalen i etterkant. Då kan han sjå kva som er strategiske rekkefølger dei ulike gruppene får dela arbeidet sitt

med resten. Dette gjorde han m.a. når elevane arbeidde med likningssett-oppgåva som var tema i kap. 4.3.3.

Mykje av det lærarane trekkjer fram frå eigen praksis som kan knytast til problemløysing og elementa frå algoritmisk tenking, er felles for alle lærarane. Noko som gjekk att i refleksjonane rundt praksis var at samtalen fort dreidde over til å handla om matematikkundervisinga generelt, og ikkje direkte knytt til algoritmisk tenking eller programmering. Dette kan ha samanheng med intervju spørsmåla sine formuleringar. Kanskje kunne dei vore annleis, og fått fram andre refleksjonar. Det kan òg tenkjast at moderne matematikkundervising inneheld mange av elementa frå problemløysing og utforsking. Som tidlegare nemnt er det sosiokulturelle aspektet med *læring som deltaking* sentralt i matematikkdiraktikken, og problemløysing var stort allereie i den norske skulen på 80-talet. Det kan tenkjast at desse elementa har vore med på å gjera matematikkundervisinga ganske lik det ein finn att i algoritmisk tenking. Konsekvensen av dette kan vera at det, i funna mine, vert vanskeleg å skilja mellom undervising med fokus på *algoritmisk tenking* og den *vanlege* matematikkundervisinga.

5.2.2 Programmering og problemløysing?

Utfordringa som er nemnt over er ikkje like aktuell om ein ser på korleis lærarane nyttar programmering i undervisinga si, og på den måten legg opp til å løysa problem gjennom algoritmisk tenking. Det er kome fram i funna at lærarane har ganske ulike måtar å nytta programmering på i undervisinga. Når Tove går gjennom opplegget sitt der elevane skal lesa kode, er det lite samanheng med problemløysing og dei ulike nøkkelomgrepa og arbeidsmåtene i definisjonen av algoritmisk tenking. Ser me opplegget opp mot rammeverket for Computational Thinking, så er det i hovudsak fokus på *CT Concepts* i desse oppgåvene. Det er forståing av løkker og grunnleggjande omgrep innan programmering som er målet for elevane.

Læreplanen legg, som nemnt tidlegare i oppgåva, opp til at elevane skal «...formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering» (Utdanningsdirektoratet, 2019b). Når ein legg opp til å lese kode og tolka den, kan ein stilla seg spørsmål om det er snakk om å løysa problem. Det same gjeld når ein, som Torrey trekkjer fram, fokuserer på heilt grunnleggjande programmering som det han gjer i undervisinga. Dette fører til at

lærarane føler dei står i ein spagat mellom programmering og problemløysing. Elevane kan ikkje tilstrekkeleg programmering til at lærarane ser at dei kan ha nytte av det til problemløysing.

På den andre sida fortel Lars om dei gode erfaringane han har med å la elevane nytta programmering som valfritt verktøy og hjelpemiddel i problemløysingsoppgåver. Dette finn me òg att i oppgåva om likningssettet samla inn frå Lars (Kap. 4.3.3). Sjølv om opplegget presentert der ikkje lar programmering vera valfritt verktøy, så er programmeringa ein del av ei større problemløysingsoppgåve. Det er ikkje nokon grunn til å tru at elevane til Lars, som gruppe, har andre føresetnader knytt til programmering (frå ungdomsskulen) enn Tove og Torrey sine elevar, då alle desse tre lærarane arbeider ved ulike skular i ulike fylkeskommunar i landet. I oppgåva om løysing av likningssett finn me element som er tett knytt til algoritmisk tenking. Elevane skal sjølv koma fram til algoritmen ved å sjå etter samanhengar og generelle uttrykk, før dei seinare skal programmere algoritmen og testa han. I tillegg har Lars utvida med spørsmål som opnar for evaluering og refleksjon. Sett opp mot rammeverket for Computational Thinking så legg oppgåva opp til aktivitetar som kan knytast til fleire nivå. Den grunnleggjande kodinga heng gjerne saman med *CT concepts*, medan feilsøking og innsamling av informasjon gjerne er knytt opp til *CT practices*. Oppgåva legg opp til å nytta det store fleirtalet av nøkkelomgrep og arbeidsmåtar i definisjonen av algoritmisk tenking. Dette heng saman med det Weintrop et al. (2016, s. 139) trekkjer fram, om at elevane kan ha god nytte av programmering i undervisinga utan å vera ekspertar.

Dette skiljet mellom oppfatninga lærarane har av koplinga mellom programmering og problemløysing er interessant. På den eine sida fortel lærarar om at elevane ikkje kan nok til å nytta programmering som verktøy til problemløysing, medan Lars på si side argumenterer for at han ikkje vil nytta vanskelegare kode enn elevane skal kunna skriva sjølv (kap. 4.1.1), og fortel om god effekt av bruk av programmering til problemløysing.

5.3 Kompetanse og praksis

Tidlegare forskning peikte på lærarane sin kompetanse, eller manglande erfaringar med å undervisa i programmering, som moglege årsaker til at elevane ikkje utvikla ferdigheiter på eit høgare nivå enn grunnleggjande programmering (Nouri et al., 2020,

s. 12). Dette har ein tenkja har samanheng med korleis programmering vert nytta som didaktisk verktøy i undervisinga. Funna frå studien min viser at det er Lars, utan relevant vidareutdanning, som legg mest til rette for problemløysing gjennom programmering og algoritmisk tenking. Tove, som har vidareutdanning, legg meir opp til opplæring i grunnleggjande ferdigheiter knytt til programmering. Det har òg kome fram gjennom intervjua at Lars er van med å arbeida utforskande frå tidlegare, medan Tove trekkjer fram dette som noko ho finn utfordrande. Kanskje kan dette ha ein verknad på korleis dei sjølve nyttar programmering og algoritmisk tenking i undervisinga si.

I funna kjem det òg fram at kompetansen til lærarane i denne samanhengen har fleire sider. For å nytta programmering som didaktisk verktøy i matematikken treng ein både matematisk kompetanse og programmeringskompetanse (Figur 15).



Figur 15 - Den didaktiske kompetansen

Refleksjonane til lærarane peikar i retning av at det ikkje er eit problem å utvikla programmeringskompetansen sin. Alle lærarane gjev uttrykk for at dei har ein del grunnleggjande kompetanse i programmering. Dei fem lærarane har òg utdanning som kvalifiserer til å vera matematikklærarar, så det er grunn til å tru at den matematiske kompetansen er god nok. Det fleire av lærarane i studien trekkjer fram som utfordrande er det å kopla programmeringa og matematikken saman. Dette finn me att både i utsegner frå intervjua og i måten det vert lagt til rette for å nytta programmering og algoritmisk tenking i undervisinga.

6 Avslutning

Gjennom denne oppgåva har eg freista å få eit innblikk i noko av det som er nytt i matematikkfaget i Fagfornyinga. Der er algoritmisk tenking og programmering sentrale omgrep. Problemstillinga som har vore leiande for arbeidet har vore:

P: Korleis kan programmering nyttast som didaktisk verktøy i matematikkundervisinga?

I dette kapitlet vil eg freista å gje ei oppsummering av funna eg har gjort, samt trekkja fram kva for implikasjonar og avgrensingar denne studien har. Avslutningsvis vil eg sjå på kva som er vegen vidare etter denne oppgåva.

6.1 Konklusjon

Forskingsspørsmåla som er freista svart på gjennom oppgåva har vore:

F1: Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?

F2: Korleis legg lærarar til rette for å løysa problem ved hjelp av algoritmisk tenking?

I det følgjande vil eg oppsummera funna mine til dei to forskingsspørsmåla.

6.1.1 Kompetanseutvikling knytt til algoritmisk tenking

Gjennom å nytta endringsmiljømodellen til Clarke og Hollingsworth (2002) som utgangspunkt, har eg sett nærare på refleksjonane lærarane gjer seg kring utvikling av kompetanse knytt til algoritmisk tenking. Det kan sjå ut som at lærarane både ønskjer og treng å utvikla den didaktiske kompetansen sin. Sjølv om lærarane har god matematisk kompetanse og grunnleggjande programmeringskompetanse, vitnar funna mine om at lærarane meiner dei ikkje *kan nok* for å nytta programmering og algoritmisk tenking i undervisinga si. Funna indikerer at kurs og vidareutdanning kan opna for refleksjonar som kan føra til endringar i praksis, eller til endra haldningar og

oppfatningar. Det kan òg sjå ut som klasserommet fungerer som ein test-arena for lærarane, og at det som skjer der kan føra til refleksjonar knytt både til læringsutbyttet til elevane og haldningane og kunnskapen til lærarane. Her kan ein sjå at lærarane kan utvikla kompetansen sin iterativt gjennom fleire rundar i denne sløyfa av refleksjon og endra eller oppdatert praksis. Desse prosessane og refleksjonane kan igjen vera med på å auka den didaktiske kompetansen.

Studien har òg vist at ulike typar samarbeid i profesjonsfelleskapet kan vera avgjerande for utvikling av kompetanse. Eit samarbeid som fokuserer på framdrift og progresjon vil ikkje gje læraren mykje ekstern stimuli, sett opp mot eit samarbeid som fokuserer på felles forståing av læreplan, og didaktiske val knytt til implementering av programmering og algoritmisk tenking i matematikkundervisinga. Sett opp mot *kompetanshjulet* til Irgens (2010, s. 136) er det viktig at begge desse formene for samarbeid er til stades for utvikling. Funna i studien min viser at eit einseitig fokus på framdrift ikkje er med på å leggja til rette for utvikling av didaktisk kompetanse. Funna mine viser òg at ein kan få eit utvida nettverk til å samarbeida med gjennom å delta på vidareutdanning og kurs.

Med tanke på utviklinga ein vil sjå hjå elevane som kjem inn i den vidaregåande skulen dei komande åra, har dei fem lærarane vist at det er mange ulike måtar å reflektera rundt dette på. Nokre av lærarane tenkjer kompetansen ikkje er god nok, og uroar seg litt, med eller utan planar og tankar om korleis han kan utviklast. Andre tenkjer at grunnleggjande programmering er god nok kompetanse, så får elevane bli så gode som dei vil. Den siste varianten er lærarar som tenkjer at kompetanseutvikling er noko som vil skje parallelt (og nesten automatisk) med at elevane sin kompetanse aukar, og undervisinga skal tilpassast elevane sitt nivå.

Mykje av det som kjenneteiknar lærarane sitt arbeid med å utvikla kompetansen sin knytt til algoritmisk tenking, kan minna om arbeidsmetodar og omgrep knytt til nettopp algoritmisk tenking. Utforsking og eksperimentering, evaluering, samarbeid, dekomponering er nokre av desse elementa ein ser att i kompetanseutviklinga. Det kjem ikkje klårt fram om det er eit bevisst val at lærarane løyser *problemet* med å implementera algoritmisk tenking gjennom å bruka *algoritmisk tenking*, men funna mine tyder på at dette er noko som skjer.

6.1.2 Løse problem ved hjelp av algoritmisk tenking

Når det kjem til å leggja til rette for å løysa problem ved hjelp av algoritmisk tenking viser funna mine at lærarane gjer dette ved å nytta, eller forsøka å nytta, ei utforskande tilnærming til matematikken. Lærarane legg til rette for ein klasseromskultur som støtter problemløysing, om ein nyttar Pennant (2013) sine aspekt for å vurdere dette. Likevel er det mykje av matematikkundervisinga til lærarane som ein kan tenkja er lik som tidlegare, men at undervisinga passar med ein del av dei omgrepa og arbeidsmåtane me finn i algoritmisk tenking. Når det kjem til å bruka programmering som verktøy, for å bruka algoritmisk tenking som problemløysingsstrategi, viser funna at dette skjer med vekslende hell. Fleire av lærarane syns det er utfordrande å få til gode koplingar mellom programmering og algoritmisk tenking, og sjølve matematikken. Elevane sin manglande kompetanse får noko av skulda for den utfordringa. Dette fører til at nokre av lærarane berre arbeider med grunnleggjande programmering, det som fell inn under Computational Thinking Concepts i rammeverket til Brennan og Resnick (2012, s. 5). Andre lærarar legg opp til undervising med oppgåver der ein nyttar om lag alle elementa frå definisjonen av algoritmisk tenking ein finn i Fagfornyninga (Figur 1).

Det er vanskeleg å seie kva variasjonane i utvikling og refleksjon rundt kompetanse, samt praksisen, til lærarane skuldast. Lærarane har nok ulike erfaringar, og ulike haldningar, som er med på å prega dei didaktiske vala dei gjer. Kanskje kan noko skuldast ei manglande forståing av omgrepet *algoritmisk tenking*. Inntrykket etter fleire av intervjua mine var at dei språklege utfordringane med omgrepet gjer at det er *algorithmical thinking*, eller algoritmisk tenking som *programmeringskompetanse*, som vert forstått av ein del lærarar, ikkje *Computational Thinking* som Utdanningsdirektoratet har tenkt er det som skal inn i matematikkfaga.

Korleis lærarane nyttar *programmering som didaktisk verktøy* handlar om korleis dei legg opp til at elevane skal bruka det for læring. Funna mine viser at dette vert gjort på to ulike måtar: 1) programmering som ein separat del i undervisinga med fokus på å lese og forstå kode (og programmeringsspråk) og 2) enkel programmering som ein del av større utforskande matematiske oppgåver. Eg ser ingen klår samanheng mellom programmeringskompetansen, eller den matematiske kompetansen, og korleis lærarane legg til rette for å løse problem ved å nytta algoritmisk tenking og programmering, i undervisinga si. Samstundes er det grunn til å tru at den didaktiske kompetansen til

lærarane er viktig når det kjem til det å kopla saman matematikken og programmeringa. Eg har vist at den didaktiske kompetansen kan utvikla seg gjennom ulike refleksjonar knytt til både praksis og læringsutbytte, samt kurs og vidareutdanning. Samarbeid og utvikling i eit profesjonelt fellesskap ser òg ut til å leggja til rette for refleksjonar som kan vera med på å utvikla den didaktiske kompetansen til lærarane.

6.2 Implikasjonar og avgrensingar

Denne oppgåva viser at det er ulike måtar ein kan reflektera rundt utvikling av kompetanse knytt til algoritmisk tenking, som lærar. Oppgåva har vist døme på ulike arenaer som kan gje refleksjonar som er med på å utvikla kompetanse, både i kollegasamarbeid og knytt til klasserommet og vidareutdanning. Gjennom oppgåva har eg òg presentert, og reflektert rundt, ulike måtar ein kan nytta algoritmisk tenking og programmering i undervisinga, og freista å kopla dette opp mot kompetansen til lærarane.

Det er mykje som kan vera spennande å sjå vidare på. Studien min har vist at det er litt uvisst kva oppfatning lærarane har av omgrepet algoritmisk tenking. Eg meiner at ein må sjå nærare på korleis norske matematikklærarar oppfattar omgrepet, og om dette påverkar praksisen deira. Om det viser seg at eit fleirtal av lærarane ikkje har forståing for kva som ligg i dette omgrepet kan ein tenkja at det igjen i ytste konsekvens kan gå ut over læringsutbyttet til elevane.

Tidlegare forskning viste at dei svenske lærarane ikkje hadde nok erfaring eller kompetanse (Nouri et al., 2020, s. 12). I tillegg viste tidlegare forskning at lærarane sine didaktiske prinsipp og pedagogiske intensjonar var avgjerande for elevane sitt utbytte (Misfeldt & Ejsing-Duun, 2015; Popat & Starkey, 2019, s. 368). Funna mine viser at kompetansen til lærarane, og då især den didaktiske kompetansen mellom anna kan utviklast gjennom samarbeid som fokuserer på undervising og didaktiske val, samt kurs og vidareutdanning. Det er trong for å forska meir på korleis norske skuleleiarar og skuleeigarar legg til rette for utvikling av didaktisk kompetanse på skulane sine.

Det vil òg vera av interesse å sjå meir på korleis lærarane sin kompetanse og praksis utviklar seg, etterkvart som elevane får eit betre kompetansegrunnlag knytt til

algoritmisk tenking og programmering frå grunnskulen. Ikkje minst vil det vera interessant å sjå på kva rolle algoritmisk tenking og programmering har i matematikkfaga i den vidaregåande skulen nokre år fram i tid!

Ein studie som denne har sine avgrensingar. Eg har gjort eit lite djupdykk i fem lærarar sin praksis. Det er ikkje grunnlag for å seie at deira erfaringar og refleksjonar er representative for alle matematikklærarane i den vidaregåande skulen. Sjølv om det ikkje er grunnlag for ei slik generalisering meiner eg at oppgåva legg til rette for naturalistisk generalisering, som vil seie at lesaren kan kjenna att sin eigen situasjon og sjå overføringsverdi til eigen praksis (Postholm, 2010, s. 131). Dermed kan lærarar nytta denne oppgåva for å sjå om det er refleksjonar eller døme på praksisar dei kan ta med seg attende til sine egne klasserom og profesjonsfellesskap.

Ei masteroppgåve som dette har og eit avgrensa omfang både i tid og i tema ein kan sjå på. I intervjuguiden hadde eg spørsmål som gjekk på andre tilhøve som t.d. støtte frå leiinga knytt til kompetanseutvikling. I dataanalysen såg eg tidleg at det var optimistisk å tru at eg kunne ha fokus på dette og. Her kjem gjerne mine manglande erfaringar frå den kvalitative forskinga inn som ein avgrensande faktor. Hadde eg vore meir erfaren ville eg nok hatt eit enno tydelegare fokus i datainnsamlinga, og betre kjennskap til kor mykje eg kunne gapa over. Det vert nok lettare neste gong.

6.3 Veggen vidare

Dei **personlege måla** mine med denne oppgåva var knytt til jobben eg har som matematikklærer. Gjennom arbeidet med denne oppgåva har eg tileigna meg eit breitt teoretisk grunnlag for å forstå kva som ligg i omgrepet algoritmisk tenking. Denne kunnskapen gjer meg i stand til å reflektera rundt eigen praksis, når det vert min tur til å undervisa i matematikk etter dei nye læreplanane. Gjennom samtalar med lærarane, og grundig analyse av deira refleksjonar og undervisningsmateriell, har eg gjort meg mange tankar om korleis eg kan leggja opp til programmering og algoritmisk tenking i matematikkfaget. Eg har òg vorte merksam på kva rolle eg kan få i eit profesjonsfellesskap med den kunnskapen eg har opparbeidd meg.

Det **intellektuelle målet** med prosjektet mitt var å sjå nærare på korleis lærarane handterer Fagfornyinga. Som skildra i førre delkapittel er den fordjupinga eg har gjort i nokre få lærarar sin praksis ikkje noko ein kan dra generelle konklusjonar og generalisera ut frå. Samstundes har oppgåva vist døme på ulike praksisar og refleksjonar som kan vera med på å utvikla didaktisk kompetanse. Eg har og peika på ulike praksisar ein finn hjå fem av dei mange lærarane som underviser i IT i år. Praksisane eg har sett på, har hatt mange likskapar knytt til algoritmisk tenking, men samstundes har det vore store skilnadar på korleis programmering har vore nytta som didaktisk verktøy i faget.

Det **praktiske målet** mitt har vore å skriva ei oppgåve som kan inspirera og gje kunnskap til matematikklærarar som treng det, i møte med Fagfornyinga og innføring av programmering og algoritmisk tenking som ein ny del av praksisen sin. Utgreiinga av omgrepet algoritmisk tenking, samt presentasjon av praksis og tilhøyrande døme på oppgåver, tenkjer eg kan vera med på nettopp dette.

Referansar

- Balanskat, A. & Engelhardt, K. (2015). *Computing our future. Computing programming and coding - Priorities, school curricula and initiatives across Europe*. European Schoolnet.
- Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. <https://doi.org/10.1145/1929887.1929905>
- Befring, E. (2015). *Forskningsmetoder i utdanningsvitenskap*. Cappelen Damm akademisk.
- Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada,
- Clarke, D. & Hollingsworth, H. (2002). Elaborating a model of teacher professional growth. *Teaching and Teacher Education*, 18(8), 947-967. [https://doi.org/https://doi.org/10.1016/S0742-051X\(02\)00053-7](https://doi.org/https://doi.org/10.1016/S0742-051X(02)00053-7)
- Costa, E. J. F., Campos, L. M. R. S. & Guerrero, D. D. S. (2017). Computational thinking in mathematics education: A joint approach to encourage problem-solving ability. 2017 IEEE Frontiers in Education Conference (FIE),
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). Computational thinking-A guide for teachers.
- Finstad, M. (2020). *Programmering og algoritmisk tenking i matematikk: Ein kvalitativ studie av ni matematikklærarar sitt syn på integreringa av programmering og algoritmisk tenking i matematikkfaget* [Masteroppgave, Universitetet i Bergen]. Bergen Open Research Archive. <https://bora.uib.no/bora-xmlui/handle/1956/23008>
- Forsström, S. E. & 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.org/10.26803/ijlter.17.12.2>
- Fuglseth, K. (2018). Vitskapsteori og hermeneutikk. I M. Krogtoft & J. Sjøvoll (Red.), *Masteroppgaven i lærerutdanninga : temavalg, forskningsplan, metoder* (2. utg., s. 245-264). Cappelen Damm akademisk.

- Gadamer, H.-G. (2004). *Truth and method* (J. Weinsheimer & D. G. Marshall, Overs.; 2. utg.). Continuum.
- García Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A. & Jormanainen, I. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers.
- Gjøvik, Ø. & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten - tidsskrift for matematikkundervisning*, 30(3), 31-37.
- Grover, S. & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.
<https://doi.org/10.3102/0013189X12463051>
- Irgens, E. J. (2010). Rom for arbeid: Lederen som konstruktør av den gode skole. I R. A. Andreassen, E. J. Irgens & E. M. Skaalvik (Red.), *Kompetent skoleledelse* (s. 125 - 145). Tapir.
- Jakobsen, R. E. (2019). *Programmering i matematikk – muligheter og utfordringer: En studie rundt innføringen av programmering som del av matematikkfaget i den norske skolen og hvilke argumenter som taler for eller imot innføringen*. [Masteroppgåve, Universitetet i Agder]. <https://uia.brage.unit.no/uia-xmlui/bitstream/handle/11250/2646381/Renate%20Elise%20Jakobsen.pdf?sequence=1&isAllowed=y>
- Kaufmann, O. T. & Stenseth, B. (2020). Programming in mathematics education. *International journal of mathematical education in science and technology*, 1-20. <https://doi.org/10.1080/0020739X.2020.1736349>
- Kilpatrick, J., Swafford, J. & Findell, B. (2001). *Adding It Up: Helping Children Learn Mathematics*. National Academy Press.
- Kluge, A. (2020). Hva tenker vi på når vi tenker digitalt? I B. K. Engen (Red.), *Digitalisering, kompetanse og læring* (1. utg., s. 115-136). Gyldendal.
- Krumsvik, R. J. (2019). Validitet i kvalitativ forskning. I R. J. Krumsvik (Red.), *Kvalitativ metode i lærarutdanninga* (s. 191-204). Fagbokforlaget.
- Kunnskapsdepartementet. (2017). *Framtid, fornyelse og digitalisering. Digitaliseringsstrategi for grunnsopplæringen 2017–2021*. Kunnskapsdepartementet.
https://www.regjeringen.no/contentassets/dc02a65c18a7464db394766247e5f5fc/kd_framtid_fornyelse_digitalisering_nett.pdf

- Kunnskapsdepartementet. (2018). Fornyer innholdet i skolen [Pressemelding].
Regjeringen.no. <https://www.regjeringen.no/no/aktuelt/forny-er-innholdet-i-skolen/id2606028/>
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg.). Gyldendal akademisk.
- Kvarv, S. (2014). *Vitenskapsteori : tradisjoner, posisjoner og diskusjoner* (2. utg.). Novus.
- Lye, S. Y. & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Maxwell, J. A. (2008). Designing a qualitative study. *The SAGE handbook of applied social research methods*, 2, 214-253.
- Misfeldt, M. & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls.
- National Council of Teachers of Mathematics. (2014). *Principles to actions: ensuring mathematical success for all* (M. National Council of Teachers of, Red.). NCTM, National Council of Teachers of Mathematics.
- NESH. (2016). *Forskningsetiske retningslinjer for samfunnsvitenskap, humaniora, juss og teologi*. <https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-humaniora-juss-og-teologi/>
- Niss, M. & Højgård Jensen, T. (2002). *Kompetencer og matematiklæring: ideer og inspiration til udvikling af matematikundervisning i Danmark*. Undervisningsministeriet : Undervisningsministeriets forlag.
- NOU 2015:8. (2015). *Fremtidens skole - Fornyelse av fag og kompetanser*. Kunnskapsdepartementet.
<https://www.regjeringen.no/contentassets/da148fec8c4a4ab88daa8b677a700292/no/pdfs/nou201520150008000dddpdfs.pdf>
- Nouri, J., Zhang, L., Mannila, L. & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education enquiry*, 11(1), 1-17.
<https://doi.org/10.1080/20004508.2019.1627844>
- Oldervoll, T. (2020). *Sinus IT matematikk, studieforberedende vg1*. Cappelen Damm.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books.

- Pennant, J. (2013). *Developing a Classroom Culture That Supports a Problem-solving Approach to Mathematics*. Henta 20.01.2021 frå <https://nrich.maths.org/10341>
- Polya, G. (1973). *How to solve it: A new aspect of mathematical method* (2. utg.). Princeton University Press.
- 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>
- Postholm, M. B. (2010). *Kvalitativ metode : en innføring med fokus på fenomenologi, etnografi og kasusstudier* (2. utg.). Universitetsforl.
- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G. & Voll, L. O. (2016). Teknologi og programmering for alle-En faggjennomgang med forslag til endringer i grunnopplæringen-august 2016.
- Schoenfeld, A. H. (2016). Learning to Think Mathematically: Problem Solving, Metacognition, and Sense Making in Mathematics (Reprint). *Journal of Education*, 196(2), 1-38. <https://doi.org/10.1177/002205741619600202>
- Selwyn, N. (2016). *Education and technology : key issues and debates* (2. utg.). Bloomsbury.
- Sentance, S. & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22(2), 469-495. <https://doi.org/10.1007/s10639-016-9482-0>
- Sfard, A. & Leron, U. (1996). Just give me a computer and i will move the earth: Programming as a catalyst of a cultural revolution in the mathematics classroom. *International Journal of Computers for Mathematical Learning*, 1(2), 189-195. <https://doi.org/10.1007/BF00571078>
- Skogen, K. (2018). Caseforskning. I M. Krogtoft & J. Sjøvoll (Red.), *Masteroppgaven i lærerutdanninga : temavalg, forskningsplan, metoder* (2. utg., s. 79-91). Cappelen Damm akademisk.
- Skott, J. (2018). *Matematik for lærerstudierende: delta 2.0 : fagdidaktik, 1.-10. klasse*. Samfundslitteratur.
- Säljö, R. (2013). Støtte til læring – tradisjoner og perspektiver. I R. Säljö & R. J. Krumsvik (Red.), *Praktisk-pedagogisk utdanning: en antologi* (s. 145-169). Fagbokforlaget.
- Tjora, A. (2021). *Kvalitative forskningsmetoder i praksis* (4. utg.). Gyldendal.

- Tolo, A. (2017). Kompetanseutvikling i lærerprofesjonen. I *Kompetanse og lærerprofesjonalitet* (s. 135-155). Fagbokforlaget.
- Trilling, B. & Fadel, C. (2009). *21st century skills: learning for life in our times* (1. utg.). Jossey-Bass.
- Utdanningsdirektoratet. (2017). Overordnet del av læreplanverket. <https://www.udir.no/laring-og-trivsel/lareplanverket/overordnet-del/>
- Utdanningsdirektoratet. (2019a). *Algoritmisk tenkning*. Henta 09.05.2020 frå <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2019b). *Læreplan i matematikk fellesfag Vg1 teoretisk* (MAT09-01). <https://www.udir.no/lk20/mat09-01>
- Voogt, J., Fisser, P., Good, J., Mishra, P. & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. <https://doi.org/10.1007/s10639-015-9412-6>
- Vygotskii, L. S. & Cole, M. (1978). *Mind in society: the development of higher psychological processes*. Harvard University Press.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7-14.
- Yin, R. K. (2018). *Case study research and applications : design and methods* (6. utg.). SAGE.
- Zimmermann, J. (2015). *Hermeneutics: A very short introduction*. OUP Oxford.

Oversikt over vedlegg

- Vedlegg 1: Intervjuguide
- Vedlegg 2: Kvittering NSD
- Vedlegg 3: Invitasjon med samtykkeerklæring
- Vedlegg 4: Tangram-oppgåve

Vedlegg 1: Intervjuguide

P: Korleis kan programmering nyttast som didaktisk verktøy i matematikkundervisinga?».

F1: Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?

F2: Korleis legg lærarar til rette for å løyse problem ved hjelp av algoritmisk tenking?

Oppstart

Kort intro og info

- Lydopptak
- Intervjusituasjonen - snakke fritt
- Notater.
- Evt spørsmål før oppstart.

Bakgrunn

- Kan du fortelja kort om bakgrunnen din?
 - Utdanning, arbeidserfaring (fag osv). Kurs/etterutdanning
 - Erfaring med programmering? (yrke/utdanning/fritid)
- Kor lenge (/ofte) har du undervist i matematikk 1T?

Utvikling av eigen kompetanse

I den nye læreplanen for matematikk 1T seier eit av dei nye kompetansemåla at eleven skal «...formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering»

- Korleis har du førebudd deg på innføringa av algoritmisk tenking i undervisinga di?
 - Kva har arbeidsgjevaren din gjort for å førebu dykk [som matematikklærarar] på den nye læreplanen?

Dersom kurs eller vidareutdanning :
 - Kan du fortelja litt om bakgrunnen for å ta kurs/vidareutdanning
 - Eigen interesse? Oppfordring frå leing? KFK?
 - Nytteverdi?
- Kan du sei noko om korleis går du fram når du skal planleggja ei undervisingsøkt/opplegg der algoritmisk tenking skal vera ein del av opplegget?
 - Innhente kunnskap/informasjon/fagstoff
- Kan du fortelja litt om samarbeid med kollegaar?
 - Planlegging av undervising?
 - Dele på undervising?

Dei kommande åra vil elevane som kjem til vgs ha meir erfaring med algoritmisk tenking frå grunnskulen.

- Kva tenkjer du om kompetansen din opp mot denne utviklinga?
 - Kurs? Sjølvstudium?

Algoritmisk tenking i matematikkundervisinga

- Kan du gje nokre døme på korleis du nyttar algoritmisk tenking i undervisinga di?
 - Matematisk tema?

Nokre spørsmål som omhandlar økter der det vert arbeidd med algoritmisk tenking.

- Korleis føregår samtalen i klasserommet - kven er det som snakkar mest?
 - Samarbeid? Deling?
- Kva type spørsmål stiller du elevane?
 - Kven er det som svarar på spørsmåla du stiller? (elevane(dei same?) evt. sjølv?)
- Kva gjer du når elevane svarar på spørsmål?
 - Gjentar / bed om utdjuping / omformulerer til «betre svar»?
 - Kva gjer du med svaret? (ros/fortsette vidare/hente kommentarar frå andre elevar?)
- Korleis legg du til rette for læring?
 - Hint, lede på rett spor?
 - Forklarar oppgåva utdjupande?
 - Rom for utforsking?
 - Legg du opp til at elevane skal laga noko? (skape/lage/designe?)
- Kan du fortelja kort om elevane?
 - Korleis er dei når dei arbeider?
 - Ta sjansar, gjere feil, vise uthald (gjer dei lett opp?)
 - Ta eigne val? Kopiere dine (lærar) sine idear og løysingar?
- Kva tenkjer du om kroppsspråket ditt i denne type økter? Korleis er det?
 - Interesse, glede, frustrasjon, misnøye?

Avslutning

- Er det noko meir du vil leggja til, eller noko anna du tenkjer på knytt til programmering og algoritmisk tenking som du trur kan vera relevant?
- Informere om
 - kontaktinfo for spørsmål e.l.
 - Anonymisering og handsaming av lydopptak osb.
- **Takke for deltakinga**

Vedlegg 2: Kvittering NSD

NSD NORSK SENTER FOR FORSKNINGSDATA

NSD sin vurdering

Prosjekttittel

Programmering som didaktisk verktøy i matematikken

Referansenummer

410756

Registrert

27.01.2021 av Torodd Flatråker Ottestad - 245987@stud.hvl.no

Behandlingsansvarlig institusjon

Høgskulen på Vestlandet / Fakultet for lærerutdanning, kultur og idrett / Institutt for pedagogikk, religion og samfunnsfag

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Anders Grov Nilsen, anders.nilsen@hvl.no, tlf: 53491511

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Torodd F. Ottestad, 245987@stud.hvl.no, tlf: 41452933

Prosjektperiode

01.01.2021 - 31.08.2021

Status

04.02.2021 - Vurdert

Vurdering (1)

04.02.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 04.02.2021, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

MELD VESENTLIGE ENDRINGER

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

<https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-imeldeskjema>

Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 31.08.2021

LOVLIG GRUNNLAG

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

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

TAUSHETSPLIKT

Deltagerne i prosjektet har taushetsplikt. Intervjuene må gjennomføres uten at det fremkommer opplysninger som kan identifisere elever.

PERSONVERNPRINSIPPER

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

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon omog samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitteog berettigede formål, og ikke behandles til nye, uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante ognødvendige for formålet med prosjektet
- lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for åoppfylle formålet

DE REGISTRERTES RETTIGHETER

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), og dataportabilitet (art. 20).

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

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

FØLG DIN INSTITUSJONS RETNINGSLINJER

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

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

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

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

Vedlegg 3: Invitasjonsskriv

Vil du delta i forskingsprosjektet «*Programmering som didaktisk verktøy i matematikk*»?

Dette er eit spørsmål til deg om å delta i eit forskingsprosjekt der føremålet er å undersøkje og presentera korleis programmering kan nyttast i matematikkfaget. I dette skrivet gjev me deg informasjon om måla for prosjektet og kva deltaking vil innebera for deg.

Føremål

Eg heiter Torodd F. Ottestad og arbeider med ei masteroppgåve i IKT i læring ved Høgskulen på Vestlandet (HVL). Studien omhandlar bruk av programmering i matematikkundervisinga.

I samband med den pågåande fagfornyinga kjem programmering og algoritmisk tenking inn i matematikkfaget. Mange matematikklærarar i den vidaregåande skulen er nok usikre på korleis ein kan inkludera desse verktøya og tenkjemåtane i eiga undervising. Mange har nok òg trong for å oppdatere kompetansen sin for å framleis undervisa i matematikk. Gjennom denne studien er målet å sjå på korleis matematikklærarar nyttar programmering som didaktisk verktøy i matematikkfaget, og få eit innblikk i kompetanseutviklinga til lærarane. Problemstillinga til studien er «*Korleis kan programmering nyttast som didaktisk verktøy i matematikkundervising?*» med tilhøyrande forskings spørsmål:

- F1: Korleis reflekterer lærarar kring utvikling av eigen kompetanse knytt til algoritmisk tenking?
- F2: Korleis legg lærarar til rette for å løyse problem ved hjelp av algoritmisk tenking?

I etterkant av datainnsamlinga vil data frå ulike lærarar samanliknast med kvarandre og vurderast opp mot aktuell teori. Gjennom å lesa den ferdige masteroppgåva er det eit mellom anna eit mål at matematikklærarar kan få inspirasjon til å endra og betra eigen praksis og utvikla kompetansen sin.

Kven er ansvarleg for forskingsprosjektet?

Høgskulen på Vestlandet er behandlingsansvarleg institusjon for prosjektet.

Kvifor får du spørsmål om å delta?

Utvalet er tenkt sett saman av matematikklærarar med undervisingserfaring frå IT etter ny læreplan (LK20. Direkte kontakt med aktuelle personar i eit utvida kontaktnett, samt utlysing ved bruk av didaktikkfora på sosiale media vert nytta for å koma i kontakt med aktuelle informantar. Du som får invitasjon til å delta innfrir desse utvalskriteria.

Kva inneber deltakinga for deg?

Datainnsamlinga vil i hovudsak gå føre seg gjennom intervju. Desse vil gjennomførast fysisk, eller gjennom videokonferanseverktøy som Zoom, alt etter nærleik, smittepress i samband med covid-19 osv. Intervjua vil ha eit omfang på 30 - 45 minutt. Intervjuet vert gjennomført på eit tidspunkt som passar for oss begge. Eg vil gjera lydopptak av intervjua til hjelp i seinare transkribering av intervjuet. Det er òg tenkt innsamling av undervisingsmateriell nytta i undervisinga di. Oppgåver som er nytta, relevant utdelt fagstoff, pensum etc. Utgangspunktet for intervjuet vil vera ein intervjuguide med aktuelle tema og spørsmål knytt til problemstillinga og forskings spørsmåla. Fokus er på utvikling av kompetanse, og på undervising knytt til algoritmisk tenking.

Deltakinga er frivillig

Det er frivillig å delta i prosjektet. Dersom du vel å delta, kan du trekkja samtykket ditt kva tid som helst utan å oppgje grunn. Alle peronsopplysingar vil då bli sletta. Det vil ikkje ha nokon negative konsekvensar for deg om du ikkje vil delta, eller om du seinare vel å trekkja deg.

Personvernet ditt - korleis me oppbevarer og nyttar opplysingane dine

Opplysingane om deg vil berre bli nytta til føremåla som er presentert i dette skrivet. Me handsamar opplysingane konfidensielt og i samsvar med personvernregelverket.

Det er berre eg og min rettleiar, Anders Grov Nilsen ved HVL, som vil ha tilgang til personidentifiserbare data. Dette gjeld namn i samband med samtykkeerklæring og lydopptak. Frå intervjuar vert transkriberte og ut forskingsprosjektet vil det verta nytta fiktive namn på lærarane. I den ferdige oppgåva vil det ikkje vera mogleg å knytta opplysingane opp til person. Namn på lærar vil vera fiktivt og namnet på skulen/arbeidsgjevar vil vera anonymisert. Lydopptak og personopplysingar vert lagra etter dei forskningsetiske retningslinjene ved HVL. Lydopptaka vil slettast så snart transkriberinga er gjennomført.

Kva skjer med opplysingane dine når me avsluttar forskingsprosjektet?

Opplysingane vert anonymisert når prosjektet vert avslutta/når oppgåva er godkjent, noko som etter planen er innan 31. august 2021.

Dine rettar

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

- innsyn i kva for personopplysingar som er registrert om deg, og å få utlevert ein kopi av opplysingane
- å få retta personopplysingar om deg
- å få sletta personopplysingar om deg
- å sende klage til Datatilsynet om handsaminga av dine personopplysingar

Kva gjev oss rett til å handsama personopplysingar om deg?

Me handsamar opplysingane om deg basert på samtykket ditt. På oppdrag frå Høgskulen på Vestlandet har NSD – Norsk senter for forskningsdata AS vurdert at handsaminga av personopplysingar i dette prosjektet er i samsvar med personvernregelverket.

Kvar kan eg finna ut meir?

Om du har spørsmål til studien, eller ønskjer å nytta deg av dine rettar, ta kontakt med:

- Høgskulen på Vestlandet ved meg på tlf. 414 52 933 eller epost: 245987@stud.hvl.no eller rettleiaren min Anders Grov Nilsen på anders.nilsen@hvl.no.
- Personvernombodet til HVL, Trine Anniken Larsen på personvernombod@hvl.no eller tlf: 55 58 76 83

Har du spørsmål knytt til NSD si vurdering av prosjektet, kan du ta kontakt med:

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

Beste helsing

Torodd F. Ottestad

Samtykkeerklæring

Eg har motteke og forstått informasjon om prosjektet «Programmering som didaktisk verktøy i matematikk», og har fått høve til å stilla spørsmål. Eg samtykker til:

- å delta i intervju med lydopptak
- å delta i innsamling av undervisningsmateriell

Eg samtykker til at opplysingane mine vert handsama fram til prosjektet er avslutta.

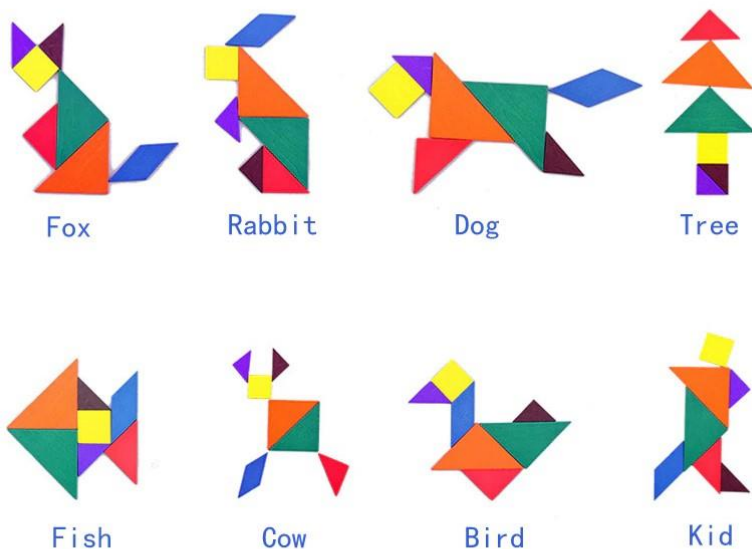
(Signert av prosjektdeltakar, dato)

Vedlegg 4: Tangram-oppgåve

Praktisk øvelse med tangram

Dere har sikkert jobbet mye med tangram i barneskolen

Et tangram består av 7 brikker som utgjør et kvadrat. Vi har 5 rettvinklede, likebeinte trekkanter, et kvadrat og et parallelogram. Regler sier at man må benytte alle 7 brikkene, de skal ligge plant og berøre hverandre, men kan ikke overlape hverandre⁷. Under ser dere eksempler på figurer man kan lage med et tangram.



Oppgaver å gjøre:

- 1) Gå sammen to og to. Sitt med ryggen mot hverandre. Den ene skal først lage en figur med tangram. Deretter skal den andre muntlig hvordan den skal legge brikkene for å skape samme form. Det er ikke lov til å si eksempelvis "det er en katt", men skal bare fortelle hvilke brikker som skal plasseres hvor, og det er ikke lov til å bruke ansikts- eller kroppsspråk. Hvordan gikk det?
- 2) Gjenta punkt 2, men nå bytter dere rolle

⁷ "Tangram - Matematikksenteret."

<https://www.matematikksenteret.no/sites/default/files/attachments/Novemberkonferansen/Tangram.pdf>