



**Høgskolen  
i Innlandet**

Fakultetet for lærerutdanning og pedagogikk

**Frida Tuftedal**

## **Masteroppgave**

# **Digital kompetanse for fremtiden – Læreres forståelse av algoritmisk tenkning i naturfag**

Digital competence for the future – Teachers'  
perceptions of Computational Thinking in science

Grunnskolelærerutdanning 1-7  
2MASTER17 - Masteroppgave

**2023**

# Forord

Ved innleveringen av denne masteroppgaven markeres slutten på et fem år langt studie ved Høgskolen i Innlandet. Da jeg startet på denne mastergraden virket dette tidspunktet uendelig langt borte, og det er helt surrealistisk at disse fem årene nå er over. Det er både rart, fint, skummelt, spennende og nervepirrende at denne epoken er over og en helt ny hverdag venter til høsten.

Denne masteroppgaven hadde ikke vært en mulighet hvis det ikke var for all støtte og hjelp jeg har fått underveis. Det er derfor flere mennesker jeg ønsker å takke. Først vil jeg takke de lærerne som tok seg tid i en travel hverdag for å delta i intervjuene som la grunnlaget for denne masteroppgaven. Uten dere hadde det ikke blitt noen oppgave! Jeg vil også takke veileder min, Thomas Frågåt, som gjennom hele prosessen har gitt grundig og god veiledning, raske tilbakemeldinger og gode innspill. Dette har gjort at jeg hele tiden har hatt god fremdrift i arbeidet med oppgaven, og det har spart med for mye frustrasjon og stress.

I løpet av disse fem årene har jeg også blitt kjent med en mange flotte medstudenter og fått flere nye vennskap. Jeg kommer til å savne samlingene våre på Hamar fylt med spennende diskusjoner, latter og minnerike opplevelser. Av disse medstudenter må jeg rette en spesiell takk til Marit Sørsveen Sæther, som har vært en fantastisk venn og god støtte gjennom hele studiet. Uten deg hadde ikke disse fem årene vært det samme!

Jeg vil takke min mann, Henrik Skaar Giske, som hele tiden har støttet valget mitt om å starte på en helt ny utdanning. Du har i disse fem årene, og som alltid ellers, vært en bauta. Jeg vil takke mamma, Tove Solli, som flere ganger enn jeg kan telle har hjulpet til med å få hverdagen til å gå rundt. Til slutt må jeg takke min beste venninne, Tara Ståhl, for alle turer og samtaler som har gitt avbrekk i hverdagen og påfyll av ny energi.

Nå er jeg ferdig utdannet lærer, men på ingen måte utlært. Til høsten starter en ny hverdag, og det blir spennende å endelig få prøve seg «på ekte».

Lillehammer, mai 2023

Frida Tuftedal

# Sammendrag

Temaet for denne masteroppgaven er algoritmisk tenkning i naturfag.

Algoritmisk tenkning ble et begrep i norske læreplaner etter Fagfornyelsen i 2020 og ble inkludert i naturfag gjennom to kompetansemål tilknyttet programmering, ett kompetansemål etter 7.trinn og ett kompetansemål etter 10. trinn. I denne masteroppgaven er fokuset på kompetansemålet tilknyttet programmering etter 7. trinn.

Både algoritmisk tenkning og programmering er begreper hentet fra informatikk, et fagfelt få lærere har bakgrunn fra. Oppgavens formål er derfor å undersøke hvordan naturfagslærere på mellomtrinnet forstår kompetansemålet etter 7.trinn, begrepet algoritmisk tenkning, og om de opplever å ha nok kompetanse til å undervise i disse nye konseptene. Ved å undersøke dette er målet å besvare den overordnede problemstillingen som ligger til grunn for denne masteroppgaven: *«I hvilken grad har man innført algoritmisk tenkning i naturfag etter Fagfornyelsen i 2020?»*

Studien bygger på et kvalitativt forskningsdesign med semistrukturert intervju av seks naturfagslærere på mellomtrinnet som metode. Intervjuene ble gjennomført en-til-en med utgangspunkt i en forhåndsbestemt intervjuguide. Intervjuguiden var utformet med utgangspunkt i problemstilling og forskningsspørsmål.

Resultatene viser at lærerne som ble intervjuet i stor grad forstod kompetansemålet som et rent programmeringsmål, og i liten grad klarte å aktualisere programmeringen opp mot naturfag for øvrig. Lærernes forståelse av kompetansemålet ut over dette var variert, samt begrenset og mangelfull sett opp mot det analytiske rammeverket. Lærerne hadde vansker med å forklare begrepet algoritmisk tenkning, og de forklaringene som ble gitt var smale og mangelfulle sett opp mot det analytiske rammeverket. Kun et fåtall av lærerne benyttet begrepet algoritmisk tenkning spontant for å beskrive kompetansemålet, noe som forsterker inntrykket av at lærerne i liten grad knytter algoritmisk tenkning og programmering til det etablerte naturfagspensumet. Et flertall av lærerne tok på eget initiativ opp behovet for mer tid til å jobbe med temaet, samt kurs og videreutdanning. Dette tyder på at lærerne selv opplever å ikke ha nok kompetanse innenfor området algoritmisk tenkning og programmering.

## Abstract

The title of this master's thesis is «Digital competence for the future – Teachers' perceptions of Computational Thinking in science». The topic is *Computational Thinking* in science.

The term *Computational Thinking (CT)* entered the Norwegian curriculum after the renewal in 2020. In science, *CT* indirectly appears in two competence aims, realized through the term *programming*, one after grade 10 and one after grade 7. Due to the scope of this thesis, only the latter will be discussed throughout this study.

Both *CT* and *programming* are terms that originate from computer science, a field in which few teachers have educational background. Hence, the purpose of this master's thesis is to investigate how middle school science teachers understand the competence aim after grade 7, the term *CT*, and whether or not they feel competent to teach these new concepts. By examining these aspects of teachers' perceptions, the aim is to answer the research question that forms the basis of this thesis: *To what extent has Computational Thinking been implemented in science after the curriculum reform in 2020?*

The study is based on a qualitative research design with semi-structured interviews of six middle school science teachers. The interviews were conducted individually, based on a predetermined interview guide. The interview guide was designed based on the research aim and purpose and the above-mentioned research question.

The results show that the teachers who were interviewed mainly understood the competence aim after the grade 7 to be strictly about programming. Programming, however, was not primarily understood as a way to teach and learn science. In addition, the teachers' perceptions of the competence aim were different and deficient according to the analytical framework used in this study. The teachers struggled to define the concept of *CT*, and their attempts to explain the term were limited and inadequate compared to the analytical framework. Only a few of the teachers used the term *CT* spontaneously to describe the competence aim, which strengthen the impression that teachers do not link *CT* or programming to the established science curriculum. A majority of the teachers called for more time to work with *CT* and programming, as well as more education related to these concepts. This implies that teachers do not feel like they have enough competence to teach *CT* and programming sufficiently.

## Tabeller og figurer

Tabell 1. Oversikt over det teoretiske rammeverket til Shute et al. (2017) med ferdigheter tilknyttet algoritmisk tenkning, samt beskrivelse av disse. Rammeverket er oversatt fra engelsk.....	24
Tabell 2. Oversikt over det teoretiske rammeverket til Weintrop et al. (2016). Rammeverket er oversatt fra engelsk.....	27
Tabell 3. Tabell over lærere med antall år som lærer, hvorvidt de underviser i matematikk, samt videreutdanning. ....	33
Tabell 4. Oversikt over det analytiske rammeverket med beskrivelser av hovedtema og tilhørende undertema.....	39
Tabell 5. Eksempel på koding av sitater fra intervju 5.....	42
Tabell 6. Utdrag fra oversikten over alle relevante sitater fra intervjuene plassert inn i det analytiske rammeverket.....	42
Tabell 7. Oversikt over temaspesifikke spørsmål og hvilke undertema, samt hovedtema (HT), lærernes besvarelse passet inn under.....	44
Figur 1. Oversikt over analytisk rammeverk med hovedtema og tilhørende undertema.	39

# Innholdsfortegnelse

<b>Sammendrag</b> .....	iii
<b>Abstract</b> .....	iv
<b>1. Innledning</b> .....	1
1.1 Introduksjon .....	1
1.2 Problemstilling .....	3
1.3 Studiens oppbygning .....	4
<b>2. Teori</b> .....	6
2.1 Algoritmisk tenkning .....	6
2.2 Algoritmisk tenkning og programmering i norsk skole .....	10
<b>3. Tidligere forskning</b> .....	19
3.1 Læreres forståelse av algoritmisk tenkning .....	19
3.2 Implementering av algoritmisk tenkning .....	21
<b>4. Teoretisk rammeverk</b> .....	24
4.1 Algoritmisk tenkning som et sett med generelle ferdigheter .....	24
4.2 Algoritmisk tenkning sin rolle i matematikk og naturfag .....	26
<b>5. Metode</b> .....	29
5.1 Kvalitativ metode .....	29
5.2 Kvalitativt forskningsintervju .....	30
5.3 Utvalg og rekruttering .....	31
5.4 Gjennomføring av intervjuer .....	33
5.5 Transkripsjon .....	35
5.6 Tolkning og analytisk tilnærming .....	35
5.7 Analytisk rammeverk .....	38
5.8 Eksempler på analyse og koding .....	42
5.9 Kvalitet i forskningen .....	45

5.10	Forskningsetikk .....	47
<b>6.</b>	<b>Resultat</b> .....	<b>49</b>
6.1	Programmering i fokus .....	49
6.2	Begrenset eller manglende forståelse av algoritmisk tenkning.....	52
6.3	Behov for kurs og etterutdanning.....	55
<b>7.</b>	<b>Drøfting</b> .....	<b>57</b>
7.1	Lærernes forståelse av kompetansemålet i naturfag .....	57
7.2	Lærernes forståelse av algoritmisk tenkning .....	63
<b>8.</b>	<b>Konklusjon</b> .....	<b>70</b>
	<b>Referanseliste</b> .....	<b>73</b>
	<b>Vedlegg</b> .....	<b>78</b>
	<b>Vedlegg 1: Intervjuguide</b> .....	<b>78</b>
	<b>Vedlegg 2: Informasjonsskriv til lærere om studien</b> .....	<b>79</b>
	<b>Vedlegg 3: Samtykkeerklæring</b> .....	<b>81</b>
	<b>Vedlegg 4: Godkjenning fra NSD</b> .....	<b>82</b>

# 1. Innledning

## 1.1 Introduksjon

For over 20 år siden påpekte Andrea diSessa (2000) at den stadig økende rollen datamaskiner hadde i samfunnet ikke ble gjenspeilet i skolen. Siden den gang har situasjonsbildet endret seg, både i form av teknologisk utvikling og hvordan man forsøker å gjenspeile dette i utdanning. Samfunnet har blitt enda mer digitalisert, og den teknologiske utviklingen går raskere og raskere. Digitalisering gjennomsyrrer i dag alle sider av samfunnet, økonomien og vanlige folks hverdag (Bocconi et al., 2022). Dette gjør at behovet for teknologisk kompetanse hos befolkningen har økt, og trolig vil bli enda større i fremtiden. På bakgrunn av dette trekkes stadig ferdigheter innenfor teknologi, problemløsning og algoritmisk tenkning frem av flere som viktige kompetanser for det 21. århundre (Angeli & Giannakos, 2020; Larson & Miller, 2011; Wing, 2017). Kompetanser for det 21. århundre er kompetanser som anses som viktige for å kunne leve meningsfulle liv i fremtiden. På bakgrunn av den digitale verdenen vi lever i, anses digitale ferdigheter og algoritmisk tenkning å være ferdigheter som i fremtiden vil være like viktige som lesing, skriving og regning er i dag (Voogt et al., 2015; Wing, 2017). Det teknologiske samfunnet vi lever i, samt ferdighetene som trengs for å beherske dette samfunnet, må gjenspeiles i utdanning (Sanne et al., 2016).

Behovet for ferdigheter og kompetanser innenfor teknologi er anerkjent av flere land, og ble ytterligere aktualisert under korona-epidemien (Bocconi et al., 2022). Det er derfor stadig flere land som ser viktigheten av at ferdigheter og kompetanse innenfor teknologi implementeres i utdanning. Algoritmisk tenkning blir sett på som en viktig ferdighet i en enda mer digital fremtid, og er derfor et av begrepene som gjerne blir tatt inn i læreplaner og dermed knyttet til utdanning (Bocconi et al., 2022). Algoritmisk tenkning er den norske oversettelsen av begrepet *Computational thinking*, og algoritmisk tenkning vil derfor i denne masteroppgaven benyttes som tilsvarende *Computational thinking*. Konseptet algoritmisk tenkning, og hva dette egentlig innebærer, er svært omdiskutert (Andersen et al., 2023). Det finnes flere ulike definisjoner, men ingen tydelig konsensusdefinisjon (Bocconi et al., 2022; Shute et al., 2017). Eksempelvis kan algoritmisk tenkning ses på som et sett med generelle ferdigheter (Shute et al., 2017), som praksiser innenfor ulike hovedområder tilknyttet matematikk og naturfag (Weintrop et al., 2016), eller det kan defineres smalere ved at det i større grad knyttes opp mot programmering og problemløsning (Brennan & Resnick, 2012;



Grover & Pea, 2013). Overordnet kan de ulike typene definisjoner på algoritmisk tenkning gjerne plasseres innenfor tre ulike kategorier (Bocconi et al., 2022). Den ene kategorien inneholder definisjoner hvor algoritmisk tenkning blir sett på som å utvikle løsninger på et problem som kan gjennomføres av en datamaskin. Den andre kategorien omhandler definisjoner hvor algoritmisk tenkning blir sett på som en form for problemløsningsprosess, mens i den siste kategorien blir algoritmisk tenkning gjerne sett på som et sett med ferdigheter som kan generaliseres og benyttes innenfor flere områder. En annen overordnet måte å sortere definisjonene på algoritmisk tenkning er ved å se på de som domene-spesifikke, eller generelle (Bocconi et al., 2022). Det finnes altså mange ulike definisjoner på algoritmisk tenkning, noe som betyr at begrepet og konseptet blir forstått ulikt. Det at flere land velger å innføre algoritmisk tenkning i sine læreplaner betyr derfor ikke nødvendigvis at de innfører det samme. Det betyr derfor heller ikke at man automatisk vet hva en slik innføring innebærer, hvordan det skal undervises, eller hva det er tenkt at elevene skal lære.

Også i Norden har flere land innført ulike informatikkinspirerte konsepter i sine læreplaner (Andersen et al., 2023), men hvordan dette har blitt gjort er ulikt fra land til land. Finland innførte i 2016 både programmering og algoritmisk tenkning i sine læreplaner. Danmark innførte i 2019 et nytt fag tilknyttet teknologiforståelse, hvor algoritmisk tenkning var en del av dette faget (Bocconi et al., 2022). Det nye faget var en del av et pilot-prosjekt hvor målet var å få økt kunnskap og ekspertise rundt hvordan man på best måte kan undervise teknologiforståelse på barneskolen. Sverige har også i nyere tid revidert sine læreplaner, der den reviderte utgaven var gjeldene fra høsten 2018 (Bocconi et al., 2022). Den nye læreplanen inkluderte en definisjon på digital kompetanse, hvor programmering inngikk som en del av denne definisjonen. Til tross for at stadig flere land tar inn algoritmisk tenkning i sine læreplaner er det fortsatt ikke klart for lærere, lærerutdannere eller forskere hva dette egentlig innebærer (Andersen et al., 2023).

Algoritmisk tenkning i utdanningssammenheng er et forskningsområde hvor det også foregår større forskningsprosjekt. Denne masteroppgaven er en del av et slikt større forskningsprosjekt ved navn MASCOT (Mathematics, Science and Computational thinking) (OsloMet, u.å.). MASCOT-prosjektet er et tverrfaglig forskningsprosjekt som gjennomføres som et samarbeid mellom Norge, Danmark og Finland. Prosjektet er finansiert av Norsk forskningsråd. MASCOT-prosjektet ble påbegynt i mai 2021 og avsluttes i oktober 2024. Målet med prosjektet er å «... utvikle kunnskap om lærings- og vurderingsprosessene i algoritmisk tenkning i lærerutdanning og skole, primært innen matematikk og naturfag.

Prosjektets forskning skal videre danne grunnlag for utvikling av nye undervisnings- og vurderingspraksiser.» (OsloMet, u.å.). Dataene samlet inn i arbeidet med denne masteroppgaven, samt resultatene som presenteres, vil derfor kunne bli brukt som en del av MASCOt-prosjektet i etterkant av innlevering.

## 1.2 Problemstilling

Høsten 2020 ble de informatikkinspirerte konseptene algoritmisk tenkning, programmering og algoritmer også en del av den norske læreplanen (Andersen et al., 2023). Anbefalingen i forkant av at disse konseptene ble tatt inn i LK20 var å opprette et nytt obligatorisk teknologifag (Sanne et al., 2016). I dette teknologifaget skulle algoritmisk tenkning være en kjernekompetanse, og ble beskrevet som evnen til å kunne abstrahere, systematisk gjennomgå informasjon, tolke, forstå, modellere og resonnere. Begrunnelsen for å opprette et helt nytt fag var å unngå nedprioritering slik man hadde erfart med det tverrfaglige emnet *Teknologi og design* i LK06 (Sanne et al., 2016).

Da LK20 ble innført var det ikke opprettet et nytt teknologifag. Istedenfor var algoritmisk tenkning og programmering blitt inkludert i fagene matematikk, naturfag og kunst og håndverk (Andersen et al., 2023; Vinnervik & Bungum, 2022). I LK20 nevnes algoritmisk tenkning eksplisitt kun én gang (Utdanningsdirektoratet, 2020a). Implisitt nevnes algoritmisk tenkning gjennom kompetansemål i fagene matematikk, naturfag og kunst og håndverk tilknyttet algoritmer og, eller programmering (Vinnervik & Bungum, 2022). I læreplanen for matematikk blir algoritmisk tenkning hovedsakelig beskrevet som en problemløsningsprosess (Utdanningsdirektoratet, 2020a). Det er også den måten algoritmisk tenkning hovedsakelig blir forklart på Utdanningsdirektoratet sin ressursnettside for lærere (Utdanningsdirektoratet, 2019). Som nevnt finnes det flere definisjoner på og forståelser for hva som ligger i algoritmisk tenkning, og det er uklart for lærere hva algoritmisk tenkning egentlig innebærer. I tilfellet for den norske læreplanen samsvarer ikke beskrivelsen fra anbefalingene i forkant av innføring, nødvendigvis med beskrivelsen av algoritmisk tenkning i LK20. En tredje forståelse er den lærerne sitter på, og denne er viktig ettersom det er lærerne som skal videreformidle innholdet i læreplanen til elevene. På bakgrunn av dette ble følgende problemstilling formulert:

*«I hvilken grad har man innført algoritmisk tenkning i naturfag etter Fagfornyelsen i 2020?»*

Det er problemstillingen presentert ovenfor som ligger til grunn for denne masteroppgaven. Ettersom det er læreres forståelse av læreplanen som avgjør hvordan læreplanen aktualiseres, ble det vurdert som viktig å undersøke læreres forståelse av algoritmisk tenkning for å kunne besvare problemstillingen. Med bakgrunn i problemstillingen ble det derfor utarbeidet tre forskningsspørsmål som omhandlet læreres forståelse. Forskningsspørsmålene utgjorde også utgangspunktet for intervjuguiden.

1. Hvordan forstår naturfagslærere på mellomtrinnet etter 7.trinn i naturfag: «*utforske, lage og programmere teknologiske systemer som består av deler som virker sammen*»?
2. Hvordan forstår naturfagslærere på mellomtrinnet begrepet algoritmisk tenkning?
3. I hvor stor grad opplever naturfagslærere på mellomtrinnet å ha tilstrekkelig kompetanse til å undervise i algoritmisk tenkning?

Formålet med denne studien er derfor å undersøke i hvilken grad man har innført algoritmisk tenkning i naturfag gjennom å undersøke naturfagslærere på mellomtrinnet sin forståelse av dette begrepet, samt tilhørende kompetansemål i naturfag. Lærernes forståelse vil bli belyst med bakgrunn i deres besvarelser under intervjuet, samt en abduktiv tematisk analyse av disse besvarelsene. Ettersom algoritmisk tenkning og programmering ikke ble en del av den norske læreplanen før høsten 2020, er det fortsatt få studier som ser på norske læreres forståelse av disse begrepene. Gjennom denne masteroppgaven er målet å bidra med litt økt kunnskap innenfor nettopp dette området.

### 1.3 Studiens oppbygning

I denne innledningen har algoritmisk tenkning som temaet for denne masteroppgave blitt presentert i korte trekk. Teamets relevans, både som en kompetanse for fremtiden og som et konsept innenfor utdanning, har også blitt presentert. Videre har algoritmisk tenkning blitt satt i en norsk kontekst, samt aktualisert i problemstillingen som legger grunnlaget for denne masteroppgaven. Sammen med problemstilling har også forskningsspørsmålene blitt presentert.

I neste kapittel beskrives først algoritmisk tenkning mer i dybden. Jeg har valgt å starte teori-kapittelet med en grundig beskrivelse av begrepet, ettersom det er et begrep som er nytt eller ukjent for de fleste som ikke har bakgrunn fra informatikk. Videre beskrives sammenhengen

mellom algoritmisk tenkning og LK20. Etter teori kommer tidligere forskning. Her presenteres tidligere forskning, både norsk og internasjonal, som har sett på læreres forståelse av algoritmisk tenkning og implementeringen av algoritmisk tenkning i utdanning. Gjennom tidligere forskning blir også problemstillingen og forskningsspørsmålenes relevans enda tydeligere.

I kapittel 4 presenteres to teoretiske rammeverk tilknyttet algoritmisk tenkning. Disse to teoretiske rammeverkene legger grunnlaget for oppgavens analytiske rammeverk. Neste kapittel omhandler oppgavens metodiske design og gjennomføring. Her redegjøres det for ulike metodiske beslutninger som har blitt gjort både i forkant og underveis i studien, den analytiske tilnærmingen beskrives og det analytiske rammeverket presenteres. I tillegg diskuteres oppgavens validitet og reliabilitet, samt forskningsetiske perspektiver.

Etter metodekapittelet presenteres resultatene fra analysen i kapittel 6. Disse blir presentert som tre hovedfunn med utgangspunkt i rekkefølgen på forskningsspørsmålene. I kapittel 7 blir disse funnene drøftet opp mot forskningsspørsmålene, teori og tidligere forskning. Målet i dette kapittelet er å belyse forskningsspørsmålene, samt besvare problemstillingen på en god måte. I kapittel 8 presenteres en kortfattet oppsummering, samt en konklusjon. I dette kapittelet legges det vekt på å tydelig belyse studiens forskningsspørsmål, samt å gi en god og tydelig besvarelse på studiens problemstilling: *I hvilken grad har man innført algoritmisk tenkning i naturfag etter Fagfornyelsen i 2020?*

## 2. Teori

### 2.1 Algoritmisk tenkning

På norsk har begrepet *Computational thinking* blitt oversatt til algoritmisk tenkning. En oversettelse enkelte hevder er problematisk (Nordby et al., 2022; Vinnervik & Bungum, 2022). Vinnervik og Bungum (2022) mener, med utgangspunkt i sitt analytiske rammeverk, at begrepet *Computational thinking* internasjonalt er bredere og mer omfattende enn den norske oversettelsen. Internasjonalt omtales *algorithmic thinking* gjerne som en ferdighet innenfor *Computational thinking*, sammen med koding og evnen til å designe algoritmiske løsninger på et problem (Angeli & Giannakos, 2020). Samtidig finnes det også internasjonalt eksempler på rammeverk for *Computational thinking* som er betydelig smalere enn det rammeverket Vinnervik og Bungum (2022) tar utgangspunkt i. Se for eksempel Brennan og Resnick (2012) eller Grover og Pea (2013). Nordby et al. (2022) mener på sin side at det problematiske ved den norske oversettelsen er at begrepet algoritmisk tenkning hos matematikklærere kan forveksles med standard algoritmer i matematikk. Uavhengig av en potensielt problematisk norsk oversettelse finnes det ikke en konsensusdefinisjon på hva *Computational thinking* omfatter (Barr & Stephenson, 2011; Brennan & Resnick, 2012; Ketelhut et al., 2020; Shute et al., 2017). Videre i denne masteroppgaven vil derfor den norske oversettelsen algoritmisk tenkning bli brukt som tilsvarende *Computational thinking*.

Algoritmisk tenkning ble først tatt i bruk av Seymour Papert i 1980 (Papert, 1980). I boka «Mindstorms: Children, Computers, and Powerful Ideas» lanserte Papert begrepet som et verktøy for tenkning og læring innenfor utdanning. Det å lære algoritmisk tenkning var, ifølge Papert, ikke en ferdighet forbeholdt de som skulle jobbe innenfor informatikk. Denne ferdigheten ville kunne bidra til å øke alle elevers resonneringsferdigheter og problemløsningsferdigheter, med programmering som et middel mot målet. Flere år senere introduserte Andrea diSessa begrepet *Computational literacy* (diSessa, 2000). I sin bok tok han opp den stadig økende rollen datamaskiner hadde både i og på samfunnet og samfunnsutviklingen, og hvordan dette ikke ble gjenspeilet i utdanning. På bakgrunn av dette argumenterte han for at det burde legges til rette for en ny type litterasitet i samfunnet og i utdanning. Den nye formen for litterasitet kalte han *Computational literacy*, og den skulle bestå av tre grunnpilarer. Den første pilaren var den materielle og omhandlet «språket», altså de tegnene, symbolene, skildringene og representasjonene som gjøre det mulig for oss å

tenke, huske og kommunisere. Den neste pilaren var den kognitive, som innebar en spesialisert måte å tenke på. Den siste pilaren omfattet det sosiale, nemlig hvordan man er en del av et fellesskap av likesinnede med tilsvarende kunnskap og ferdighet innenfor området. Papert og diSessa brukte ulike begreper og forklart dem også ulikt. Likevel har de noe til felles. De tok begge utgangspunkt i konsepter og begrepet som tidligere hadde vært forbeholdt informatikken, og gjorde dem til noe mer generelt og nyttig for allmennheten.

I etterkant av Papert og diSessa kom begrepet algoritmisk tenkning virkelig på dagsorden og ble aktualisert utenfor informatikken av Wing (2006). I sin artikkel beskrev Wing algoritmisk tenkning som et sett med ferdigheter med høy nytteverdi innenfor flere områder, ikke bare innenfor informatikk. I artikkelen ble disse ferdighetene beskrevet som like grunnleggende i dagens samfunn som det å kunne lese, skrive og regne. Av den grunn burde algoritmisk tenkning være en del av alle barns opplæring. Til tross for en noe vag forklaring om hva begrepet og tilhørende ferdigheter innebar, fikk begrepet stor oppmerksomhet og ble av flere brukt som et springbrett for å få algoritmisk tenkning inn i utdanning (Lodi & Martini, 2021).

Ferdigheter innenfor teknologi og problemløsning går igjen på de fleste oversikter over kompetanser for det 21 århundre (Larson & Miller, 2011). Med kompetanser for det 21 århundre menes ferdigheter fremtidens borgere vil ha behov for, og som derfor må inkluderes i utdanning. Algoritmisk tenkning blir i flere av disse oversiktene trukket frem som en nøkkelferdighet for å kunne håndtere de utfordringene man står ovenfor i det 21 århundre (Angeli & Giannakos, 2020; Wing, 2017). Dette har bidratt til at stadig flere land har tatt algoritmisk tenkning inn i sine læreplaner (Bocconi et al., 2022). Gjennom koronapandemien ble det enda tydeligere hvilke muligheter teknologien kan gi oss, noe som aktualiserte behovet for teknologiske kompetanse hos befolkningen ytterligere (Bocconi et al., 2022).

En av hovedutfordringene med begrepet algoritmisk tenkning generelt og i utdanningssammenheng, er at det ikke finnes noen tydelig konsensusdefinisjon på hva begrepet innebærer (Barr & Stephenson, 2011; Brennan & Resnick, 2012; Ketelhut et al., 2020; Shute et al., 2017). Den manglende konsensusdefinisjon på algoritmisk tenkning handler ikke om at det ikke finnes definisjoner. Tvert imot har det blitt gjort flere forsøk på å definere hva begrepet innebærer generelt og i utdanningssammenheng (Shute et al., 2017). Mangelen på en konsensusdefinisjon gjør at når det gjennomføres studier som omhandler algoritmisk tenkning, er det stor variasjon i hvilke definisjoner som benyttes. Dette fører igjen til at de ulike studiene har ulike utgangspunkt når de skal gjennomføre, tolke og vurdere ulike konsepter og praksiser (Bocconi et al., 2022). Til tross for ulikt utgangspunkt er

det flere studier de siste årene som har argumentert for å implementere algoritmisk tenkning i grunnskoleutdanninger (Yadav et al., 2017).

I utdanningssammenheng er det ikke nok med bare en definisjon på et nytt begrep, man trenger også konkrete eksempler på hvordan begrepet skal realiseres i klasserommet (Barr & Stephenson, 2011). Den manglende definisjonen på algoritmisk tenkning fører derfor til utfordringer med å forstå hvordan man i praksis skal undervise rundt dette begrepet, og hva elevene helt konkret skal lære (Voogt et al., 2015). Hvilke konsepter fra informatikken som innføres i de ulike landene under begrepet algoritmisk tenkning er også varierende. Det enkelte land må derfor ta stilling til hvilke konsepter de ønsker å ta inn i utdanningen, når de skal gjøre det og hvordan (Bocconi et al., 2022). Et viktig aspekt ved innføring av nye begreper eller konsepter i utdanningssammenheng er naturligvis at styringsdokumentene endres, men dette i seg selv er ikke nok hvis ikke lærerne får ressurser til å gjennomføre endringen (Barr & Stephenson, 2011). Flere land har nå endret sine læreplaner for å legge til rette for at elevene skal utvikle den digitale kompetansen som trengs i fremtiden, men med denne endringen må også lærerne få ressurser hvis det skal gjennomføres i praksis. En viktig del av slike ressurser er en klar definisjon på begrepet, samt eksempler på aktiviteter tilpasset den aldersgruppen som skal undervises (Barr & Stephenson, 2011). En ting er hva som står i læreplanene, men hvordan dette kommer til uttrykk i klasserommet er avhengig av læreres tolkninger og prioriteringer (Vinnervik & Bungum, 2022).

Algoritmisk tenkning er mer enn å programmere en datamaskin (Wing, 2006). Når man skal undervise i algoritmisk tenkning benyttes ofte datamaskiner og ulike applikasjoner som verktøy, men det er da viktig å være bevisst at algoritmisk tenkning ikke oppfattes som synonymt med programmering og koding (Yadav et al., 2017). Programmering er et nyttig verktøy for å lære elever algoritmisk tenkning, men det bør ikke være det eneste verktøyet (Voogt et al., 2015). Lu og Fletcher (2009) mener at en av de største pedagogisk utfordring ved å undervise i algoritmisk tenkning er å klare og skille algoritmisk tenkning fra ren informatikk og programmering. For å unngå en slik begrenset forståelsen av begrepet bør elever få erfaring med algoritmisk tenkning lenge før de introduseres for ulike programmeringsspråk. Ifølge Lu og Fletcher (2009) bør hovedfokuset i starten være på å gi elevene et ordforråd og symboler de kan benytte for å lage instruksjoner, hjelpe dem å abstrahere, foreslå løsninger på et problem, samt beskrive hvordan de tenker. Dette betyr at man må finne måter å undervise i algoritmisk tenkning på tvers av fag, både med og uten programmering eller andre digitale verktøy. Ved å gi elevene erfaringer med algoritmisk

tenkning over tid, som er fristilt fra programmering og koding, vil elevene være bedre forberedt til å lære nettopp dette senere i utdanningsløpet sitt. Ved å lære seg algoritmisk tenkning vil man få gode forutsetninger for å lære seg programmering senere i livet (Shute et al., 2017). Det er likevel ikke alle som deler dette synet på algoritmisk tenkning og programmering. Grover og Pea (2013) mener at programmering er en helt grunnleggende ferdighet innenfor algoritmisk tenkning, og et av de viktigste verktøyene for å støtte utviklingen av algoritmisk tenkning. Ifølge dem vil aktiviteter hvor elevene skal jobbe med datakonsepter og algoritmisk tenkning utenfor konteksten av en datamaskin kunne hindre elevene i å få autentiske erfaringer med hvordan algoritmisk tenkning faktisk benyttes.

Teknologi har blitt en stor del av hverdagen og dette må gjenspeiles i utdanningen (Sanne et al., 2016). Den digitale utviklingen har skjedd fort og påvirker alle nivåer av samfunnet og økonomien, samt har stor innflytelse på folks hverdagsliv (Bocconi et al., 2022). Innenfor spesifikke fagdisipliner som matematikk og naturvitenskap har den digitale utviklingen gitt helt nye muligheter, og dermed også blitt en viktig del av arbeidet innenfor disse fagdisiplinene (Weintrop et al., 2016). Ved å gi elevene et realistisk bilde av hvilke muligheter som ligger i teknologi innenfor ulike fag gjør man dem mer forberedt på hva det innebærer å jobbe innenfor en gitt fagdisiplin. For elever som ikke skal jobbe innenfor disse fagområdene i fremtiden er opplæring i teknologiske konsepter viktig for allmenndannelsen (Weintrop et al., 2016). Sagt med andre ord er teknologi gjennomgripende i dagens samfunn, både generelt i hverdagen, men også mer spesifikt innenfor ulike fagområder. Digital kompetanse i vår tid handler om mer enn å mestre noen enkeltstående, grunnleggende digitale ferdigheter (Bocconi et al., 2022). Skal elevene bli rustet til å møte fremtiden må de i dag kunne mer enn å bruke teknologien rundt dem. Elevene må i tillegg til å kunne bruke teknologien forstå de konseptene og prinsippene som ligger bak. På denne måten vil de i fremtiden kunne bidra til å videreutvikle dem. En slik forståelse krever innsikt i og erfaring med grunnleggende teknologiske prinsipper, digital teknologi og programmering (Sanne et al., 2016). Behovet for at utdanning og skole skal speile dagens teknologiske samfunn har ført til at flere Europeiske land, i ulik grad, har innført konsepter hentet fra informatikk i sine læreplaner (Bocconi et al., 2022). Eksempelvis ble det innført programmering som en del av matematikkfaget i den finske læreplanen i 2016 og i den svenske læreplanen i 2018 (Vinnervik & Bungum, 2022). Høsten 2020 inkluderte også Norge algoritmisk tenkning og programmering som nye begrep i sine læreplaner.



## 2.2 Algoritmisk tenkning og programmering i norsk skole

I Læreplanverket for Kunnskapsløftet 2020, heretter referert til som LK20, ble programmering inkludert i fagene kunst og håndverk, musikk, naturfag og matematikk (Bocconi et al., 2022; Vinnervik & Bungum, 2022). I arbeidet med Fagfornyelsen fikk en ekstern arbeidsgruppe i oppdrag av Utdanningsdirektoratet å foreta en faggjennomgang av teknologi i grunnopplæringen. Målet var å utarbeide en rapport som skulle gi beslutningstakere et kunnskapsgrunnlag om hvilke kompetanse fremtidens elever trengte innenfor teknologi og teknologirelaterte emner, hvor også digitale ferdigheter og programmering skulle inngå. Rapporten fra dette arbeidet, «Teknologi og programmering for alle – En faggjennomgang med forslag til endringer i grunnopplæringen – august 2016» (Sanne et al., 2016), har gjennom MASCOT-prosjektet blitt identifisert som det ledende dokumentet bak innføringen av algoritmisk tenkning og programmering i LK20 (Andersen et al., 2023).

Forståelsen av teknologi som viktig for alle var ikke nytt i arbeidet med Fagfornyelsen. I LK06 inngikk teknologi som en del av undervisningen under det flerfaglige emnet *Teknologi og design*. Et emne som erfaringsmessig ble nedprioritert (Sanne et al., 2016). Det at nye temaer i læreplanen nedprioriteres til fordel for mer etablerte faglige temaer er en kjent problematikk (Bocconi et al., 2022). Et viktig aspekt ved innføring av et nytt tema i læreplanen er derfor tilrettelegging i form av tid og ressurser, slik at det nye temaet ikke velges bort eller nedprioriteres (Bocconi et al., 2022). For å unngå nedprioritering av nye emner og fag må disse gjøres synlig på egne premisser i læreplanen (Sanne et al., 2016). Anbefalingene etter faggjennomgangen ble derfor å opprette et nytt, obligatorisk fag i grunnskolen som skulle omfatte teknologi og programmering. Dette faget måtte få egne læringsmål, det måtte bli gitt dedikerte ressurser til å jobbe med faget, samt avsatt tid til å utvikle læreres kompetanse og faglige identitet. Overordnet var anbefalingen at det nye faget skulle inkludere både programmering og digital teknologi, men også at det skulle gi et felles grunnlag for å fornye det faglige innholdet både i realfagene og de andre skolefagene. En slik fornyelse måtte tilpasses det enkelte fags egenart, samtidig som det faglige innholdet ble koblet opp mot teknologi og de mulighetene som fantes innenfor teknologi og programmering i det enkelte fag (Sanne et al., 2016). Denne anbefalingen får også støtte i annen litteratur. Blant annet trekker Weintrop et al. (2016) frem at teknologi og digitale verktøy utgjør en viktig del av både matematikk og naturvitenskap, samtidig som man ikke nødvendigvis

benytter den samme teknologien eller digitale verktøy innenfor de ulike fagområdene. Betydningen og bruksområdet til algoritmisk tenkning må derfor defineres innenfor hvert enkelt fagområde. Også Wing (2006) poengterte at man må hente ut konseptene og ferdigheten fra algoritmisk tenkning og undervise eksplisitt i hvordan disse kan benyttes innenfor andre kontekster. Å lære algoritmisk tenkning handler ikke om å lære alle å bli fremtidige informatikere, men å vise hvordan ferdigheter innenfor algoritmisk tenkning også kan komme til nytte innenfor andre yrker (Voogt et al., 2015).

I rapporten ble det spesifisert at algoritmisk tenkning skulle være en kjernekompetanse innenfor digital teknologi, og at algoritmisk tenkning derfor måtte ha betydelig plass i faget (Sanne et al., 2016). Algoritmisk tenkning ble definert som evnen til å kunne abstrahere, gjennomgå informasjon systematisk, tolke og forstå ulike representasjonsformer, modellere, samt resonnere iterativt og på tvers av parallelle strukturer. Flere av disse ferdighetene, slik som abstraksjon, gjentakelse, generalisering, samt å forstå og beherske ulike representasjoner og modeller, finner man igjen i forskningsbaserte rammeverkene, slik som hos Shute et al. (2017), eller Weintrop et al. (2016). Algoritmisk problemløsning ble også anbefalt som en del av et det nye, obligatoriske faget, og ble beskrevet som evnen til å kunne omforme et generelt problem til et problem som kunne løses gjennom programmering (Sanne et al., 2016). Denne evnen trekkes av flere inn som en viktig del av algoritmisk tenkning (Bocconi et al., 2022; Wing, 2006).

Programmering defineres av Sanne et al. (2016) som evnen til å oversette en algoritme til et programmeringsspråk slik at en datamaskin kan gjennomføre kommandoene. En mer anerkjent definisjon på programmering er at programmering er en aktivitet hvor man analyserer et problem, utvikler en løsning på problemet og implementerer denne løsningen (Bocconi et al., 2022). Det Sanne et al. (2016) beskriver som programmering, er det som internasjonalt gjerne betegner som koding (Bocconi et al., 2022). Å utforme en algoritme handler om evnen til å lage en presis sekvens av kommandoer (Sanne et al., 2016). Algoritme, slik det beskrives i rapporten til Sanne et al. (2016), er altså ikke synonymt med algoritmisk tenkning eller standard algoritmer i matematikk (Nordby et al., 2022). Begrepene programmering, koding og algoritme har ikke nødvendigvis fastsatte definisjoner, eller blir forstått på samme måte innenfor ulike kontekster eller i litteratur. Nordby et al. (2022) påpekte hvordan oversettelsen av algoritmisk tenkning kan føre til forvirring fordi den kan bli forvekslet med andre, lignende begrep som standard algoritmer i matematikk. Det er ikke utenkelig at man kan oppleve samme forvirring rundt de andre informatikkinspirerte

begrepene. Både programmering, koding, og algoritmer er begreper hentet fra et fagområdet få lærere har erfaring fra, og er derfor begrepet få lærere har god kjennskap til fra tidligere (Bocconi et al., 2022). Når disse begrepene blir tatt ut av sin opprinnelige kontekst og over i andre fagområder, uten å ha klare konsensusdefinisjoner, kan dette tenkes å føre til misforståelser og forvirring for hva begrepene egentlig innebærer.

Det ble i rapporten spesifisert at å formulere presise algoritmer ikke bare skulle være et mål i det nye faget, men det skulle også være et verktøy for å løse relevante problemer i de andre skolefagene (Sanne et al., 2016). I rapporten ble det anbefalt at elevene skulle lære å skrive algoritmer og programmere, men dette skulle ikke være mål i seg selv. Algoritmer og programmering skulle i det nye faget være verktøy som elevene kunne ta med seg inn i andre fag, til andre problemstillinger og andre kontekster. Dette samsvarer med annen forskning som også poengterer at man må skille mellom algoritmisk tenkning og programmering (Lu & Fletcher, 2009; Shute et al., 2017; Vinnervik & Bungum, 2022; Voogt et al., 2015; Yadav et al., 2017). Som nevnt tidligere er det ingen konsensusdefinisjon på hva som ligger i begrepet algoritmisk tenkning (Barr & Stephenson, 2011; Bocconi et al., 2022; Brennan & Resnick, 2012; Ketelhut et al., 2020; Shute et al., 2017). Likevel er det gjerne tre typer definisjoner som går igjen i forskningslitteratur (Bocconi et al., 2022). Den første definisjonen handler om algoritmisk tenkning som en tenkemåte hvor man kommer frem til løsninger som kan realiseres ved hjelp av en datamaskin. En slik definisjon finner vi igjen hos Sanna et al. (2016) sin beskrivelse av algoritmisk problemløsning. Den andre typen definisjon i forskningslitteratur ser på algoritmisk tenkning om en ren problemløsningsprosess (Bocconi et al., 2022). I rapporten står det «Nøkkelkompetanser i digital teknologi er systematisk problemløsning, programmering og forståelse av sammenhenger og komponenter.» (Sanne et al., 2016, s. 7). Denne beskrivelsen er av nøkkelkompetanser som inngår i det anbefalte faget, og ikke nødvendigvis av algoritmisk tenkning. Samtidig er det tydelig at forfatterne anser ren problemløsning som viktig i et nytt teknologibasert fag. Den siste typen definisjonen i forskningslitteratur handler om hvordan ulike ferdigheter innenfor algoritmisk tenkning kan overføres og benyttes for å løse problemer innenfor flere ulike kontekster og fagdisipliner (Bocconi et al., 2022). I rapporten til Sanne et al. (2016) presiseres det at algoritmisk tenkning også må ses på som et verktøy for problemløsning på tvers av fag og aktualiseres i andre fag enn bare realfagene. Beskrivelsene av hva som skal inngå i det nye, anbefalte faget kan derfor sies å være svært bred og til en viss grad omfatte alle de tre typene definisjoner på algoritmisk tenkning som går igjen i forskningslitteraturen. Bocconi et al. (2022) sier også at man kan se

på algoritmisk tenkning som en spesiell eller generell ferdighet. Som en spesiell ferdighet innebærer algoritmisk tenkning problemløsning innenfor informatikk eller programmering. Algoritmisk tenkning som en generell ferdighet derimot handler om å systematisk problemløsning av mer hverdagslige og generelle utfordringer. I rapporten til Sanne et al. (2016) blir algoritmisk tenkning beskrevet som en generell ferdighet.

For å få en vellykket implementering av et nytt tema i utdanning kreves det både endring i styringsdokumenter, samt kompetanseheving hos lærerne (Barr & Stephenson, 2011; Yadav et al., 2017). Sanne et al. (2016) poengterte også i sin rapport at skulle det nye faget bli vellykket ville det kreve utvikling av lærerkompetanse, dedikerte ressurser og engasjement fra skoleeier og skoleledelse. Et slikt fag ville både kreve generell kompetanse hos lærerne innenfor teknologi, men også mer spesialisert programmeringskompetanse. Dette poenget underbygges også i annen forskningslitteratur. Bocconi et al. (2022) trekker frem at de færreste lærere har bakgrunn fra informatikk. Dette gjør at de vil ha behov for faglig og pedagogisk videreutdanning for å kunne undervise i informatikkinspirerte konsepter på en god måte. Det finnes flere eksempler på hvordan man kan undervise for å utvikle elevens evne til algoritmisk tenkning på en morsom og motiverende måte. Utfordringen med alle disse aktivitetene er at det krever høy kompetanse innenfor det spesifikke fagområdet, noe de færreste lærerne som skal undervise i algoritmisk tenkning har per dags dato. Det må derfor tilrettelegges for kurs og utdanning som øker lærernes faglige og pedagogiske kompetanse. I tillegg vil et viktig tiltak for å heve lærernes kompetanse være å etablere gode forum for samarbeid rundt tematikken innad i kollegiet og på tvers av skoler. Dette vil gi en god arena for å øke kompetansen hos lærerne gjennom erfaringsdeling og konkrete tips, noe som igjen kan bidra til å øke lærernes selvtillit på området.

Da LK20 ble innført var det ikke opprettet et nytt obligatorisk fag slik det ble anbefalt i rapporten til Sanne et al. (2016). I læreplanen inngår digitale ferdigheter som en grunnleggende ferdighet i alle fag på lik linje med å lese, skrive og regne. Beskrivelsen av digitale ferdigheter fokuserer hovedsakelig på elever som brukere av digitale verktøy og ikke på hvilke konsepter som ligger bak digitale verktøy, eller hvordan elevene selv kan utvikle lignende verktøy (Vinnervik & Bungum, 2022). I stedet for opprettelsen av et nytt fag var begrepene algoritmisk tenkning, programmering og algoritme plassert inn i læreplanene til matematikk, naturfag, kunst og håndverk og musikk (Bocconi et al., 2022; Vinnervik & Bungum, 2022). Algoritmisk tenkning som begrepet nevnes i LK20 kun i den overordnede delen av matematikkfaget. Her defineres begrepet hovedsakelig som en

problemløsningsferdighet (Vinnervik & Bungum, 2022). Sitatet under er tatt direkte fra læreplanen i matematikk:

Problemløsning i matematikk handler om at elevene utvikler en metode for å løse et problem de ikke kjenner fra før. Algoritmisk tenkning er viktig i prosessen med å utvikle strategier og framgangsmåter for å løse problemer og innebærer å bryte ned et problem i delproblemer som kan løses systematisk. Videre innebærer det å vurdere om delproblemene best kan løses med eller uten digitale verktøy. Problemløsning handler også om å analysere og omforme kjente og ukjente problemer, løse dem og vurdere om løsningene er gyldige. (Utdanningsdirektoratet, 2020a)

Begrepet algoritme benyttes kun i ett kompetansemål, og dette er i matematikk etter 4.trinn: «Lage algoritmer og uttrykke dem ved bruk av variabler, vilkår og løkker.»

(Utdanningsdirektoratet, 2020a). Etersom algoritme her knyttes til sentrale konsepter innenfor programmering, (variabler, vilkår og løkker) omhandler dette kompetansemålet algoritmer innenfor programmering, og er ikke det samme som algoritmisk tenkning.

Begrepet programmering derimot nevnes i flere kompetansemål i LK20. I læreplanen for matematikk benyttes programmering i ett kompetansemål både etter 5., 6., og 7. trinn.

Begrepet algoritmer i forbindelse med programmering benyttes i kompetansemålene i matematikk etter 8. trinn, mens begrepet programmering benyttes i kompetansemålene etter 9., og 10.trinn (Utdanningsdirektoratet, 2020d, 2020f, 2020e). Programmering trekkes også frem i ett kompetansemål i kunst og håndverk etter 7. trinn.

I læreplanen for naturfag brukes programmering som begrep i et kompetansemål etter 7.trinn og et kompetansemål etter 10. trinn (Utdanningsdirektoratet, 2020c, 2020g).

Kompetansemålet i naturfag etter 7. trinn innebærer at elevene skal «Utforske, lage og programmere teknologiske systemer som består av deler som virker sammen.»

(Utdanningsdirektoratet, 2020c). Med utgangspunkt i formuleringen av dette kompetansemålet er ikke inntrykket at elevene skal programmere for programmerings skyld, men heller at de skal benytte programmeringen som et verktøy i arbeidet med større oppgaver. I kompetansemålet etter 10. trinn står det at elevene skal kunne «Bruke programmering til å utforske naturfaglige fenomener» (Utdanningsdirektoratet, 2020g). Slik kompetansemålet etter 10. trinn er formulert kommer det tydelig frem at formålet med programmering på ungdomstrinnet er at programmering skal benyttes som et verktøy for å lære fag. Under kjerneelementet «Teknologi» i naturfag står det at «Elevene skal forstå, skape og bruke teknologi, inkludert programmering og modellering, i arbeid med naturfag.»

(Utdanningsdirektoratet, 2020b). Kompetansemålene i naturfag, sammen med formuleringen

under kjerneelementet «Teknologi» i naturfag, gir uttrykk for at programmering og teknologi skal aktualiseres i naturfag og brukes som et faglig verktøy.

For at programmering skal kunne benyttes som et faglig verktøy må elevene ha en viss programmeringskompetanse (Vinnervik & Bungum, 2022). En slik bruk av programmering vil også øke kravene til lærerne betraktelig, ettersom dette ikke bare betyr at lærerne må kunne programmere, men også at de har nok kompetanse til å koble programmeringen opp mot fag. Dette er krevende ettersom kun svært få lærere har erfaring med algoritmisk tenkning, og da spesielt programmering, fra egen skolegang eller høyere utdanning (Andersen et al., 2023). Samtidig står det ikke noe sted i læreplanen for naturfag hva programmering egentlig innebærer. Det som står er at programmering ikke er et mål i seg selv, men en ferdighet elevene kan benytte i arbeidet med faget. Noe som ikke nødvendigvis gir lærerne en forklaring på hva programmering egentlig innebærer. At programmering er en del av en større teknologisk forståelse og ses på som et faglig verktøy er til en viss grad i tråd med anbefalingene til Sanne et.al (2016). Samtidig har teknologi og programmering fått mye mindre plass i LK20 enn det som ble anbefalt i rapporten. Dette kommer også frem i studien til Vinnervik og Bungum (2022), hvor det ble konkludert med at programmering er den mest fremtredende ferdigheten tilknyttet algoritmisk tenkning i LK20. Ifølge denne studien blir programmering i LK20 beskrevet som en metode og et verktøy i arbeid med andre fag, og ikke som et eget kunnskapsområde.

Leser man LK20 overordnet og på tvers av fag signaliserer beskrivelsene av digitale ferdigheter høyere ambisjoner tilknyttet hva dette innebærer at elevene skal lære enn det som kommer til uttrykk i det enkelte fags kompetansemål (Vinnervik & Bungum, 2022).

Læreplanen legger til rette for en bredere innfallsvinkel når man skal undervise i algoritmisk tenkning som går utover å bare lære elevene og programmere, men en slik innfallsvinkel krever høy lærerkompetanse og tydelig avsatt tid (Vinnervik & Bungum, 2022). Andersen et al. (2023) har også undersøkt de norske styringsdokumentene for å få et bilde av hvordan algoritmisk tenkning defineres. Konklusjonene deres er at algoritmisk tenkning i norske styringsdokumenter blir beskrevet på to måter. Den ene beskrivelsen innebærer algoritmisk tenkning som en problemløsningsprosess og programmeringsprosess, mens den andre beskrivelsen handler om algoritmisk tenkning som et verktøy for å lære fag. Samtidig, med utgangspunkt i hvor fremtredende programmering er i kompetansemålet, er det ikke lærerne er like bevisst de andre aspektene ved algoritmisk tenkning i LK20. Ettersom det er

kompetansemålene som blir mest brukt i lærernes hverdag, som ofte er førende for hvordan lærebøker designes (Vinnervik & Bungum, 2022), er det sannsynligheten at programmeringsfokuset i kompetansemålene da vil bli gjenspeilet i undervisningen og i lærebøker. En konsekvens av programmeringsfokuset kan derfor tenkes å bli et rent programmeringsfokus, noe som da vil kunne gi en fragmentert og begrenset realisering av intensjonen i læreplanen (Vinnervik & Bungum, 2022).

Som en ressurs for lærere har Utdanningsdirektoratet laget en nettside som omhandler algoritmisk tenkning (Utdanningsdirektoratet, 2019). Her presenteres en overordnet beskrivelse av hva som menes med algoritmisk tenkning, nøkkelbegreper og arbeidsmåter, samt en beskrivelse av «den algoritmiske tenkeren» med en tilhørende figur. Figuren av «den algoritmiske tenkeren» baserer seg på Barefoot Computing sin modell (Barefoot Computing, 2014). På nettsiden beskrives algoritmisk tenkning innledningsvis som en problemløsningsmetode, og det står at «litt forenklet kan vi si at det er ‘å tenke som en informatiker’ når vi skal løse problemer eller oppgaver.» (Utdanningsdirektoratet, 2019). Wing (2006, s. 2) bruker en lignende formulering når hun skriver «Thinking like a computer scientist means more than being able to program in computer. It requires thinking at multiple levels of abstraction». Hun skriver også «Professors of computer science should teach a course called «Ways to Think Like a Computer Scientist» [...]» (Wing, 2006, s. 3). Utdanningsforbundets beskrivelse ‘å tenke som en informatiker’, gir derfor tydelige assosiasjoner til Wing sin artikkel, og dermed også hennes forklaring og definisjon på algoritmisk tenkning. Etter innledningen står det videre på nettsiden at det å tenke algoritmisk også innebærer å bruke sin teknologiske kompetanse til å løse deler av et problem. Det spesifiseres at det med dette menes en forståelse for hvilke oppgaver teknologi kan hjelpe oss med å løse, og hvilke oppgaver som bør løses av mennesker. En formulering som også har store likhetstrekk med formuleringene i Wing sin artikkel (2006). Samtidig som beskrivelsen av algoritmisk tenkning på Utdanningsdirektoratet (2019) sin nettside virker å være inspirert av Wing (2006), er deres beskrivelsen smalere da den hovedsakelig fokuserer på algoritmisk tenkning som en problemløsningsprosess. Det står også at den algoritmiske tenkeren må være skapende, samt ha en nysgjerrig og utforskende tilnærming til problemløsning (Utdanningsdirektoratet, 2019). Å være skapende tolkes her som tilsvarende det å kunne lage en form for produkt. Å lage og å utforske forstås derfor også, basert på Utdanningsdirektoratet (2019) sin ressursnettside for lærere, som viktige ferdigheter tilknyttet det å være en algoritmisk tenker.

På nettsiden til Utdanningsdirektoratet (2019) står det at «Det finnes flere definisjoner av algoritmisk tenkning, men hovedtrekkene er sammenfallende». Det er ingen tvil om at det finnes mange ulike definisjoner på hva som ligger i algoritmisk tenkning (Bocconi et al., 2022; Shute et al., 2017). Hvorvidt de ulike definisjonene er sammenfallende derimot, kan diskuteres. Det er forskjell på smalere definisjoner og rammeverk tilknyttet algoritmisk tenkning slik som hos Brennan og Resnick (2012) og Grover og Pea (2013), sammenlignet med bredere og mer generelle rammeverk som hos Weintrop et al. (2016) og Shute et al. (2017). Tar man utgangspunkt i Weintrop et al. (2016) sitt rammeverk passer beskrivelsene på Utdanningsdirektoratet sine sider inn under kategorien «algoritmisk tenkning og problemløsningspraksiser» (Vinnervik & Bungum, 2022). Samtidig dekker ikke beskrivelsen til Utdanningsdirektoratet noen av de tre andre kategoriene i Weintrop et al. (2016) sitt rammeverk. Ifølge Grover og Pea (2013) er evnen til abstraksjon en nøkkelferdighet innenfor algoritmisk tenkning, og det som skiller algoritmisk tenkning fra andre tenkemåter. På nettsiden til Utdanningsdirektoratet (2019) trekkes abstraksjon frem som ett av seks nøkkelbegrep tilknyttet «den algoritmiske tenkeren», men får lite fokus ut over dette. De andre nøkkelbegrepene tilknyttet «den algoritmiske tenkeren» er logikk, algoritmer, dekomposisjon, mønstre og evaluering. Når det gjelder typisk aktiviteter tilknyttet algoritmisk tenkning nevnes det å fikle, skape, feilsøke, holde ut og samarbeide. I Shute et al. (2017) sitt rammeverk for algoritmisk tenkning utgjør dekomposisjon, abstraksjon, algoritmer og feilsøking fire av seks kategoriene. Beskrivelsene av nøkkelbegrep og aktiviteter tilknyttet «den algoritmiske tenkeren» er derfor til en viss grad sammenfallende med Shute et al. (2017) sitt rammeverk.

Det er altså ulikt i hvor stor grad beskrivelsene til Utdanningsdirektoratet (2019) av algoritmisk tenkning dekker de ulike rammeverkene som har blitt brukt som eksempler her. Både Shute et al. (2017) og Weintrop et al. (2016) sine rammeverk er basert på omfattende datainnsamling tilknyttet hva som legges i algoritmisk tenkning internasjonalt, og utviklet med utgangspunkt i dette. Hvis definisjonene på algoritmisk tenkning da i stor grad er sammenfallende slik Utdanningsdirektoratet (2019) skriver, er det interessant at deres definisjon ikke i større grad dekker rammeverkene til eksempelvis Shute et al. (2017) eller Weintrop et al. (2016). Samtidig skriver Bocconi et al. (2022) at de ulike definisjonene på algoritmisk tenkning i forskningslitteraturen gjerne har likhetstrekk, og derfor kan plasseres innenfor tre ulike kategorier. Algoritmisk tenkning som problemløsningsmetode, som beskrevet hos Utdanningsdirektoratet (2019), er en av disse tre kategoriene.



Det har også blitt utviklet kompetansepakker for digital kompetanse i skolen. En av disse kompetansepakkene omhandler spesifikt programmering og algoritmisk tenkning og er utviklet for å gi lærere en forståelse av hva programmering er og hvordan man kan jobbe med programmering i fagene (Utdanningsdirektoratet, 2023). Kompetansepakken tar utgangspunkt i kompetansemålene fra LK20. På bakgrunn av oppgavens omfang diskuteres ikke kompetansepakkenes innhold i denne oppgaven.

### 3. Tidligere forskning

#### 3.1 Læreres forståelse av algoritmisk tenkning

Det er interessant å undersøke hvordan lærere forstår begrepet algoritmisk tenkning. Ettersom det er et til dels nytt begrep i de fleste læreplaner, og det ikke finnes én tydelig definisjon på hva begrepet innebærer, har flere studier undersøkt hvordan nyutdannede og praktiserende lærere forstår algoritmisk tenkning. Hvordan lærerne forstår algoritmisk tenkning som konsept har direkte innvirkning på hvordan lærerne velger å anvende konseptet i egen undervisning (Cabrera, 2019). Lærernes forforståelse av konseptet kan også påvirke hvordan lærerne tilegner seg ny kunnskap innenfor området. Yadav et al. (2017) undersøkte ved bruk av spørreskjema hvordan lærerstudenter forstod begrepet algoritmisk tenkning. I denne studien ble det identifisert tre ulike forståelser som gikk igjen hos lærerstudentene. Den mest fremtredende forståelsen var algoritmisk tenkning som en problemløsningsstrategi. En annen forståelse var at algoritmisk tenkning bestod av flere andre tenkemetoder som eksempelvis matematisk tenkning, eller «å tenke som en datamaskin». Lærerstudentene ga inntrykk av en forståelse for matematisk tenkning som en ferdighet innenfor algoritmisk tenkning. Til tross for at både algoritmisk tenkning og matematisk tenkning har problemløsning til felles er det likevel andre aspekter som skiller dem fra hverandre, og gjør at de beskrives som to ulike tenkemåter (Shute et al., 2017). Den siste forståelsen som ble identifisert hos lærerstudentene handlet om at algoritmisk tenkning som en måte å løse et problem eller en oppgave ved hjelp av en datamaskin (Yadav et al., 2017).

Forståelsen av algoritmisk tenkning hos praktiserende lærere har også blitt undersøkt. Cabrera (2019) fant, i likhet med Yadav et al. (2017), at flere praktiserende lærere også oppfattet algoritmisk tenkning som en problemløsningsstrategi. Forståelsen av algoritmisk tenkning som «å tenke som en datamaskin» var ble også identifisert hos de praktiserende lærerne (Cabrera, 2019). En av hovedforskjellene mellom forståelsen hos de nyutdannede lærerne og de praktiserende lærerne var at flere av de praktiserende lærerne hadde en veldig smal forståelse av algoritmisk tenkning (Cabrera, 2019; Yadav et al., 2017). Denne forståelsen ble ikke identifisert hos de nyutdannede lærerne. Den smale forståelsen innebar at algoritmisk tenkning ble forstått som synonymt med informatikk og programmering. Hos de praktiserende lærerne med denne oppfattelsen ble ikke algoritmisk tenkning sett på som en ferdighet eller tenkemåte som kunne benyttes på tvers av fag, eller utenfor rene programmeringsaktiviteter.

Å lære programmering ble sett på som selve målet med algoritmisk tenkning, og ikke som et verktøy for å utvikle algoritmisk tenkning hos elevene. En annen forskjell mellom de nyutdannede lærerne og de praktiserende lærerne var at flere praktiserende lærere forstod algoritmisk tenkning som synonymt med å benytte teknologi og digitale verktøy i undervisningen (Cabrera, 2019). Hos disse lærerne ble det oppfattet som å jobbe med algoritmisk tenkning når elevene brukte applikasjoner på iPad, ulike søkemotorer for å finne informasjon, eller se på videoer på YouTube.

Studier i Norge har også undersøkt læreres forståelse av algoritmisk tenkning. Kravik et al. (2022) gjennomførte intervjuer med lærere i matematikk, naturfag, kunst og håndverk og musikk på mellomtrinnet. Målet med studien var å undersøke hvordan disse lærerne forstod begrepene programmering, algoritmisk tenkning og hvordan de vurderte sin egen kompetanse innenfor dette området. Studien viste at lærerne i stor grad brukte begreper tilknyttet algoritmer og mønstre for å forklare programmering. Ut over dette ga lærerne begrensede forklaringer på hva programmering innebar. Når lærerne videre skulle forklare algoritmisk tenkning uttrykte de usikkerhet rundt begrepet, og det stor variasjon i hvordan det ble forstått. Algoritmisk tenkning ble i stor grad forklart på samme måte som programmering, noe som tydet på en begrenset forståelse for begrepet. Dette til tross for at de fleste lærerne som ble intervjuet selv anså å ha middels eller høy kompetanse på området. Studien viste at det var stor variasjon i hvor mye kursing lærerne hadde gjennomgått og lærerne selv trakk frem begrenset tilgang på ressurser og opplæring som en utfordring. Nordby et al. (2022) har også undersøkt læreres forståelse av algoritmisk tenkning, men i denne studien ble det gjennomført intervjuer med matematikklærere. I likhet med funnene til Kravik et al. (2022), hadde også lærerne hos Nordby et al. (2022) problemer med å definere hva algoritmisk tenkning innebar. De diskuterte algoritmisk tenkning som algoritmer i form av å følge en oppskrift, å lage en oppskrift, eller som programmering. Dette er kanskje ikke overraskende ettersom det er disse begreper som benyttes mest i tilknytning til algoritmisk tenkning i LK20 (Nordby et al., 2022; Vinnervik & Bungum, 2022).

Både internasjonal forskning og forskning i norsk kontekst viser at det er stor variasjon i hvordan lærere forstår begrepet algoritmisk tenkning (Cabrera, 2019; Kravik et al., 2022; Nordby et al., 2022; Yadav et al., 2017). Forståelsen av algoritmisk tenkning som i stor grad synonymt med programmering ble funnet i tre av de fire studiene som er gjennomgått her. I de internasjonale studiene påpekes viktigheten av kompetanseheving hos lærere for at undervisning i algoritmisk tenkning skal bli vellykket. Lærerne i de norske studiene oppgir

behov for økt kompetanse og at de har fått lite tid og ressurser til å drive med kompetanseheving. Manglende kompetanseheving hos de norske lærerne kan derfor være en forklaring på den manglende forståelsen av algoritmisk tenkning som ble identifisert i begge de norske studiene.

### 3.2 Implementering av algoritmisk tenkning

Bakgrunnen for hvorfor ulike land velger å implementere algoritmisk tenkning i sine læreplaner varierer (Bocconi et al., 2022). Samtidig var konklusjonen til Bocconi et al. (2022) at hovedmotivasjonen hos landene for å implementere algoritmisk tenkning i sine læreplaner, handlet om å fremme kompetanser for det 21 århundre. En viktig del av disse kompetansene innebærer ferdigheter innenfor informatikkinspirerte konsepter som algoritmisk tenkning (Angeli & Giannakos, 2020; Larson & Miller, 2011). Flere land rapporterte at de gjennom å implementere algoritmisk tenkning i sine læreplaner ønsket å øke elevers problemløsningsferdigheter og evnen til logisk tenkning (Bocconi et al., 2022). I tillegg ønsket nesten alle landene som deltok i denne studien også å øke elevers kompetanse innenfor programmering og koding. Disse tre perspektivene vitner om et syn på algoritmisk tenkning som en allmenn ferdighet relevant for alle elever. En annen grunn for implementering som ble trukket frem av flere land var ønsket om å få elevene inn i arbeidslivet i fremtiden, og da gjerne innenfor informatikk. I Norge er implementeringen av algoritmisk tenkning og programmering i LK20 basert på antagelsen om at dette vil tilrettelegge for å utvikle elevenes problemløsningsferdigheter, logiske ferdigheter og digitale kompetanse (Bocconi et al., 2022).

En vellykket implementering av algoritmisk tenkning i grunnskolen vil kreve at lærerne som skal undervise er godt forberedt (Cabrera, 2019; Yadav et al., 2017). En slik forberedelse innebærer at lærerne får økt faglig- og pedagogisk kompetanse tilknyttet temaet. Som en del av den faglige kompetansen må lærerne få en god forståelse for hva begrepet algoritmisk tenkning innebærer. Den pedagogiske kompetansen handler om hvordan denne forståelsen kan videreformidles til elevene på best mulig måte. I tillegg vil teknologisk kompetanse hos lærerne være viktig ettersom lærerne må vite hvilke teknologiske verktøy som er tilgjengelige, og hvordan disse kan benyttes som hjelpemidler i undervisningen tilknyttet algoritmisk tenkning. Dette innebærer en forståelse hos lærerne for hvordan de teknologiske verktøyene representerer ulike ferdigheter innenfor algoritmisk tenkning, hvordan man på best mulig måte kan benytte disse som et pedagogisk verktøy, samt hvordan disse verktøyene gjør det mulig for elever å anvende algoritmisk tenkning (Cabrera, 2019). Til tross for viktigheten av å

forberede lærerne rapporterte både lærerne hos Kravik et al. (2022) og hos Nordby et al. (2022) å ha fått lite tid til å videreutvikle sin kunnskap rundt algoritmisk tenkning og programmering. Nordby et al. (2022) konkluderte med at det mangler kunnskap hos lærerne om hvordan algoritmisk tenkning kan benyttes som verktøy for å hjelpe elevene med å lære matematikk.

Hvordan lærere ser for seg å innføre algoritmisk tenkning i egen undervisning har blitt undersøkt i flere studier. Yadav et.al (2017) fant at hvordan nyutdannede lærerne så for seg å implementere algoritmisk tenkning hadde stor sammenheng med hvordan de selv oppfattet og definerte algoritmisk tenkning. Flere av de nyutdannede lærerne så for seg at de ville innføre algoritmisk tenkning gjennom å la elevene benytte teknologi som en del av undervisningen. De andre nyutdannede lærerne så for seg å innføre algoritmisk tenkning gjennom ulike problemløsningsaktiviteter, da også gjerne på tvers av fag og ulike temaer. Hvordan de nyutdannede lærerne så for seg å innføre algoritmisk tenkning hadde derfor stor sammenheng med om lærerne hovedsakelig forstod algoritmisk tenkning som en problemløsningsstrategi, eller som problemløsning ved hjelp av datamaskiner og teknologi (Yadav et al., 2017). Lignende funn har også blitt gjort hos praktiserende lærere hvor man fant en tydelig sammenheng mellom hvilke utfordringer lærerne forventet å møte i forbindelse med implementering av algoritmisk tenkning, og hvordan de selv forstod dette konseptet (Cabrera, 2019). Her fant man at lærere som var skeptiske til å innføre algoritmisk tenkning gjerne forstod begrepet som synonymt med programmering, noe de igjen opplevde som krevende å lære seg. Programmering ble av disse lærerne igjen oppfattet som noe kun en begrenset og begavet del av befolkningen drev med, og som noe som krevde mye erfaring og kunnskap. Hos Nordby et al. (2022) fant man at ettersom matematikklærerne som ble intervjuet var usikre på begrepet algoritmisk tenkning, var de også usikre på hvordan de skulle implementere algoritmisk tenkning i egen undervisning. I denne studien ble det identifisert to ulike syn på hva algoritmisk tenkning innebar i undervisning (Nordby et al., 2022). Det første synet handlet om en forståelse av algoritmisk tenkning som et tillegg til det allerede eksisterende pensumet, og derfor fristilt fra resten av faget. Lærerne med denne forståelsen underviste da utelukkende i programmering uten å knytte det opp mot resten av matematikkfaget. Det andre synet innebar en forståelse av algoritmisk tenkning som noe de allerede drev i form av undervisning om algoritmer i matematikk. Resultatet av disse to synene på hva algoritmisk tenkning innebar i praksis førte til at lærerne enten kun hadde fokus på matematikk, eller utelukkende på programmering og koding.

Algoritmisk tenkning og programmering nevnes ofte sammen i forskningslitteratur (Bocconi et al., 2022). Dette skyldes at programmering er et godt verktøy og hjelpemiddel for å utvikle algoritmisk tenkning (Bocconi et al., 2022; Voogt et al., 2015). Hovedforskjellen mellom de nyutdannede lærerne og de praktiserende lærerne var forståelsen hos de praktiserende lærerne av algoritmisk tenkning som synonymt med informatikk og programmering. Det er flere land som de siste årene har innført ulike varianter av algoritmisk tenkning eller andre konsepter fra informatikk i sine lærerplaner (Bocconi et al., 2022). I forbindelse med denne implementeringen har flere land valgt å vektlegge nettopp programmering i sine læreplaner istedenfor algoritmisk tenkning. Bocconi et al. (2022) fant at hele 27 av de 28 europeiske landene i deres studie refererte til programmering og koding i sine læreplaner, noe som gjorde dette til de mest brukte konseptene fra informatikk i læreplansammenheng.

## 4. Teoretisk rammeverk

I dette kapittelet presenteres to ulike teoretiske rammeverk tilknyttet algoritmisk tenkning. Disse to teoretiske rammeverkene danner grunnlaget for oppgavens analytiske rammeverk.

### 4.1 Algoritmisk tenkning som et sett med generelle ferdigheter

Den digitale utviklingen gjør at det ikke lenger er nok å bare kunne lese og skrive. For å henge med på den raske teknologiske utviklingen er det behov for andre ferdigheter, slik som algoritmisk tenkning. Samtidig er det slik at hvis algoritmisk tenkning er noe alle skal lære må begrepet være tydelig definert, både hva det innebærer generelt og hva det betyr i en utdanningssammenheng (Shute et al., 2017). Dette var bakgrunnen for at Shute et al.(2017) utviklet sitt rammeverk og sin definisjon på algoritmisk tenkning. Målet med rammeverket var å trekke frem konseptuelle ferdigheter som assosieres med algoritmisk tenkning og hvordan disse ferdighetene kan benyttes i ulike fag på grunnskolenivå. Rammeverket ble laget med på bakgrunn av en omfattende litteraturstudie. I studien ble det identifisert seks ferdigheter som ofte ble trukket frem i forskningslitteratur i forbindelse med algoritmisk tenkning. Disse ferdighetene med tilhørende beskrivelser utgjør det teoretiske rammeverket. Rammeverket skiller seg fra tidligere rammeverk tilknyttet algoritmisk tenkning fordi det ikke ble knyttet opp mot et spesifikt fag, område eller tema. Tvert imot skulle ferdighetene i dette rammeverket være generelle og gjeldene for algoritmisk tenkning uavhengig av kontekst.

Tabell 1. Oversikt over det teoretiske rammeverket til Shute et al. (2017) med ferdigheter tilknyttet algoritmisk tenkning, samt beskrivelse av disse. Rammeverket er oversatt fra engelsk.

<b>Ferdighet</b>	<b>Norsk oversettelse</b>	<b>Beskrivelse</b>
<i>Decomposition</i>	<b>Dekomposisjon</b>	Å dele et komplekst problem/system inn i mindre, mer håndterbare problemer. De mindre problemene er ikke tilfeldige deler, men funksjonelle elementer som samlet omfatter hele problemet/systemet. Dekomposisjon handler også om å kunne bruke systematiske prosesser for å takle hvert av de mindre problemene.

*Abstraction*

## **Abstraksjon**

Handler om å kunne abstrahere og trekke ut essensen av et komplekst system. Abstraksjon har tre underkategorier:

1. **Datainnsamling og analyse:** Hente inn den mest relevante og viktige informasjonen fra flere kilder og forstå forholdet mellom datasett med flere lag.
2. **Mønsterkjennning:** Identifisere mønstre/regler som ligger til grunn for datastrukturen/informasjonsstrukturen.
3. **Modellering:** Bygge modeller eller simuleringer som representerer hvordan systemet opererer, og/eller hvordan et system vil virke i fremtiden.

*Algorithms*

## **Algoritmer**

Designe logiske og ordnede instruksjoner for å kunne gjengi løsninger på et problem. Instruksjonene kan gjøres av mennesker eller datamaskiner. Fire underkategorier:

1. **Algoritmisk design:** Lage en rekke med ordnede steg for å løse et problem.
2. **Parallellisme:** Gjennomføre et gitt antall steg simultant.
3. **Effektivitet:** Benytte færrest mulig steg for å løse et problem, fjerne redundans og unødvendige steg.
4. **Automatisering:** Automatisere kjøringen av prosedyren når det er behov for å løse lignende problemer.



<i>Debugging</i>	<b>Feilsøking</b>	Oppdage og identifisere feil. Rette opp i disse feilene når en løsning ikke fungerer som ønsket.
<i>Iteration</i>	<b>Gjentakelse</b>	Handler om å gjenta prosessen flere ganger. Gå tilbake og gjenta designprosessen for å hele tiden forbedre løsninger helt til man har oppnådd det ideelle resultatet. Dette sikrer at hvert delproblem løses på best mulig måte og at man unngår løse tråder.
<i>Generalization</i>	<b>Generalisering</b>	Overføre ferdigheter fra algoritmisk tenkning til et bredere spekter av situasjoner/domener for å kunne løse problemer effektivt og nøyaktig.

## 4.2 Algoritmisk tenkning sin rolle i matematikk og naturfag

Etter hvert som samfunnet har blitt stadig mer digitalisert har også matematikk og naturvitenskap endret seg og tilpasset seg denne digitaliseringen. I større grad enn tidligere utgjør i dag bruk av digitale verktøy, teknologiske løsninger, programmering og algoritmisk tenkning en stor del av disse fagområdene (Weintrop et al., 2016). For å gjenspeile teknologiens økende plass innenfor matematikk og naturvitenskap, ønsket Weintrop et al. (2016) å utvikle et teoretisk rammeverk for algoritmisk tenkning innenfor matematikk og naturvitenskap, og hvordan dette kan aktualiseres i skolen. Rammeverket skulle tydeliggjøre hvilken plass algoritmisk tenkning har i matematikk og naturvitenskap, samt hvilke ferdigheter innenfor algoritmisk tenkning elevene kan jobbe med i matematikk og naturfag.

Rammeverket ble utviklet med bakgrunn i eksisterende forskningslitteratur tilknyttet algoritmisk tenkning, intervjuer med matematikere og naturvitere, samt eksisterende undervisningsmateriale tilknyttet algoritmisk temaet. Gjennom dette arbeidet ble det identifisert fire hovedpraksiser for algoritmisk tenkning innenfor matematikk og naturvitenskap. Disse er datapraksiser, modellering og simulering, problemløsning via dataprosessering og systemtenkning. Hver av disse praksisene består igjen av fem til syv underkategorier. Oversikt over det teoretiske rammeverket med hovedpraksiser og tilhørende

underkategorier er beskrevet i tabellen under. De fire hovedpraksisene er beskrevet under tabellen. For beskrivelse av hver underkategori se Weintrop et al. (2016).

Tabell 2. Oversikt over det teoretiske rammeverket til Weintrop et al. (2016). Rammeverket er oversatt fra engelsk.

<b>Data praksiser</b>	<b>Modellering og simulering</b>	<b>Problemløsning via dataprosessering</b>	<b>Systemtenkning</b>
Innsamling av data	Bruke datamodeller for å forstå et konsept	Klargjøre problemer for teknologiske løsninger	Undersøke et komplekst system som en helhet
Generere data	Bruke datamodeller for å finne og teste løsninger	Programmere	Forstå sammenhengende innenfor et system
Manipulere data	Vurdere datamodeller	Velge effektive teknologiske verktøy	Nivåbasert tenkning
Dataanalyse	Designere datamodeller	Vurdere ulike tilnærminger/løsninger på et problem	Viderefremme informasjon om et system
Datavisualisering	Konstruere datamodeller	Utvikle modulbaserte teknologiske løsninger	Definere systemer og håndtere kompleksitet
		Abstraksjon	
		Feilsøking	

Datainnsamling og evidens utgjør grunnlaget for all forskning (Weintrop et al., 2016). Data praksiser i rammeverket handler om hvordan data innenfor matematikk og naturvitenskap samles inn, skapes, analyseres og deles, prosesser som i dag i stor grad utføres ved hjelp av teknologi. Å kunne benytte denne teknologien for å håndtere og forstå store mengder data er derfor en viktig del av matematikk og naturvitenskap, og krever ulike ferdigheter innenfor algoritmisk tenkning. Den andre praksisen i rammeverket handler om praksiser innenfor modellering og simulering. Evnen til å skape, tilpasse og bruke modeller for å beskrive og forklare fenomener og verden rundt oss er en viktig del matematikk og naturvitenskap. Også dette blir i dag i stor grad utført digitalt og ved hjelp datamaskiner. Problemløsning via dataprosesser handler om å utforske problemstillinger innenfor matematikk og naturvitenskap gjennom algoritmisk tenkning. I matematikk og naturvitenskap har man adoptert ulike

praksiser og strategier tilknyttet problemløsning fra informatikk. Denne praksisen innebærer derfor ferdigheter som koding, programmering, utvikling av algoritmer, feilsøking og evnen til abstraksjon. Den siste praksisen er systemtenkning, og handler om hvordan de fleste problemene man møter i dag er komplekse. Disse problemstillingene inkluderer gjerne flere variabler, har mange konsekvenser både direkte og indirekte, og består av flere deler som virker sammen og påvirker hverandre. Mange fundamentale naturvitenskapelige konsepter kan ses på som systemer, og systemtenkning blir derfor viktig. Systemtenkning handler ikke nødvendigvis om algoritmisk tenkning, men ettersom teknologiske verktøy gjør det mulig å utforske og forstå slike komplekse systemer er algoritmisk tenkning viktig for forståelsen av komplekse systemer.

## 5. Metode

Dette kapittelet tar for seg prosessen for datainnsamling og analyse som har dannet grunnlaget for funnene som presenteres i denne masteroppgaven. I dette kapittelet beskrives metoden som har blitt brukt, den analytiske tilnærmingen til datamaterialet og det analytiske rammeverket som har blitt benyttet. Underveis forklares og begrunnes de valgene og som har blitt tatt. Avslutningsvis vurderes oppgavens reliabilitet, validitet og forskningsetiske perspektiver. En slik systematisk og transparent gjennomgang av metode er det som skiller forskning fra hverdags erfaringer (Postholm & Jacobsen, 2018).

### 5.1 Kvalitativ metode

I denne masteroppgaven var målet å få en forståelse for hvordan naturfagslærere på mellomtrinnet oppfatter begrepet algoritmisk tenkning. Når man ønsker å undersøke kvalitet, spesielle kjennetegn eller egenskaper ved et fenomen er kvalitativ metode hensiktsmessig (Johannessen et al., 2016). Kvalitativ metode er også godt egnet til å undersøke et fenomen mer i dybden. Det var nettopp dette som var målet med denne studien; å gå i dybden på læreres forståelse av algoritmisk tenkning. Beskrivelse, forståelse og mening er viktige begreper i kvalitativ forskning (Postholm & Jacobsen, 2018). Et kvalitativ forskningsdesign ble derfor valgt som utgangspunkt for denne masteroppgaven.

Målet med forskning er å få økt kunnskap om virkeligheten (Postholm & Jacobsen, 2018). Hva som anses som virkelighet er avhengig av forskningens ontologiske utgangspunkt. Det ontologiske utgangspunktet bestemmer hvilken virkelighet forskningen ønsker å beskrive, forstå og forklare. I denne masteroppgaven var virkeligheten som skulle studeres skolen sett i et prosessperspektiv. Denne ontologiske tilnærmingen innebærer at skolen blir sett på som et sosialt fellesskap i stadig bevegelse. Dette innebærer blant annet at læreplanen ikke anses som noe «fast», men som noe som oppfattes, diskuteres og fortolkes fortløpende (Postholm & Jacobsen, 2018). I denne masteroppgaven var det nettopp oppfattelsen og forståelsen hos lærere av en bestemt del av læreplanen som skulle studeres.

Det ontologiske utgangspunktet påvirker igjen det epistemologiske utgangspunktet, som handler om hvordan man får kunnskap om denne virkeligheten. Det ontologiske og epistemologiske utgangspunktet er derfor avgjørende for hvilke metoder man benytter i forskningen, som igjen bestemmer hvilken data som samles inn. Utgangspunktet for denne oppgaven er at det er menneskene, i dette tilfellet lærerne, som danner virkeligheten. Det er

deres tolkninger og prioriteringer som avgjør hvordan læreplanen realiseres i klasserommet (Vinnervik & Bungum, 2022). Oppgaven har derfor en fenomenologisk tilnærming ettersom målet blir å beskrive hvordan lærerne opplever verden (Johannessen et al., 2016; Kvarv, 2021). En måte å beskrive lærernes opplevelse av verden på er gjennom det Kvarv (2021) kaller for et «semistrukturert livsverdensintervju». Dette kan oppnås gjennom et kvalitativt forskningsintervju med et fenomenologisk inspirert perspektiv. Postholm og Jacobsen (2018) trekker også frem semistrukturert intervju som en god metode for å få innblikk i en persons perspektiv og forståelse. Det er når mennesker selv får mulighet til å bestemme hva som tas opp i intervjuet at deres erfaringer og opplevelser kommer best frem (Johannessen et al., 2016). Ettersom målet med oppgaven var å beskrive læreres forståelse av et fenomen ble dette valgt som metode for datainnsamling. Nærmere beskrivelse av det kvalitative forskningsintervjuet er gitt under.

## 5.2 Kvalitativt forskningsintervju

I denne masteroppgaven ble den valgte kvalitative metoden semistrukturerte én-til-én intervjuer. Én-til-én intervjuer benyttes når man ønsker dypere og mer detaljerte beskrivelser av en persons forståelse, følelser, erfaringer, oppfatninger, meninger, holdninger og refleksjoner tilknyttet et gitt fenomen (Johannessen et al., 2016). I denne masteroppgaven ble én-til-én intervju benyttet for å få en detaljert beskrivelse av lærernes forståelse av algoritmisk tenkning og tilknyttede kompetansemål. At et intervju er semistrukturert betyr at det gjennomføres med utgangspunkt i en intervjuguide med bestemte temaer, samtidig som det også kan inneholde forslag til spørsmål (Kvale & Brinkmann, 2015). I denne intervjuformen er man også åpen for at personene som intervjues kan ta opp andre temaer enn de som er beskrevet i intervjuguiden (Postholm & Jacobsen, 2018). I forkant av intervjuene ble det utformet en intervjuguide med utgangspunkt i forskningsspørsmålene (For fullstendig intervjuguide se vedlegg 1.). Valget om å ikke ha åpne intervju var basert på at det var flere lærere som skulle intervjues, og ved å gi alle lærerne i stor grad de samme spørsmålene ville det bli lettere å sammenligne de ulike svarene i etterkant (Johannessen et al., 2016). Temaene for intervjuet var kompetansemålet innenfor programmering i naturfag, ferdigheter og aktiviteter tilknyttet dette, samt algoritmisk tenkning. I naturfag er ikke algoritmisk tenkning nevnt, men leser man læreplanen overordnet og på tvers av fag er også dette et kompetansemål knyttet opp mot å lære elevene algoritmisk tenkning (Vinnervik & Bungum, 2022). For å undersøke om lærerne selv var bevisst denne koblingen ble ikke algoritmisk

tenkning nevnt innledningsvis i intervjuet. I en fenomenologisk tilnærming ønsker man at informantene skal gi en beskrivelse av sin virkelighet, og derfor prøver man å begrense egen innvirkning på denne beskrivelsen (Kvale & Brinkmann, 2015). Det var derfor et helt bevisst valg å ikke spørre informantene om algoritmisk tenkning innledningsvis for å unngå å påvirke svarene deres. Fokuset i starten av intervjuet var derfor på kompetansemålet og lærernes forståelse og beskrivelse av innholdet i dette kompetansemålet. Hvis lærerne selv tok i bruk begrepet algoritmisk tenkning fikk de videre spørsmål om hva de la i dette begrepet. Hvis lærerne ikke selv tok i bruk algoritmisk tenkning, ble oppfølgingsspørsmålet om de hadde noen kjennskap til dette begrepet. Teamene for intervjuet var derfor tydelige, men det var lagt inn forslag til spørsmål basert på lærernes svar, noe som er typisk for det semistrukturerte intervjuet.

### 5.3 Utvalg og rekruttering

Lærerne som ble intervjuet i denne masteroppgaven måtte oppfylle to kriterier. Å starte med og definere kriterier for målgruppen, for så å velge ut personer som oppfyller disse kriteriene, kalles gjerne for strategisk utvelgelse (Johannessen et al., 2016). Det første kriteriet for lærerne som skulle delta i denne masteroppgaven var at de underviste naturfag på mellomtrinnet. Begrunnelsen for mellomtrinns lærere var at programmering ikke nevnes i kompetansemålene for naturfag før etter 7.trinn. Man kan derfor ikke forvente at naturfagslærere på småtrinnet har kjennskap til algoritmisk tenkning og programmering. Det andre kriteriet for utvelgelse var at lærerne hadde undervist i naturfag etter innføringen av LK20. Det er først i LK20 at algoritmisk tenkning og programmering har blitt en del av læreplanen, og lærere som har undervist i naturfag før 2020 kan derfor ikke forventes å ha inngående kjennskap til det aktuelle kompetansemålet. Naturfagslærere på mellomtrinnet som har undervist i naturfag etter innføringen i LK20 kan man derimot forvente at har kjennskap til det nye kompetansemålet. Når det kommer til antallet informanter i en kvalitativ studie er tommelfingerregelen at man må ha nok informanter til å belyse problemstillingen (Johannessen et al., 2016). En annen mening er at man har nok informanter når flere informanter ikke lenger bidrar til ny informasjon. Det er formålet med studien som er avgjørende for hvor mange personer som bør intervjues (Kvale & Brinkmann, 2015). På bakgrunn av masteroppgavens begrensede omfang og varighet ble målet i denne oppgaven å få intervjuet minimum fem lærere i naturfag på mellomtrinnet. Ved å også inkludere lærere i andre fag, eller på småtrinnet, ville jeg kunne økt antall informanter betraktelig. Samtidig er

ville ikke dette vært hensiktsmessig for å belyse min problemstilling. Valget ble derfor å begrense det til naturfaglærere på mellomtrinnet ettersom det er viktigere med et relevant utvalg av informanter, enn et stort antall (Johannessen et al., 2016).

Etter å ha definerte kriteriene for deltakelse var neste steg å finne lærere som oppfylte disse kriteriene, og som var villige til å stille på intervju. Gjennom lærerutdanningen har jeg hatt praksis på flere ulike skoler. I disse praksisperiodene ble jeg kjent med lærere som oppfylte kriteriene for deltakelse. Det første som ble gjort var derfor å kontakte de lærerne jeg kjente som var aktuelle. Denne formen for rekruttering, hvor man tar direkte kontakt med aktuelle informanter, kalles for personlig rekruttering (Johannessen et al., 2016). Gjennom personlig rekruttering fikk jeg fire lærere som var villige til å stille til intervju. For å øke antallet informanter ytterligere ble snøballmetoden benyttet (Johannessen et al., 2016). I denne rekrutteringsformen spør man personer man kjenner om de vet om andre personer innenfor målgruppen som kan tenkes å stille til intervju. Gjennom de fire lærerne som hadde takket ja fikk jeg kontaktinformasjonen til ytterligere fem lærere som kunne være aktuelle. Det viste seg imidlertid at en av disse lærerne ønsket ikke å delta, mens én annen av lærerne ikke oppfylte kriteriene for deltakelse. Av de syv lærerne som til slutt takket ja til intervju jobbet seks innenfor samme kommune. Jeg valgte derfor å ikke intervju den ene læreren som jobbet i en annen kommune. Begrunnelsen for dette var at de seks lærerne i samme kommune utgjorde en mer homogen gruppe. Med homogen gruppe i dette tilfellet menes at de hadde likere forutsetning for opplæring, etterutdanning og kursing etter innføringen av LK20. Et relativt homogent utvalg gjør det lettere å avdekke potensielle felles og ulike erfaringer (Johannessen et al., 2016). Ved å holde meg til lærere innenfor samme kommune var tanken at ulike erfaringer i mindre grad ville skyldes ulike retningslinjer for opplæring, videreutdanning og kursing innad i kommunen. Utvalget av lærere i denne masteroppgaven endte derfor på seks naturfaglærere på mellomtrinnet.

En oversikt over hvor lenge de seks naturfaglærerne har jobbet som lærere, hvorvidt de også underviser i matematikk og om de har tatt videreutdanning i programmering eller tilsvarende er vist i tabell 3. Hvorvidt lærerne underviser i matematikk er interessant ettersom det er dette undervisningsfaget som har flest kompetansemål tilknyttet programmering (Vinnervik & Bungum, 2022). Det er også interessant å vite om noen av lærerne har gjennomført videreutdanning innenfor programmering eller tilsvarende tema. Da videreutdanning antageligvis vil spille inn på lærernes forståelse av algoritmisk tenkning og programmering.

For å opprettholde lærernes anonymitet omtales lærer 1 som L1, lærer 2 som L2 osv. Dette gjelder også videre i oppgaven.

Tabell 3. Tabell over lærere med antall år som lærer, hvorvidt de underviser i matematikk, samt videreutdanning.

Lærer	Antall år som lærer	Underviser også i matematikk	Videreutdanning
L1	35 år	Nei	Nei
L2	24 år	Ja	15 stp i programmering
L3	24 år	Nei	Nei
L4	12 år	Ja	Nei
L5	7 år	Ja	Nei
L6	5 år	Ja	Nei

Det ble avtalt tidspunkt for intervju med hver enkelt lærer, og alle lærerne som deltok fikk tilsendt et informasjonsskriv med informasjon om oppgaven, samt samtykkeerklæring minimum én uke før selve intervjuet. Informasjonsskrivet som ble sendt ut kan ses i sin helhet under vedlegg 2.

#### 5.4 Gjennomføring av intervjuer

Det ble avtalt tidspunkt for intervju ut fra når den enkelte lærer hadde mulighet. Intervjuene ble gjennomført innenfor lærerens arbeidstid på skolen hvor læreren jobbet. Totalt ble det gjennomført seks intervju på tre ulike skoler. Alle intervjuene ble gjennomført i løpet av en måned. Innsamlingen av datamaterialet foregikk derfor over en relativt kort periode. En fysisk kopi av informasjonsskrivet ble medbrakt til hvert intervju, og intervjuene startet med en gjennomgang av informasjonsskrivet. Det ble her presisert at deltakelsen var frivillig, at lærerne når som helst kunne trekke seg fra studien og at all informasjon ville bli anonymisert. At intervjueren starter med å definere situasjonen for intervjupersonen, gi informasjon og la intervjupersonen komme med eventuelle spørsmål er en vanlig måte å starte et intervju (Kvale & Brinkmann, 2015). Informasjonsskrivet lærerne fikk utdelt inneholdt to ulike samtykkeerklæringer (Se vedlegg nr. 3). Det første samtykket handlet om at innsamlet data kunne benyttes i masteroppgaven, mens det andre samtykke handlet om at dataen i etterkant



av kunne benyttes som en del av MASCOT-prosjektet. Gjennom denne ekstra gjennomgangen av studiens omfang og formål fikk lærerne muligheten til å gi et informert samtykke. Alle lærerne signerte frivillig på begge samtykkeerklæringene. For å dokumentere datainnsamlingen ble alle intervju tatt opp med mobiltelefon gjennom «Diktafon-appen», som er en applikasjon for lydopptak utviklet av Universitetet i Oslo. De ferdige lydfilene ble i etterkant av intervjuene sendt til «Nettskjema», som er en nettside med høyt sikkerhetsnivå for lagring av data. Lærerne ble informert om når lydopptaket startet og når det ble avsluttet, samt hvordan den ferdige lydfilen ble oppbevart. Samtlige intervju ble gjennomført uten forstyrrelser og lydfilene hadde derfor god kvalitet. Avsatt tid til hvert intervju var 20 minutter, og samtlige intervju varte i underkant av dette. Intervjuguiden utgjorde utgangspunktet for alle intervjuene.

Innledningsvis i intervjuet ble lærerne spurt om å fortelle litt om lærerbakgrunnen sin og hvilke fag og trinn de underviser. Dette var en bevisst inngang i håp om å få lærerne til og slappe av og bli komfortable i situasjonen. De første minuttene av et intervju er nemlig avgjørende ettersom intervjupersonen her lager seg en oppfatning av intervjuer, noe som igjen kan påvirke hvordan de svarer på videre spørsmål (Kvale & Brinkmann, 2015). Kvale og Brinkmann (2015) trekker frem at for å skape god kontakt til intervjupersonen bør intervjueren lytte oppmerksomt, vise interesse, forståelse og respekt for det intervjupersonen sier. Dette prøvde jeg å være bevisst på under alle intervjuene gjennom kroppsholdning, unngå avbrytelser og gjennom å gjenta viktige poeng intervjuobjektene kom med. Under alle intervjuene opplevde jeg at det var en god tone mellom meg og lærerne som ble intervjuet og at jeg la til rette for en trygg opplevelse for lærerne. Samtidig er det viktig å være bevisst at det i et intervju alltid vil være en asymmetrisk maktrelasjon mellom intervjuer og intervjuobjekt (Kvale & Brinkmann, 2015). Det er intervjueren som definerer intervjusituasjonen, hvilke temaer som er viktig og må følges opp ytterligere, hvilke innspill som ikke skal følges opp og når intervjuet avsluttes. I tillegg går utspørringen bare én vei i form av at intervjueren spør og intervjuobjektet svarer. Det er også intervjueren som fortolker det intervjupersonen sier og rapportere hva intervjuobjektet virkelig mente (Kvale & Brinkmann, 2015). Disse faktorene kan føre til at intervjuobjektet holder tilbake informasjon, eller snakke utenom temaet. Det er ikke mulig for meg å avgjøre om lærerne som ble intervjuet holdt tilbake informasjon, men jeg opplevde at alle lærerne holdt seg til temaet for de ulike spørsmålene og prøvde å svare etter beste evne.

Under intervjuet ble det ikke spurt hvorvidt lærerne hadde videreutdanning innenfor algoritmisk tenkning eller programmering. Under skriveprosessen av masteroppgaven oppstod det et behov for å vite hvorvidt lærerne hadde gjennomført videreutdanning innenfor disse temaene. Det ble derfor sendt ut SMS til lærerne som hadde deltatt i intervju i etterkant hvor de ble spurt om de hadde gjennomført videreutdanning. Svarene på dette spørsmålet ble vist i forrige kapittel i tabell 3.

## 5.5 Transkripsjon

I etterkant av de seks intervjuene ble lydopptakene fra intervjuene transkribert. Transkribering av kvalitative forskningsintervju er svært vanlig, og det er transkripsjonene som utgjør datagrunnlaget for den videre analysen (Kvale & Brinkmann, 2015). Transkripsjon er også et viktig steg for å bli kjent med datamaterialet sitt (Braun & Clarke, 2006). Samtidig er det viktig å være bevisst på de utfordringene man kan møte når man skal oversette noe fra muntlig til skriftlig form, og derfor også de vurderingene og beslutningene man må ta underveis i transkriberingsprosessen (Kvale & Brinkmann, 2015). I en intervjusituasjon kan man observere kroppsspråk, mimikk og ulike gester, men disse sidene ved intervjupersonens svar blir borte i lydopptaket og transkripsjonen (Kvale & Brinkmann, 2015). Det er derfor viktig å være tydelig på hvordan transkripsjonen er blitt utført. Ettersom intervjuene mine hadde blitt tatt opp hadde jeg muligheten til å pause avspillingen og høre ulike svar om igjen og notere tilleggsinformasjon jeg mente var relevant. I transkripsjonsprosessen prøvde jeg, så langt det lot seg gjøre, å skrive ned ordrett det lærerne sa. Begrunnelsen for dette valget var å unngå å tillegge utsagn en annen mening enn det som var tiltenkt ved å omskrive utsagnene. Jeg valgte heller ikke å transkribere pauser, følelsesuttrykk eller andre ikke-verbale utsagn. Dette valget ble tatt på bakgrunn av målet med intervjuet var å undersøke hvordan lærerne uttrykte sin forståelse språklig, og ikke nødvendigvis følelser eller holdninger til temaene som ble tatt opp ut over det som kom til uttrykk gjennom språket. Det finnes ingen fasit for hvordan en transkripsjon skal gjennomføres og hvilke valg man tar vil være avhengig av hva transkripsjonen skal brukes til (Kvale & Brinkmann, 2015).

## 5.6 Tolkning og analytisk tilnærming

Etter at data har blitt samlet inn, og i dette tilfellet transkribert, er neste steg å analysere dataene. Denne analysen innebærer også at dataene må tolkes ettersom kvalitative data ikke taler for seg selv (Johannessen et al., 2016). Analysearbeidet blir derfor et forsøk på å forstå

gjennom tekstlig fortolkning, også kalt en hermeneutisk tilnærming (Kvarv, 2021). I hermeneutikken snakker man gjerne om den som tolker sin forståelseshorisont, nemlig de ubevisste antakelsene som ligger til grunn hos dem som utfører tolkningsarbeidet. Tolkerens forståelseshorisont gjør at arbeidet starter med en foreløpig tolkning av fenomenet som skal undersøkes. Videre undersøkelse kan enten forsterke eller endre på den foreliggende tolkningen (Kvarv, 2021). Det er viktig å være bevisst at man aldri er helt objektiv i et tolkningsarbeid. Dette vil derfor også være tilfellet for tolkningsarbeidet gjort i denne masteroppgaven. Analysene som presenteres er et resultat av mine tolkninger, og dermed også min forståelseshorisont. Tolkningsprosessen handler mellom å vekselvis se på helheten og delene, en prosess som kalles den hermeneutiske sirkelen (Kvarv, 2021). Gjennom analysen av datamaterialet i denne masteroppgaven har det flere ganger blitt vekslet mellom å se på hvert intervju for seg, samt helheten ved å se på intervjuene samlet.

Analysemetoden i denne masteroppgaven er inspirert av Braun og Clarke (2006) sin beskrivelse av tematisk analyse. En av fordelene med denne analyseformen er at den er fleksibel og uavhengig av teoretisk og epistemologisk utgangspunkt. Tematisk analyse benyttes for å identifisere mønstre, såkalte temaer, i et datamateriale (Braun & Clarke, 2006). Det første steget i analyseprosessen innebærer å gjøre seg kjent med datamaterialet. Dette kan eksempelvis gjøres gjennom transkribering. I tillegg innebærer dette steget flere gjennomlesninger dataene samtidig som man notere ned tanker og ideer som dukker opp underveis i gjennomlesningen (Braun & Clarke, 2006). Underveis i transkriberingen ble jeg oppmerksom på forklaringer og beskrivelser som gikk igjen hos lærerne. Da jeg i etterkant leste gjennom transkripsjonene på nytt noterte jeg ned slike utsagn og tanker rundt disse. I det andre steget av tematisk analysen skal man identifisere innledende koder (Braun & Clarke, 2006). For å identifisere koder gjennomgikk jeg de transkriberte intervjuene, samt de tidligere notatene mine gjort underveis i gjennomlesningen og fant utsagn og sitater som var relevante for min problemstilling. Videre forsøkte jeg å oppsummere disse sitatene i enkeltord som var beskrivende for sitatets innhold. Enkeltordene utgjorde da kodene. I det tredje steget av tematisk analyse skal man enten lete etter tema i kodene man har identifisert, eller prøve å plassere kodene inn i forhåndsbestemte tema (Braun & Clarke, 2006). Når man leter etter tema i datamateriale gjennomfører man en såkalt induktiv analyse. Ved forhåndsbestemte tema er analyseprosessen deduktiv.

I utgangspunktet var planen i denne oppgaven å gjøre en deduktiv analyse basert på rammeverket til Shute et al. (2017). Dette innebar at de identifiserte kodene skulle plasseres

inn i forhåndsbaserte tema som da var de ulike kategoriene i Shute et al. (2017) sitt rammeverk. Dette rammeverket ble valgt fordi det gir en generell og overordnet beskrivelse av hvilke ferdigheter som inngår i algoritmisk tenkning basert på en omfattende litteraturstudie. Underveis i analyseprosessen dukket det opp koder som ikke passet inn i dette rammeverket, men som likevel ble vurdert viktige for resultatet. Noen av disse kodene passet inn i Weintrop et al. (2016) sitt rammeverk. Dette rammeverket tar utgangspunkt i hvilken rolle algoritmisk tenkning har, og hva det innebærer innenfor ulike praksiser i matematikk og naturfag. I tillegg dukket det gjennom analysen opp koder som ikke kunne plasseres innenfor noen av disse rammeverkene, men som likevel ble vurdert som viktige for resultatet. Å sortere disse kodene i temaer som ikke var forhåndsdefinerte ble derfor en del av en induktiv analyse (Braun & Clarke, 2006). De induktive kodene som dukket opp i analysen ble til slutt slått sammen til temaet «Algoritmisk tenkning og faktorer som påvirker undervisningskvaliteten». Analysen ble derfor abduktiv, noe som innebærer at det består av både deduktiv og induktiv analyse.

Ved bruk av kvalitative metoder kan man gjøre endringer underveis i forskningsprosessen når ny kunnskap dukker opp (Kvarv, 2021). Det er også en del av den tematiske analysen at man stadig revurderer temaene sine og hvorvidt disse er dekkende for de kodene man har identifisert (Braun & Clarke, 2006). For å få med alle temaene som dukket opp underveis i analysen ble det derfor besluttet å utarbeide et analytisk rammeverk som både inneholdt deler av Shute et al. (2017) sitt rammeverk, deler av Weintrop et al. (2016) sitt rammeverk, samt det induktive temaet. Denne beslutningen utgjorde da det fjerde steget i den tematiske analysen som innebærer at man vurderer hvor vidt temaene man har passer til kodene man har identifisert, det totale datamaterialet, samt problemstillingen man ønsker å besvare. Det femte steget i analyseprosessen ble derfor å utarbeide et analytiske rammeverk med tydelige definisjoner og beskrivelser av hvert tema. Det avsluttende steget i tematisk analyse innebærer en siste analyse for å forsikre seg om at resultatet er relevant for å besvare problemstillingen man har som utgangspunkt.

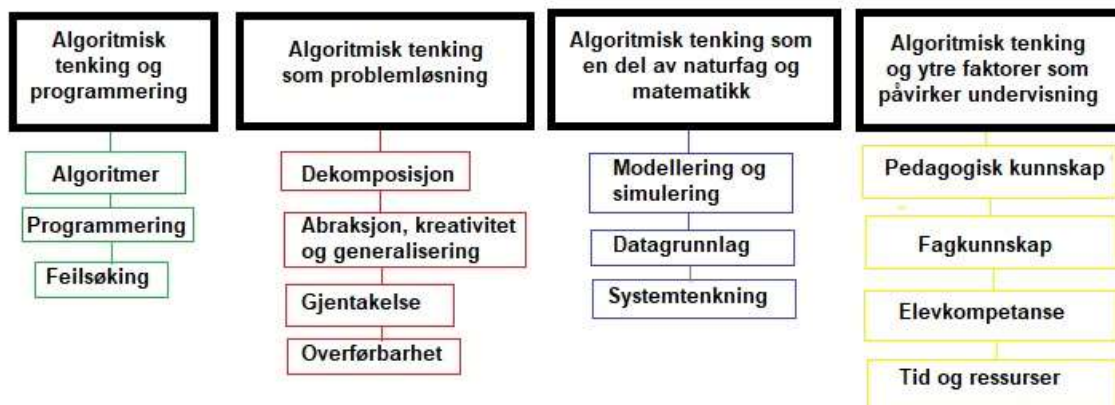
I arbeidet med tematisk analyse er det noen fallgruver man må unngå (Braun & Clarke, 2006). Først og fremst må man passe på at man faktisk gjennomfører en analyse. Det er ikke nok å kun trekke ut deler av datamaterialet uten at det gjøres noen form for analyse. De delene av datamaterialet som velges ut har en tematisk betydning, og tolkningen av disse dataene legger grunnlaget for resultatene. Dette var noe jeg i starten av analyseprosessen opplevde som vanskelig å få til, og til tider frustrerende. Gjennom god veiledning fikk jeg hjelp som gjorde

at jeg ut over i prosessen klarte å gjennomføre en tematisk analyse, og med det unngå å bare hente ut deler av datamaterialet uten å gjøre noen form for analyse. En annen vanlig feil er at man bruker spørsmålene fra datainnsamlingen som temaer (Braun & Clarke, 2006). I disse tilfellene blir også problemet at det ikke gjennomføres noen faktisk analyse for å identifisere temaer i datamaterialet. I mitt tilfelle var ikke dette et problem ettersom jeg i utgangspunktet skulle gjøre en deduktiv analyse, og det ble da tydelig gjennom analyseprosessen at det var temaer i datamaterialet som ikke inngikk i de temaene som opprinnelig var tenkt. En tredje fallgrube er at temaene man gjør en svak analyse hvor temaene enten ikke passer i forhold til det man ønsker å undersøke, at temaene overlapper, eller at temaene ikke er tydelig nok definert (Braun & Clarke, 2006). Dette oppleves også å ha blitt unngått i denne studien da alle temaene er tydelig definerte i det analytiske rammeverket, et rammeverk utarbeidet for å kunne besvare problemstillingen og samtidig dekke funnene i datamaterialet. Andre typiske fallgruver i tematisk analyse handler om at analysen som presenteres ikke samsvarer med datamaterialet, eller at den formen for tematisk analyse som har blitt benyttet ikke passer til problemstillingen som ligger til grunn (Braun & Clarke, 2006). I denne studien har datamaterialet dannet grunnlaget for resultatene, og det presenteres ikke påstander eller funn som ikke kan begrunnes i datamaterialet. Den analytiske tilnærmingen som har blitt benyttet er også nøye gjennomtenkt med utgangspunkt i oppgavens problemstilling.

## 5.7 Analytisk rammeverk

I det analytiske rammeverket ble ulike kategorier fra de teoretiske rammeverkene benyttet, samt en kategori som ble identifisert gjennom en induktiv analyse av datamaterialet. De kategoriene som ble valgt ut fra de teoretiske rammeverkene ble valgt ut både på bakgrunn av problemstillingen og forskningsspørsmålene, og som et resultat av ulike koder i datamaterialet. Lignende kategoriene fra de to teoretiske rammeverkene ble slått sammen, eller det ble gjort små endringer eller tilpasninger på definisjonene. Det resulterte til slutt i et analytisk rammeverk bestående av fire hovedtemaer; algoritmisk tenkning og programmering, algoritmisk tenkning som problemløsning, algoritmisk tenkning som en del av naturfag og matematikk, samt algoritmisk tenkning og ytre faktorer som påvirker undervisning. Hver av disse hovedtemaene bestod av tre eller fire undertema. De tre første hovedtemaene var inspirert av de teoretiske rammeverkene, mens det siste temaet ble identifisert gjennom induktiv analyse.

Det analytiske rammeverket med fire hovedtemaer, samt tilhørende undertemaer er vist under.



Figur 1. Oversikt over analytisk rammeverk med hovedtema og tilhørende undertema.

Det analytiske rammeverket med definisjoner på hovedtemaer og undertemaer er vist under. I beskrivelsen av hvert undertema er det også spesifisert hvilket av de teoretiske rammeverkene undertemaet er hentet fra, samt hvordan undertemaet defineres i det analytiske rammeverket.

Tabell 4. Oversikt over det analytiske rammeverken med beskrivelser av hovedtema og tilhørende undertema.

Hovedkategori	Underkategori	Beskrivelse av kategori
<b>1. Algoritmisk tenkning og programmering</b>  Denne hovedkategorien beskriver de ferdighetene innenfor algoritmisk tenkning som kan knyttes til algoritmer, programmering og koding.	<b>Algoritmer</b>	Denne kategorien er inspirert av Shute et al. (2017) sin tredje kategori «algoritmer».  I denne kategorien ligger det både at man kan følge en oppskrift eller mal utarbeidet av noen andre, og at man selv klarer å utarbeide logiske og ordnede instruksjoner for å løse et problem. Disse instruksjonene kan gjennomføres av enten mennesker eller datamaskiner. Det innebærer å øve seg på å lage logiske og ordnede instruksjoner både som en del av en forberedelse til koding, men også uavhengig av koding, som i arbeidet med analog programmering.
	<b>Programmering</b>	Denne kategorien er inspirert av Weintrop et. al (2016) sin kategori «Programmering» som er en underkategori av «Problemløsning via dataprosesser».  Denne kategorien handler om å kunne skrive instruksjoner i et programmeringsspråk på en slik måte at en datamaskin klarer å kjøre dem. Det innebærer å kunne forstå og modifisere programmer skrevet av andre, i tillegg til å kunne skrive egne programmer fra bunn. Det å lære seg syntaksen innenfor et programmeringsspråk er derfor en viktig del av denne kategorien.
	<b>Feilsøking</b>	Denne kategorien er inspirert av Shute et. al (2017) sitt fjerde kategori, «Feilsøking», samt underkategorien «Feilsøking» under «Problemløsning via

		<p>dataprosessering» i Weintrop et. al (2016) sitt rammeverk.</p> <p>Denne kategorien handler om å oppdage og identifisere hvorfor noe ikke virker som ønsket, eller ikke oppfører seg som forventet. Det innebærer å jobbe systematisk for å identifisere problemet, samt for å løse problemet. Det handler om å kunne identifisere, isolere, reprodusere og korrigere utforutsette problemer og gjøre dette på en systematisk og effektiv måte.</p>
<p><b>2. Algoritmisk tenkning som problemløsning</b></p> <p>Denne hovedkategorien tar for seg ulike underkategorien som beskriver de ferdigheten som inngår i algoritmisk tenkning som en problemløsningsprosess.</p>	<p><b>Dekomposisjon</b></p> <p>Denne kategorien er inspirert av første kategori i Shute et. al (2017) sitt rammeverk, «Dekomposisjon».</p> <p>Kategorien handler om evnen til å kunne dele komplekse problemer inn i mindre, mer håndterbare problemer. De mindre problemene er ikke tilfeldige deler, men funksjonelle deler som sammen utgjør hele problemet. For å løse hvert delproblem brukes systematiske prosesser.</p>	
	<p><b>Abstraksjon, kreativitet og generalisering</b></p> <p>Denne kategorien er inspirert av Shute et.al (2017) sin andre kategori, «Abstraksjon», samt Weinstrop et. al (2016) sin underkategori «Abstraksjon» under «Problemløsning via dataprosesser».</p> <p>Kategorien handler om å kunne trekke ut essensen av et komplekst system, eller problemstilling. For å klare det må man kunne trekke frem de viktigste aspektene ved løsningen eller systemet, samtidig som man klarer å se bort fra det som er mindre viktig. Det handler også om å kunne gjøre en løsning eller idé mer generell, slik at den kan benyttes for å løse lignende problemstillinger i fremtiden. I denne kategorien ligger også abstraksjon og kreativitet i form av idémyldring, det å kunne se for seg hvilke ulike muligheter man har, velge ut den muligheten som trolig vil fungere best og videreutvikle denne idéen. Utvikle og lage et produkt basert på en idé.</p>	
	<p><b>Gjentakelse</b></p> <p>Inspirert av Shute et. al (2017) sin femte kategori, «Gjentakelse». Handler om å gjenta en prosess flere ganger for å optimalisere en løsning. Prosessen gjentas til man har oppnådd et resultat som på best mulig måte svarer til målet, eller til det ikke lenger er hensiktsmessig å utbedre løsningen.</p>	
	<p><b>Overførbarhet</b></p> <p>Inspirert av Shute et. al (2017) sin sjettede kategori «Generalisering».</p> <p>Denne kategorien handler om evnen til å overføre ferdighetene fra algoritmisk tenkning over til et bredere spekter av situasjoner og problemstillinger og benytte disse ferdighetene for å løse problemer effektivt og nøyaktig. I denne kategorien ligger også det å kunne benytte algoritmisk tenkning i flere fag og å jobbe tverrfaglig. Elevene må gruble og prøve seg frem i et forsøk på å anvende det de har lært innenfor algoritmisk tenkning i en ny kontekst.</p>	
<p><b>3. Algoritmisk tenkning som en del av naturfag og matematikk</b></p> <p>De siste årene har det skjedd en enorm teknologisk utvikling, en</p>	<p><b>Modellering og simulering</b></p> <p>Inspirert av Weintrop et. al (2016) sin andre hovedkategori, «Modellering og simulering».</p> <p>Denne kategorien handler om hvordan modellering og simulering er essensielt i både matematikk og naturvitenskap i arbeidet med ulike fenomener. Med dagens teknologiske utvikling har teknologiske løsninger og dataprogrammer blitt viktige både for å lage, tilpasse og bruke modeller for å beskrive og forklare fenomener.</p>	

<p>utvikling som også har endret hvordan man jobber innenfor matematikk og naturvitenskap. Weinstrop et.al (2016) trekker frem hvordan denne utviklingen har gjort det essensielt å kunne håndtere teknologiske løsningene som en del av naturvitenskapelig- og matematisk forskning, noe som igjen har ført til at algoritmisk tenkning også er blitt en viktig del av naturvitenskapelig- og matematisk forskning. Denne kategorien handler derfor om hvordan hvilke ferdigheter innenfor algoritmisk tenkning som spesifikt kan knyttes opp mot matematikk og naturvitenskap.</p>	<p><b>Datagrunnlag</b></p>	<p>Inspirert av Weintrop et.al (2016) sin første Hovedkategori «Datapraksiser».</p> <p>For å løse et problem eller utforske en problemstilling i matematikk eller naturvitenskap kreves det et godt datagrunnlag. Datagrunnlag og evidens utgjør grunnlaget for forskning innenfor matematikk og naturvitenskap. Denne kategorien handler om å kunne benytte seg av teknologiske løsninger for å samle inn data, generere data, manipulere data, analysere data og visualisere data.</p>
	<p><b>System tenkning</b></p>	<p>Denne kategorien er inspirert av Weintrop et. al (2016) sin fjerde hovedkategori.</p> <p>Denne kategorien handler om å forstå store og komplekse systemer. Dette innebærer å forstå et system som en helhet, forstå hvordan komponentene innenfor et system virker sammen, kunne forstå og forklare hvordan systemet virker på ulike nivå, videreformidle kunnskapen om et system, samt definere rammene for et system.</p>
<p><b>4. Algoritmisk tenkning og faktorer som påvirker undervisningskvaliteten</b></p> <p>Denne hovedkategorien består av underkategorier som ble tydelige under den tidligere analysen av datamaterialet. Disse kategoriene er ikke en del av definisjonen på algoritmisk tenkning, men utgjør ytre faktorer som påvirker lærernes undervisning av temaet, og dermed også elevenes læringsutbytte.</p>	<p><b>Pedagogisk kunnskap</b></p>	<p>Pedagogisk kunnskap omhandler skjæringspunktet mellom lærerens fagkunnskap og lærerens fagdidaktiske kunnskap. I denne kategorien ligger den kunnskapen lærerne har om hvordan de på best mulig måte kan legge til rette for best mulig læringsutbytte hos elevene innenfor et bestemt fag, tema eller fagområde. I denne inngår hvordan man strukturerer undervisningen, hvordan man legger opp den faglige progresjonen, hvordan man legger frem nytt fagstoff for elevene og lignende. Læreren må både ha god fagkunnskap og god didaktisk kompetanse for å legge opp undervisningen slik at elevene får best mulig læringsutbytte.</p>
	<p><b>Fagkunnskap</b></p>	<p>Denne kategorien omhandler lærernes fagkunnskap innenfor algoritmisk tenkning og programmering og i hvor stor grad lærerne opplever at de selv har nok kompetanse til å undervise i dette emnet.</p>
	<p><b>Elevkompetanse</b></p>	<p>Denne kategorien omhandler i hvor stor grad lærerne opplever at de selv har nok kompetanse til å undervise i algoritmisk tenkning og programmering.</p> <p>Denne kategorien handler generelle ferdigheter lærerne mener er viktige i arbeidet med algoritmisk tenkning og programmering, og ferdigheter lærerne mener elevene får jobbet med. Eksempelvis tålmodighet, konsentrasjon, motivasjon, nysgjerrighet, fantasi og samarbeid.</p>
	<p><b>Tid og ressurser</b></p>	<p>Denne kategorien omhandler på hvilken måte ytre faktorer i form av tid og tilgjengelige ressurser påvirker undervisningen i algoritmisk tenkning og programmering, samt hvordan disse faktorene oppleves.</p>



## 5.8 Eksempler på analyse og koding

Ettersom det teoretiske rammeverket som i utgangspunktet var tiltenkt for den deduktive analysen ikke var dekkende, ble det femte steget i analyseprosessen å utvikle et dekkende analytisk rammeverk. Det analytiske rammeverket ble tydelig beskrevet i forrige kapittel. I etterkant av dette steget ble alle transkripsjonene gjennomgått på nytt og kodene ble sortert inn under de endelige under- og hovedtemaene i det analytiske rammeverket. Tabellen under viser eksempel på denne delen av analyseprosessen hvor noen av sitatene fra intervju 5 ble omgjort til koder, for så å sorteres innenfor riktig under- og hovedtema.

Tabell 5. Eksempel på koding av sitater fra intervju 5.

<b>Sitat</b>	<b>Kode</b>	<b>Underkategori i analytisk rammeverk</b>	<b>Hovedkategori i analytisk rammeverk</b>
«Altså at du følger en viss mal, en viss oppskrift, en viss strategi for å få ønsket resultat da.»	Mal Oppskrift	Algoritmer	Algoritmisk tenkning og programmering
«Så vi sjekker, vi har sånne økter hvor vi liksom sjekker at alt går som det skal, får ønsker resultat.»	Feilsøke	Feilsøking	Algoritmisk tenkning og programmering
«Nei, i det kompetansemålet der så trekker jeg paralleller veldig til programmering jeg [...]»	Programmering	Programmering	Algoritmisk tenkning og programmering

Etter å ha analysert hvert enkelt intervju som vist ovenfor oppstod det et behov for å se intervjuene i sammenheng og identifisere hvilke undertemaer og hovedtemaer som ble mest brukt. Dette er et eksempel på den hermeneutiske sirkelen hvor man veksler mellom å se på delene og helheten (Kvarv, 2021). For å få en oversikt over helheten i datamaterialet ble derfor alle de utvalgte sitatene fra hvert intervju plassert inn i det analytiske rammeverket.

Tabell 6. Utdrag fra oversikten over alle relevante sitater fra intervjuene plassert inn i det analytiske rammeverket.

Hovedkategori	Underkategori	Beskrivelse	Sitater
Algoritmisk tenkning og programmering	Algoritmer	<p>Denne kategorien er inspirert av Shute et al. (2017) sin tredje kategori «algoritmer».</p> <p>I denne kategorien ligger det både at man kan følge en oppskrift eller mal utarbeidet av noen andre, og at man selv klarer å utarbeide logiske og ordnede instruksjoner for å løse et problem. Disse instruksjonene kan gjennomføres av enten mennesker eller datamaskiner. Det innebærer å øve seg på å lage logiske og ordnede instruksjoner både som en del av en forberedelse til koding, men også uavhengig av koding, som i arbeidet med analog programmering.</p>	<p>«Jeg tenker jo at i første omgang så er det det her å ta en instruksjon. Følge en oppskrift. Også i neste omgang bruke det til å utvikle seg videre kreativt.» <b>(Intervju 1)</b></p> <p>«Ja, det er en formel eller en metode for å komme frem til et svar.» <b>(Intervju 3)</b></p> <p>«Det å kunne, det er det her å lage, forholde seg til en slags oppskrift. Det å forstå instruksjoner, kunne lage instruksjoner.» <b>(Intervju 4)</b></p> <p>«Nei, det er nærliggende å tenke programmering da i en eller annen form. Men både analog og, ja, hva kaller man vanlig programmering? [...] Koding kaller man vanlig programmering. For å gi den forståelsen for hva det dreier seg om.» <b>(Intervju 4)</b></p> <p>«Altså at du følger en viss mal, en viss oppskrift, en viss strategi for å få ønsket resultat da.» <b>(Intervju 5)</b></p> <p>«Og da har vi hatt ren opplæring i det i forhold til hva en kode er for noe, hva er, ikke brukt ordet algoritme enda, men fulgt liksom en oppskrift da og mal og for å liksom sette dem i riktig rekkefølge slik vi ønsker at hendelsen skal skje da.» <b>(Intervju 5)</b></p>

			«Og da er det for eksempel hefter, de har jobbet med hefter og fulgt steg for steg.» (Intervju 5)
--	--	--	---

Den fullstendige versjonen av tabell 5 ga god oversikt over hvilke undertemaer og hovedtemaer i det analytiske rammeverket som i størst grad ble trukket frem av lærerne, og hvilke som i mindre grad ble brukt. Det oppstod også et behov for å få en oversikt over hvilke hovedtema besvarelsene til lærerne passet inn under på hvert av de temaspesifikke spørsmålene i intervjuet. Dette var for å få en oversikt over om lærerne brukte andre hovedtemaer når de beskrev kompetansemålet enn hva de eksempelvis gjorde når de snakket om ferdigheter tilknyttet kompetansemålet, eller algoritmisk tenkning. Denne oversikten er vist i tabellen under. Her er hovedtema forkortet HT, og vist sammen med undertemaet sitatene ble plassert inn under.

Tabell 7. Oversikt over temaspesifikke spørsmål og hvilke undertema, samt hovedtema (HT), lærernes besvarelse passet inn under.

Spørsmål	L1	L2	L3	L4	L5	L6
4. <i>Kompetansemål</i>	Programmering (HT1)	Programmering (HT1) Gjentakelse (HT2) Overførbarhet (HT2) Elevkompetanse (HT4)	Programmering (HT1)	Programmering (HT1) Abstraksjon, kreativitet og generalisering (HT2)	Programmering (HT1)	Programmering (HT1)
5. <i>Ferdigheter</i>	Algoritmer (HT1)	Overførbarhet (HT2) Elevkompetanse (HT4) Tid og ressurser (HT4)	Systemtenkning (HT3)	-	Elevkompetanse (HT4)	-
5a, 5b <i>Algoritmisk tenkning</i>	-	Overførbarhet (HT2) Gjentakelse (HT2)	Algoritmer (HT1)	Algoritmer (HT1)	Algoritmer (HT1) Programmering (HT1)	-
6. <i>Aktiviteter</i>	Overførbarhet (HT2)	Systemtenkning (HT3)	Programmering (HT1)	Algoritmer (HT1)	Programmering (HT1)	-

	System tenking (HT3)		Progresjon (HT4)	Programmering (HT1)	Algoritmer (HT1)	
7.	Strukturering av undervisning (HT4)	Systemtenkning (HT3)	-	Overførbarhet (HT2)	Programmering (HT1)	Programmering (HT1)
<i>Undervisning</i>	Elevkompetanse (HT4)			Abstraksjon, kreativitet og generalisering (HT2)	Feilsøking (HT1)	
	Progresjon (HT4)				Algoritmer (HT1)	
					Progresjon (HT4)	
8.	Lærerkompetanse (HT4)	Tid og ressurser (HT4)	Lærerkompetanse (HT4)	Lærerkompetanse (HT4)	Lærerkompetanse (HT4)	-
<i>Annet</i>		Elevkompetanse (HT4)				

Alle tabellene det er vist utdrag fra i dette kapittelet la grunnlaget for resultatene som presenteres i neste kapittelet. Igjen er det viktig å påpeke at analysearbeidet som har blitt gjort er en tolkningsprosess. Gjennom tematisk analyse tolkes ulike aspekter ved datamaterialet sett opp mot den aktuelle problemstillingen (Braun & Clarke, 2006). Denne tolkningsprosessen starter allerede gjennom transkriberingen, eller det første steget i tematisk analyse, hvor man gjør seg kjent med datamaterialet (Braun & Clarke, 2006). I de videre stegene i analysen har sitater som blir ansett som viktige i henhold til problemstillingen blitt kodet, i dette tilfellet til enkeltord som oppsummerer eller beskriver hovedinnholdet i sitatet. Videre har disse kodene blitt forsøkt plassert inn i forhåndsbestemte tema som har vist seg å ikke være dekkende. Av den grunn har temaene blitt revurdert, endret og tydelig definert i henhold til steg 4 og 5 i den tematiske analysen (Braun & Clarke, 2006). Avslutningsvis har sitatene og kodene igjen blitt analysert med utgangspunkt i det analytiske rammeverket, samt sortert på ulike måter for å få et helhetsbilde som vist i tabell 5 og 6.

## 5.9 Kvalitet i forskningen

Forskningens kvalitet kan ikke utelukkende knyttes til det resultatet som blir presentert, men handler i større grad om hvordan denne kunnskapen har blitt produsert (Postholm & Jacobsen, 2018). I dette kapittelet har jeg derfor prøvd å gi en tydelig gjennomgang av hvordan kunnskapen som presenteres i neste kapittel har blitt konstruert. For å vurdere kvaliteten på et forskningsprosjekt ser man gjerne på forskningens validitet og reliabilitet. Forskningens validitet handler om hvor gyldig den er, og handler om hvilke slutninger det er mulig å trekke ut i fra den dataen man har samlet inn (Postholm & Jacobsen, 2018). Sagt med andre ord om

forskningen har undersøkt det den var ment til å undersøke (Johannessen et al., 2016). I denne master oppgaven handler det om jeg gjennom de metodevalgene som har blitt tatt har klart å undersøke i hvilken grad man har innført algoritmisk tenkning i naturfag etter LK20. Et viktig del av validiteten handler om å bli godt kjent med feltet man skal undersøke slik at man klarer å skille mellom informasjon som er relevant og ikke (Johannessen et al., 2016). I forkant av masteroppgaven leverte jeg en prosjektskisse og i arbeidet med denne satt jeg meg godt inn i algoritmisk tenkning, både hvilke definisjoner som har blitt brukt, samt hvordan det er implementert i norsk skole. I utarbeidelsen av intervjuguiden fikk jeg også god hjelp av veileder til å forsikre meg om at intervju spørsmålene var gode for å undersøke de forskningsspørsmålene jeg ønsket å besvare. Det analytiske rammeverket, samt kodingen av sitater fra intervjuene, har blitt diskutert både med veileder og medstudenter. At flere personer kritisk har vurdert både intervjuguiden, kodingen og analysen styrker oppgavens validitet. En svakhet ved oppgavens validitet er derimot at det kun har blitt benyttet én metode, i dette tilfellet intervju (Johannessen et al., 2016). Metodetriangulering, altså bruk av flere metoder, er noe som styrker forskningens validitet. Metodetriangulering kan også innebære at man undersøker mer enn bare én setting (Johannessen et al., 2016). Som eksempel har det i denne masteroppgaven blitt intervjuet seks lærere fra tre ulike skoler. Et annet viktig aspekt ved en oppgaves validitet er hvorvidt den kunnskapen man har kommet frem til kan overføres til andre områder som ikke er studert. I kvalitativ forskning handler overførbarheten i stor grad om hvorvidt beskrivelsene er gjenkjennbare for andre (Postholm & Jacobsen, 2018). En transparent og utfyllende beskrivelse av forskningen som har blitt gjort vil også øke overførbarheten (Postholm & Jacobsen, 2018). Gjennom dette metodekapittelet har jeg forsøkt å gi en transparent og utfyllende beskrivelse av prosessen som ligger bak resultatene med et ønske om at dette vil kunne bidra til økt overførbarhet.

En forsknings reliabilitet handler om studiens pålitelighet og i hvilken grad selve undersøkelsen og forskeren kan ha påvirket resultatet (Postholm & Jacobsen, 2018). Når man benytter intervju som metode handler dette eksempelvis om personene som ble intervjuet ville gitt andre svar til en annen forsker (Kvale & Brinkmann, 2015). Samtidig er intervjukunnskap kontekstuell, noe som betyr at intervjusituasjonen i seg selv er en spesifikk kontekst for å produsere kunnskap (Kvale & Brinkmann, 2015). Som Johannessen et al. (2016) påpeker vil det være umulig for en forsker å duplisere en kvalitativ studie, nettopp fordi det ofte er samtaler som danner datagrunnlaget og disse er kontekstavhengige. Likevel tror jeg at hadde noen andre stilt de samme lærerne de samme spørsmålene som ble gjort i disse intervjuene

ville de fått lignende svar. Denne antagelsen er basert på at jeg under intervjuene var bevisst på å prøve og unngå å påvirke lærerne som ble intervjuet og lot den snakke fritt når de besvarte spørsmålene. Tolkningen av dataene påvirker også forskningens reliabilitet, men ettersom ingen andre har akkurat samme bakgrunn og erfaring som forskeren som gjennomfører tolkningen vil heller ingen andre kunne tolke på samme måte (Johannessen et al., 2016). For å øke tolkningens reliabilitet er det derfor viktig at man gir en åpen og detaljert beskrivelse av hvordan man har gått frem. Dette gjør det mulig for andre å vurdere de valgene som har blitt tatt, og igjen det endelige resultatet (Johannessen et al., 2016).

## 5.10 Forskningsetikk

I arbeidet med en masteroppgave har studenten som gjennomfører studien et etisk ansvar (Postholm & Jacobsen, 2018). Dette etiske ansvaret handler om forskerens ansvar for forskningsdeltakerne, for undersøkelsen og for forskeren selv. I utgangspunktet er det tre hovedkrav som må være ivaretatt mellom forsker og dem det forskes på. Disse er «informert samtykke, krav på privatliv og krav på å bli korrekt gjengitt.» (Postholm & Jacobsen, 2018, s. 247).

For at de som deltar i studien skal kunne gi et informert samtykke må de informeres om hva som er formålet med studien, hovedtrekkene i forskningsdesignet, hva dataene som brukes til og eventuelle fordeler og risikoer ved deltakelse (Kvale & Brinkmann, 2015). Frivillig deltakelse og muligheten for å trekke seg er også viktige prinsipper innenfor informert samtykke. I forkant av deltakelse fikk lærerne i denne studien tilsendt et informasjonsskriv hvor dette ble gjennomgått (Se vedlegg 2.). Den samme informasjonen ble også gjennomgått muntlig i forkant av intervjuene, og lærerne fikk her mulighet til å stille spørsmål hvis noe var uklart, eller det var noe de lurte på.

Krav på privatliv handler om hvor følsom informasjonen som skal samles inn er, hvor privat informasjonen er og hvor stor sannsynlighet det er for å identifisere deltakerne ut fra datamaterialet (Postholm & Jacobsen, 2018). I god tid før datainnsamlingen meldte jeg prosjektet mitt til NSD, Norsk samfunnsvitenskapelig datatjeneste. Stemme regnes som en personopplysning (Johannessen et al., 2016), og ettersom jeg ønsker å gjennomføre intervjuer var min masteroppgaven meldepliktig. Vedlagt i søknaden til NSD lå informasjonsskrivet som skulle informere lærerne om prosjektet, intervjuguide, samt en detaljert beskrivelse av prosjektet. På bakgrunn av denne informasjonen ble masteroppgaven godkjent av NSD (Se vedlegg 4.). Oversikten over navnene på lærerne hadde jeg på en liste lagret lokalt på

datamaskinen og som blir slettet ved endt prosjektslutt. Lærerne oppga ikke navn under intervjuene. Informasjon knyttet til hvor lenge lærerne hadde undervist og hvilke fag og trinn de underviser på ble anonymisert i analysen ved at lærerne ble referert til som L1, L2, L3 osv. Hvis noen av lærerne nevnte hvilken skole eller kommune de jobbet i underveis i intervjuene ble dette fjernet under transkripsjonen. På bakgrunn av disse tiltakene føler jeg at kravet om privatliv i denne studien har blitt ivaretatt på en god måte.

Det siste etiske kravet mellom forsker og forskningsdeltaker handler om kravet til å bli korrekt gjengitt (Postholm & Jacobsen, 2018). Dette innebærer at man forsøker å gjengi resultatene så fullstendig som mulig og at man ikke forfalsker data eller resultater. Gjengivelse av resultater er vanskelig ettersom all analyse innebærer en utvelgelse av data (Postholm & Jacobsen, 2018). Underveis i transkriberingen har jeg derfor vært bevisst å gjengi så ordrett som mulig det læreren har sagt. I analysen har jeg vært bevisst på å ikke ta enkeltsiter eller beskrivelser ut av kontekst, men å analysere disse med bakgrunn i konteksten det ble sagt. Jeg har heller på ingen måte prøvd å utelate data, eller endre på data, for at resultatene mine skal fremstå «bedre».

## 6. Resultat

I dette kapittelet presenteres resultatene fra analysen av den de seks intervjuene med naturfagslærere på mellomtrinnet.

### 6.1 Programmering i fokus

På det første temaspesifikke spørsmålet under intervjuet fikk lærerne spørsmål om hva de la i kompetansemålet «*utforske, lage og programmere teknologiske systemer som består av deler som virker sammen*». I dette kompetansemålet benyttes de tre verbene *utforske, lage og programmere*. Et mindretall av lærerne benyttet verbene *utforske* eller *lage* i sine besvarelser på dette spørsmålet. På den andre siden ga alle lærerne svar som kan knyttes til *Programmering*. Fire av lærerne ga svar som kun omhandlet *Programmering*. Et eksempel på en slik ren programmeringsforståelse av kompetansemålet var da L3 svarte:

L3: Programmering og data og koding og sånne ting tenker jeg at det har med å gjøre.

L5 hadde i likhet med L3 en ren programmeringsforståelse, men ga en mer utfyllende beskrivelse av sin forståelse:

L5: Nei, i det kompetansemålet der så trekker jeg paralleller veldig til programmering jeg og det faget som er relativt nytt og som vi har satt oss litt inn i nå hvor vi bruker Ipader og vi bruker programmeringsutstyr som vi har på huset. Micro:bit, Scratch har vi brukt en del av og de appene som inngår i programmeringsfaget og bruker det for å liksom gjøre oss kjent med det kompetansemålet som du viser til da.

Det som er verdt å merke seg i sitatet over er at L5 omtaler arbeidet med kompetansemålet og ulike programmeringsverktøy som en del av et «programmeringsfag». Det at fire av seks lærere kun forbinder kompetansemålet med *Programmering*, gir inntrykk av at det aktuelle kompetansemålet er fristilt fra de andre temaene i naturfag, og at arbeidet med dette kompetansemålet er uavhengig av den resterende naturfagundervisningen. Det at L5 i sin besvarelse omtaler kompetansemålet som en del av et «programmeringsfag» bygger opp under dette inntrykket.

To av lærerne, L2 og L4, ga svar som kunne tolkes som en bredere forståelse av kompetansemålet.



L2: Fordi at, de skal først finne et problem, så skal de gruble, så skal de tilbake igjen og få ny kunnskap gjennom den måten å jobbe på. Den måten kan man jobbe, den er kommet gjennom programmering i skolen, men den kan også adopteres i andre fag ved at man gir dem problemer hvor de ikke får fasit med en gang, men der de på en måte greier å finne løsninger av seg selv og går tilbake og frem, og frem og tilbake.

L2 trakk i likhet med de andre lærerne inn *Programmering* i sin besvarelse, men som vist i sitatet over inkluderte denne læreren også *Gjentakelse* og *Overførbarhet* i sin forståelse. Begge disse undertemaene er en del av *Algoritmisk tenkning som problemløsning*, og viser derfor er bredere forståelse av kompetansemålet. L4 snakket først om *Programmering*, men gikk så videre og ga et svar som tydet på en bredere forståelse av kompetansemålet. I denne delen av besvarelsen benytter L4 også verbet *utvikle* som også er nevnt spesifikt i kompetansemålet.

L4: At elevene får lov og utvikle og lage noe selv [...] Og det å på en måte få lov å begynne på papiret og ha en prosess med hva er det vi vil lage. [...] Og da det her å få lov og få mulighet til å utvikle den ideen, teste det ut, ja, men som et tverrfaglig prosjekt da. Det å både designe og, ja, selve ja, dingsen, produktet.

I tillegg til *Programmering* knytter L4 kompetansemålet opp mot *Abstraksjon, kreativitet og generalisering*, et undertema innenfor *Algoritmisk tenkning som problemløsning*.

Etter spørsmålet tilknyttet kompetansemålet fikk lærerne spørsmål om hvilke ferdigheter elevene får øvd på i arbeidet med kompetansemålet, og hvilke aktiviteter man kan jobbe med for å nå kompetansemålet. På disse spørsmålene gikk også *Programmering* igjen hos samtlige lærere, både som ferdighet og aktivitet, men i tillegg dukket *Systemtenkning* opp i besvarelsene til tre av lærerne. *Systemtenkning* er et undertema innenfor *Algoritmisk tenkning som en del av naturfag og matematikk*. Svar som passet inn i dette hovedtemaet hadde ikke kommet frem hos noen av lærerne innledningsvis i intervjuet. Et eksempel på dette er da L1 skulle beskrive aktiviteter tilknyttet kompetansemålet og ga en besvarelse som passet inn under *Systemtenkning*.

L1: Jeg tror vel det at det, at det, de kan jo lage seg. For eksempel brytere og ting som, ja, egentlig er funnet opp før da, men finne ut hvordan ting er laget, bygd opp. Hvordan ting blir, ja styringer da av enkelte ting.

Her uttrykker L1 en forståelse av at elevene gjennom programmering kan lære få kunnskap om hvordan ulike systemer består av deler som virker sammen. Samme forståelse kom til uttrykk i L3 sin besvarelse.

L3: Det blir jo å se hvordan deler kan bli satt sammen til en helhet da. Og få det til å funke. Vil jeg tro.

L2 gir også et eksempel på en aktivitet hvor elevene må kombinere sin naturfaglige kunnskap og programmeringskunnskap:

L2: Da må man kanskje for eksempel lage et vanningsanlegg gjennom programmering og bruke micro:biter og programmere til at man vanner. [...] Og så er det jo klart at da bruker de jo både naturfagskunnskapen sin, hvor mye de skal vanne, og programmeringssystemet sitt for å få det her til å fungere. Også er det jo ganske vanskelig å få det til, så de må jo jobbe med det for å få det til. Hvordan skal de stå, hvor skal planten sitte, hvor vannet skal, hvor det skal helles ut vann og sånn.

Da det i starten av intervjuet kunne virke som om lærerne oppfattet det aktuelle kompetansemålet som et rent programmeringsmål fristilt fra resten av naturfagen tyder sitatene over på at dette ikke nødvendigvis er tilfellet for alle lærerne. *Systemtenkning* regnes som en viktig ferdighet innenfor både matematikk og naturfag, og knytter da forståelsen av kompetansemålet opp mot naturfag for øvrig. L2 sier også helt eksplisitt at elevene må kombinere sin programmeringskunnskap med sin naturfaglige kunnskap, noe som tyder på en forståelse for at programmering kan være et verktøy i naturfag. En forståelse for programmering som et faglig verktøy blir enda tydeligere når L2 senere sier:

L2: Også må man kanskje bli mer bevisste blant lærere, at dette her ikke er noe farlig, folk tror, å det er programmering ja, men det her kan også brukes i norsk, matte, samfunnsfag og KRLE.

Ut over en til dels felles forståelse av *Programmering* og *Systemtenkning* som en del av ferdighetene og aktivitetene tilknyttet kompetansemålet, ga lærerne i stor grad veldig ulike svar. Felles for alle lærerne var derimot at svarene på disse to spørsmålene tydet på en bredere og mer variert forståelse av kompetansemålet enn det som ble uttrykt innledningsvis i intervjuet. Som eksempel på dette mener L5 at elevene først og fremst får jobbet med de grunnleggende ferdighetene:

L5: Nei, her er det grunnleggende ferdigheter. Det er lesing, det er skriving, det er muntlig, men ikke minst så er det jo praktiske ferdigheter.

L1 på sin side mener elevene får øvd på *Algoritmer*:

L1: Jeg tenker jo at i første omgang så er det det her å ta en instruksjon. Følge en oppskrift.

L6 beskriver derimot at elevene får jobbet med det motsatte av det L1 beskriver:

L6: Nei, det er jo det her å, jeg tenker at de blir jo utfordret til å tenke selv og utfordre seg på noe mer enn det at de bare får en oppskrift da, kanskje.

Ifølge L6 er det ikke å følge en oppskrift elevene får øvd på, men heller det å tenke selv. L6 sin beskrivelse tolkes som at elevene heller får øvd seg på å utforske og undersøke selv. Under undertemaet *Abstraksjon, kreativitet og generalisering* ligger idémyldring, samt å se for seg hvilke muligheter man har når man skal løse et problem. Dette undertemaet kan derfor sies å inneholde utforskning i form av at elevene må bruke kreativiteten sin til å komme med ulike forslag til idéer for å løse et problem. L6 beskriver altså at elevene får øvd på en helt annen ferdighet enn det L1 beskriver, men til tross for at disse to besvarelsene kan virke til å være motsetninger utelukker de ikke hverandre ettersom begge beskrivelsene inngår i det analytiske rammeverket for algoritmisk tenkning.

På det første temaspesifikke spørsmålet var det en felles, men begrenset forståelse, av kompetansemålet som i stor grad kun omhandler *Programmering*. Alle lærerne brukte *Programmering* for å beskrive hva de la i kompetansemålet. To av lærerne knyttet i tillegg kompetansemålet opp mot problemløsning. Verbene *lage, utvikle og programmere* benyttes spesifikt i kompetansemålet, men det var verbet *programmering* som hyppigst ble brukt da lærerne skulle forklare sin forståelse. De to andre verdiene ble i liten grad brukt av lærerne for å beskrive kompetansemålet. Under spørsmålet tilknyttet ferdigheter og aktiviteter tilknyttet kompetansemålet ga de fleste lærerne fortsatt besvarelser som kunne kategoriseres under *Programmering*. Til forskjell fra de innledende besvarelsene ble det i denne delen av intervjuet også gitt beskrivelser på ferdigheter og aktiviteter som kunne kategoriseres under *Systemtenkning*. Svar innenfor dette undertemaet var ikke blitt nevnt tidligere. Lærerne viste på disse spørsmålene derfor en bredere forståelse for kompetansemålet enn de hadde gjort innledningsvis, men ut over *Programmering* og *Systemtenkning*, var det veldig varierende hva som lå i denne bredere forståelsen. At svarene på de tre spørsmålene tilknyttet kompetansemålet varierer så mye kan tolkes som at det er lite felles forståelse for hva som ligger i kompetansemålet.

## 6.2 Begrenset eller manglende forståelse av algoritmisk tenkning

Under intervjuene fikk lærerne som på eget initiativ tok i bruk begrepet *algoritmisk tenkning* et oppfølgingsspørsmål der de ble spurt om hva de legger i dette begrepet. Lærere som ikke selv tok i bruk *algoritmisk tenkning*, fikk spørsmål om hvilken kjennskap de hadde til begrepet. Kun to av de seks lærerne brukte på eget initiativ *algoritmisk tenkning* som begrepet i spørsmålene knyttet til kompetansemålet. L1 var en av fire lærere som ikke selv tok i bruk begrepet *algoritmisk tenkning*. På spørsmålet om sin kjennskap til begrepet svarte læreren:

L1: Veldig lite. Det er ikke, det har ikke vært et tema egentlig. Det har ikke vært noe fokus på det.

Dette svaret kan tyde på at L1 har liten eller ingen kjennskap til begrepet. Noe av det samme var tilfellet for L6. som heller ikke kunne gi noen forklaring på hva som lå i begrepet.

L6: Jeg er veldig dårlig på begreper så algoritmisk tenking så tenker jeg, egentlig sånn i utgangspunktet så bare, algoritmisk tenking, altså algoritmer det er jo på en måte, ja, hvordan skal jeg forklare det med normale ord da. Men der er jo, nei jeg klarer ikke å forklare det akkurat nå jeg.

L3 og L5 benyttet heller ikke begrepet spontant i sine besvarelser, men ved videre utspørring prøvde begge disse to lærerne seg på en forklaring.

L3: Ja, det er en formel eller en metode for å komme frem til et svar.

L3 viser gjennom dette sitatet en forståelse for begrepet som går inn under undertemaet *Algoritmer*.

L5: Altså at du følger en viss mal, en viss oppskrift, en viss strategi for å få ønsket resultat da. [...] Og da har vi hatt ren opplæring i det i forhold til hva en kode er for noe, hva er, ikke brukt ordet algoritme enda, men fulgt liksom en oppskrift da og mal og for å liksom sette dem i riktig rekkefølge slik vi ønsker at hendelsen skal skje da.

L5 knytter, i likhet med L3, begrepet *algoritmisk tenkning* til undertemaet *Algoritmer* ved å trekke frem evnen til å følge en bestemt oppskrift eller mal for å komme frem til et resultat. Samtidig kan besvarelsen til L5 i sitatet over også kategoriseres under undertemaet *Programmering* ettersom L5 trekker inn kodeopplæring i form av koding og syntaks innenfor et programmeringsspråk. L3 og L5 viser i sine besvarelser at de har en formening om hva som ligger i begrepet, og til dels en felles forståelse, men denne forståelsen dekker bare en liten del av det analytiske rammeverket.

To av lærerne, L2 og L4, benyttet begrepet algoritmisk tenkning spontant i sine besvarelser innledningsvis i intervjuet. Da disse to lærerne fikk oppfølgingsspørsmålet om hvordan de forstod dette begrepet oppga de ulik forståelse. På oppfølgingsspørsmålet svarte L4:

L4: Det å kunne, det er det her å lage, forholde seg til en slags oppskrift. Det å forstå instruksjoner, kunne lage instruksjoner.

L4 sin forklaring av begrepet kan plasseres innenfor *Algoritmer*. L4 viste derfor samme forståelse av begrepet som L3 og L5. Eneste forskjellen her var at L4, i motsetning til de to andre lærerne, benyttet begrepet spontant. Dette kan tyde på at L4, i større grad enn de to andre lærerne, assosierer begrepet med kompetansemålet. L2 nevne også begrepet spontant, men hadde ved oppfølgingsspørsmålet en helt annen forståelse av begrepet enn det de andre lærerne hadde oppgitt tidligere.

L2: At man skal, litt mer nær det å jobbe som et vanlig voksent menneske, ved at du skal få et problem som du skal løse. Så skal du bruke kunnskapen din for å løse problemet. Også møter du motstand, også må du evaluere og gå tilbake igjen og hele tiden frem og tilbake, frem og tilbake. Og til slutt evaluere og gå helt tilbake og se hva var egentlig intensjonen.

Denne forståelsen av begrepet skiller seg fra de andre lærerne. Mens de andre lærerne knytter begrepet direkte opp mot *Algoritmer*, knytter L2 sin forståelse til undertemaet *Overførbarhet og Gjentakelse*, et undertema under hovedtemaet *Algoritmisk tenkning som problemløsning*. L2 trakk hverken frem *Programmering* eller *Algoritmer* i sin forklaring slik de andre lærerne hadde gjort. Ettersom L2 bruker begrepet spontant kan dette tyde på en assosiasjon mellom begrepet og kompetansemålet slik som hos L4.

Flertallet av lærerne som ble intervjuet brukte ikke begrepet algoritmisk tenkning spontant i sine besvarelser tilknyttet kompetansemålet i naturfag. Av de lærerne som ikke selv brukte begrepet, var det to av lærerne som ikke kunne gi noen forklaring på begrepet ved nærmere utspørring. De to andre lærerne som heller ikke hadde brukt begrepet spontant ga beskrivelser for algoritmisk tenkning som kan kategoriseres innenfor *Programmering* og *Algoritmer*. To av lærerne tok i bruk algoritmisk tenkning på eget initiativ, men forståelsen for hva dette begrepet innebar var helt ulik hos disse lærerne. Dette tyder på en begrenset og manglende forståelse av begrepet hos alle lærerne. Ingen av besvarelsene som ble gitt kunne kategoriseres inn under hovedtemaet *Algoritmisk tenkning som en del av matematikk og naturfag*. Dette underbygger inntrykket fra forrige funn om at lærerne opplever algoritmisk tenkning og programmering som et tema fristilt fra resten av naturfagundervisningen.

### 6.3 Behov for kurs og etterutdanning

Avslutningsvis i intervjuet ble lærerne spurt om det var noe de ønsket å utdype eller tilføye. På dette spørsmålet valgte fire av lærerne spontant å trekke frem behovet for kurs og muligheten til å oppdatere seg faglig. Kurs, videreutdanning og etterutdanning innenfor algoritmisk tenkning og programmering var ikke et tema i intervjuguiden. Det var derfor overraskende at så mange av lærerne valgte å trekke frem dette på eget initiativ. Et slikt behov går inn under *Lærerkompetanse* som er en underkategori av *Algoritmisk tenkning og ytre faktorer som påvirker undervisning*.

L1: Generelt sett så er det dette her at det er et godt stykke fra lærerplan til praksisen i klasserommet. [...] Og kurs har vi minimalt på. Vi hadde, vi var i gang med litt kursing i forhold til kunnskapsløftet før koronaen kom, men så stoppet det opp og det har jo egentlig stoppet opp, ikke begynt igjen. Kunnskapsløftet innført, men vi har egentlig fått jobbet alt for lite med det. Og det gjør at du på en måte føler at du, en ting er at du ikke får noe faglig påfyll, men stagnerer litt, du gjør det du stort sett har gjort de siste årene. Vi har jo gode fagplaner og årsplaner og alt dette der, og ting blir jo bra, men det blir ikke akkurat dette. Så det er litt, jeg syns det har stoppet opp litt i utviklingen jeg altså, den faglige utviklingen.

L1 forteller om manglende kursing og for lite tid til å jobbe med temaet etter at det ble innført i LK20. L1 beskriver en følelse av å ha stagnert og stoppet opp i den faglige utviklingen, noe som kan tolkes som at læreren opplever å ha ønsket kompetanse innenfor temaet. L3 gir uttrykk for samme opplevelse.

L3: Nei. Egentlig ikke, men det har jo vært veldig lite, altså det står i fagplaner og sånn at vi skal gjøre det, men det har jo vært veldig lite kursing for oss lærerne på hvordan vi faktisk skal jobbe med det. Så man føler jo at man begynner litt på bar bakke. Og særlig når du er liksom 50 år og skal begynne å sette seg inn i det her. Så skulle det ha vært fint med litt mer kurs og samarbeid innad og på hvordan vi skal jobbe med det.

L3 tar også opp mangelen på kurs og veiledning knyttet til praktisk gjennomføring av undervisning for å nå kompetansemålet. L3 sitt utsagn om å føle at man begynner på bar bakke tolkes, på lik linje som hos L1, som en opplevelse av å ikke ha tilstrekkelig kompetanse innenfor emnet. Både L1 og L3 nevner at det står i styringsdokumenter som fagplaner og årsplaner, men begge sitatene kan tyde på en opplevelse av at det har stoppet her. L3 nevner i tillegg til kurs et behov for samarbeid internt rundt hvordan temaet skal jobbes med.

L5 trekker også frem viktigheten av kurs og opplæring og knytter dette direkte opp mot kvaliteten på undervisning:

L5: Så det er veldig, tenker at programmering som og i naturfag og matematikk er et viktige kompetansemål, men at jeg ser viktigheten av at vi får kurs og opplæring i det vi skal gjøre. For at det skal bli kvalitet ut av det og i undervisningen da. Rett og slett.

L4 tar også opp dette behovet i sin besvarelse, men denne besvarelsen beskriver et annet utgangspunkt enn hos de tre andre.

L4: Nei, jeg tenker at jeg har vært på kurs i dag, og nå begynner jeg å få sånn grei oversikt over programmering i barneskolen. Men jeg er jo en av få som har fått såpass mye som jeg har, også tenker jeg at, og merker selv hvor mye jeg må tenke på hva er det som ligger i det her, hva er det som ligger i den undervisningen også de som da ikke har det og heller ikke programmering i det hele tatt. Hvordan vil de møte det her målet? Det er vel det eneste.

L4 beskriver å ha fått en del kurs, og at dette begynner å gi en grei oversikt over emnet. Samtidig forsterker L4 sitt svar det de andre tre lærernes beskrivelse av manglende kursing, ved å presisere at det er få lærere som har fått tilsvarende mengde kurs og opplæring. L4 sin beskrivelse tyder også på en opplevelse av at det krever mye av lærerne å undervise i innenfor dette temaet, og at manglende kunnskap og kompetanse hos lærere kan gjøre det vanskelig for å gjennomføre undervisning tilknyttet algoritmisk tenkning og programmering.

Fire av lærerne som ble intervjuet oppga behovet for kurs, etterutdanning og videreutdanning tilknyttet temaet. Dette temaet ble tatt opp av lærerne på eget initiativ, og var uventet ettersom det ikke var et tema i intervjuguiden. Dette behovet ble blant annet knyttet til å øke kvaliteten på undervisningen, samt et behov for faglig påfyll og utvikling hos lærerne. Det beskrives også en manglende forklaring på hvordan føringene i årsplaner og fagplaner skal overføres til undervisning.

## 7. Drøfting

### 7.1 Lærernes forståelse av kompetansemålet i naturfag

I naturfag er det to kompetansemål tilknyttet algoritmisk tenkning, et kompetansemål etter 7. trinn og et kompetansemål etter 10. trinn (Utdanningsdirektoratet, 2020c, 2020g). På bakgrunn av masteroppgavens omfang og problemstilling er det naturfagslærere på mellomtrinnet sin forståelse av kompetansemålet etter 7.trinn som drøftes. Kompetansemålet etter 10. trinn blir derfor ikke diskutert i denne oppgaven. Kompetansemålet etter 7. trinn består av tre verb: *utforske*, *lage* og *programmere*. Innledningsvis i intervjuet skulle lærerne forklare hvordan de forstod dette kompetansemålet. Alle lærerne brukte programmering for å beskrive kompetansemålet, mens de to andre verbene i liten grad ble nevnt. Dette gir inntrykk av at det legges mye større vekt på programmeringsaspektet i kompetansemålet, enn den delen som handler om å utforske og å lage. Dette til tross for at både å utforske og å lage trekkes frem som viktige ferdigheter for den algoritmiske tenkeren på Utdanningsdirektoratet (2019) sin ressursnettside for lærere.

I noen av lærernes besvarelser var fokuset på programmering så fremtredende at det kunne virke som om lærerne oppfattet kompetansemålet som et rent programmeringsmål. Lignende funn har også blitt gjort av Cabrera (2019), hvor flere av lærerne som deltok i hennes studie forstod programmering som et mål i seg selv. Av lærerne med denne forståelsen ble programmering sett på som en ferdighet som ikke var relevant utenfor rene programmeringsaktiviteter. Inntrykket av at lærerne som deltok i denne masteroppgaven så på kompetansemålet som et rent programmeringsmål ble ytterligere forsterket da en av lærerne valgte å beskrive kompetansemålet som et mål innenfor det «relativt nye programmeringsfaget». Til tross for at kompetansemålet ikke nevner algoritmisk tenkning, er den overordnede intensjonen at kompetansemålene i LK20 som omhandler programmering skal inngå som en del av en bredere intensjon tilknyttet algoritmisk tenkning og elevenes digitale kompetanse (Bocconi et al., 2022; Vinnervik & Bungum, 2022). Det å undervise i algoritmisk tenkning uten at det for elevene oppleves som synonymt med programmering, er en av de største pedagogiske utfordringen innenfor dette området (Lu & Fletcher, 2009). Denne pedagogiske utfordringen blir mest sannsynlig ikke mulig å løse hvis også lærerne oppfatter at det elevene skal lære er ren programmering.



Til tross for anbefalingene til Sanne et al. (2016) ble det i LK20 ikke opprettet noe eget fag tilknyttet algoritmisk tenkning, programmering eller teknologi. Istedenfor ble det integrert i enkelte av de allerede etablerte skolefagene (Andersen et al., 2023; Bocconi et al., 2022). At én av lærerne likevel omtaler kompetansemålet som en del av «et nytt programmeringsfag», kan kanskje tyde på at integreringen av algoritmisk tenkning og programmering som deler av de allerede etablerte skolefagene har vært helt vellykket. Begrunnelsen for at det ble anbefalt å opprette et nytt fag var for å unngå nedprioriteringer slik man hadde erfart ved tidligere tverrfaglige emner. Når lærerne beskriver kompetansemålet etter 7.trinn i programmering virker det ikke som om dette kompetansemålet nedprioriteres. Alle lærerne gir beskrivelser av sin forståelse av kompetansemålet, samt ferdigheter elevene får jobbet med og aktiviteter man kan benytte i arbeidet. Det virker innledningsvis i intervjuene heller som at lærerne kanskje ikke er bevisst den overordnede intensjonen i LK20 og tilhørende styringsdokumenter, hvor programmering og problemløsning inngår som ferdigheter i algoritmisk tenkning og at målet med programmering er at det skal benyttes som et verktøy for å lære fag (Andersen et al., 2023)

At lærernes forståelse av kompetansemålet i hovedsak er knyttet til programmering er ikke nødvendigvis så overraskende. Programmering er nemlig den ferdigheten som i størst grad trekkes frem i LK20 tilknyttet algoritmisk tenkning (Vinnervik & Bungum, 2022). I læreplanen for naturfag nevnes ikke algoritmisk tenkning eksplisitt, og av de informatikkinspirerte begrepene er det bare programmering som blir nevnt i kompetansemålene for naturfag. Kompetansemålene innenfor et fag er det lærerne i størst grad forholder seg til i hverdagen (Vinnervik & Bungum, 2022). I tillegg vil læreres forståelse av algoritmisk tenkning som konsept mest sannsynlig påvirke hvordan de underviser i dette konseptet (Cabrera, 2019). Hvis lærernes forståelse av kompetansemålet, og da indirekte algoritmisk tenkning, er at dette betyr programmering er det derfor sannsynlig at det også blir programmering elevene får undervisning i. At elevene får undervisning i ren programmering er ikke nødvendigvis problematisk. Grover og Pea (2013) mener at programmering er helt grunnleggende innenfor algoritmisk tenkning, og det viktigste verktøyet når man skal lære elever denne tenkemåten. Tar man utgangspunkt i Grover og Pea (2013) kan man derfor anta at elevene til dels vil lære algoritmisk tenkning indirekte gjennom arbeidet med ren programmering. Spørsmålet er bare hvor vidt elevene gjennom ren programmeringsundervisning oppnår den kompetansen man så for seg da man innførte algoritmisk tenkning og programmering i LK20.

For å få en vellykket innføring av et nytt tema i læreplanen holder det ikke kun å endre styringsdokumentene (Barr & Stephenson, 2011). Skal det nye temaet bli formidlet slik det er

tiltenkt må lærerne få en tydelig innføring i det nye temaet, noe som innebærer økt faglig kompetanse, økt pedagogiske kompetanse, samt økt teknologisk kompetanse (Barr & Stephenson, 2011; Cabrera, 2019; Yadav et al., 2017). Styringsdokumentene ble endret da LK20 ble innført, men spørsmålet er om man har klart å holde en rød tråd fra det som var hensikten bak styringsdokumentene, til utformingen av læreplanen, og avslutningsvis det i dag praktiseres i klasserommet. Hva algoritmisk tenkning innebar i anbefalingen til Sanne et al. (Sanne et al., 2016) virker ikke å være det samme som hvordan algoritmisk tenkning beskrives i LK20 (Bocconi et al., 2022). I LK20 beskrives algoritmisk tenkning hovedsakelig som en problemløsningsprosess (Bocconi et al., 2022), en beskrivelse som er smalere enn den som ble gitt i anbefalingene (Sanne et al., 2016). Samtidig fokuserer kompetansemålene tilknyttet algoritmisk tenkning på programmering, og funnene fra denne studien indikerer at også lærerne i stor grad oppfatter kompetansemålet etter 7.trinn i naturfag som et programmeringsmål. Det at lærerne fokuserer på programmering er ikke nødvendigvis problematisk, men gir dette elevene den kompetansen de trenger i fremtiden?

I LK20 er programmering beskrevet som en metode og et verktøy for å jobbe med de ulike fagene, og ikke en ferdighet som et mål i seg selv (Andersen et al., 2023; Vinnervik & Bungum, 2022). Dette var også anbefalingen i forkant av at algoritmisk tenkning og programmering ble innført i den norske læreplanen (Sanne et al., 2016). En av utfordringene med LK20 er at hvis man skal få med seg denne intensjonen må læreplanen studeres overordnet og på tvers av fag (Vinnervik & Bungum, 2022). Bruken av programmering som et faglig verktøy har derfor ikke blitt så tydelig i LK20 som det som var anbefalingen i forkant av innføring. Lærernes innledende beskrivelser av kompetansemålet ga inntrykk av at de forstod programmering som et tilleggstema i naturfag, fristilt fra det øvrige pensumet i faget. Det ble i liten grad vist noen forståelse for den overordnede intensjonen med programmering i LK20 som Vinnervik og Bungum (2022) viser til, eller som ble anbefalt av Sanne et al. (2016). Et lignende funn har også blitt gjort blant lærere i matematikk, hvor lærerne fokuserer på programmering uten å knytte det opp mot resten av matematikkfaget (Nordby et al., 2022). Også i denne studien oppgis den hyppige bruken av programmeringsbegrepet i LK20 som en forklaring på denne forståelsen blant matematikklærerne (Nordby et al., 2022). En annen mulig forklaring kan være at det å utnytte programmering som et faglig verktøy stille høye krav til lærernes programmeringskompetanse (Andersen et al., 2023). I denne studien var det kun én av lærerne som hadde videreutdanning tilknyttet programmering. En forklaring på at

lærerne sliter med å koble programmeringen opp mot naturfag kan derfor være at lærerne ikke har den nødvendige kompetansen som skal til for å få til dette.

Programmering er et konsept hentet fra informatikk, og vil derfor være et nytt og lite kjent begrep og konsept for de fleste lærere (Andersen et al., 2023; Bocconi et al., 2022). I tillegg er det ulikt hvordan programmering blir forklart og definert. Da programmering ble anbefalt som et nytt konsept i LK20 ble det forklart som koding, nemlig å oversette en algoritme til et programmeringsspråk (Bocconi et al., 2022; Sanne et al., 2016). En slik smal forklaring oppmuntrer ikke nødvendigvis til en forståelse for programmering som et faglig verktøy. Programmering som en problemløsningsferdighet derimot kan i større grad legge til rette for en forståelse for programmering som et faglig verktøy. I læreplanen i naturfag gis det ikke noen forklaring av hva programmering innebærer, annet enn at programmering presenteres som et verktøy for å lære fag, noe som krever en høy kompetanse innenfor programmering hos både elever og lærere for å være mulig (Andersen et al., 2023; Vinnervik & Bungum, 2022). Det er derfor ikke nødvendigvis så uventet at lærerne har problemer med å se hvordan programmeringen kan knyttes opp mot faget. Uten en bakgrunn fra informatikk, eller tidligere kjennskap til konseptet programmering, kan det tenkes at det er omfattende nok bare å sette seg inn i hva programmering egentlig innebærer. Ettersom programmering heller ikke er tydelig forklart i læreplanen for naturfag, og veldig smalt forklart i anbefalingene til Sanne et al. (2016), er ikke det å forstå konseptet heller uproblematisk. Ut fra hvor man leter er det stor sannsynlighet for at man vil få ulik forklaring på hva programmering innebærer, noe som igjen kan gjøre det vanskelig å forstå helt eksplisitt hvordan man skal koble programmering opp mot det allerede etablert pensum.

To av lærerne beskrev, i tillegg til programmering, problemløsning som en del av sin forståelse for kompetansemålet i naturfag. Dette til tross for at problemløsning ikke er nevnt i dette kompetansemålet. Samtidig inngår både programmering og problemløsning i beskrivelsen av algoritmisk tenkning i norske styringsdokumenter (Andersen et al., 2023). I overordnet del av læreplanen i matematikk, det eneste stedet hvor algoritmisk tenkning er nevnt, beskrives begrepet hovedsakelig som en problemløsningsprosess (Bocconi et al., 2022; Vinnervik & Bungum, 2022). Som vist i tabell 3 underviser fire av de seks lærerne som ble intervjuet også i matematikk. Det kan derfor tenkes at lærerne som trakk frem problemløsning gjorde dette på bakgrunn av hvordan algoritmisk tenkning blir beskrevet i læreplanen for matematikk (Utdanningsdirektoratet, 2020a). Hvis dette er tilfellet tyder det på en forståelse hos disse lærerne for at kompetansemålet i naturfag også er tilknyttet algoritmisk tenkning. En

slik forståelse fordrer at lærerne har høy kompetanse og har fått nok tid til å jobbe med temaet (Vinnervik & Bungum, 2022). Ettersom lærerne selv trekker frem at de føler å ikke ha fått nok tid til å sette seg inn i algoritmisk tenkning og programmering, samt etterspør mer kurs, er denne forklaringen i mindre grad sannsynlig.

En annen mulig forklaring kan være at algoritmisk tenkning i stor grad er beskrevet som en problemløsningsressurs på Utdanningsdirektoratet sin ressurside for lærere (Utdanningsdirektoratet, 2019). Samtidig fordrer også denne forklaringen at lærerne knytter kompetansemålet i naturfag opp mot algoritmisk tenkning, noe som med utgangspunkt i diskusjonen i forrige avsnitt er mindre sannsynlig. En kanskje mer sannsynlig forklaring kan være at lærerne assosierer problemløsning med kompetansemålet, fordi de forstår programmering som en problemløsningsprosess i seg selv. Problemløsning inngår som en del av flere internasjonale definisjoner på programmering (Bocconi et al., 2022). Samtidig har tidligere norsk forskning vist at lærere ofte knytter programmering til algoritmer og mønstre, og at de ut over dette har utfordringer med å definere hva programmering egentlig innebærer (Kravik et al., 2022).

Under intervjuene ble lærerne også spurt om hvilke ferdigheter elevene fikk øvd på i arbeidet med kompetansemålet, og hvilke aktiviteter som kunne benyttes. På disse spørsmålene ble også programmering trukket frem av de fleste lærerne som eksempel på både ferdighet og aktivitet. Til forskjell fra tidligere ble det på disse spørsmålene gitt besvarelser som dekket større deler av det analytiske rammeverket enn tidligere. Halvparten av lærerne trakk inn systemtenkning som eksempel på ferdigheter og aktiviteter tilknyttet kompetansemålet. Gjennom beskrivelsene av systemtenkning ble programmeringen i større grad enn tidligere aktualisert opp mot det allerede etablerte naturfagspensumet. Disse beskrivelsene var mer i tråd med den overordnede intensjonen i LK20 som Vinnervik og Bungum (2022) viser til, samt de anbefalingene som ble gitt i forkant av innføring (Sanne et al., 2016). Ut over programmering og systemtenkning var det veldig ulikt hvilke ferdigheter og aktiviteter lærerne beskrev, og hvor disse kunne plasseres i det analytiske rammeverket. Utover programmering var det derfor ingen tydelig, felles forståelse hos lærerne for hva som lå i dette kompetansemålet. Forståelsen for kompetansemålet fremstod derfor som mangelfull både sett opp mot det analytiske rammeverket, den overordnede intensjonen i LK20 (Vinnervik & Bungum, 2022) og anbefalingene i forkant av innføring (Sanne et al., 2016).

Den begrensede forståelsen for kompetansemålet kan tyde på at lærerne ikke har høy nok kompetanse innenfor algoritmisk tenkning og programmering, og at de heller ikke har fått nok

tid til å øke egen kompetanse (Andersen et al., 2023; Vinnervik & Bungum, 2022). Styringsdokumentene er endret, men det kan virke som at den delen av vellykket implementering som innebærer å heve lærernes kompetanse ikke enda ikke er helt på plass (Barr & Stephenson, 2011). En slik kompetanseheving handler ikke bare om å gi lærerne øke lærernes faglige kompetanse tilknyttet algoritmisk tenkning og programmering, men også deres pedagogiske- og teknologiske kompetanse (Cabrera, 2019; Yadav et al., 2017). Manglende tid til å jobbe med temaet, samt behov for kurs, etterutdanning og videreutdanning var tema lærerne selv trakk frem uoppfordret under intervjuene. Dette var uventet ettersom dette var tema som ikke inngikk i intervjuguiden. Når lærerne selv i så stor grad trekker frem og poengterer behov for tid, kurs og utdanning gir dette inntrykk av at dette er et reelt behov. Samtidig styrker det mistanken om at lærernes begrensede forståelse for kompetansemålet skyldes manglende kompetanse. Behovet for tid til å jobbe med det nye temaet, samt tilgang på opplæring og kurs, ble også adressert av lærerne i både Nordby et al. (2022) og Kravik et al. (2022) sine studier. Dette kan tyde på at manglende etter- og videreutdanning ikke bare er et problem for lærerne i denne masteroppgaven, men at dette problemet er mer generelt og gjelder for flere lærere i Norge.

I tillegg til muligheten for å heve egen kompetanse er en viktig støtte for lærere i arbeidet med algoritmisk tenkning og programmering at det også settes av tid til å jobbe med temaet sammen i kollegiet (Bocconi et al., 2022). Det bør også etableres gode muligheter for lærere innad i kollegiet og på tvers av skoler til å jobbe med faget, dele erfaringer og konkrete undervisningstips (Bocconi et al., 2022). Den varierte forståelsen av kompetansemålet og tilhørende arbeidsmåter kan tyde på at det ikke har blitt satt av nok tid til å jobbe med det nye temaet felles blant lærerne i denne studien. Hvis det heller ikke har blitt tilrettelagt for kompetanseheving i form av etter- og videreutdanning blir det vanskelig å oppnå den bredere innfallsvinkelen LK20 legger til rette for (Vinnervik & Bungum, 2022). Hvis det ikke har blitt satt av tid til å øke kompetansen tilstrekkelig hverken hos den enkelte lærer, eller i kollegiet, er det ikke rart om forståelsen for kompetansemålet blir begrenset, mangelfull og variert. Uten avsatt tid kan det tenkes at det i stor grad blir opp til hver enkelt lærer å sette seg inn i det nye temaet og tilhørende kompetansemål. En oppgave som nødvendigvis ikke er så enkel ettersom beskrivelsene av konseptene i læreplanen er tynne, og definisjonene på programmering og algoritmisk tenkning utenfor læreplanen er mange og ulike.

I rapporten til Sanne et al. (2016) var anbefalingen å opprette et nytt, obligatorisk fag som skulle omfatte teknologi og programmering. Dette var for å unngå samme nedprioritering som

man hadde erfart ved det tverrfaglige Teknologi-emnet i LK06. For å unngå at nye temaer i læreplanen nedprioriteres må disse tilrettelegges for ved at det blir gitt tilstrekkelig tid og ressurser (Bocconi et al., 2022). Dette inkluderer endring av styringsdokumenter, samt faglig-, pedagogisk-, og teknologisk kompetanseheving hos lærerne (Barr & Stephenson, 2011; Cabrera, 2019; Yadav et al., 2017). I LK20 ble det ikke opprettet noe nytt fag som anbefalt. I stedet ble algoritmisk tenkning og programmering implementert i noen bestemte fag (Andersen et al., 2023). Når lærerne i denne studien skulle uttrykke sin forståelse for kompetansemålet etter 7.trinn i naturfag var programmering den forståelsen som felles for alle lærerne og som det tydeligst ble gitt uttrykk for. Problemløsning og systemtenkning ble også trukket frem, men disse forståelsene var ikke like fremtredende som programmeringsforståelsen. Sett opp mot anbefalingene i forkant av innføring (Sanne et al., 2016), intensjonen med algoritmisk tenkning i LK20 (Andersen et al., 2023; Vinnervik & Bungum, 2022), og det analytiske rammeverket for algoritmisk tenkning benyttet i denne oppgaven er forståelsen lærerne uttrykket begrenset og mangelfull. Med bakgrunn i tidligere diskusjon er det flere faktorer som tyder på at det heller ikke ble satt av nok tid og ressurser til å øke lærernes kompetanse i forkant eller etterkant av innføringen av algoritmisk tenkning og programmering. Hvis dette er tilfellet mangler en viktig del av forutsetningene for at implementeringen skal bli vellykket (Barr & Stephenson, 2011; Yadav et al., 2017).

## 7.2 Lærernes forståelse av algoritmisk tenkning

I norsk utdanningssammenheng blir begrepet algoritmisk tenkning hovedsakelig forklart som en problemløsningsprosess, eller en problemløsningsferdighet (Utdanningsdirektoratet, 2019, 2020a). I anbefalingen til Sanne et al. (2016) ble algoritmisk tenkning blant annet definert som evnen til å kunne abstrahere, gjennomgå informasjon systematisk, tolke og forstå ulike representasjonsformer, modellere og resonnerer. Den anbefalte forklaringen på hva algoritmisk tenkning innebar var altså mye bredere i anbefalingene enn det som ble resultatet i LK20. Hvertfall hvis man kun tar utgangspunkt i den eneste eksplisitte beskrivelsen av algoritmisk tenkning i læreplanen til matematikk hvor algoritmisk tenkning hovedsakelig beskrives som en problemløsningsprosess (Utdanningsdirektoratet, 2020a). Overordnet i norske styringsdokumenter blir algoritmisk tenkning forklart på to måter. Det blir forklaring som problemløsning og programmering, og som et faglig verktøy (Andersen et al., 2023). Anbefalingen fra Sanne et al. (2016) og det som faktisk ble beskrivelsen av algoritmisk tenkning i norske styringsdokumenter (Andersen et al., 2023), gir derfor uttrykk for to ulike

forståelser av hva algoritmisk tenkning innebærer. En tredje forståelse vil være den lærerne sitter med. Dette er kanskje også den viktigste forståelse ettersom det er de som gjør teori til praksis gjennom å formidle sin forståelse av læreplanen og konseptet algoritmisk tenkning til elevene (Cabrera, 2019; Vinnervik & Bungum, 2022).

Det var ønskelig å undersøke om lærerne selv knyttet kompetansemålet i naturfag opp mot algoritmisk tenkning. Alle kompetansemålene i LK20 hvor programmering inngår kan nemlig knyttes til algoritmisk tenkning hvis man ser på læreplanen overordnet og på tvers av fag (Vinnervik & Bungum, 2022). For å undersøke om lærerne hadde en slik overordnet forståelse ble det under intervjuene gjort et bevisst valg om å ikke nevne algoritmisk tenkning innledningsvis. Målet var å se om lærerne selv var bevisst den overordnede intensjonen med kompetansemålet. Fire av lærerne tok ikke i bruk begrepet algoritmisk tenkning spontant innledningsvis i intervjuet. Det at så få lærere tok i bruk begrepet spontant kan ha flere mulige forklaringer. Det kan skyldes at lærerne har liten eller ingen kjennskap til begrepet.

Manglende kjennskap til begrepet kan være et resultat av at begrepet i liten grad omtales i læreplanen (Vinnervik & Bungum, 2022). Det kan også et resultat av at begrepet er tatt fra et fagområde få lærere har bakgrunn fra, eller god kjennskap til (Andersen et al., 2023; Bocconi et al., 2022). Samtidig underviste fire av lærerne også i matematikk og bør derfor ha lest definisjonen på algoritmisk tenkning i læreplanen til matematikk. En mulig forklaring kan derfor være at de ikke assosierer denne beskrivelsen av algoritmisk tenkning med kompetansemålet tilknyttet programmering i naturfag. Det kan også hende at de ikke assosierer algoritmisk tenkning med programmering i det hele tatt. Dette kan igjen være et resultat av at de ikke har fått jobbet nok med den nye læreplanen og heller ikke har nok kompetanse rundt programmering og algoritmisk tenkning (Bocconi et al., 2022; Vinnervik & Bungum, 2022). Alternativt kan det være at de har en annen forståelse for hva algoritmisk tenkning innebærer. Ulik forståelse for hva som ligger i begrepet kan skyldes at det ikke finnes en tydelig og felles vedtatt definisjon på algoritmisk tenkning (Barr & Stephenson, 2011; Brennan & Resnick, 2012; Ketelhut et al., 2020; Shute et al., 2017).

At lærerne ikke benyttet begrepet algoritmisk tenkning spontant i sine beskrivelser av kompetansemålet betyr ikke uten videre at de ikke har en formening om hva begrepet innebærer. De lærerne som ikke tok i bruk begrepet spontant, ble etter de innledende spørsmålene i intervjuet spurt om hvilken kjennskap de hadde til begrepet algoritmisk tenkning. To av lærerne som deltok i denne studien klarte ikke å gi noen forklaring på hva som menes med begrepet algoritmisk tenkning. Det at disse to lærerne ikke forsøkte seg på en

forklaring tolkes som at de enten er veldig usikre på begrepet, eller at de rett og slett ikke har kjennskap til det. Usikkerhet rundt begrepet algoritmisk tenkning og problemer med å definere det har også blitt identifisert hos andre lærere (Kravik et al., 2022; Nordby et al., 2022). At begrepet oppleves som ukjent kan som nevnt tidligere være et resultat av at det er et informatikkinspirert begrepet (Bocconi et al., 2022). Som påpekt i forrige avsnitt blir heller ikke begrepet algoritmisk tenkning vektlagt i LK20. At to av lærerne ikke klarer å gi noen forklaring på hva begrepet betyr er likevel verdt å merke seg. Anbefalingene i forkant av innføring var at algoritmisk tenkning skulle være en kjernekompetanse innenfor digital teknologi (Sanne et al., 2016). Hvis dette hadde vært tilfellet er det nærliggende å tenke at begrepet har blitt tatt opp i forbindelse med kurs, etter- og videreutdanning, samtaler i kollegiet, eller i det innledende arbeidet med ny læreplan. Når algoritmisk tenkning som begrepet kun nevnes ett sted i LK20, og to av lærerne som deltok i denne studien ikke kan gi noen forklaring på hva dette begrepet innebærer, kan det vel diskuteres hvorvidt anbefalingene til Sanne et al. (2016) har blitt oppnådd. Som nevnt tidligere er ikke dette nødvendigvis et problem. Igjen er det avgjørende hva man egentlig ønsket å oppnå med innføringen av algoritmisk tenkning og programmering i LK20, og hvor vidt det man står igjen med samsvarer med det opprinnelige ønsket.

De andre lærerne som deltok i denne studien, ga beskrivelser av deres forståelse for algoritmisk tenkning. Algoritmisk tenkning ble hovedsakelig forklart på to måter. Enten som algoritme eller som programmering. Når lærerne knyttet algoritmisk tenkning opp mot algoritmer var det i form av å følge en oppskrift eller en mal for å oppnå ønsket resultat. En lignende forståelse av algoritmisk tenkning ble også identifisert hos matematikklærerne i Nordy et al. (2022) sin studie. Nordy et al. (2022) knyttet denne forståelsen opp mot oversettelsen av *Computational thinking* til algoritmisk tenkning, og mente at denne oversettelse gjør at begrepet forveksles med standard algoritmer i matematikk. Som vist i tabell 3 underviser fire av de seks lærere som deltok i denne masteroppgaven i matematikk i tillegg til naturfag. Det er derfor en mulighet for at forståelsen av algoritmisk tenkning som algoritmer i denne studien også handler om at matematikklærere gjennom oversettelsen til algoritmisk tenkning får assosiasjoner til standard algoritmer i matematikk. Samtidig fant Kravik et al. (2022) i sin studie at algoritmer ble brukt av lærere både i matematikk, naturfag, kunst og håndverk og musikk for å beskrive både programmering og algoritmisk tenkning. Dette kan tyde på at også lærere innenfor andre fag enn matematikk forstår algoritmisk tenkning som algoritmer. Det kan være et resultat av at algoritmer ikke entydig assosieres



med standard algoritmer i matematikk, men også med algoritmer i form av å følge en oppskrift som lærerne i denne studien forklarte begrepet.

Ut over algoritmer brukte lærerne i denne studien også programmering som forklaring på algoritmisk tenkning. En slik forståelse er heller ikke ny, men har blitt identifisert i tidligere studier både i Norge og internasjonalt (Cabrera, 2019; Nordby et al., 2022). Hos lærerne som deltok i Nordby et al. (2022) sin studie ble også algoritmisk tenkning i stor grad forstått som programmering. Lærerne i denne studien underviste i matematikk, og matematikkfaget er det faget i LK20 med fleste kompetansemål tilknyttet programmering. Samtidig er det kun i matematikkfaget i LK20 at algoritmisk tenkning som begrepet blir brukt, og da forklart som en problemløsningsprosess (Vinnervik & Bungum, 2022). Hos Cabrera (2019) var forståelsen av algoritmisk tenkning som synonymt med programmering én av fire forståelser som ble identifisert. Denne forståelsen ble funnet hos lærere som ikke hadde gjennomgått kurs eller etterutdanning tilknyttet temaet. Den ble også lagt frem som den mest begrensede forståelsen av begrepet, ettersom lærerne med denne forståelsen ikke oppfattet algoritmisk tenkning som en ferdighet med nytteverdi utenfor programmeringsaktiviteter. Programmering ble sett på som selve målet med algoritmisk tenkning, og ikke som et verktøy for å lære fag. Som diskutert i forrige kapittel ga også lærerne intervjuet i denne masteroppgaven besvarelsen som ga inntrykk av programmering som et mål i seg selv, og programmeringen ble i liten grad knyttet opp mot fag. I tillegg ble algoritmisk tenkning av flere beskrevet nettopp som programmering. Det er derfor flere likhetstrekk mellom forståelsen identifisert hos lærerne i denne studien, og den tilsvarende forståelsen identifisert hos Cabrera (2019). Den interessante forskjellen er at lærerne hos Cabrera (2019) ikke hadde gjennomgått noen form for opplæring eller utdanning tilknyttet temaet. Som vist i tabell 3 er det kun én av lærerne i denne masteroppgaven som har gjennomført videreutdanning tilknyttet programmering. Til tross for at de andre fem lærerne ikke har noen videreutdanning kan det forventes at de har fått etterutdanning eller kurs etter at LK20 ble innført. Det ville også vært naturlig at algoritmisk tenkning og programmering ble diskutert i kollegiet under arbeidet med å sette seg inn i LK20 ettersom dette er helt nye begreper i norske læreplaner. Det at lærerne i denne studien likevel har samme forståelse som lærerne i studien til Cabrera (2019), kan tyde på at den opplæringen som har blitt gitt etter innføringen av LK20 ikke har vært tilstrekkelig.

To av lærerne som deltok i denne masteroppgaven brukte begrepet algoritmisk tenkning spontant i sin forklaring av kompetansemålet i naturfag. Dette kan tyde på at de assosierer begrepet med kompetansemålet, og kanskje at de ser den overordnede intensjonen i LK20.

Samtidig er det påfallende at når disse to lærerne skal forklare hva de legger i begrepet er forklaringene helt ulike. Den ene læreren beskriver algoritmisk tenkning som algoritmer, i likhet med flere av de andre lærerne. Den andre læreren derimot beskrev algoritmisk tenkning som en problemløsningsprosess. Denne beskrivelsen sammenfaller med hvordan algoritmisk tenkning blir beskrevet i læreplanen for matematikk og Utdanningsdirektoratet sin ressursnettside for lærere (Utdanningsdirektoratet, 2019, 2020a). Til tross for at begge disse lærerne nevner algoritmisk tenkning spontant i forbindelse med kompetansemålet, har de altså ikke lik forståelse for hva som ligger i dette begrepet. Dette styrker igjen inntrykket diskutert tidligere av at lærerne kanskje har fått muligheten til å jobbe sammen i kollegiet, eller på tvers av skoler, med det nye temaet i LK20. Hvis lærerne hadde møttes jevnlig og diskutert hva som ligger i de nye kompetansemålene, hva som er intensjonen med disse og hva som er det overordnede temaet er tanken at lærernes forståelse av begrepet ville likere. De to ulike forståelsene dekker også bare hver sin lille del av det analytiske rammeverket, noe som tyder på at begge lærerne har en begrenset og manglende forståelse for begrepet. Hvis lærerne hadde møttes jevnlig for å diskutere det nye faget uten innspill utenifra er det ikke sikkert forståelsen hadde blitt noe bredere, men det kan tenkes at forståelsen ville vært mer sammenfallende.

Ingen av lærernes beskrivelser av algoritmisk tenkning kunne plasseres innenfor teamet *Algoritmisk tenkning som en del av matematikk og naturfag* i det analytiske rammeverket. Dette er interessant i og med at algoritmisk tenkning ble anbefalt å aktualiseres opp mot det enkelte fag for å tydeliggjøre hvordan denne ferdigheten var nyttig innenfor dette faget (Sanne et al., 2016). Det er også interessant fordi det finnes flere praksiser innenfor både matematikk og naturfag hvor algoritmisk tenkning er svært aktuelt, og hvor det derfor ville vært gode muligheter for å trekke autentiske koblinger mellom algoritmisk tenkning og naturfag for øvrig (Weintrop et al., 2016). Eksempler på slike praksiser er datainnsamling, dataprosessering, modellering og simulering (Weintrop et al., 2016). Gjennom å jobbe med disse praksisene kunne programmering blitt brukt som et faglig verktøy for å lære algoritmisk tenkning slik det var tiltenkt (Andersen et al., 2023; Vinnervik & Bungum, 2022). Det finnes derfor flere muligheter for å aktualisere algoritmisk tenkning i naturfag. Samtidig bør ikke programmering være den eneste innfallsvinkelen til algoritmisk tenkning (Voogt et al., 2015). I introduksjonen av algoritmisk tenkning til elevene kan man fokusere på å gi elevene et ordforråd og symboler for å lage instruksjoner, man kan gjøre aktiviteter for å trene abstraksjon, eller jobbe med problemløsning (Lu & Fletcher, 2009). Samtidig mener enkelte

at å skille algoritmisk tenkning fra programmering kan gi elevene misoppfatninger rundt hvordan algoritmisk tenkning faktisk benyttes (Grover & Pea, 2013). Det er likevel ansett av flere som viktig å skille mellom algoritmisk tenkning og programmering, samt å klare og knytte disse konseptene opp mot de faktiske praksisene som benyttes innenfor det aktuelle fagområdet (Lu & Fletcher, 2009; Weintrop et al., 2016). Når lærerne i denne studien ikke knytter algoritmisk tenkning opp mot teamet *Algoritmisk tenkning som en del av matematikk og naturfag* kan det virke som de kanskje ikke er bevisst denne sammenhengen. Dette kan igjen være et resultat av at intensjonen med algoritmisk tenkning og programmering ikke har blitt gjort tydelig nok. Samtidig underbygger dette det tidligere funnet tilknyttet lærernes forståelse av kompetansemålet, hvor lærerne slet med å knytte programmeringen opp mot faget. Dette virker også å være tilfellet for begrepet algoritmisk tenkning. Den pedagogiske utfordringen med å klare og skille programmering fra algoritmisk tenkning blir betraktelig vanskeligere hvis ikke lærerne selv er bevisst dette skillet (Lu & Fletcher, 2009). Summen av at programmering og algoritmisk tenkning ikke blir sett i sammenheng med naturfag, og at algoritmisk tenkning i stor grad oppfattes som programmering, kan derfor før til at det som blir praktisert i klasserommet er ren programmeringsundervisning.

Lærernes forståelse av kompetansemålet i naturfag, samt deres beskrivelse av algoritmisk tenkning, gir uttrykk for begrenset og manglende forståelse for den overordnede intensjonen med programmering og algoritmisk tenkning i LK20. I tillegg vises en manglende og begrenset forståelse for hva algoritmisk tenkning innebærer, og hvordan dette konseptet kan aktualiseres i naturfag. Sett i sammenheng tyder dette på at lærerne ikke har nok kompetanse innenfor området og at de heller ikke har fått nok tid til å sette seg inn i den nye læreplanen, eller de nye konseptene. Av lærerne i denne studien var det kun én av lærerne som hadde videreutdanning. I tillegg blir inntrykket av manglende kompetanse hos lærerne enda tydeligere når fire av lærerne helt spontant og på eget initiativ trekker frem at de har behov for mer kurs og økt kompetanse innenfor området. For å kunne undervise i algoritmisk tenkning og programmering på en god måte må lærerne ha høy fagkompetanse, pedagogisk kompetanse, teknologisk kompetanse og samarbeid mellom lærere for kompetanseutveksling og kompetanseheving (Bocconi et al., 2022; Cabrera, 2019). Lærerne må få en tydelig definisjon på hva algoritmisk tenkning innebærer, konkrete eksempler på hvordan man underviser i temaet og eksempler på alderstilpassede aktiviteter (Barr & Stephenson, 2011). Etersom det ikke finnes noen konsensusdefinisjon på hva algoritmisk tenkning innebærer er det vanskelig for lærere å forstå hva elevene helt konkret skal lære (Voogt et al., 2015).

Lærerne som deltok i denne studien, var i stor grad bevisst egen manglende kompetanse ettersom flertallet av dem uoppfordret trakk frem behovet for kurs og kompetanseheving. Dette behovet var et resultat av at lærerne ønsket å heve kvaliteten på egen undervisning, samt et ønske om faglig påfyll og faglig utvikling. Det at lærerne i stor grad trakk frem programmering både i sin forståelse av kompetansemålet og i sin forståelse av algoritmisk tenkning kan skyldes at det er programmering som står i fokus i LK20 (Vinnervik & Bungum, 2022). Samtidig vekker det en mistanke om at de kursene lærerne eventuelt har fått i forbindelse med innføringen av LK20 også i stor grad har omhandlet programmering. Hvis dette er tilfellet legges tilrettelegges det ytterligere for en forståelse for programmering som et mål i seg selv, og programmering som synonymt med algoritmisk tenkning. En forståelse det i forskningslitteratur advares mot (Lu & Fletcher, 2009; Voogt et al., 2015; Yadav et al., 2017). For å motvirke en slik misoppfatning anbefales det blant annet at elevene får erfaringer med algoritmisk tenkning før de blir introdusert for programmering (Lu & Fletcher, 2009). Spørsmålet da blir hvordan lærere skal kunne tilrettelegge for en slik introduksjon hvis de selv hovedsakelig får opplæring i ren programmering. Hvis elevene kun lærer å programmere får de en fragmentert og begrenset opplæring i forhold til det som i utgangspunktet er intensjonen i LK20 (Vinnervik & Bungum, 2022). At lærerne selv etterspør mer kurs og økt kompetanse innenfor området er ikke unikt for denne masteroppgaven. Også lærerne som deltok i Kravik et.al (2022) og Nordy et al. (2022) trakk frem samme ønske og behov. I sin anbefaling poengterte Sanne et al. (2016) at man i arbeidet med et nytt fag måtte sette av god tid til å utvikle lærernes kompetanse og faglige identitet. Når lærerne mangler den nødvendige kompetansen for å vellykket kunne implementere algoritmisk tenkning og selv etterspør økt kompetanse er det mulig å påstå at heller ikke denne anbefalingen ble fulgt.

## 8. Konklusjon

Siden diSessa (2000) påpekte at rollen datamaskiner hadde i samfunnet ikke ble gjenspeilet i skolen har det skjedd en enorm digital og teknologisk utvikling. Samfunnet har i enda større grad blitt digitalisert, og teknologi tar enda større plass i folks hverdag enn det som var tilfellet på starten av 2000-tallet. Samtidig har også utdanningsbildet endret seg i samme tidsperiode. Stadig flere land har forsøkt å gjenspeile den teknologiske utviklingen gjennom å innføre informatikkinspirerte konsepter i sine læreplaner (Bocconi et al., 2022). Gjennom LK20 gjorde også Norge dette ved å ta i bruk begrepene algoritmisk tenkning, programmering og algoritme i læreplansammenheng. Som påpekt flere ganger i denne oppgaven er algoritmisk tenkning er et begrep uten en tydelig konsensusdefinisjon (Barr & Stephenson, 2011; Brennan & Resnick, 2012; Ketelhut et al., 2020; Shute et al., 2017). Det er også slik at hvordan lærere forstår algoritmisk tenkning mest sannsynlig vil påvirke hvordan de velger å undervise i dette konseptet (Cabrera, 2019). Læreres forståelse av algoritmisk tenkning vil derfor ha betydning for hvordan de underviser i algoritmisk tenkning, noe som igjen vil ha betydning for i hvilken grad dette konseptet har blitt innført i norsk skole. Utgangspunktet for denne kvalitative studien ble derfor å intervju naturfagslærere på mellomtrinnet for å få et bilde av deres forståelse av algoritmisk tenkning. Dette vil da igjen kunne si noe om i hvilken grad dette konseptet har blitt innført i naturfag. Problemstillingen som har blitt forsøkt belyst gjennom hele oppgaven er: *«I hvilken grad har man innført algoritmisk tenkning i naturfag etter Fagfornyelsen i 2020?»*

Etttersom det kun er ett kompetansemål etter 7.trinn i naturfag som er knyttet til algoritmisk tenkning var det interessant å undersøke hvordan lærere forstod dette kompetansemålet. Funnene fra denne studien viser at alle lærerne knytter dette kompetansemålet opp mot programmering og at flere av lærerne ser på det som et rent programmeringsmål fristilt fra andre tema i naturfag. Lignende funn har også blitt gjort i andre studier (Cabrera, 2019; Nordby et al., 2022). På bakgrunn av dette kan det virke som om anbefalingen og intensjonen om programmering som et verktøy for å lære fag, samt et verktøy for å lære algoritmisk tenkning ikke har blitt formidlet tydelig nok. Det var også enkelte lærere som knyttet kompetansemålet til problemløsning og systemtenkning, men disse forståelsene var ikke like fremtredende som programmeringsforståelsen. Ut over dette hadde lærerne ulike forståelser for hva som ligger i kompetansemålet, men felles for alle forståelsene var at de var begrensede og mangelfulle sett opp mot det analytiske rammeverket.

Lærernes forståelse av algoritmisk tenkning ble også undersøkt i denne studien. Her var det to lærere som ikke kunne gi en beskrivelse av hvordan de forstår begrepet, mens de andre lærerne hovedsakelig beskrev det som programmering, eller algoritmer i form av å følge en oppskrift. Lignende funn har også blitt gjort i andre studier (Cabrera, 2019; Kravik et al., 2022; Nordby et al., 2022). Det var ingen av beskrivelsene som passet inn under hovedtemaet i det analytiske rammeverket *Algoritmisk tenkning som en del av matematikk og naturfag*. Dette styrker funnet om at kompetansemålet, og dermed indirekte algoritmisk tenkning, i liten grad blir knyttet opp mot andre temaer i naturfag. Også her ble lærernes forståelse tolket som begrenset og mangelfull sett opp mot det analytiske rammeverket.

Den begrensede og manglende forståelsen av både kompetansemålet i naturfag, samt begrepet algoritmisk tenkning, tyder på at lærerne har for lite kompetanse innenfor dette temaet. Dette er ikke nødvendigvis overraskende ettersom kun én av lærerne har noen form for videreutdanning innenfor et relatert tema, og at disse konseptene er hentet fra informatikk, et fagområdet det er få lærere som har erfaring fra (Bocconi et al., 2022). Samtidig er den manglende kompetansen noe lærerne selv er bevisst, og noe som flere av lærerne på eget initiativ trekker frem under intervjuet. Det at hele fire av lærerne spontant trekker frem behov for økt kompetanse innenfor emnet, samt mer tid til å sette seg temaet, kurs og utdanning, tolkes som at lærerne ikke opplever å ha tilstrekkelig kompetanse til å undervise i algoritmisk tenkning og programmering. Lærernes besvarelser gir inntrykk av at de ikke har nok kompetanse innenfor emnet, et inntrykk lærerne selv bekrefter ved å uttrykke et helt klart ønske om mer kompetanse og faglig utvikling.

*Så i hvilken grad har man innført algoritmisk tenkning i naturfag etter Fagfornyelsen i 2020?*

På bakgrunn av resultatene i denne studien virker det som at man til en viss grad har lyktes med å innføre programmering. Hvis målet var at programmering skulle være et verktøy for å lære fag og for å lære algoritmisk tenkning, tyder resultatene på at man heller ikke er i mål med innføringen av programmering. Når det kommer til algoritmisk tenkning er forståelsen hos lærerne begrenset og mangelfull sett opp mot det analytiske rammeverket, og med dette som utgangspunkt vil konklusjonen være at man i liten grad har innført algoritmisk tenkning i naturfag. På bakgrunn av lærernes uttrykte forståelse, samt tydelige etterspørsel etter økt kompetanse, virker det heller ikke som man har lyktes med å innføre algoritmisk tenkning i form av tilstrekkelig kompetanse hos lærerne. Det er ikke nødvendigvis slik at det man har lyktes med å innføre er feil. Spørsmålet blir bare hvor vidt det man har innført var det man

ønsket, og om det læringsutbyttet elevene sitter igjen med gir dem den kompetansen og de ferdighetene de vil ha behov for i fremtiden.

Utvalget av informanter i denne masteroppgaven var en relativt homogen gruppe ettersom alle jobbet innenfor samme kommune. I videre forskning hadde det vært interessant å undersøke om man hadde fått tilsvarende resultater hvis man tok utgangspunkt i lærere innenfor andre kommuner. I denne oppgaven har fokuset vært på kompetansemålet etter 7.trinn i naturfag, men som nevnt er det også et tilsvarende kompetansemål etter 10.trinn i naturfag. Det kunne derfor også vært interessant å gjennomføre en tilsvarende studie blant ungdomsskolelærere, for å undersøke i hvilken grad man har innført algoritmisk tenkning i naturfag på ungdomsskolen.

## Referanseliste

- Andersen, Frågåt, Boğar, Jensen, & Mifsud. (2023). Representations of Computational Thinking in Policy Documents in an Educational Context: The Case of Denmark, Finland, and Norway. *International Society of the Learning Sciences Conference (ISLS)*. The International Society of the Learning Sciences.
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185.  
<https://doi.org/10.1016/j.chb.2019.106185>
- Barefoot Computing. (2014). *Barefoot Computing delivering digital dreams to the classroom*. Barefootcomputing.Org. <https://www.barefootcomputing.org/about-barefoot>
- 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*. <https://doi.org/10.1145/1929887.1929905>
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. Publications Office of the European Union.  
<https://doi.org/10.2760/126955>
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. American Educational Research Association meeting, Vancouver, BC, Canada.  
<http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>



- Cabrera. (2019). Teachers Preconceptions of Computational Thinking: A Systematic Literature Review. *Journal of Technology and Teacher Education*, 27(3), 305–333.
- diSessa. (2000). *Changing Minds: Computers, Learning and Literacy*. The MIT Press.
- Grover, S., & Pea, R. (2013). Computational Thinking in K—12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <http://www.jstor.org/stable/23360476>
- Johannessen, Tufte, & Christoffersen. (2016). *Introduksjon til samfunnsvitenskaplig metode* (5. utg.). Abstrakt forlag AS.
- Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher Change Following a Professional Development Experience in Integrating Computational Thinking into Elementary Science. *Journal of Science Education and Technology*, 29(1), 174–188. <https://doi.org/10.1007/s10956-019-09798-4>
- Kravik, R., K Berg, T., & Siddiq, F. (2022). Teachers’ understanding of programming and computational thinking in primary education – A critical need for professional development. *Acta Didactica Norden*, 16(4). <https://doi.org/10.5617/adno.9194>
- Kvale & Brinkmann. (2015). *Det kvalitative forskningsintervju* (3. utgave). Gyldendal akademisk.
- Kvarv. (2021). *Vitenskapsteori—Tradisjoner, posisjoner og diskusjoner*. Novus.
- Larson, L. C., & Miller, T. N. (2011). 21st Century Skills: Prepare Students for the Future. *Kappa Delta Pi Record*, 47(3), 121–123. <https://doi.org/10.1080/00228958.2011.10516575>
- Lodi, M., & Martini, S. (2021). Computational Thinking, Between Paper and Wing. *Science & Education*, 30(4), 883–908. <https://doi.org/10.1007/s11191-021-00202-5>
- Lu, J., & Fletcher, G. (2009). *Thinking about Computational Thinking*. 41, 260–264. <https://doi.org/10.1145/1539024.1508959>

- Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Primary Mathematics Teachers' Understanding of Computational Thinking. *Künstliche Intelligenz*.  
<https://doi.org/10.1007/s13218-021-00750-6>
- OsloMet. (u.å.). *Matematikk, naturfag og algoritmisk tenkning (MASCOT)*. Hentet 5. mai 2023, fra <https://www.oslomet.no/forskning/forskningsprosjekter/matematikk-naturfag-algoritmisk-tenkning>
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books Inc.
- Postholm & Jacobsen. (2018). *Forskningsmetode for masterstudenter i lærerutdanning* (1. utg.). Cappelen Damm AS.
- 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* (s. 91).
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158.  
<https://doi.org/10.1016/j.edurev.2017.09.003>
- Utdanningsdirektoratet. (2019, mars 27). *Algoritmisk tenkning*. [udir.no](http://udir.no).  
<https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2020a). *Kjerneelementer—Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. <https://www.udir.no/lk20/mat01-05/om-faget/kjerneelementer?lang=nob>
- Utdanningsdirektoratet. (2020b). *Kjerneelementer—Læreplan i naturfag (NAT01-04)*.  
<https://www.udir.no/lk20/nat01-04/om-faget/kjerneelementer?lang=nob>
- Utdanningsdirektoratet. (2020c). *Kompetansemål etter 7. Trinn—Læreplan i naturfag (NAT01-04)*. <https://www.udir.no/lk20/nat01-04/kompetansemaal-og-vurdering/kv79?lang=nob>

- Utdanningsdirektoratet. (2020d). *Kompetansemål etter 8. Trinn—Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. <https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv16?lang=nob>
- Utdanningsdirektoratet. (2020e). *Kompetansemål etter 9. Trinn—Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. <https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv15?lang=nob>
- Utdanningsdirektoratet. (2020f). *Kompetansemål etter 10. Trinn—Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. <https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv14?lang=nob>
- Utdanningsdirektoratet. (2020g). *Kompetansemål etter 10. Trinn—Læreplan i naturfag (NAT01-04)*. <https://www.udir.no/lk20/nat01-04/kompetansemaal-og-vurdering/kv78>
- Utdanningsdirektoratet. (2023, mars 27). *Kompetansepakker for digital kompetanse i skolen*. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/kompetansepakker-for-digital-kompetanse-i-skolen/>
- Vinnervik, P., & Bungum, B. (2022). Computational thinking as part of compulsory education: How is it represented in Swedish and Norwegian curricula? *Nordic Studies in Science Education*, 18(3), Artikkel 3. <https://doi.org/10.5617/nordina.9296>
- 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>
- 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://www.jstor.org/stable/43867736>

Wing. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.

<https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2017). Computational thinking's influence on research and education for all.

*Italian Journal of Educational Technology*, 25(2), Artikel 2.

<https://doi.org/10.17471/2499-4324/922>

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). *Computational Thinking in Teacher*

*Education* (s. 205–220). [https://doi.org/10.1007/978-3-319-52691-1\\_13](https://doi.org/10.1007/978-3-319-52691-1_13)

# Vedlegg

## Vedlegg 1: Intervjuguide

### Intervjuguide for masteroppgave

*Innføringen av begrepet «algoritmisk tenking» i norske læreplaner og klasserom.*

1. Kan du fortelle litt om lærerbakgrunnen din?
2. Hvilket trinn underviser du på?
3. Hvilke fag underviser du i?
4. Etter 7.klasse er det et kompetansemål i naturfag hvor elevene skal «utforske lage og programmere teknologiske systemer som består av deler som virker sammen». Hva tenker du ligger i dette kompetansemålet?
5. Hvilke ferdigheter får elevene øvd på i arbeidet med dette kompetansemålet?
  - a. Hvilken kjennskap har du til begrepet algoritmisk tenkning?
  - b. Hva tenker du ligger i begrepet algoritmisk tenkning?
6. Hvilke aktiviteter tenker du elevene kan jobbe med for å nå dette kompetansemålet?
7. Hvordan kan en undervisningsøkt eller undervisningsopplegg tilknyttet dette kompetansemålet se ut?
8. Vi er nå ferdige med de spørsmålene jeg hadde. Er det noe du har lyst å legge til eller utdype før vi avslutter?

## Vedlegg 2: Informasjonsskriv til lærere om studien

### **Vil du være med i en masteroppgave om innføringen av «algoritmisk tenkning» og programmering i norske læreplaner og klasserom?**

#### **Hei!**

Ønsker du å delta i en masteroppgave tilknyttet innføringen av begrepet «algoritmisk tenkning»? Deltakelsen innebærer å være med på et kort intervju, og krever ingen forberedelse. Jeg håper du vil være med!

#### **Hva er dette?**

Dette er et spørsmål til deg om å delta i masteroppgaven om innføringen av «algoritmisk tenkning» og programmering i norske læreplaner og klasserom. Formålet med oppgaven er å undersøke intensjonen bak innføringen av «algoritmisk tenkning» og programmering i LK20, og hvordan denne intensjonen stemmer overens med lærernes opplevelse. I dette skrivet vil du få informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

#### **Formål**

Masteroppgaven skrives som en avsluttende oppgave for Grunnskolelærerutdanningen 1 – 7 trinn ved Høgskolen i Innlandet. Masteroppgaven er tilknyttet spesialisering i naturfagdidaktikk.

Ved LK20 ble algoritmisk tenkning innført som et begrep i flere fag, blant annet naturfag. I denne masteroppgaven ønsker jeg å undersøke hva som var intensjonen bak innføringen av algoritmisk tenkning i læreplanen, samt hvordan dette begrepet blir forstått blant naturfagslærere på mellomtrinnet. Å delta i intervjuet vil ikke kreve noen form for forberedelse og det er ikke ønskelig at de som skal intervjues leser seg opp på temaet i forkant. Dette er fordi formålet med intervjuet er å få et bilde av hvordan lærere forstår begrepet basert på den etter- og videreutdanningen de har fått.

Masteroppgaven inngår som en del av MASCOT-prosjektet (Matematikk, naturfag og algoritmisk tenkning) som er et norsk forskningsrådsprosjekt. Prosjektet har internasjonale samarbeidspartnere. Data som samles inn i denne masteroppgaven vil derfor også kunne bli brukt videre i MASCOT-prosjektet.

#### **Hvem er ansvarlig for forskningsprosjektet?**

Høgskolen i Innlandet, avdeling Hamar, er ansvarlig for prosjektet.

#### **Hvorfor får du spørsmål om å delta?**

Som informanter til masteroppgaven ønskes naturfaglærere som underviser, eller har undervist, i naturfag på mellomtrinnet etter innføringen av LK20.

#### **Hva innebærer det for deg å delta?**

Hvis du velger å delta i masteroppgaven innebærer det gjennomføring av et fysisk eller digitalt intervju på maksimalt 30 minutter. Det vil bli tatt lydopptak av intervjuet som lagres sikkert. Spørsmålet i intervjuet vil være tilknyttet din oppfatning av programmering og algoritmisk tenkning i naturfag på mellomtrinnet.

#### **Det er frivillig å delta**

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket

tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

### **Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger**

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrevet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- *Det vil være masterstudent Frida Tuftedal, samt forsker/veileder Thomas Frågåt som vil ha tilgang til dataene.*
- *Lyddopptakene gjøres med diktafonapp fra Nettskjema og lagres sikkert på Nettskjema i etterkant. Navnet og kontaktopplysningene dine vil jeg erstatte med en kode som lagres på egen navneliste adskilt fra øvrige data.*

I selve masteroppgaven vil alle informantene presenteres anonymt.

### **Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?**

Masteroppgaven leveres innen 15.mai 2023. Datamaterialet med personopplysninger vil ikke bli lagret. Anonymiserte opplysninger vil ikke bli slettet, men kan bli gjenbrukt i MASCOT-prosjektet.

### **Hva gir oss rett til å behandle personopplysninger om deg?**

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Høgskolen i Innlandet har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

### **Dine rettigheter**

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

- innsyn i hvilke opplysninger vi behandler om deg, og å få utlevert en kopi av opplysningene
- å få rettet opplysninger om deg som er feil eller misvisende
- å få slettet personopplysninger om deg
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger

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

- Ansvarlig student, Frida Tuftedal [[fridatuftedal@hotmail.no](mailto:fridatuftedal@hotmail.no), 46804908]
- Høgskolelektor Thomas Frågåt [[thomas.fragat@inn.no](mailto:thomas.fragat@inn.no), mobil: 99593053]
- Vårt personvernombud: Usman Asghar [[usman.asghar@inn.no](mailto:usman.asghar@inn.no), 61287483]

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

- Personverntjenester på epost ([personverntjenester@sikt.no](mailto:personverntjenester@sikt.no)) eller på telefon: 53 21 15 00.

Med vennlig hilsen

Thomas Frågåt  
(Forsker/veileder)

Frida Tuftedal  
(Masterstudent)

## Vedlegg 3: Samtykkeerklæring

### Samtykkeerklæring

Jeg har mottatt og forstått informasjon om masteroppgaven om innføringen av «algoritmisk tenking» og programmering i norske læreplaner og klasserom, og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta på intervju tilknyttet masteroppgaven.
- at data fra dette intervjuet kan brukes i MASCOT-prosjektet.

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

-----  
(Signert av prosjektdeltaker, dato)



## Vedlegg 4: Godkjenning fra NSD

### Vurdering av behandling av personopplysninger

#### Referansenummer

718402

#### Vurderingstype

Standard

#### Dato

21.09.2022

#### Prosjekttittel

Masteroppgave om innføringen av «algoritmisk tenking» og programmering i norske læreplaner og klasserom

#### Behandlingsansvarlig institusjon

Høgskolen i Innlandet / Fakultet for lærerutdanning og pedagogikk / Institutt for matematikk, naturfag og kroppsøving

#### Prosjektansvarlig

Thomas Frågåt

#### Student

Frida Tuftedal

#### Prosjektperiode

18.08.2022 - 15.05.2023

#### Kategorier personopplysninger

- Almennelige

#### Lovlig grunnlag

- Samtykke (Personvernforordningen art. 6 nr. 1 bokstav a)

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 15.05.2023.

## **Kommentar**

### OM VURDERINGEN

Personverntjenester har en avtale med institusjonen du forsker eller studerer ved. Denne avtalen innebærer at vi skal gi deg råd slik at behandlingen av personopplysninger i prosjektet ditt er lovlig etter personvernregelverket. Personverntjenester har nå vurdert den planlagte behandlingen av personopplysninger. Vår vurdering er at behandlingen er lovlig, hvis den gjennomføres slik den er beskrevet i meldeskjemaet med dialog og vedlegg.

### VIKTIG INFORMASJON TIL DEG

Du må lagre, sende og sikre dataene i tråd med retningslinjene til din institusjon. Dette betyr at du må bruke leverandører for spørreskjema, skylagring, videosamtale o.l. som institusjonen din har avtale med. Vi gir generelle råd rundt dette, men det er institusjonens egne retningslinjer for informasjonssikkerhet som gjelder.

### TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til den datoen som er oppgitt i meldeskjemaet.

### LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

## PERSONVERNPRINSIPPER

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

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

## DE REGISTRERTES RETTIGHETER

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

Personverntjenester 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

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

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

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

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

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

Du må vente på svar fra oss før endringen gjennomføres.

#### OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!