

MASTEROPPGAVE
MGLU18
Mai 2023

Læreres mestringstro i programmeringsundervisning i naturfag

Teachers' self-efficacy towards teaching programming in science

30 stp. oppgave

Antall ord: 18 667

Mai Willoch

OSLOMET

OsloMet – storbyuniversitetet

Fakultet for lærerutdanning og internasjonale studier
Institutt for grunnskole- og faglærerutdanning

Forord

Denne masteroppgaven markerer slutten på min fem år lange lærerutdanning ved OsloMet. Det har vært fem lærerike år jeg er svært takknemlig for, spesielt årene med fordypning i naturfag der jeg har fått mulighet til å studere med en helt herlig gjeng.

Da jeg startet på studiet, hadde jeg aldri trodd at jeg kom til å skrive en masteroppgave rettet mot programmering, fordi det ikke var et tema jeg hadde spesielt mye kunnskaper om fra før. Noen morsomme prosjektarbeider med elever fikk meg til å innse at det var et tema jeg ønsket å lære mer om. I praksis møtte jeg også på lærere som ikke følte seg klare for å undervise i programmering, og nysgjerrigheten min ble dermed rettet mot lærerne som skal integrere programmering i fagene de underviser.

Arbeidet med masteroppgaven har vært krevende, og jeg hadde aldri kommet i mål hvis det ikke hadde vært for masse hjelp fra en utrolig flink og tålmodig veileder. Jeg må derfor sende en stor takk til veilederen min Per Øyvind Sollid som har vært tilgjengelig gjennom hele prosessen, kommet med gode tips og råd og svart på både store og små spørsmål.

Jeg må også takke Kirsti som har vært min nærmeste venn gjennom studiet, og alltid vært tilgjengelig om jeg har trengt å diskutere deler av oppgaven med noen, og David som har støttet meg gjennom hele prosjektet med oppmuntrende ord når jeg har trengt det. En stor takk går også til alle venner og familiemedlemmer som har hjulpet meg med alt fra å tenke høyt om formulering av forskningsspørsmål til korrekturlesing av oppgaven. Dere har hjulpet meg å se lyset i enden av mastertunellen.

Oslo, mai 2023

Mai Willoch

Sammendrag

I 2020 kom det nye nasjonale læreplaner i Norge. En endring fra de gamle læreplanene er at programmering blir nevnt eksplisitt i flere fag, deriblant naturfag. Det betyr at naturfagslærere nå må ha kompetanse for å undervise i programmering. I tillegg til kompetanse vil det være interessant å undersøke om lærere selv føler at de kommer til å mestre og undervise i programmering. Her blir begrepet mestringstro relevant, som handler om troen på at man klarer å mestre en oppgave eller situasjon.

Hensikten med denne masterstudien er å undersøke læreres mestringstro i programmeringsundervisning i naturfag. For å oppnå hensikten ble det utarbeidet to forskningsspørsmål som skulle svares på:

- Hvordan beskriver respondentene sine opplevelser av mestring i programmeringsundervisning i naturfag?
- Hvilke strategier oppgir respondentene at de bruker på tross av lav forestilt mestring?

For å samle data fra så mange lærere som mulig, ble det utviklet et kvalitativt spørreskjema som ble delt i ulike facebook-grupper for lærere, samt sendt til en gruppe lærere som har deltatt på programmeringskurs og godtatt å motta informasjon om en masterstudie. For å analysere datamaterialet ble det gjort en kvalitativ innholdsanalyse.

Resultatene indikerer at det er varierende grad av mestringstro blant respondentene. Noe som går igjen, er at flere respondenter føler at de mangler kunnskaper om og kompetanse i programmering, og at dette blant annet påvirker deres evne til å hjelpe elever i undervisningen. På tross av dette indikerer resultatene også at en del av disse respondentene har strategier de bruker for å mestre situasjoner der de føler at de mangler kunnskaper. Strategiene respondentene bruker er blant annet å bruke elever som ressurser til å hjelpe elever de selv ikke klarer å hjelpe og å bruke ressurser de finner på internett.

Studien konkluderer med at den varierende mestringstroen blant respondentene kan føre til varierende kvalitet på undervisningen til elevene, og at dette kan indikere at lærere som skal undervise i programmering trenger omfattende kompetanseheving og oppfølging i programmering. På tross av dette virker det som at flere av respondentene er overordnet positive til at de kan klare å undervise i programmering i naturfag. At respondentene ser på

elever som ressurser i undervisningen blir også satt i et positivt lys, da det kan tenkes at dette kan føre til at utforskning, kreativitet og nyskaping får større plass i opplæringen.

Abstract

In 2020, new national curricula were introduced in Norway. A change from the old curricula is that programming is mentioned explicitly in several subjects, including natural sciences. This means that science teachers need competence to teach programming. In addition to teacher competence, it would be interesting to investigate whether the teachers themselves feel that they will master teaching programming. This is where the concept of self-efficacy becomes relevant, which is about one's belief that one can master a task or situation.

The purpose of this master's thesis is to investigate teachers' self-efficacy towards teaching programming in science. In order to achieve this purpose, the following two research questions were to be answered:

- How do the respondents describe their experience of mastery when teaching programming in science?
- What strategies do the respondents state that they use despite low perceived mastery?

To collect data from as many teachers as possible, a qualitative questionnaire was developed and shared in various Facebook groups for teachers, as well as sent to a group of teachers who had attended a programming course and given their consent to receive information about a master's study. A qualitative content analysis was used to analyze the data.

The results indicate that there is a varying degree of self-efficacy among the respondents. Several respondents feel that they lack knowledge about and competence in programming, and this, amongst other things, affect their ability to help the pupils they are teaching. Despite this, the results also indicate that some of these respondents have strategies they use to master situations where they feel they lack knowledge. The strategies the respondents use are, amongst others, using pupils as resources to help pupils they cannot help themselves, and to use online resources.

The conclusion of the study is that the varying self-efficacy amongst the respondents may lead to a varying quality on the teaching the pupils receive, and that this may indicate that teachers who will teach programming need extensive training in programming. Despite this, it seems like most of the respondents are generally positive in their capability to teach programming in science. That the respondents view students as teaching-resources is also put

in a positive light, as it is thought that this may lead to more room for inquiry, creativity, and innovation in their learning.

Innholdsfortegnelse

INTRODUKSJON	1
BAKGRUNN – FRA DIGITALE FERDIGHETER TIL PROGRAMMERING I LÆREPLANEN	1
HENSIKT OG FORSKNINGSSPØRSMÅL	4
TEORI	5
HVA ER MESTRINGSTRO?.....	5
ULIKE KILDER TIL MESTRINGSTRO.....	6
TIDLIGERE FORSKNING	8
HVORDAN KAN VI MÅLE MESTRINGSTRO?	12
METODE	15
BEGRUNNELSE FOR STUDIENS METODOLOGI	15
UTVIKLINGEN AV SPØRRESKJEMA	15
UTVALG	19
ANALYSE	21
RELIABILITET OG VALIDITET	25
FORSKNINGSETISKE REFLEKSJONER	27
RESULTATER	28
HVORDAN BESKRIVER RESPONDENTENE SINE OPPLEVELSER AV MESTRING I PROGRAMMERINGSUNDERVISNING I NATURFAG?.....	29
<i>Mestringserfaringer</i>	29
<i>Emosjonell tilstand</i>	31
<i>Forestilt didaktisk mestring</i>	31
<i>Forestilt faglig mestring</i>	32
<i>Rammefaktorer</i>	34
<i>Selvregulering</i>	34
<i>Tilgjengelighet til hjelp</i>	35
<i>Kognitiv selvmodellering</i>	36
HVLKE STRATEGIER OPPGIR RESPONDENTENE AT DE BRUKER PÅ TROSS AV LAV FORESTILT MESTRING?.....	36
DISKUSJON	38
HVORDAN BESKRIVER LÆRERNE SINE OPPLEVELSER AV MESTRING I PROGRAMMERINGSUNDERVISNING I NATURFAG?	38
HVA GÅR IGJEN I SVARENE TIL RESPONDENTENE SOM TYDER PÅ LAV MESTRINGSTRO?.....	39
HVLKE KOMPETANSER UTTRYKKER RESPONDENTENE SELV AT DE TRENGER FOR Å UNDERVISE I PROGRAMMERING?	41
HVLKE STRATEGIER HAR RESPONDENTENE PÅ TROSS AV LITE FORESTILT MESTRING?	42
KILDER TIL MESTRINGSTRO SOM BLE FUNNET I ANALYSEN	45
KILDER TIL MESTRINGSTRO SOM IKKE BLE FUNNET I ANALYSEN	47
BEGRENSNINGER	47
OPPSUMMERING OG KONKLUSJON	49
FORSLAG TIL VIDERE FORSKNING	51
LITTERATURLISTE	52
VEDLEGG	55
VEDLEGG 1 – SPØRREUNDERSØKELSE TIL MASTERPROSJEKTET.....	55
VEDLEGG 2 – VURDERING FRA SIKT.....	57

Introduksjon

I dette kapitlet introduserer jeg bakgrunnen for studien, samt hensikten og forskningsspørsmålene. Først beskriver jeg hvordan flere rapporter og utredninger anbefalte at digital kompetanse og programmering burde få en større plass i opplæringen i den norske skolen, og hvordan programmering til slutt ble integrert i de nye læreplanene i LK20. Videre kommer jeg til å presentere hensikten med studien og forskningsspørsmålene, før jeg går gjennom studiens avgrensninger.

Bakgrunn – fra digitale ferdigheter til programmering i læreplanen

Et av formålene med opplæringen i skolen er at den skal åpne dører mot verden og fremtiden for elevene, og de skal utvikle kunnskap for å kunne mestre livet og delta i arbeid og i samfunnet (Opplæringslova, 1998, § 1-1). Den teknologiske utviklingen i samfunnet er rask, og det er lite man møter i samfunnet og hverdagen som ikke er helt eller delvis avhengig av en eller annen form for teknologi. Derfor er teknologisk kompetanse viktig for å kunne fungere og bidra i dagens og fremtidens samfunn. I læreplanverket for Kunnskapsløftet 2006, også kalt LK06, ble det introdusert fem grunnleggende ferdigheter som ble ansett som viktig for elever gjennom hele opplæringsløpet, og en av disse ferdighetene er digitale ferdigheter. I 2013 kom en utredning fra Digiutvalget (NOU 2013: 2) der det kom frem at de mente at digitale ferdigheter ikke var tilstrekkelig inkludert i skolen. Utvalget mente at det var for lite fokus på *forståelse*, både av teknologisamfunnet og av hvordan IKT fungerer. Digiutvalget anbefalte at elever skulle få mulighet til å lære om programmering allerede fra grunnskolen av, og ikke først på videregående som var vanlig da. Begrunnelsen for dette var at det ville føre til at elevene fikk mulighet til å bli skapere av digitale tjenester, i stedet for å bare bli oppdratt til videreformidlere og konsumenter av teknologi.

Ludvigsen-utvalget (NOU 2015: 8) ble oppnevnt i 2013 for å vurdere blant annet hvilke kompetanser som ville være viktige for elever fremover, og hvordan fagene i skolen måtte endres for at elevene skulle utvikle disse kompetansene. I utredningen står det blant annet at det vil være behov for at elevene utvikler kompetanse innenfor teknologi, og utvalget ser digital kompetanse som en sentral del av fagområdene i skolen. De skriver at det vil variere hvilke teknologiske verktøy som er relevant å bruke i de ulike fagene, men at digital kompetanse også må ses som en fagovergripende kompetanse som er relevant på tvers av fagområder. Utvalget mente teknologiutviklingen ville føre til endringer i alle fag, men at

opplæringen i noen deler av digital kompetanse burde legges til et bestemt eller noen flere fag for å unngå at ansvaret for opplæringen blir uklart.

I 2017 kom en digitaliseringsstrategi for grunnopplæringen for 2017-2021 (Kunnskapsdepartementet, 2017). Målet med strategien var at elever skulle få digitale ferdigheter som skulle hjelpe dem med å lykkes i utdanning, arbeid og samfunnsdeltakelse, og at IKT skulle brukes på en måte i opplæringen som økte elevenes læringsutbytte. Det var blant annet ønsket at elever skulle lære mer programmering enn det som da var tilgjengelig i programfag på videregående og valgfag på ungdomstrinnet. Strategien nevnte også at det i fagfornyelsen skulle vurderes hvordan teknologi, programmering og algoritmisk tenkemåte kunne bli integrert i allerede eksisterende fag, særlig matematikk og naturfag.

I de fleste andre land der de har lagt til teknologi, algoritmisk tenkning og/eller programmering i de nasjonale læreplanene, har det blitt lagt til som et tema i allerede eksisterende fag, eller som et overordnet tema som skal gjelde for hele utdannelsen (Heintz et al., 2016). Dette har også vært situasjonen i Norge, der teknologi har vært et tema som blant annet har ligget under naturfag. Ifølge Heintz et al. (2016) kan det være både fordeler og ulemper med å integrere et slikt tema i allerede eksisterende fag. På en side kan det hende at lærere får utviklet sin digitale kompetanse når de blir oppfordret til å inkludere teknologi i fagene de underviser. På en annen side kan kvaliteten på undervisningen variere basert på lærerens interesse i og tid til å lære seg om teknologi på egenhånd. I en rapport om teknologi og programmering i den norske skolen av Sanne et al. (2016) ble det anbefalt at det skulle opprettes et nytt fag i grunnskolen som skulle omfatte teknologi og programmering.

Fagfornyelsen førte til nye læreplaner, LK20, som ble tatt i bruk fra 2020. Digitale ferdigheter er fremdeles en av de fem grunnleggende ferdighetene (Utdanningsdirektoratet, 2017, 2.3). Noen av endringene i den nye læreplanen, er at algoritmisk tenkning og programmering har fått en større plass i opplæringen. Algoritmisk tenkning er først og fremst en problemløsningsmetode som går ut på å bryte et komplekst problem ned i mindre løsbare problemer (Utdanningsdirektoratet, 2019). I denne forklaringen til Udir er ikke programmering en nødvendighet for å kunne jobbe med algoritmisk tenkning, men de skriver at en del av algoritmisk tenkning også kan være å ha en forståelse av hva slags problemer som kan løses med teknologi og hva som bør overlates til mennesker. I læreplaner i en del andre land legges det opp til at algoritmisk tenkning som en metode for problemløsning er tenkt å

kunne utvikles gjennom programmering (Sanne et al., 2016). Det er ikke uvanlig å bruke programmering som en metode for å utvikle ferdigheter innenfor algoritmisk tenkning siden begge deler krever mange av de samme ferdighetene, for eksempel å analysere og bryte ned et problem, abstraksjon, generalisering og feilsøking (Shute et al., 2017). Med andre ord er ikke algoritmisk tenkning og programmering det samme, men å kunne tenke algoritmisk er en forutsetning for å kunne programmere.

I de nye læreplanene for grunnskolen blir programmering eksplisitt nevnt i kompetansemålene til naturfag, matematikk, musikk og kunst og håndverk. Det betyr at lærerne som underviser disse fagene må ha kompetanse til å undervise i de nye temaene. I tillegg til kompetanse, vil det være viktig å undersøke om lærerne selv opplever at de mestrer å skulle undervise i programmering. Her blir begrepet mestringstro (self-efficacy) relevant. Mestringstro handler om troen på at man klarer å mestre en oppgave eller situasjon (Bandura, 1997). Studier fra andre land har funnet at lærere har lav mestringstro når det kommer å lære å programmere, bruke egne programmeringsferdigheter til problemløsning og å undervise i programmering (Hartell et al., 2019; Mannila et al., 2018). Denne masterstudien undersøker derfor læreres mestringstro i å skulle undervise i programmering i naturfag for elever i barneskolen.

Hensikt og forskningsspørsmål

Hensikten med denne masterstudien er å undersøke læreres mestringstro i programmeringsundervisning i naturfag. For å kunne oppnå hensikten med studien, var det nødvendig å finne ut av hvordan naturfaglærere beskriver sine opplevelser av mestring når de skal undervise i programmering i naturfaget. Det første forskningsspørsmålet jeg utarbeidet for studien var derfor:

- Hvordan beskriver respondentene sine opplevelser av mestring i programmeringsundervisning i naturfag?

Videre var planen i utgangspunktet å undersøke hva respondentene selv trakk frem som viktig for at de skal oppleve å få økt mestringstro. Denne planen ble endret under analysen av datamaterialet. Under analysen kom det frem at det var sider ved programmeringsundervisning der respondentene hadde lavere mestringstro, men flere av respondentene trakk selv frem strategier de brukte for å finne en løsning på situasjoner de i utgangspunktet ikke mestret. Derfor ble det andre forskningsspørsmålet til studien:

- Hvilke strategier oppgir respondentene at de bruker på tross av lav forestilt mestring?

Jeg har valgt å avgrense studien til å kun undersøke programmering i naturfag, og ikke algoritmisk tenkning. Den første begrunnelsen for dette var at studien skulle være overkommelig for en masteroppgave skrevet på ett semester. Den andre grunnen var at programmering nevnes eksplisitt i et kompetansemål og et kjerneelement i naturfag, og det er derfor nødvendig at naturfaglærere lærer seg å programmere. Selv om programmering blir nevnt i flere fag enn naturfag, er studien avgrenset til å handle om lærere som underviser i naturfag. Begrunnelsen for dette er at det er naturfag som er min fagfordypning.

Teori

I dette kapittelet presenterer jeg den teoretiske og empiriske bakgrunnen for studien. Først presenterer jeg begrepet mestringstro og gir en liten innføring i hva det er. Så går jeg gjennom hvilke kilder til mestringstro som har blitt definert av ulike forskere. Videre presenterer jeg tidligere studier om mestringstro. Hovedfokuset her vil ligge på studier som har undersøkt læreres mestringstro i undervisningssituasjoner, med ekstra vekt på studier som undersøker lærere som underviser i programmering. Til slutt presenterer jeg ulike instrumenter som har blitt tatt i bruk for å måle mestringstro.

Hva er mestringstro?

Mestringstro (engelsk: self-efficacy) er et begrep innen psykologi som refererer til «(...) troen på egen evne til å organisere og gjennomføre de handlinger som kreves for å oppnå gitte mål.» (Bandura, 1997, s. 3, egen oversettelse). Med andre ord er det troen på at en klarer å mestre en oppgave eller situasjon. Mestringstro er et viktig grunnlag for handling, fordi det blant annet kan påvirke hvilke handlinger en velger å gjøre, hvor lenge en holder ut i møte med hindringer, hvor mye stress en opplever i krevende situasjoner og nivået på prestasjonene en oppnår (Bandura, 1997, s. 3). Med mindre mennesker tror de kan oppnå ønsket effekt av en handling, har de lite insentiv til å utføre handlingen.

Mestringstro er kontekstspesifikt (Goddard et al., 2000; Hartell, 2017). En lærer kan for eksempel ha høy mestringstro når de underviser i et fag, men lavere mestringstro når de underviser i et annet fag (Goddard et al., 2000). Mestringstroen til lærere kan også variere mellom forskjellige temaer innad i faget. Det er ikke bare faget i seg selv som påvirker lærerens mestringstro for å undervise i faget. Lærerens mestringstro kan også variere avhengig av blant annet hvilke elever de underviser (Goddard et al., 2000). Mestringstro handler om ens egen oppfatning av egen kompetanse, heller enn det faktiske nivået på egen kompetanse (Tschannen-Moran et al., 1998). Så om en lærer har høy mestringstro når det kommer til å undervise i et fag, så sier det ingenting om det faktiske nivået på undervisningen. Allikevel kan mestringstro være med på å påvirke kvaliteten på undervisningen. Mennesker har tendenser til å ville unngå situasjoner som de vurderer at de ikke kommer til å mestre, og heller velge aktiviteter de vurderer at de mestrer (Bandura, 1977). Med andre ord kan en lærers mestringstro påvirke i hvilken grad ulike temaer innad i et fag blir vektlagt i undervisningen.

Det er flere faktorer som påvirker en person sine handlinger. To begreper det er viktig å skille mellom er en persons mestringstro og utfallsforventning (outcome expectancy). En utfallsforventning er « (...) en persons antakelser om at en gitt atferd vil føre til visse utfall» (Bandura, 1977, s. 193, egen oversettelse). I en klasseromssammenheng ville dette for eksempel være i hvilken grad læreren tror at lærerens undervisning påvirker elevenes læring. Mestringstro er i hvilken grad læreren tror at hen selv kommer til å klare å gjennomføre undervisningen som lærer elevene det de skal lære. Utfallsforventning og mestringstro er forskjellige ting, men kan påvirke hverandre. Hvis man tror at en viss handling kan føre til et bestemt utfall, men ikke tror at man selv klarer å gjennomføre denne handlingen, så er det mindre sjans for at du vil gjennomføre handlingene som du tror fører til det ønskede resultatet (Bandura, 1977). Med andre ord, om en naturfagslærer mener elevene burde lære å programmere, men mangler mestringstro i å undervise i programmering, så er det mulig hen velger å bruke mindre tid på å undervise i programmering. Læreren kan også ha høy mestringstro, men om utfallsforventningene er lave og hen ikke tror elevene kommer til å lære å programmere, så kan dette også føre til mindre programmeringsundervisning for elevene.

Når lærere vurderer egen mestringstro, veier de vurderingen av egen undervisningskompetanse opp mot hva de tenker kreves av den gitte situasjonen de skal undervise i (Tschannen-Moran et al., 1998). De analyserer situasjonen for å vurdere vanskelighetsgraden på oppgaven og hva som skal til for å lykkes. I denne vurderingen inkluderer de blant annet elevenes evner og motivasjon, strategier for undervisningen, tilgjengelighet til undervisningsmateriale og tilgang til teknologi (Tschannen-Moran et al., 1998).

Ulike kilder til mestringstro

Det har blitt definert flere kilder til mestringstro. Bandura (1997) definerte noen, og senere har også andre hatt teorier om og funnet flere kilder til mestringstro. I denne delen kommer jeg til å presentere noen forskjellige kilder til mestringstro som har blitt definert.

Bandura (1997) definerte fire kilder til mestringstro; mestringserfaringer (enactive mastery experiences), vikarierende erfaringer (vicarious experiences), verbal overtalelse (verbal persuasion) og fysiologiske og emosjonelle tilstander (physiological and affective states).

Mestringserfaringer er erfaringene en person har med å mestre en situasjon. Om erfaringene er positive, vil mestringstroen til personen øke. Vikarierende erfaringer handler om at man observerer at en annen person mestrer en situasjon. Å se at en annen person mestrer en situasjon kan øke troen på at en selv også vil klare å mestre en liknende situasjon. Verbal overtalelse er når en person klarer å overbevise en annen person at de vil klare å mestre en situasjon. Fysiologiske og emosjonelle tilstander er en persons vurdering av egen fysiologiske og emosjonelle tilstand, og hvordan de vurderer at dette kan påvirke mestringen av en situasjon. Ifølge Bandura (1997) er det mestringserfaringer som er den mest innflytelsesrike kilden til mestringstro.

Det kan være flere måter å oppnå høyere mestringstro gjennom Bandura (1997) sine definerte kilder. Vikarierende erfaringer kan for eksempel være faktisk modellering, symbolsk modellering, selv-modellering og kognitiv selv-modellering (Bandura, 1997). Faktisk modellering er at man observerer en annen person mestre en situasjon. Under symbolsk modellering observerer man også en annen person mestre en situasjon, men da er ikke personen man observerer fysisk til stede. Man observerer personen gjennom et annet media, for eksempel TV. Selvmodellering er når man får se et opptak av seg selv i en situasjon man mestrer. Kognitiv selvmodellering er når en person visualiserer at de mestrer en situasjon.

I tillegg til de fire kildene til mestringstro definert av Bandura (1997) har også flere kilder til mestringstro blitt funnet. Butz og Usher (2015) undersøkte elevers selvsikkerhet innenfor lesing og matematikk. De fant at i tillegg til de fire kildene til mestringstro definert av Bandura (1997) så var blant annet selvregulering, tilgjengelighet til hjelp og fremtidsperspektiv viktig for elevenes mestringstro. Selvregulering handler om hvor godt man klarer å organisere arbeidet sitt og tiden sin. Tilgjengelighet til hjelp handler om at man føler at det er hjelp tilgjengelig om man trenger det. For elevene i studien til Butz og Usher (2015) kunne det handle om hjelp fra lærere, andre voksne eller medelever. Fremtidsperspektiv handlet om at mestringstroen til elevene ble høyere når de så nytten av arbeidet i et fremtidsperspektiv, for eksempel at de innså at de måtte være flinke i et fag for å få drømmejobben sin.

Noen flere kilder til mestringstro har også blitt funnet i spesifikke grupper. Palmer (2006) undersøkte mestringstro blant lærerstudenter, og argumenterte for at forestilt faglig mestring (cognitive content mastery) og forestilt didaktisk mestring (cognitive pedagogical mastery) er

viktig for mestringstroen deres. Forestilt faglig mestring handler om følelsen av å ha forstått det faglige innholdet i faget man skal undervise. Palmer (2006) argumenterer for at en følelse av å ha forstått faginnholdet kan øke mestringstroen for å undervise i faget. Forestilt didaktisk mestring handler om følelsen av kontroll over pedagogisk kunnskap, altså hvordan man skal lære bort et fag. Palmer (2006) skriver at det ikke er uvanlig at lærerstudenter ikke vet hvordan de skal undervise et fag (i dette tilfellet naturfag) på en måte som fenger elevene. Derfor vil følelsen av å ha forstått hvordan man motiverer elever og effektivt underviser i faget være med på å øke mestringstroen. I Palmer (2006) sin studie var det forestilt didaktisk mestring som var hovedkilden til mestringstro blant lærerstudentene.

Tidligere forskning

Siden mestringstro handler om ens egen oppfatning av egen kompetanse (Tschannen-Moran et al., 1998), kan det være interessant å se hvor viktig lærere selv mener det er å føle at de har kompetanse i faget de skal undervise i. I Nilsson (2008) sin doktoravhandling undersøkte hun lærerstudenters overgang fra å være studenter til å være lærere. En av studiene hun gjorde var å undersøke lærerstudenters holdninger til fysikk, viktigheten av kunnskaper i emnet man skal undervise og studentenes selvtillit i naturfagundervisning. Hun fant at lærerstudentene merket allerede under lærerutdannelsen at de trenger kunnskap om faget de underviser i (i dette tilfellet fysikk) for å føle seg selvsikre når de skal undervise i faget (Nilsson, 2008). Også blant lærere har det blitt funnet at en mer solid kunnskap om et fag gjør at de har større tro på egne evner til å undervise i faget (Finger & Houguet, 2009; Vinnervik, 2022). Så både før de er i jobb, og når de har erfaring med lærerjobben, er det lærere som kjenner på viktigheten av fagkunnskaper i faget de skal undervise.

Når læreplaner endres og nye temaer legges til, er det viktig å undersøke om lærere har kompetansen som trengs og føler seg klare for å undervise i de nye temaene, eller om de har behov for videre kompetanseheving. Kravik et al. (2022) har gjort en studie om læreres forståelse av programmering og algoritmisk tenkning. De konkluderer blant annet med at det trengs en systematisk og gjennomgående kompetanseheving av lærere dersom forventningene i Fagfornyelsen om å utvikle norske elevers kompetanse i algoritmisk tenkning skal oppfylles. Alle lærerne i studien deres var enige i at de trengte mer trening (Kravik et al., 2022). I en svensk studie om læreres holdninger til å undervise i programmering kom det frem at en del av lærerne som deltok i studien hadde manglende selvtillit til å undervise i programmering

(Hartell et al., 2019). De observerte at lærerne var avhengige av ferdigladd materiale, og fikk bekreftet i intervjuer med lærerne at det var mangel på erfaring med programmeringsundervisning som førte til dette. Dette viste seg å være noe som gikk igjen hos de fleste lærerne, med unntak av de lærerne som hadde tidligere erfaring med programmering. Vinnervik (2022) gjorde en studie der han undersøkte læreres utfordringer ved å implementere programmering i teknologi og matematikk. Han fant også at majoriteten av lærerne som ble berørt da programmering ble integrert i matematikk- og teknologiundervisningen i Sverige hadde lite eller ingen tidligere erfaring med programmering. Også en finsk studie om læreres mestringstro i digital kompetanse har vist at lærere føler seg usikre når det kommer til å undervise i programmering (Mannila et al., 2018). Studien sier ingenting om hva grunnen er, men den fant at både lærere med høy og lav mestringstro innen digital kompetanse hadde til felles at programmering var ett av temaene innenfor digital kompetanse der de hadde lavest mestringstro.

Siden teknologi er et fagområde som endrer seg raskt, er det nødvendig med kontinuerlig faglig utvikling (Hartell, 2017), og det er viktig at lærere får etterutdanning i teknologi og pedagogikk knyttet til informatikk (Finger & Houguet, 2009; Kert, 2019). Når programmering innføres i en ny læreplan, er det anbefalt at lærere får en grundig opplæring for å lære de nye kompetansene de trenger for å integrere programmering i undervisningen sin (Bocconi et al., 2018). Hartell et al. (2015) gjorde en studie der de undersøkte teknologilæreres mestringstro i å vurdere elever i teknologifaget. Resultatene i studien viste stor forskjell i lærernes oppfatning av egne evner til å undervise og vurdere elever i teknologi. Denne forskjellen var tydelig mellom lærerne som oppga at de har deltatt på lærerkurs i teknologi og lærerne som oppga at de ikke hadde. De fant med andre ord at teknologilærere som har fått både fagspesifikk og pedagogisk opplæring også får høyere mestringstro.

Selv om kompetanseheving av lærere blir sett på som viktig, har lærere i andre land opplevd at programmering blir implementert i læreplanen, og at det forventes at de skal integrere programmering som en del av andre fag selv om de har lite eller ingen tidligere erfaring med programmering (Vinnervik, 2022). Stigberg og Stigberg (2020) har i sin studie blant annet undersøkt hvordan lærere reflekterer over utfordringer ved å undervise i programmering i matematikk. De fant at selv om det har blitt innført egne programmeringskurs for lærere, så er det fremdeles lærere som ikke har fått den nødvendige opplæringen de burde ha fått. Thorsnes et al. (2020) undersøkte læreres mestringstro i programmeringsundervisning etter at de hadde

deltatt på et kompetansehevende kurs i programmering. Et av funnene deres var at lærerne som deltok i studien fikk høyere mestringstro i å undervise i programmering etter kurset, men de påpekte allikevel at det ikke holder med et introduksjonskurs i programmering for lærere. Kompetanseheving av lærere mener de at burde bli sett på som en kontinuerlig prosess for at lærere skal klare å holde seg oppdatert og i stand til å håndtere utfordringene de møter hver dag. Mason og Rich (2019) gjennomførte en litteraturstudie der de undersøkte nåværende forskning på kurs for lærere og lærerstudenter som skulle øke holdninger, mestringstro og kunnskap for å undervise i computing, koding eller algoritmisk tenkning. De påpekte blant annet at de fleste studiene de fant strakk seg over et relativt kort tidsperspektiv, og at det er behov for flere studier som undersøker de langsiktige utfallet av slike kompetansehevende kurs for lærere.

Når programmering skal integreres i læreplanen, vil det alltid være et spørsmål om hvilke fag man skal inkludere det i. Noen av lærerne i Stigberg og Stigberg (2020) sin studie stilte spørsmål ved hvorfor programmering har blitt en del av andre skolefag, og at det heller ville vært passende at programmering som en del av teknologiundervisningen, siden de har et eget teknologifag i Sverige. Fagerlund et al. (2022) fant i sin studie om læreres og elevers motivasjon for programmering at flere av lærerne som deltok studien ikke nødvendigvis var personlig interesserte i programmering, og det var relativt få lærere som kunne tenke seg å drive med programmering på fritiden. Allikevel mente flere at programmering er et viktig tema for elevene å lære i utdannelsen sin. Noen av lærerne i studien var dog ikke overbeviste om verdien av å lære elever å programmere i skolen. Også i Hartell et al. (2019) sin studie er det lærere som stiller spørsmål ved hvorfor programmering har blitt lagt til i en allerede overfylt læreplan, selv om mange av lærerne i utgangspunktet var enige i at programmering var relevant å lære elevene. Med andre ord er det uenigheter rundt hvor relevant det er å integrere programmering i allerede eksisterende skolefag.

Siden den raske utviklingen av teknologi også endrer hva lærerne må kunne undervise elevene sine, er mestringstro viktig for at lærere skal fortsette å selvstendig utforske hva som er nytt og relevant, og hvordan de skal inkludere digital kompetanse i sin undervisning (Mannila et al., 2018). Vannatta og Nancy (2004), som undersøkte hvordan forskjellige personlige egenskaper hos lærere kan påvirke hvordan de brukte teknologi i klasserommet, fant at i tillegg til kompetanseheving, så er det viktig at lærere selv er villige til å lære seg nye ting. De påpeker at det tar tid å lære seg hvordan man skal bruke teknologi, og at det trengs

kurs, men at det også er viktig å bruke tid på å leke med og utforske teknologi på egenhånd. De nevner også at denne villigheten til å bruke tid på å utforske på egenhånd kan komme til syne ved at lærerne er villige til å bruke tid utenfor sin vanlige arbeidstid til å forberede undervisning. Nesten alle lærerstudentene i Nilsson (2008) sin undersøkelse om lærerstudenters oppfatninger om utvikling av fagkunnskaper svarte at de var mer motiverte til å øke sine kunnskaper om et fag (i dette tilfellet fysikk) når de følte at kunnskapen var meningsfull for dem, og kunne brukes i deres fremtidige undervisning. Dette har også blitt funnet blant lærere, der noen har kunnskap om programmering fra før det ble integrert i den nye læreplanen, enten på grunn av egeninteresse eller fordi de har bestemt seg for å være klare for endringene i læreplanen (Vinnervik, 2022). Om en lærer mangler kunnskaper i faget hen underviser, og samtidig er usikker på hvordan å endre undervisningen for å inkludere nye temaer, kan det påvirke hvor mye entusiasme læreren føler for faget eller det spesifikke faglige innholdet (Vinnervik, 2022). Dette går også andre veien; entusiasme kan være en viktig del av prosessen for å ta til seg mer eller ny kunnskap.

Om en lærer føler at de mangler kompetanse i å undervise i et fag eller et tema, kan det fort bli tydelig for læreren selv i en undervisningssituasjon. Nilsson (2008) fant at lærerstudentene i studien hennes kunne føle på frustrasjon og ubehag om elever stilte spørsmål de ikke kunne svare på, og dette var en av grunnen til at lærerstudentene anså det som viktig med kunnskaper i faget de skulle undervise. Da lærerne som deltok i Thorsnes et al. (2020) sin studie ble spurt om elevspørsmål, var det flere som svarte at de ikke trodde de kom til å klare å svare på spørsmål om programmering fra elever med høy faglig mestring, men de uttrykte ikke samme frustrasjon og ubehag som lærerstudentene i studien til Nilsson (2008). De svarte heller at de trodde de kunne komme tilbake til eleven med et svar på et senere tidspunkt, eller finne frem til et svar sammen med eleven.

Når det er deler av undervisningen læreren selv ikke mestrer, må de noen ganger bruke andre strategier. I en studie fra Norge (Kravik et al., 2022) fant forskerne ut at noen lærere er avhengige av hjelp fra elever når de skal undervise i programmering, og at dette er et av tegnene på at lærerne trenger mer kompetanseheving. Andre studier har funnet ut at lærere tar i bruk ressurser og undervisningsopplegg de finner på internett og bruker dette når de selv skal undervise i programmering (Hartell et al., 2019; Stigberg & Stigberg, 2020). For noen lærere blir det dog så mye å velge mellom av alt av ressurser rettet mot programmering som man finner på internett, at det blir vanskelig å vite hvilken ressurs som er mest egnet for

programmeringsundervisning (Thorsnes et al., 2020). Stigberg og Stigberg (2020) fant at til tross for at flere av lærerne i studien deres brukte internett for å finne inspirasjon, så var det ingen av lærerne som brukte et kurs som var utviklet av Skoleverket (en del av det svenske utdanningsdirektoratet) og spesifikt rettet mot å undervise i programmering i skolene. Også i Norge finnes det et slikt kurs i form av en kompetansepakke som er utviklet for å gi lærere en innføring i programmering og algoritmisk tenkning, samt idéer til hvordan de kan integrere programmering i fagene de underviser i (Utdanningsdirektoratet, 2021). Hvor mange skoler og lærere som har tatt i bruk denne kompetansepakken er ukjent for forfatteren av denne oppgaven.

For noen lærere er det ikke nødvendigvis sånn at mestringstroen deres i å undervise i et fag kun avhenger av deres egne kunnskaper og ferdigheter i faget. Thorsnes et al. (2020) fant at selv om noen lærere uttrykte at de var bekymret for at de ikke hadde gode nok ferdigheter i programmering, så var det ikke nødvendigvis slik at mestringstroen ble lavere av den grunn. Lærerne i denne studien hadde tro på at de kom til å klare å undervise i programmering, gjennomføre gode timer og tilrettelegge undervisningen til elever på ulike nivåer. En av strategiene de nevnte for å oppnå dette var å ha åpne oppgaver. Noen av lærerne i studien uttrykte også at det ikke var en utfordring om elevene kunne mer programmering enn de kunne selv. Hvis de opplevde dette brukte de strategier som å ha åpne oppgaver, la elevene velge selv hva de skal jobbe med, eller fant passende oppgaver sammen med elevene. Noen av lærerne syntes det var vanskeligere å lage oppgaver som engasjerte elevene som slet med å få til programmering, og samtidig gav disse elevene mestringsfølelse.

Hvordan kan vi måle mestringstro?

Et mye brukt instrument for å måle læreres mestringstro i naturfag er STEBI (Science Teaching Efficacy Belief Instrument) (Riggs & Enochs, 1990). Instrumentet er utviklet for å måle både læreres mestringstro og utfallsforventning (outcome expectancy). Begge deler er viktige faktorer som er med på å bestemme en person sin atferd (Bandura, 1977). En utfallsforventning er en person sin forventning om at en gitt atferd vil føre til visse utfall, og mestringstro er personens tro på at de selv klarer å gjennomføre atferden som trengs for å oppnå disse utfallene (Bandura, 1977). For å ta et eksempel fra programmeringsundervisning; en lærer kan ha en forventning om at hyppig underveisvurdering med tilbakemeldinger som hjelper elevene videre, vil føre til at de presterer på et høyere nivå og får mer motivasjon for å

jobbe med programmering. Men den samme læreren kan også tenke at hen selv ikke vil ha tid til å følge opp elevene og gi den undervisvurderingen som trengs for at elevene skal oppnå dette. En person kan ha slike tanker om at noen handlinger vil føre til bestemte utfall, men om de tviler på at de selv vil klare å utføre handlingene, så vil ikke utfallsforventningene påvirke handlingene i like stor grad (Bandura, 1977). Siden mestringstro og utfallsforventninger er to forskjellige ting, så trengs det også forskjellige spørsmål for å måle dem hver for seg. Derfor inneholder STEBI noen spørsmål som måler mestringstro og noen spørsmål som måler utfallsforventninger (Riggs & Enochs, 1990).

Et annet instrument som kan brukes er DAS (Dimensions of Attitude Toward Science) (Aalderen-Smeets & Molen, 2013). Dette instrumentet måler ikke bare mestringstro, men også læreres holdninger til naturfag. De har delt inn læreres holdninger til naturfag i tre dimensjoner, der mestringstro ligger under dimensjonen «oppfattet kontroll». Siden dette instrumentet også er delt inn for å måle forskjellige aspekter ved holdninger, så er det utviklet egne spørsmål som kun skal måle mestringstro.

Begge instrumentene som er nevnt over er utviklet for å måle mestringstro og holdninger til naturfag. Dette vil ikke være det samme som å måle mestringstro i programmering. Det er allikevel mulig å ta utgangspunkt i disse instrumentene for å utvikle et instrument for å måle mestringstro i programmeringsundervisning i naturfag. Hartell et al. (2019) gjorde dette ved å ta utgangspunkt i DAS instrumentet, men å endre ordet «science» til «programmering» slik at alle spørsmålene rettet seg mot programmering i stedet for naturfag. De la også til noen spørsmål for å gjøre instrumentet mer relevant for en svensk skolekontekst.

Siden mestringstro er kontekstspesifikt, er det viktig at de som skal vurdere egen mestring vurderer den i en gitt kontekst (Goddard et al., 2000). Når lærere skal vurdere egen mestringstro, må de da vurdere egne evner opp mot kravene for å lykkes i den gitte undervisningssituasjonen (Goddard et al., 2000; Tschannen-Moran et al., 1998). Med tanke på det, vil en ulempe med instrumentene beskrevet over være at man ikke får noen informasjon om respondentens tanker rundt dette. Om man ønsker å vite mer om hva en respondent tenker at kreves for å lykkes i en gitt situasjon, og hvordan hen vurderer egne evner mot disse kravene, så kan det lønne seg å undersøke fenomenet mer kvalitativt. Tschannen-Moran et al. (1998) undersøker blant annet forskjellige instrumenter som har blitt utviklet for å måle mestringstro. Ifølge dem trengs det flere kvalitative studier av læreres mestringstro, for å blant

annet kunne gi en rikere beskrivelse av læreres mestringstro og få en bedre forståelse av hvordan mestringstro utvikler seg.

Metode

I dette kapittelet presenterer og begrunner jeg de metodiske valgene i denne oppgaven. Først går jeg gjennom studiens overordnede metodologi. Så tar jeg for meg metodene for datainnsamling og analyse, samt kontekst og utvalg. Til slutt drøfter jeg reliabiliteten, validiteten og etiske problemstillinger knyttet til oppgaven.

Begrunnelse for studiens metodologi

Denne studien av læreres mestringstro i programmeringsundervisning i naturfag er basert på en kvalitativ spørreundersøkelse av naturfaglærere. Å rette undersøkelsen direkte mot naturfaglærere var naturlig siden jeg ønsket å undersøke læreres mestringstro innenfor ulike deler av programmeringsundervisning i naturfag. Jeg ønsket å undersøke om det er noen svar som går igjen hos flere respondenter, for eksempel om flere av respondentene føler at de mestrer og ikke mestrer de samme tingene, og jeg ville derfor ha svar fra et større antall lærere. Dermed ble spørreundersøkelse et naturlig valg fremfor eksempelvis intervju, siden det kan sendes ut til flere lærere på en gang. Jeg ønsket også å gi lærerne mulighet til å gi utfyllende svar, og derfor falt valget på et kvalitativt spørreskjema. Spørreskjema er praktisk for respondentene, fordi de kan svare på det når det passer dem (Cohen et al., 2018). Det er også mulig at svarprosenten blir høyere og at respondentene svarer fortere (Cohen et al., 2018).

Utviklingen av spørreskjema

For å lage spørreskjemaet, tok jeg utgangspunkt i de kvantitative instrumentene STEBI (Riggs & Enochs, 1990) og DAS (Aalderen-Smeets & Molen, 2013), som begge er laget for å måle blant annet mestringstro. Begge instrumentene inneholder spørsmål som skal svares på med en likert skala. De inneholder også flere spørsmål som skal måle andre dimensjoner enn bare mestringstro. Det første jeg gjorde var å identifisere punktene som måler mestringstro (dette er ført opp i artiklene). STEBI inneholder 13 slike punkter, som er presentert i Tabell 1, og DAS inneholder 4, som er presentert i Tabell 2.

Tabell 1: spørsmål fra STEBI (Riggs & Enochs, 1990) som måler mestringsstro

Spørsmål fra STEBI som måler mestringsstro
I am continually finding better ways to teach science.
Even when I try very hard, I don't teach science as well as I do most subjects.
I know the steps necessary to teach science concepts effectively.
I am not very effective in monitoring science experiments.
I generally teach science ineffectively.
I understand science concepts well enough to be effective in teaching elementary school.
I find it difficult to explain to students why science experiments work.
I am typically able to answer students' science questions.
I wonder if I have the necessary skills to teach science.
Given a choice, I would not invite the principal to evaluate my science teaching.
When a student has difficulty understanding a science concept, I am usually at a loss as to how to help the student understand it better.
When teaching science, I usually welcome student questions.
I don't know what to do to turn students on to science.

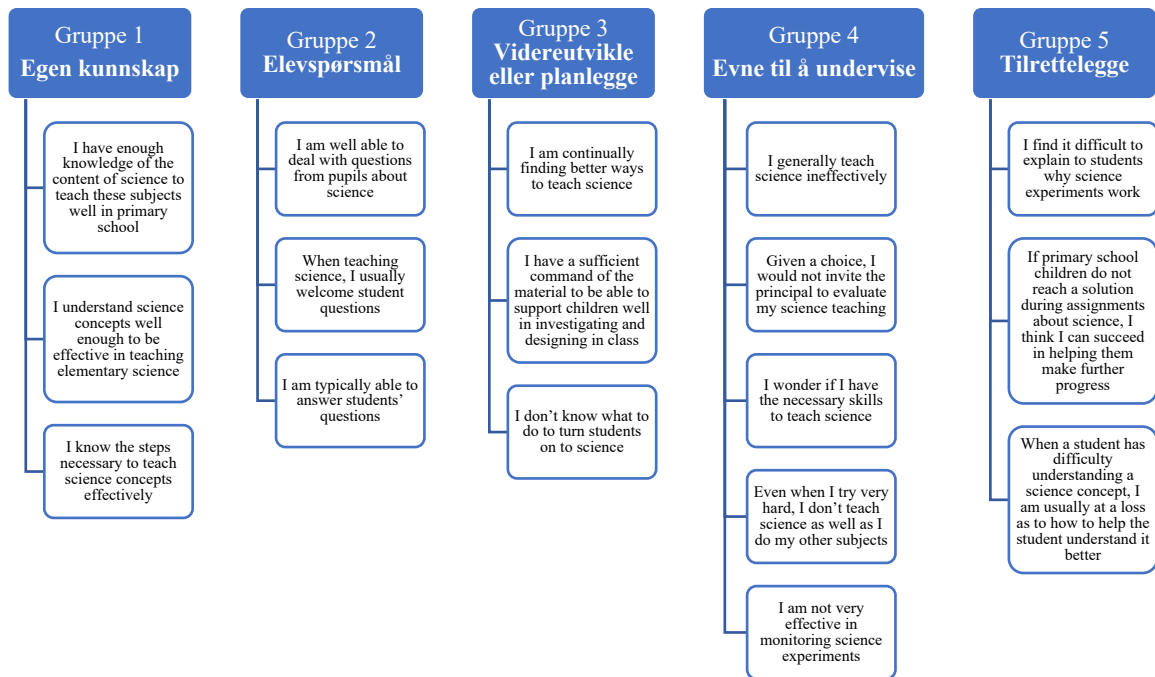
Tabell 2: Spørsmål fra DAS (Aalderen-Smeets & Molen, 2013) som måler mestringsstro

Spørsmål fra DAS som måler mestringsstro
I have enough knowledge of the content of science to teach these subjects well in primary school
I am well able to deal with questions from pupils about science
I have sufficient command of the material to be able to support children well in investigating and designing in class
If primary school children do not reach a solution during assignments about science, I think I can succeed in helping them make further progress

Videre prøvde jeg å dele spørsmålene inn i grupper ut ifra hva de handlet om. De ble først delt inn i syv grupper. Inndelingen ble så diskutert med medstudenter, noe som førte til at noen grupper ble slått sammen og endret. Til slutt satt jeg igjen med fem grupper. Disse er presentert i Figur 1. Alle gruppene fikk et navn som var ment å beskrive temaet til gruppen.

Figur 1

Spørsmål inndelt i tematiske grupper



Noen av punktene kunne ha passet inn i flere grupper, så de ble videre diskutert med medstudenter, men temaene til gruppene forble allikevel de samme og det ble derfor bestemt å beholde inndelingen over. Siden temaene i de fem gruppene gikk igjen i de to spørreundersøkelsene som var utviklet for å måle mestringstro, bestemte jeg at temaene til gruppene skulle være grunnlaget til undersøkelsen. Siden vurdering er en viktig del av undervisning og nevnes eksplisitt i læreplanen (Utdanningsdirektoratet, 2020), valgte jeg å også ha med vurdering som et eget tema i undersøkelsen. Dermed ble det totalt seks kategorier som ble utgangspunktet til undersøkelsen. Disse er presentert i Tabell 3.

Tabell 3: Kategorier til undersøkelse om læreres mestringstro

Kategorier til undersøkelse
Kunnskaper om programmering og programmeringsferdigheter
Evne til å undervise i programmering i naturfag
Evne til å planlegge og videreutvikle et undervisningsopplegg
Evne til å svare på spørsmål fra elever
Evne til å tilrettelegge programmeringsundervisningen
Evne til å vurdere elever

Både det å planlegge og videreutvikle undervisningsopplegg, svare på spørsmål fra elever, tilrettelegge undervisningen og vurdere elever er deler av det å undervise i programmering i naturfag. «Evne til å undervise i programmering i naturfag» ble allikevel valgt til å være en egen kategori til undersøkelsen. Grunnen til dette var at det kunne tenke seg at en respondent for eksempel på et mer generelt grunnlag vurderer at hen mestrer å undervise i programmering i naturfag godt, men at hen ikke føler at hen mestrer enkelte av de spesifikke delene. Det kunne for eksempel være interessant å se om noen føler at de ikke mestrer å tilrettelegge undervisningen og vurdere elevene, men allikevel svarer at de mestrer å undervise i programmering i naturfag.

Undersøkelsen (se Vedlegg 1) startet med å spørre om noe bakgrunnsinformasjon. Informantene ble spurt om hvilke trinn de har erfaring med å undervise i naturfag, om de har undervist i programmering, og hvis ja: i hvilke fag de har undervist i programmering og på hvilke trinn de har erfaring med å undervise i programmering. Dette ble gjort for å få en kort oversikt over respondentenes erfaringer med programmeringsundervisning. Etter det kom det seks spørsmål formulert etter kategoriene i Tabell 3 med åpne tekstsvare. Jeg valgte å holde undersøkelsen relativt kort, fordi det kan øke sjansen for at flere respondenter fullfører (Cohen et al., 2018). Jeg valgte også å ha alle spørsmålene på én side i stedet for at de klikket seg videre for hvert spørsmål, slik at respondentene kunne få god oversikt over hvor langt spørreskjemaet var.

I alle de seks spørsmålene i denne undersøkelsen ble respondentene bedt om å vurdere sine evner innenfor de spesifikke områdene av programmeringsundervisning i naturfag som ble beskrevet over, for eksempel «Her ønsker jeg at du skal vurdere din evne til å planlegge og videreutvikle et undervisningsopplegg i programmering i naturfag». De ble formulert på denne måten etter anbefalingen til Bandura (2006), som poengterer at respondenter bør bli bedt om å vurdere egne evner heller enn å vurdere sine intensjoner, siden mestringsstro handler om ens oppfatning av egne evner. For å være helt sikker på at respondentene forsto at det var vurdering av evner i programmeringsundervisning i *naturfag* jeg var ute etter, ble dette eksplisitt nevnt i alle punktene, med unntak av det som spurte mer generelt om kunnskaper om programmering og programmeringsferdighetene til respondentene. Jeg la også til flere detaljer der det virket hensiktsmessig, som i eksempelet over der jeg spesifiserte at «Her tenker jeg både på undervisningsopplegg du har laget selv, og undervisningsopplegg du har lånt fra andre». Slike detaljerte instruksjoner er viktige i kvalitative spørreskjemaer med åpne

tekstsvær, for at respondentene skal vite hva de skal svare på (Cohen et al., 2018). Siden jeg ønsket muligheten til å sammenlikne svarene til respondentene, valgte jeg å ikke ha med noen hjelpespørsmål eller «punkter du kan ha med er ...» og heller holde det helt åpent. Da er det mulig å sammenlikne og se om flere respondenter trekker frem noe av det samme. Det gjør også at respondentene er mer frie til å svare det de selv ønsker, og det de selv tenker er relevant for å vurdere sine egne evner.

For å forsikre meg om at ingen respondenter satt igjen med en følelse av at det var mer de ønsket å skrive som ikke passet inn i noen av de andre spørsmålene, la jeg til et ekstra spørsmål for dette der jeg spurte om det var noe mer de ønsket å legge til om programmering i naturfag, som de ikke syntes passet i de foregående spørsmålene. Dette er noe som er anbefalt i andre kvalitative metoder, som intervju (Kvale & Brinkmann, 2015), og det virket derfor hensiktsmessig å ta det med i denne undersøkelsen. Det gav også respondentene en mulighet til å ta opp noe de tenkte var relevant i sammenheng med programmering i naturfag som ikke var med i undersøkelsen.

Utvalg

Siden oppgaven retter seg mot naturfagslærere, var det nødvendig å kun rekruttere naturfagslærere som informanter. Siden det bare var de som hadde disse kvalifikasjonene som fikk delta i studien, er utvalget et ikke-randomisert formålsutvalg (Nardi, 2018). Det ble ikke stilt noe krav til at respondentene hadde erfaring med å undervise i programmering. Grunnen til dette var at jeg ville at undersøkelsen skulle være mulig å svare på også for naturfagslærere som av forskjellige grunner ikke har undervist i programmering enda. Tanken var at disse respondentene kan vurdere om de har evnene til å gjennomføre programmeringsundervisning i naturfag selv om de ikke har gjort det.

Respondentene ble rekruttert på to forskjellige måter. Den første er at lærere som deltok på et programmeringskurs høsten 2022 fikk mulighet til å melde seg på for å motta informasjon om en masterstudie. Fra disse kursene var det 24 lærere som meldte interesse. Den andre måten respondenter ble rekruttert på, var å legge ut informasjon om studien og en lenke til spørreskjemaet i flere facebook-grupper, både grupper for naturfaglærere og grupper for lærere generelt. Siden spørreskjemaet var anonymt, var det ingen måte å vite om de endelige respondentene var blant lærerne fra programmeringskurset eller lærerne fra facebook-

gruppene. For å øke antallet respondenter ble det etter to uker sendt ut en påminnelsemail og innlegget ble delt på facebook-gruppene igjen.

Det var totalt ti respondenter som svarte på spørreskjemaet. Av de ti var det én respondent som svarte veldig kort på alle spørsmålene. Svarene var av typen «Middels» heller enn å skrive utfyllende svar. Siden respondenten svarte såpass kort på alle spørsmålene, var det ikke mulig å analysere svarene med utgangspunkt i oppgavens hensikt og rammeverket jeg brukte for å analysere datamaterialet. Derfor ble ikke svarene til denne respondenten med som en del av datamaterialet til oppgaven. En annen respondent hadde noen korte svar av samme type, men hadde også svart mer utfyllende på noen spørsmål. De mer utfyllende svarene kunne analyseres, og derfor er noen av svarene til denne respondenten med som en del av datamaterialet til oppgaven. Dermed ble det totalt ni respondenter som utgjorde det faktiske utvalget som produserte datamaterialet. Disse blir presentert i Tabell 4.

Tabell 4: Utvalg

Respondent	Har erfaring med å undervise i naturfag på trinn: *	Har undervist i programmering	Har undervist i programmering i faget/fagene:	Har undervist i programmering på trinn: *
Respondent 1	1.-7. trinn	Ja	Naturfag, matematikk, musikk, kunst og håndverk, norsk	1.-7. trinn
Respondent 2	5.-7. trinn	Ja	Naturfag	5.-7. trinn
Respondent 3	5.-7. trinn	Ja	Naturfag, matematikk	5.-7. trinn
Respondent 4	1.-4. trinn	Ja	Matematikk	1.-4. trinn
Respondent 5	5.-7. trinn	Ja	Matematikk	5.-7. trinn
Respondent 6	5.-7. trinn	Ja	Naturfag, matematikk	5.-7. trinn
Respondent 7	1.-7. trinn	Ja	Naturfag	5.-7. trinn
Respondent 8	1.-4. trinn	Ja	Programmering som eget fag	1.-4. trinn
Respondent 9	1.-4. trinn	Ja	Naturfag, matematikk	1.-7. trinn

* Når respondentene fikk spørsmål om trinn, kunne de velge 1.-4. trinn, 5.-7. trinn eller begge deler. Det er derfor ikke oppgitt nøyaktig hvilket trinn de har undervist i naturfag eller programmering, kun om det er småtrinnet, mellomtrinnet eller begge deler

Analyse

For å analysere dataene gjorde jeg en kvalitativ innholdsanalyse inspirert av fremgangsmåten Hsieh og Shannon (2005) kaller rettet tilnærming (directed approach). Forhåndsbestemte midlertidige kategorier ble definert basert på Bandura (1997) sin teori om mestringstro, samt andre kilder til mestringstro som ble definert av Butz og Usher (2015) og Palmer (2006). Når en lærer vurderer hva som kreves for å klare å gjennomføre en undervisningssituasjon vurderer hen ikke bare indre faktorer, men også ytre faktorer som tilgjengelighet til undervisningsmateriale og tilgang til teknologi (Tschannen-Moran et al., 1998). For å kunne få frem slike ytre faktorer i analysen ble også *rammefaktorer* lagt til som en forhåndsbestemt kategori. Det endelige rammeverket som ble brukt for å analysere dataene består av 11 koder og er vist i Tabell 5.

Tabell 5: Forhåndsbestemte koder til analyse

Forhåndsbestemte koder
Mestringserfaring – Personlige erfaringer av å ha mestret programmeringsundervisning
Sosial påvirkning – Å bli overbevist av andre at man kan klare å gjennomføre programmeringsundervisning
Vikarierende erfaring – Å observere at andre klarer å gjennomføre programmeringsundervisning
Kognitiv selvmodellering – Å visualisere at man klarer å gjennomføre programmeringsundervisning
Emosjonell tilstand – Opplevelser av positive emosjoner i programmeringsundervisningen
Selvregulering – Hvordan man organiserer arbeidet sitt og tiden sin
Tilgjengelighet til hjelp – Opplevelser av at det er mulig å få hjelp om man trenger det
Fremtidsperspektiv – Å se viktigheten av å undervise elevene i programmering ut ifra et fremtidsperspektiv
Rammefaktorer – ytre faktorer som utstyr og tid
Forestilt faglig mestring – Forestillinger om at man selv forstår det faglige innholdet man skal lære elevene
Forestilt didaktisk mestring – Forestillinger om at man forstår hvordan man skal formidle det faglige innholdet til elevene

Kognitiv selvmodellering er en form for vikarierende erfaring (Bandura, 1997). Allikevel valgte jeg å plassere begge to som overordnede koder. Grunnen til dette var at spørreundersøkelsen rettet seg både mot naturfagslærere som har erfaring med å undervise i programmering, og naturfagslærere som ikke har det. Tanken var at det var mulig at noen av respondentene ikke kom til å ha erfaringer med å undervise i programmering. Da ville mest sannsynlig en del av svarene deres ha basert seg på at de visualiserte at de gjennomfører undervisningen, og dermed ville kognitiv selvmodellering være en kode som burde være med i rammeverket. Siden jeg benyttet meg av fremgangsmåten rettet tilnærming var det også åpent for at andre former for vikarierende erfaringer kunne bli lagt til som koder om det dukket opp i datamaterialet.

I tillegg til de 11 kodene i Tabell 5 hadde jeg også med kodene «Annet» og «Uspesifisert». Under «Annet» kodet jeg svar fra respondenter som ikke kunne brukes i analysen fordi det ikke var noe jeg kunne knytte til hensikten med oppgaven. Det var for eksempel en respondent som under ekstraspørsmålet gav en tilbakemelding om hva hen syntes om selve spørreundersøkelsen. Siden dette ikke var noe som var knyttet til hensikten til oppgaven, og dermed ikke kunne analyseres med utgangspunkt i rammeverket jeg brukte, ble svaret kodet til «Annet». Koden «Uspesifisert» ble brukt når en respondent gav et svar der hen vurderte egne evner slik som spørsmålene bad om, men ikke utdypet det noe mer. Det var heller svar av typen «Lav evne» eller «Middels pluss». Rammeverket som ble brukt i analysen baserer seg på kilder til mestringstro. Siden det i disse svarene ikke ble oppgitt noen grunn til at respondentene vurderte sine evner på nivået de gjorde, var det ikke mulig å analysere svarene med utgangspunkt i rammeverket jeg brukte. Svarene ble derfor kodet til «Uspesifisert», fordi respondentene ikke spesifiserte hva grunnlaget for vurderingen av egne evner var.

Siden svarene til respondentene kan være både positive og negative innenfor de ulike kodene, for eksempel at de skriver om positive mestringserfaringer eller negative mestringserfaringer, ble det også lagt til underkoder. Hver overordnet kode fikk to underkoder: positiv og negativ. Poenget med underkodene var å kunne sortere datamaterialet ut ifra om respondentenes erfaringer og vurderinger av egne evner var gode eller ikke. Koden mestringserfaringer fikk for eksempel de underordnede kodene positiv mestringserfaring og negativ mestringserfaring. Når respondentene har blitt spurt om å vurdere sin evne til å tilrettelegge undervisningen har for eksempel «Programmering er nok noe av det som er lettest å tilrettelegge.» blitt kategorisert som en positiv mestringserfaring fordi det ble tolket som at respondenten her

peker på faktiske erfaringer fra klasserommet som har gitt hen opplevelser av at det går greit å tilrettelegge undervisningen. Et svar fra en annen respondent var «Det er utfordrende for meg å gi oppgaver som er utfordrende nok til elever som viser stor grad av mestring.» Dette ble kategorisert som en negativ mestringserfaring. I likhet med det forrige eksempelet ble det kodet til mestringserfaringer fordi det ble tolket som at dette er noe respondenten selv har opplevd i en undervisningssituasjon. Siden erfaringen er at respondenten synes noe er utfordrende, ble det kategorisert som en negativ mestringserfaring.

Datamaterialet ble gjennomgått og analysert i flere runder. Først ble det kodet til de 11 hovedkategoriene. På et senere tidspunkt ble alt datamaterialet gått gjennom på nytt, for å dobbeltsjekke om jeg var enig med meg selv om hvordan det var kodet. Så ble alt materialet gjennomgått på nytt igjen da det ble delt inn i de positive og negative underkategoriene. Ingen andre har gått gjennom og kodet datamaterialet, men analysen har blitt diskutert med medstudenter. Jeg gikk da gjennom hvordan datamaterialet var blitt kodet og hvilke argumenter jeg hadde for å kode det på den måten for å sjekke om medstudenter var enige eller om de hadde andre tolkninger av respondentene sine svar. Det ble ikke gjort noen endringer etter diskusjonen med medstudentene; de var enige i hvordan datamaterialet var blitt analysert.

Det er vanlig å dele inn en tekst i mindre deler, såkalte analyseenheter, når man skal gjøre en innholdsanalyse, og kategorisere disse hver for seg (Bakken & Andersson-Bakken, 2021). Det var ingen fast størrelse på analyseenhetene, de var forskjellige etter hva som virket hensiktsmessig. Da respondentene skulle vurdere egne kunnskaper om programmering og programmeringsferdigheter var svaret til Respondent 2: «Gode. Har kursing, egeninteresse og lang erfaring med undervisningen.» Dette svaret ble totalt kodet til tre kategorier. «Gode. Har kursing» ble kodet til positiv forestilt faglig mestring fordi det ble tolket som at respondenten med «Gode.» gav en generell vurdering av sine kunnskaper og ferdigheter, og «har kursing» også viser til fagkunnskaper. «egeninteresse» ble kodet til positiv emosjonell tilstand fordi jeg tolket det som at informanten liker å drive med programmering og derfor har positive følelser mot det å programmere. «lang erfaring med undervisning» ble kodet som positiv mestringserfaring fordi respondenten her eksplisitt nevner erfaringer. Alle disse analyseenhetene ble kodet i de positive underkategoriene for kodene, fordi det ble tolket som at de pekte tilbake på og underbygget den første vurderingen av at respondentens kunnskaper og ferdigheter innenfor programmering er gode. Til sammenlikning ble Respondent 1 sitt svar

på samme spørsmål analysert med utgangspunkt i en større analyseenhet. Her ble hele respondents svar, «Jeg føler at jeg mestrer dette godt, men bare hvis jeg har brukt god tid på forberedelse av økta», kodet til positiv selvregulering. Begrunnelsen for dette er at respondentens vurdering av egne ferdigheter som gode avhenger av hvor godt hen forbereder seg, og dermed er det bruk av tid til forberedelse som styrer hvilken kategori den ble puttet i.

I noen tilfeller var det flere respondenter som nevnte noen av de samme tingene da de svarte på samme spørsmål, men svarene ble allikevel kodet forskjellig på grunn av en tolkning av svaret som helhet. Da de ble bedt om å vurdere egne evner til å planlegge og videreutvikle undervisningsopplegg, var det for eksempel to respondenter som begge skrev at de tar i bruk nettressurser. Respondent 3 skrev «Jeg mestrer å planlegge og videreutvikle undervisningsopplegg i programmering ved å ta i bruk ressurser som finnes på nett.» og Respondent 6 skrev «Når det gjelder faktisk programmering og det å lage program er jeg ofte avhengig av ferdige opplegg eller eksempelprogrammer fra eksterne kilder (...)». Svaret til Respondent 3 ble tolket som at respondenten følte at hen mestrer å planlegge og videreutvikle undervisningsopplegg med hjelp fra nettressurser, og det ble derfor kategorisert som positiv tilgjengelighet til hjelp. Svaret til respondent 6 ble tolket som at hen føler at hen i utgangspunktet ikke mestrer å planlegge og videreutvikle et undervisningsopplegg. Dette er i utgangspunktet også noe man kan si om Respondent 3, men siden Respondent 6 beskriver seg selv som *avhengig* av eksterne ressurser ble utsagnet som helhet tolket til å i større grad være en vurdering av egen kunnskap. Derfor ble det kategorisert som negativ forestilt faglig mestring.

Noen svar fra respondenter var litt i gråsonen for hva man kan kalle negativ og positiv. I de tilfellene ble det lagt vekt på om ordlyden var mer på den positive eller negative siden, og svarene ble kategorisert deretter. Da respondentene ble bedt om å vurdere egne kunnskaper om programmering og programmeringsferdigheter, ble for eksempel svaret til respondent 5 «Grei forståelse generelt, må ofte sjekke «oppskrifter» dersom jeg prøver nye prosjekter» kodet som positiv forestilt faglig mestring, mens svaret til respondent 4 «Programmering er ikke det jeg har mest kunnskap om» kodet som negativ forestilt faglig mestring. Også her nevner en respondent å ta i bruk ressurser fra nett, men det har ikke blitt kodet til tilgjengelighet til hjelp. Grunnen til dette var igjen en vurdering av utsagnet som helhet. I svaret til Respondent 3 over tolkes det som at mestringen kommer av at hen tar i bruk nettressurser. I svaret til Respondent 5 tolkes det som at hen vurderer at hen generelt har gode

kunnskaper. Hadde hen skrevet at hen har grei forståelse generelt *fordi* hen har nettressurser tilgjengelig, så hadde det blitt kodet under tilgjengelighet til hjelp, men siden respondentens vurderinger av evner ikke var begrunnet med tilgjengelighet til hjelp ble utsagnet som helhet tolket til å handle om respondentens forestilte faglige mestring.

I noen få tilfeller ble en respondent sitt svar i et tidligere spørsmål med på å analysere svaret i et annet spørsmål. Det var for eksempel kun ett svar som ble kodet til kognitiv selvmodellering, og det ble kodet til den negative underkoden. Respondent 3 skrev «Jeg mestrer ikke å skulle undervise i skriftprogrammering.» Grunnen til at dette ble plassert under negativ kognitiv selvmodellering, var at respondenten i et tidligere svar skrev «Har ingen ferdigheter eller kunnskap om tekstprogrammering.» Det ble derfor tolket som at hen ikke faktisk har prøvd å undervise i tekstprogrammering, men ser for seg at om hen hadde prøvd så hadde hen ikke fått det til.

Reliabilitet og validitet

Validitet handler om måleinstrumentet, og om det måler det det er ment for å måle (Nardi, 2018). Dette er den klassiske definisjonen som brukes for kvantitative studier. I en kvalitativ studie vil det handle om hvorvidt metodene som er brukt i studien undersøker det den har hensikt å undersøke, såkalt begrepsvaliditet (Johannessen et al., 2005). I denne studien blir altså dette spørsmålet om spørreundersøkelsen som ble utviklet faktisk måler læreres mestringstro i programmeringsundervisning i naturfag.

Det første steget som ble tatt for å øke begrepsvaliditeten var å ta utgangspunkt i to spørreundersøkelser som allerede var utviklet for å måle mestringstro. Disse spørreundersøkelsene er, ifølge forskerne som utviklet dem, valide og reliable instrumenter for å måle mestringstro (Riggs & Enochs, 1990; Alderen-Smeets & Molen, 2013). Siden spørreundersøkelsene ble endret og spørsmålene ble omformulert, vil ikke validiteten til disse instrumentene være gyldig i undersøkelsen som ble utviklet til denne oppgaven, men at allerede valide instrumenter ble tatt i bruk for å utvikle den nye undersøkelsen kan allikevel være med på å øke sjansen for at den nye undersøkelsen måler mestringstro. Det andre steget som ble tatt for å øke begrepsvaliditeten var å følge Bandura (2006) sine anbefalinger om hvordan man bør formulere spørsmål når man utvikler instrumenter for å måle mestringstro, og jeg jobbet målrettet for å unngå ledende spørsmål i spørreundersøkelsen.

Noe som svekker denne studiens begrepsvaliditet, er valget om å gjøre en kvalitativ spørreundersøkelse i stedet for et intervju. Ved å gjennomføre et intervju kunne respondentene ha blitt bedt om å utdype svarene sine, og kunne dermed ha kommet med en mer detaljert beskrivelse av sin mestringsstro. Valget falt allikevel på en kvalitativ spørreundersøkelse fordi jeg hadde et ønske om å få svar fra et større antall lærere, og dermed få anledning til å få frem flere forskjellige synspunkter.

En annen form for validitet er ekstern validitet, som handler om en studies overførbarhet (Johannessen et al., 2005). Generell overførbarhet, eller generalisering, er noe man som regel kun kan finne i kvantitative undersøkelser, så ekstern validitet blir i en kvalitativ undersøkelse heller hvorvidt forskeren klarer å etablere beskrivelser, fortolkninger og forklaringer som er nyttige i andre situasjoner (Johannessen et al., 2005). I denne studien har jeg forsøkt å øke den eksterne validiteten ved å forklare nøye hva jeg har gjort for å komme frem til resultatene. Både informasjon om utviklingen av spørreskjemaet og hvordan svarene er analysert har blitt forklart slik at leseren skal få nok grunnlag til å kunne vurdere om hen kan overføre resultatene til en annen kontekst.

Reliabilitet er forventningen om at resultatene av en studie blir de samme hver gang de måles (Nardi, 2018), altså at det i teorien skal bli samme resultater når man bruker samme forskningsmetode på en liknende gruppe (Cohen et al., 2018). I dette tilfellet vil det bety om resultatene av studien ville blitt den samme om den samme undersøkelsen ble gjort på flere grupper av naturfagslærere. Vanlige måter å teste dette på er å gjenta samme undersøkelse på samme gruppe på to forskjellige tidspunkt, eller at flere forskere undersøker samme fenomen og sammenlikner resultatene (Johannessen et al., 2005). Ingen av disse måtene å teste reliabilitet på har vært mulig å gjennomføre for dette masterprosjektet. Et begrep som dermed blir viktig, er overenstemmelse. Å forsøke å oppnå overenstemmelse mellom virkeligheten og resultatene betyr at resultatene kommer fra forskningen, og ikke er et resultat av forskerens subjektive holdninger (Johannessen et al., 2005). Med andre ord handler det om i hvilken grad forskeren selv har påvirket resultatene.

For å forsøke å oppnå overenstemmelse mellom virkeligheten og resultatene, har jeg først tatt grepene som ble beskrevet over da jeg utviklet spørreundersøkelsen. Det viktigste grepet når det kommer til overenstemmelse er at jeg valgte å ikke legge til noen hjelpespørsmål eller gi

eksempler til hva respondentene kunne trekke frem eller nevne i svarene sine. Dette ble gjort for å forsøke å unngå og påvirke svarene til respondentene på noen som helst måte. Da dataen senere ble analysert, valgte jeg å ta utgangspunkt i et allerede eksisterende rammeverk. Dette var også for å forsøke å holde meg objektiv, og prøve å hindre at egne biaser skulle påvirke hvordan dataen ble analysert. En annen måte å prøve å hindre at egne biaser skulle påvirke analysen var å diskutere hvordan dataen var blitt analysert med medstudenter. Da fikk jeg teste om medstudentene var enige i argumentene jeg hadde for å analysere datamaterialet slik som jeg gjorde det.

Forskningsetiske refleksjoner

Selv om undersøkelsen var anonym, ble det søkt om og gitt godkjenning hos Sikt (se Vedlegg 2). Begrunnelsen for dette var at undersøkelsen hadde åpne tekstsvare, og jeg derfor ikke kunne garantere at ingen respondenter oppga personopplysninger i de åpne tekstsvarene. For å prøve og unngå at dette skjedde fikk også informantene skriftlig beskjed i starten av undersøkelsen at de ikke skulle oppgi noen personidentifiserende informasjon. De ble også minnet på taushetsplikten de har som lærere for å unngå at de oppga taushetsbelagt informasjon.

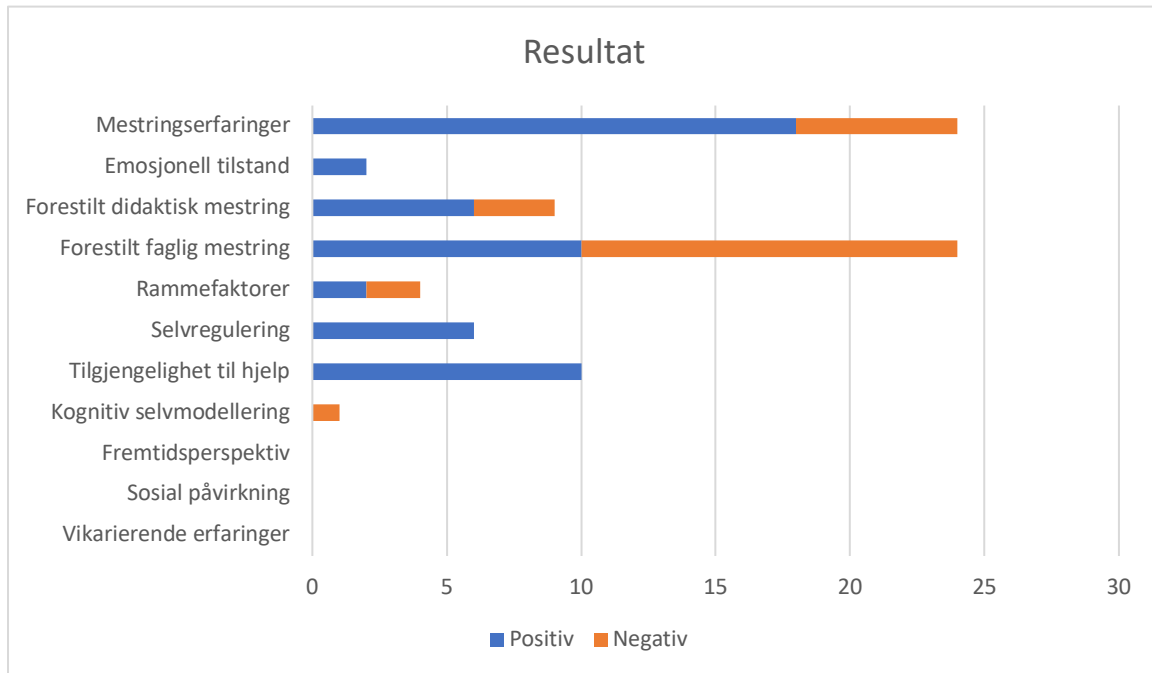
Resultater

I dette kapitlet blir funnene fra analysen av svarene til respondentene presentert. Først presenterer jeg en oversikt over resultatene av analysen. Videre går jeg gjennom resultatene med utgangspunkt i de ulike kodene som ble brukt i analysen. På denne måten ønsker jeg å legge frem hvordan respondentene beskriver sin vurdering av egne evner, og dermed svare på forskningsspørsmålet «Hvordan beskriver respondentene sine opplevelser av mestring i programmeringsundervisning i naturfag?». Siden kodene som ble brukt i analysen er ulike kilder til mestringstro, vil det også komme frem hvilke kilder til mestringstro respondentene selv trekker frem som viktige for deres egne opplevelser av å mestre eller ikke mestre programmeringsundervisningen. Til slutt presenterer jeg funnene som skal svare på det andre forskningsspørsmålet: «Hvilke strategier oppgir respondentene at de bruker på tross av lav forestilt mestring?»

I Figur 2 er det en oversikt over antall analyseenheter som er kodet til de ulike overordnede kodene, og om de er kodet til positiv eller negativ underkode. Vi kan se at de aller fleste kategoriene har flest analyseenheter kodet til den positive underkategorien, mens et par av kodene har en overvekt av analyseenheter kodet til den negative underkategorien. Det er viktig å få et mer nyansert bilde av hva innholdet i respondentene sine svar er, for eksempel hva en «positiv mestringserfaring» er ut ifra svarene til respondentene. Derfor skal jeg nå skrive om en og en kode for å gå mer i dybden til de forskjellige kodene.

Figur 2

Oversikt over resultatene av analysen



Note: Diagrammet viser antall analyseenheter kodet til de ulike hovedkodene, og hvilken andel som er kodet til de positive og negative underkodene

Hvordan beskriver respondentene sine opplevelser av mestring i programmeringsundervisning i naturfag?

Mestringserfaringer

Mestringserfaringer er personlige erfaringer respondentene har av å ha mestret eller ikke mestret programmeringsundervisningen. Antall analyseenheter og respondenter med svar som er kodet til positive og negative mestringserfaringer er vist i Tabell 6.

Tabell 6: Oversikt over antall analyseenheter og respondenter med svar kodet til mestringserfaringer

Mestringserfaringer	Positiv	Negativ	Totalt
Antall analyseenheter	18	6	24
Antall respondenter	8	4	8*

*En respondent kan ha både positive og negative utsagn. Derfor vil ikke totalt antall respondenter nødvendigvis være summen av antall respondenter med positive utsagn og antall respondenter med negative utsagn.

Mestringserfaringer var en av kodene som fikk flest svar kodet til seg. Åtte respondenter nevnte positive mestringserfaringer, men innenfor litt forskjellige områder. Blant annet har

seks respondenter svart noe positivt når de vurderer om de klarer å svare på spørsmålene fra elevene. Noe som går igjen hos flere respondenter er at de nevner noen positive mestringserfaringer, men også noe som har blitt kodet til noe negativt i samme svar.

Respondent 9 skriver for eksempel «Noen spørsmål har jeg fint klart å besvare (positiv mestringserfaring), men her kan jeg også til tider streve fordi jeg mangler kunnskap (negativ forestilt faglig mestring).» Noen respondenter kommer også med løsninger til eventuelle situasjoner de ikke mestrer. Noen skriver for eksempel at de for det meste klarer å svare greit på spørsmålene elevene har (positiv mestringserfaring), men om det er noe de ikke klarer å svare på så bruker de enten tid på å finne frem til svaret sammen med eleven, eller så skriver de at det kan være andre elever i klassen som klarer å svare (positiv tilgjengelighet til hjelp).

Fem av respondentene har hatt positive mestringserfaringer med å tilrettelegge undervisningen. En respondent skriver at programmering er noe av det som er lettest å tilrettelegge fordi en kode alltid kan lages på mer eller mindre avanserte måter. Hen opplever også at elever med høy faglig mestring i stor grad utfordrer seg selv til å forbedre sine egne koder. Respondenten mener med andre ord at programmering er lett å tilrettelegge for alle elevene siden vanskelighetsgraden på koden lett kan endres. Dette var den eneste respondenten som eksplisitt skrev at det var enkelt å tilrettelegge for de sterke elevene. Tre respondenter skriver at de synes det enkleste er å tilrettelegge for elevene som synes programmering er vanskelig. Her trekker de frem at prosjekter kan forenkles for å tilpasse disse elevene. En respondent trekker også frem at hen ofte starter med å lage flytskjema på papir før de tar frem pc for å finne blokker (til blokkprogrammering) om elevene strever.

Når det kommer til å tilrettelegge undervisningen for elever med høy faglig mestring er det ikke like mange som har positive mestringsopplevelser slik som respondenten beskrevet over. Fire respondenter har svar som har blitt kodet til negative mestringserfaringer, og tre av dem har svart under tilrettelegging at de har problemer med å tilrettelegge undervisningen for elevene som viser stor grad av mestring. Ingen av respondentene skriver noe om hva de eventuelt har prøvd å gjøre for å tilrettelegge for denne elevgruppen, med unntak av én som skriver at hen finner vanskelige oppgaver til dem på internett som de får utforske på egenhånd.

Det er også to respondenter som har hatt negative mestringserfaringer når det kommer til å svare på spørsmål fra elevene. En av de skriver at hen har vært usikker på enkelte spørsmål

elevene har stilt, og at hen erfarer at mange elever på 7. trinn kan mer programmering enn hen selv kan. Selv innehar hen ikke kompetanse utover læreplanmålene for 7. trinn, så om elevene kan mer enn dette og stiller spørsmål rundt det, så må hen lese seg opp for å kunne svare. Den andre skriver først om positive mestringserfaringer, og at hen kan svare på spørsmål om bruken av ulike blokker og hjelpe elever med feilsøking og justering av kode. utfordringene har kommet når hen har prøvd å hjelpe elever med å lage programmer fra bunnen av, hvis de ikke har eksempler å ta utgangspunkt i.

Emosjonell tilstand

Emosjonell tilstand er respondentenes opplevelser av positive eller negative emosjoner i programmeringsundervisningen. Antall analyseenheter og respondenter med svar som er kodet til positiv og negativ emosjonell tilstand er vist i Tabell 7.

Tabell 7: Oversikt over antall analyseenheter og respondenter med svar kodet til emosjonell tilstand

Emosjonell tilstand	Positiv	Negativ	Totalt
Antall analyseenheter	2	0	2
Antall respondenter	2	0	2

Det var kun to svar fra to forskjellige respondenter som ble kodet til emosjonell tilstand, og begge ble kodet til positiv emosjonell tilstand. Det ene svaret var at en respondent skrev at det er gøy når hen selv og elevene får til og mestrer programmering, og dette ble tolket som en ganske tydelig emosjonell tilstand knyttet til programmeringsundervisningen. Den andre respondenten nevnte en positiv emosjonell tilstand da hen vurderte sine egne kunnskaper om programmering og programmeringsferdigheter. Hen skrev da at hen selv har interesse for programmering, og trakk dette frem som en av grunnene til at hen vurderte sine kunnskaper og ferdigheter som gode.

Forestilt didaktisk mestring

Forestilt didaktisk mestring er forestillingene respondentene har om at de har forstått eller ikke forstått hvordan de skal formidle det faglige innholdet til elevene. Antall analyseenheter og respondenter med svar som er kodet til positiv og negativ forestilt didaktisk mestring er vist i Tabell 8.

Tabell 8: Oversikt over antall analyseenheter og respondenter med svar kodet til forestilt didaktisk mestring

Forestilt didaktisk mestring	Positiv	Negativ	Totalt
Antall analyseenheter	6	3	9
Antall respondenter	4	3	7

Det var totalt fire respondenter som fikk noen av svarene sine kategorisert som positiv forestilt didaktisk mestring. Et svar handlet om en generell følelse av å mestre pedagogisk undervisning av prinsippene bak programmering, men denne respondenten skrev også at hen ønsket seg mer kunnskap om hvordan man kan bruke programmering til å løse faglige problemstillinger. Ellers var der noen respondenter som viste positiv forestilt didaktisk mestring når de skrev at de selv ikke klarte å hjelpe eleven. I stedet for å se på dette som en negativ situasjon, fokuserte de heller på hvordan det kunne være positivt for elevens læring. Det var for eksempel noen som trakk fram viktigheten av å la elevene være aktive i å lete etter svar hvis de lurte på noe som læreren selv ikke kunne svare på, og de skrev at det ligger mye god læring i å finne svar selv. En respondent skrev også at hen mener det er viktig å ha bevissthet rundt egen læring, fordi det gjør at en har litt flere verktøy når en skal hjelpe andre ut på nye ukjente områder.

Tre respondenter hadde svar som ble kodet til negativ forestilt didaktisk mestring, og alle handlet om at respondentene syntes det er vanskelig å knytte programmering til naturfaget. En respondent skrev at hen hadde bortimot ingen mestring når det kommer til å undervise i programmering i naturfag, og begrunnet det med at hen er usikker på hvordan hen skal ta i bruk programmering i naturfaget, annet enn å lage trinnvise instruksjoner for å gjenskape forsøk. En annen respondent svarte også at det var usikkerhet rundt å inkludere programmering i faget, men hen fokuserte på at det var vanskelig å gjøre det på en *spennende* måte. Hen har selv erfart, gjennom å undervise i programmering i matematikk, at mange av oppleggene som ligger tilgjengelig på nett ikke faller helt i smak for alle mellomtrinnelever.

Forestilt faglig mestring

Forestilt faglig mestring er forestillingene respondentene har om at de selv forstår det faglige innholdet de skal lære elevene. Antall analyseenheter og respondenter med svar som er kodet til positiv og negativ forestilt faglig mestring er vist i Tabell 9.

Tabell 9: Oversikt over antall analyseenheter og respondenter med svar kodet til forestilt faglig mestring

Forestilt faglig mestring	Positiv	Negativ	Totalt
Antall analyseenheter	10	14	24
Antall respondenter	7	5	9*

* En respondent kan ha både positive og negative utsagn. Derfor vil ikke totalt antall respondenter nødvendigvis være summen av antall respondenter med positive utsagn og antall respondenter med negative utsagn.

Forestilt faglig mestring er, sammen med mestringserfaringer, koden som fikk flest svar kodet til seg. Selv om det er syv respondenter som har totalt ti svar som har blitt kodet som positiv forestilt faglig mestring, så indikerer ikke svarene spesielt høy vurdering av egne faglige evner. Det var heller svar av typen «Har helt ok kunnskaper om grunnleggende blokkprogrammering» (Respondent 3) og «Helt grei kompetanse» (Respondent 6).

Når respondentene vurderte egne evner til å undervise i programmering i naturfag var det to respondenter som skrev at de mestrer det godt, og begrunner det blant annet med at de lærte dette under utdannelsen sin. Under vurdering av egne kunnskaper om programmering og programmeringsferdigheter er det også en respondent som skriver at hen har fått kursing i programmering gjennom et vitensenter og har «grei forståelse».

Fem respondenter har totalt 14 svar som har blitt kodet til negativ forestilt faglig mestring. Av de fem respondentene var det blant annet tre stykket som skrev at de ikke har nok kunnskaper eller ferdigheter til å lage programmer på egenhånd, og at de trenger hjelp fra eksterne kilder, for eksempel nettressurser. Blant svarene som ble plassert i denne koden, var det også noen svar som ikke veldig tydelig viste lav vurdering av faglige evner, men heller svar av typen «Ikke så veldig mye kunnskap om programmering» (Respondent 9) og «Programmering er ikke det jeg har mest kunnskap om» (Respondent 4).

Det var også fire respondenter som nevnte helt eksplisitt at de manglet kompetanse, og tre av dem var også tydelige på at de ønsker seg mer kompetanse. Respondent 3 svarte under «ekstraspørsmålet» at hen «Savner et løft for lærere som har jobbet lenge i skolen og ikke lært programmering som en del av utdanningen eller fått mulighet til å delta på kurs. Jeg mener at kompetanseheving bør gå hånd i hånd med målene i læreplanene». Ellers er det blant annet utfordringer rundt å planlegge og videreutvikle undervisningsopplegg og tilrettelegge undervisningen som kommer frem i svarene, eller mangel på kompetanse for å kunne bruke programmering som en større del av faget. For eksempel skriver respondent 6 «Kunne ønsket

meg mer kompetanse (mer erfaring) i å lage programmer selv som kan benyttes til å løse faglige problemstillinger.» Hen skriver at hen selv kun kan lage enkle koder, og ofte må bruke en oppskrift hentet fra internett om hen vil lage mer avanserte programmer.

Rammefaktorer

Rammefaktorer handler om de ytre faktorene som respondentene oppgir at påvirker programmeringsundervisningen. Antall analyseenheter og respondenter med svar som er kodet til positive og negative rammefaktorer er vist i Tabell 10.

Tabell 10: Oversikt over antall analyseenheter og respondenter med svar kodet til rammefaktorer

Rammefaktorer	Positiv	Negativ	Totalt
Antall analyseenheter	2	2	4
Antall respondenter	2	2	4

Det var to informanter som fikk svar kodet til positive rammefaktorer. Det ene svaret handler om at respondenten synes at vurdering av elevene går greit, og peker på læreplanen som grunnen. Hen skriver at kompetansemålene er såpass konkrete at det ikke er noe problem å skille de ulike måloppnåelsene. Det andre svaret er en respondent som skriver at blokkprogrammering gjør at det er lettere å oppnå mestring kjapt. Her kan det tenkes at respondenten tenker det er hensiktsmessig at elevene lærer blokkprogrammering framfor tekstprogrammering når de skal lære seg å programmere.

Det var også to informanter som hadde svar som ble kodet til negative rammefaktorer. Begge svarene handlet om vurdering av elever. Respondent 2 skrev at underveisvurdering er krevende fordi det er lite tid. Respondent 5 skrev at det er «Tidvis vanskelig å plukke opp hva elevene synes er vanskelig, når man har 20-25 elever i klassen. Det er lett å se de som tar det raskt og de som har store utfordringer. Men mellomtingen er vanskelig å alltid få med seg. vi forsøker gjennom dialog med enkeltelever og oppsummering i hele klassen, men trolig kommer ikke alt frem hver gang.»

Selvregulering

Selvregulering handler om respondentenes organisering av arbeid og tid. Antall analyseenheter og respondenter med svar som er kodet til positiv og negativ selvregulering er vist i Tabell 11.

Tabell 11: Oversikt over antall analyseenheter og respondenter med svar kodet til selvregulering

Selvregulering	Positiv	Negativ	Totalt
Antall analyseenheter	6	0	6
Antall respondenter	5	0	5

Det var 5 informanter som hadde svar som ble kodet til selvregulering, og alle ble kodet til positiv selvregulering. To av respondentene svarer at de synes det går greit å vurdere elevene i programmering fordi de har klare rammer de vurderer elevene etter. De trekker frem viktigheten av at elevene vet hvordan arbeidet deres skal vurderes. Begge skriver at de er tydelige med hva de forventer av elevene, og en skriver at hen setter vurderingskriterier sammen med i elevene i starten av et prosjekt. Andre respondenter peker på egne forberedelser til undervisningen som gjør at den går greit. Under tilrettelegging av undervisningen er det to respondenter som skriver at det går greit fordi de har forberedt seg og gjort klart ulike oppgaver som kan løses på ulike nivåer.

Tilgjengelighet til hjelp

Tilgjengelighet til hjelp handler om respondentenes opplevelser av at det er mulig eller ikke mulig å få hjelp om de trenger det. Antall analyseenheter og respondenter med svar som er kodet til positiv og negativ tilgjengelighet til hjelp er vist i Tabell 12.

Tabell 12: Oversikt over antall analyseenheter og respondenter med svar kodet til tilgjengelighet til hjelp

Tilgjengelighet til hjelp	Positiv	Negativ	Totalt
Antall analyseenheter	10	0	10
Antall respondenter	6	0	6

Det var seks informanter som hadde svar som ble kodet til tilgjengelighet til hjelp, og alle ble kodet til positiv tilgjengelighet til hjelp. Fem av informantene nevner eksplisitt å bruke elevene som hjelp, og det virker som at flere av dem oppfordrer elevene til å samarbeide og lære sammen. En respondent skriver at hen selv samarbeider med eleven for å finne svar sammen om det er noe hen ikke umiddelbart kan svare på. Andre respondenter skriver at de bruker elever som mestrer programmering godt til å hjelpe andre elever når de selv ikke klarer å hjelpe. En skriver om å bruke elever på denne måten fordi hen selv synes det er vanskelig å tilrettelegge undervisningen på grunn av mangel på kompetanse. Respondentene nevner også å bruke elever som mestrer faget godt som en ressurs for å håndtere spørsmål fra elever de

selv ikke kan svare på. De skriver at de har dyktige elever og at det som regel er noen andre i klassen som kan svare om det er spørsmål de selv ikke kan svare på.

I tillegg til hjelp fra elevene, er det noen respondenter som skriver at de bruker nettressurser som hjelp. Noen bruker nettressurser for å finne oppgaver som er delt inn i forskjellige vanskelighetsgrader for å tilrettelegge for elevene. Andre skriver at de henter ressurser fra forskjellige nettsider når de skal planlegge undervisningen.

Kognitiv selvmodellering

Kognitiv selvmodellering handler om at respondentene visualiserer at de klarer eller ikke klarer å gjennomføre programmeringsundervisning. Antall analyseenheter og respondenter med svar som er kodet til positiv og negativ tilgjengelighet til hjelp er vist i Tabell 13.

Tabell 13: Oversikt over antall analyseenheter og respondenter med svar kodet til kognitiv selvmodellering

Kognitiv selvmodellering	Positiv	Negativ	Totalt
Antall analyseenheter	0	1	1
Antall respondenter	0	1	1

Det var kun én respondent som hadde et svar som ble kodet til negativ kognitiv selvmodellering. Respondenten skrev at hen ikke mestrer å undervise i skriftprogrammering. Dette ble kodet som kognitiv selvmodellering fordi respondenten i et tidligere svar skrev at hen ikke hadde noen kunnskaper eller ferdigheter om tekstprogrammering. På bakgrunn av dette ble svaret om å ikke mestre å undervise i skriftprogrammering tolket som at hen ikke har prøvd å undervise i tekstprogrammering, men ser for seg at om hen ikke hadde fått det til om hen hadde prøvd.

Hvilke strategier oppgir respondentene at de bruker på tross av lav forestilt mestring?

De fleste respondentene hadde noen deler av undervisningen der de oppga at de hadde noe lavere mestring, men de fleste oppga også strategier de bruker for å løse eventuelle situasjoner de ikke mestrer. En av de strategiene som ble nevnt av flere av respondentene var å bruke elevene som ressurser. Elevene ble nevnt som mulige ressurser både til å svare på spørsmål fra andre elever som respondentene selv ikke kunne svare på, men og for å hjelpe andre elever med oppgaver som respondentene selv ikke klarte å hjelpe med. Flere respondenter oppga

også at de samarbeidet med elevene om det var noe de selv ikke klarte. De skrev for eksempel at de utforsker blokker (i blokkbasert programmeringsspråk) og ulike programmer sammen med elevene, eller at de leter etter svar i samarbeid med elevene om det er noe de ikke umiddelbart finner ut av.

En annen strategi som flere respondenter trakk frem var å bruke eksterne ressurser for å finne oppgaver eller lage undervisningsopplegg. De fleste som nevnte dette, nevnte å ta i bruk internettressurser. Noen av respondentene brukte disse ressursene for å planlegge undervisningen, da de ikke følte at de mestret å utvikle et undervisningsopplegg i programmering fra bunnen av på egen hånd. Da oppga de at de fant både ferdiglagde opplegg og eksempeloppgaver på internett. Noen brukte også internettressurser for å tilpasse undervisningen til elever som mestret programmering på et høyt nivå. En respondent skrev at hen gjorde dette da hen selv ikke mestret å tilrettelegge for disse elevene, så da finner hen vanskeligere oppgaver på internett som elevene får utforske på egenhånd.

En siste strategi som et par respondenter nevnte var å ta seg tid til å lese seg opp på et tema. En respondent nevnte det i sammenheng med å ta seg god tid til å forberede seg til undervisningen. Hen skrev at det var nødvendig for at hen skulle mestre å undervise i programmering. En annen respondent nevner det når hen skriver om å hjelpe elevene. Om hen får spørsmål fra elever hen ikke kan svare på, tar hen seg tid til å lese seg opp og undersøke nærmere, og så går hen tilbake til eleven for å hjelpe.

Flere av respondentene trakk også frem deler av undervisningen de ikke mestret, men uten å nevne strategier de bruker eller har prøvd for å forsøke og løse det. De fleste av respondentene som gjorde dette gjorde det da de skulle vurdere egne evner til å vurdere elevene. En respondent trakk eksplisitt frem underveisvurdering som krevende fordi hen ikke føler hen har tid til å sitte med vurdering underveis i et programmeringsprosjekt som ofte består av praktiske oppgaver. Andre trakk bare frem vurdering som generelt utfordrende uten å spesifisere det noe mer. Andre deler av undervisningen som respondentene skrev de syntes var vanskelig var å ha oppgaver som elevene synes er spennende, knytte programmering til naturfag og tilpasse undervisningen til elevene som mestret programmering på et høyt nivå. Så selv om noen av respondentene hadde strategier de brukte for å for eksempel tilrettelegge for elevene som mestret programmering godt, så var det ikke alle respondentene som oppga dette.

Diskusjon

I dette kapittelet blir resultatene fra analysen knyttet opp mot teori og empiri. Først diskuterer jeg hvordan respondentene beskriver sine opplevelser av mestring i programmeringsundervisning i naturfag. Videre går jeg inn på hva respondentene trekker frem som gir dem lav mestringstro, hvilke kompetanser de uttrykker at de trenger for å undervise i programmering, og hva dette kan ha å si for eventuell kompetanseheving for lærere. Etter dette diskuteres strategiene respondentene bruker i undervisningen på tross av lav forestilt mestring opp mot annen empiri. Kilder til mestringstro som ble funnet i analysen og kilder til mestringstro som ikke kom frem i analysen legges så fram, med refleksjoner om hva dette har å si og mulige grunner til at noen kilder til mestringstro kom tydeligere frem enn andre. Til slutt diskuterer jeg studiens begrensninger.

Hvordan beskriver lærerne sine opplevelser av mestring i programmeringsundervisning i naturfag?

Det er varierende grad av mestringstro blant respondentene i undersøkelsen, og dette kan muligens påvirke kvaliteten på undervisningen elevene får. Et par respondenter skiller seg ut ved å enten vurdere sine evner som lave innenfor alle delene av programmeringsundervisning, eller ved å vurdere sine evner som gode eller svært gode innenfor de fleste delene. Ellers har de aller fleste både positive og negative innspill når de vurderer sine egne evner innenfor de forskjellige delene av programmeringsundervisning i naturfag. Varierende mestringstro i programmeringsundervisning er også noe Hartell et al. (2019) fant i sin studie om læreres holdninger til programmeringsundervisning. Spørsmålet er i hvilken grad denne variasjonen i mestringstro påvirker kvaliteten på undervisningen elevene får. Siden læreres mestringstro handler om deres oppfatning av egen kompetanse heller enn deres faktiske kompetanse (Tschannen-Moran et al., 1998), så sier ikke mestringstroen deres i seg selv noe om nivået på undervisningen deres. Men siden mennesker har en tendens til å velge aktiviteter de vurderer at de mestrer heller enn aktiviteter de ikke tror de kommer til å mestre (Bandura, 1977), så skal vi ikke se bort i fra at varierende mestringstro blant lærere kan bety varierende tid brukt på programmeringsundervisning. Dette vil i så fall påvirke kvaliteten på undervisningen elevene får, siden de ikke vil få like mye tid til å utvikle sine programmeringsferdigheter med en lærer med lav mestringstro i programmeringsundervisning. Mestringstro er også kontekstspesifikt (Goddard et al., 2000; Hartell, 2017) og kan bli påvirket av ulike faktorer (Goddard et al., 2000). Derfor er det viktig å prøve å finne ut av hva som forårsaker den

varierende mestringstroen blant lærere. Om man klarer å finne noen spesifikke grunner til lav eller høy mestringstro i programmeringsundervisning hos naturfagslærere som går igjen hos flere lærere, er det også større sjanse for at man klarer å gjøre de rette tiltakene for å øke mestringstroen deres, og med det muligens øke kvaliteten på undervisningen til elevene.

Hva går igjen i svarene til respondentene som tyder på lav mestringstro?

En tydelig negativ påvirkning av mestringstroen til flere av respondentene i denne oppgaven, var nivået på deres egne kunnskaper om programmering. Påvirkningen av egne kunnskaper var synlig blant respondentene som vurderte sine egne kunnskaper som gode, men de var mest tydelig blant respondentene som oppga at de manglet kunnskaper. Blant respondentene kom dette spesielt frem ved at de ikke alltid klarer å hjelpe elever med spørsmål og oppgaver der de står fast, eller til og med en som har erfart at elevene kan mer programmering enn hen selv. At en lærer som har mer solid kunnskap om et fag har større tro på egne evner til å undervise i faget har også blitt funnet i andre studier, både blant lærere (Vinnervik, 2022) og blant lærerstudenter (Nilsson, 2008). Det var et par respondenter i undersøkelsen til denne oppgaven som skrev at de har lært om programmering under utdannelsen sin, og disse respondentene var blant de som vurderte sine evner innenfor de fleste delene av programmeringsundervisning som gode. Siden mestringstro handler om en persons oppfatninger av egne evner, og ikke det faktiske nivået på en persons kompetanse (Tschannen-Moran et al., 1998), er det ikke mulig å si noe om disse respondentenes faktiske kompetanse i programmeringsundervisning. Det er heller ikke mulig å si om dette er noe som gjelder flere, siden utvalget i denne oppgaven er for lite til å generalisere. Det er allikevel interessant, og kan være enda et tegn på at mestringstroen til lærere blir påvirket av kunnskapene de har i faget de skal undervise. Hvis dette stemmer, vil et viktig grep for å heve mestringstroen i programmeringsundervisning hos naturfagslærere være å gi dem en form for tilbud om kompetanseheving i programmering. Det neste spørsmålet blir da hvor omfattende denne kompetansehevingen bør være.

To av respondentene nevnte at de har lært om programmering i utdannelsen sin, og et par andre skrev at de har gått på ulike typer enkeltkurs. Om de resterende respondentene har fått noen form for kompetanseheving er uvisst, men når flere av dem nevner at de ønsker seg mer kompetanse så kan det tenkes at de enten ikke har fått noen form for kompetanseheving eller at det de har fått ikke har vært bra nok. Dette har også blitt funnet i Sverige, der majoriteten

av lærerne som ble berørt da programmering ble integrert i matematikk- og teknologiundervisningen hadde lite eller ingen tidligere erfaring med programmering (Vinnervik, 2022). Om forventningene i Fagfornyelsen om å utvikle norske elevers kompetanse i algoritmisk tenking skal oppfylles, trengs det en systematisk og gjennomgående kompetanseheving av lærere (Kravik et al., 2022). Det holder ikke at lærere får kompetanseheving gjennom enkeltkurs (Mannila et al., 2018). Kursene som noen av respondentene i undersøkelsen til denne oppgaven nevner at de har vært på, er slike enkeltkurs. Selv om noen av disse kursene får veldig gode tilbakemeldinger av lærere som har deltatt (super:bit, u.å.) så ville det vært interessant å se mer på langtidseffektene av disse enkeltkursene i programmering som tilbys norske lærere. For selv om lærerne er fornøyde med kurset, så mener både Kravik et al. (2022) og Mannila et al. (2018) at det krever mer enn enkeltkurs for å forbedre kompetansen til lærere. Det er også behov for flere studier som undersøker de langsiktige utfallene av kompetansehevende kurs for lærere for å øke mestringstro og kunnskap for å undervise i programmering (Mason & Rich, 2019). Det vil være interessant å se om lærerne burde fortsette å tilbys flere enkeltkurs, eller om det er behov for en mer systematisk og langsiktig kursing av lærerne. Det vil også være interessant å undersøke om lærere er motiverte for å delta på kurs, og i så fall hva som motiverer dem til å delta.

Det at det var flere respondenter som ønsket seg mer kompetanse og/eller kunnskaper når det kom til programmeringsundervisning, kan være et tegn på at de ville ha vært motiverte til å delta om de fikk tilbud om kompetansehevende kurs. Hvor motivasjonen kommer fra, er det ikke mulig å si noe om ut ifra respondentenes svar. Et ønske om å øke egen kompetanse kan tyde på at de ønsker å inkludere programmering i undervisningen sin på en bedre måte, noe som implisitt kan peke på at de ser at det er relevant å inkludere programmering i faget. Dette var også tydelig i Nilsson (2008) sin studie om lærerstudenters oppfatninger om utvikling av fagkunnskaper, der lærerstudentene var mer motiverte til å lære et fag når de følte at kunnskapen var meningsfull for dem og kunne brukes i deres fremtidige undervisning. Det er uenigheter rundt hvor relevant det er å integrere programmering i allerede eksisterende fag (Stigberg & Stigberg, 2020), og selv lærere som i utgangspunktet er enige i at programmering er relevant å lære elevene stiller spørsmål ved hvorfor programmering har blitt lagt til i en allerede overfylt læreplan (Hartell et al., 2019). På en annen side er det ikke mulig å si om det er relevansen til naturfaget som gjør at respondentene i undersøkelsen til denne oppgaven ønsker å øke sin kompetanse. Det kan hende at ønsket kommer av at de vet at de må ha

programmeringsundervisning i naturfag fordi programmering nevnes i et av kompetansemålene til naturfag i læreplanen. Så om respondentene ønsker mer kompetanse fordi det stilles nye krav til dem i undervisningen, eller om de faktisk ser det som relevant å integrere programmering i naturfagundervisningen, er ikke mulig å si ut ifra svarene til respondentene i denne undersøkelsen. Det som *er* mulig å si, er at flere av respondentene ønsker å få økt kompetansen sin i programmering. For å kunne tilby lærere kompetanseheving, og øke sjansen for at kompetansehevingen oppleves som relevant for deltakerne, er det viktig å undersøke nøyaktig hvilke kompetanser lærerne føler at de mangler.

Hvilke kompetanser uttrykker respondentene selv at de trenger for å undervise i programmering?

Noe flere respondenter ikke mestrer, er å bygge et program fra bunnen av. Dette trekkes blant annet frem av en respondent som skriver at hen sliter med å svare om elever kommer med spesifikke spørsmål om å lage programmer fra bunnen av. I Nilsson (2008) sin studie var en av tingene som var viktig for lærerstudenter å kunne ta tak i muligheter som dukket opp underveis i undervisningen, ved å diskutere og utforske elevenes egne idéer og tanker som dukket opp underveis. Det kan antas at dette er noe også erfarne lærere synes er viktig å inkludere i undervisningen. I en undervisningssituasjon der elevene skal lære å programmere, kan det å ikke mestre å bygge et program fra bunnen av være en hindring for nettopp dette. Da har læreren mindre mulighet for å hjelpe elever som har idéer om programmer de ønsker å bygge selv.

Å ha nok ferdigheter til å kunne endre på et program eller bygge et program på egenhånd, er viktig for at elever skal føle eierskap til prosjektene de jobber med i programmeringsundervisning (Lee et al., 2011). Selv om elevene i starten lærer å programmere ved å følge en oppskrift, vil det være hensiktsmessig å øke vanskelighetsgraden for å utfordre elevene. Å la elevene endre på et program eller bygge et program fra bunnen av på egenhånd kan være en god måte å øke vanskelighetsgraden på. En av respondentene i undersøkelsen til denne oppgaven skrev at hen finner vanskelige oppgaver på nett som de flinkeste elevene kan utforske på egenhånd fordi hen selv ikke mestrer å tilrettelegge programmeringsundervisningen til disse elevene. Hva slags oppgaver dette er kommer ikke frem, men det kan tenkes at disse oppgavene gir elevene flere muligheter til å lage egne programmer. Problemene kan komme om elevene står fast og læreren ikke kan hjelpe dem

videre. Når vanskelighetsgraden til oppgavene øker, er det både viktig at de utfordres nok til at de ikke kjeder seg, men ikke så mye at de ikke mestrer oppgavene og får negative erfaringer (Lee et al., 2011). Spørsmålet blir dermed om lærerne som finner vanskeligere oppgaver på nett klarer å finne oppgaver i riktig vanskelighetsgrad for elevene. Og om oppgavene er for vanskelige og elevene ikke mestrer dem, blir det neste spørsmålet: hva slags hjelp læreren kan tilby elevene for å komme videre når læreren i utgangspunktet ikke føler at de mestrer å tilrettelegge undervisningen for disse elevene.

I tillegg til å ikke kunne bygge programmer fra bunnen av generelt, var det en respondent som eksplisitt skrev at hen ønsket seg mer kompetanse i å lage egne programmer som kan benyttes til å løse faglige problemstillinger. Hen skrev at hen trenger en oppskrift for å lage slike programmer, eller for å lage programmer som kan brukes faglig i naturfag og matematikk. Spesifikt hva slags problemer det er snakk om, kommer ikke frem, men det viser at respondenten har et ønske om å kunne bruke programmering som en større del av faget, heller enn å bare undervise i programmering som et eget tema uten å knytte det til fagene i større grad. Det å synes det er vanskelig å bruke programmering til å løse problemer er også noe som har kommet frem i en annen studie, der seksti prosent av respondentene følte seg veldig usikre når det kom til dette (Mannila et al., 2018). Det denne respondenten trekker frem, viser en annen grunn til at det kan være viktig å ha kompetanse til å kunne lage programmer fra bunnen av. For i tillegg til at læreren får flere muligheter til å hjelpe elever og tilrettelegge undervisningen, kan denne kompetansen brukes til å knytte programmeringen til flere deler av naturfaget. For selv om «programmering» kun nevnes eksplisitt i ett kompetansemål i naturfag (Utdanningsdirektoratet, 2020), er det flere kompetansemål som kan oppnås ved å ta i bruk programmering, for eksempel å jobbe med modeller og systemer. Det kan antas at en lærer med mer solid kompetanse innenfor programmering har større muligheter til å inkludere programmering som en større del av naturfaget heller enn at programmeringsundervisningen blir et eget tema som i mindre grad knyttes til andre naturfaglige tema.

Hvilke strategier har respondentene på tross av lite forestilt mestring?

Respondentene i undersøkelsen viser til flere strategier de tar i bruk på tross av lite forestilt mestring. Disse strategiene er blant annet å bruke andre elever som hjelp når de ikke selv kan hjelpe, og å ta i bruk ressurser fra internett. Nilsson (2008) fant ut at lærerstudenter kunne føle på frustrasjon og ubehag om elever stilte spørsmål de ikke kunne svare på. Thorsnes et al.

(2020) undersøkte også hvordan lærere håndterte elevspørsmål, og fant ut at lærerne var positive til at de kom til å klare å løse det selv om de ikke klarte å svare på spørsmål fra elever med en gang. Respondentene i undersøkelsen til denne oppgaven svarte også at de ikke alltid klarte å svare på spørsmål fra elever, men på samme måte som lærerne i Thorsnes et al. (2020) sin studie så virket det ikke som at respondentene syntes dette var et spesielt stort problem. Det virker heller som at de for det meste finner løsninger om det er noe de føler de mangler kunnskaper om eller spørsmål de ikke kan svare på. De nevner for eksempel at de leser seg opp og undersøker for å kunne hjelpe elevene, eller at det ofte er andre elever som klarer å svare på spørsmål de selv ikke klarer å svare på. Det kan tenkes at en av grunnene til at respondentene i undersøkelsen virker å håndtere dette greit, er at de har lengre erfaring enn lærerstudentene i Nilsson (2008) sin undersøkelse, og har utviklet noen strategier for å håndtere slike situasjoner. Det kan også forklare forskjellene mellom respondentene i Nilsson (2008) og Thorsnes et al. (2020) sine studier, siden en har respondenter som er lærerstudenter og en har respondenter som er lærere. Det kan allikevel være nyttig å sammenlikne erfaringene til lærere med erfaringene til lærerstudenter når det er snakk om at lærere skal undervise i noe nytt som de kanskje ikke har så mye erfaring med å undervise i fra før.

Selv om flere av respondentene virker å være positive til å bruke elevene som hjelp med det de ikke selv kan svare på, så er det ikke sikkert at dette bare er positivt. En studie fra Norge har også funnet at noen lærere er avhengige av hjelp fra elever når de skal undervise i programmering (Kravik et al., 2022). Kravik et al. (2022) skriver også at det er et av tegnene på at lærerne trenger mer kompetanseheving. Det samme kan sies om respondentene i undersøkelsen til denne oppgaven. Selv om det virker som at de fleste av respondentene som nevner at de bruker elever som hjelp skriver om det som noe positivt, så kan man ikke se bort ifra at grunnen til at de trenger hjelpen noen ganger er at de selv mangler kompetanse. I tillegg til kompetanseheving, er det viktig at lærere har selvtillit til å utforske på egenhånd hva som er relevant og hvordan de skal inkludere digital kompetanse i undervisningen sin (Mannila et al., 2018). Flere av respondentene viser en slik vilje til å finne ut av ting på egenhånd, for eksempel de som undersøker nærmere for å kunne svare elever som spør om noe de ikke kan eller finner ressurser på internett for å hjelpe med å planlegge og tilrettelegge undervisningen. Allikevel er det flere som skriver at de ønsker seg mer kompetanse. Så selv om en del av respondentene virker å ha både selvtillit og vilje til å utforske på egenhånd, så er det et ønske og en følelse av behov for kompetanseheving i tillegg.

Når det kommer til å ta i bruk ressurser på internett, er dette også noe andre studier har funnet at lærere gjør for å finne inspirasjon for å undervise i programmering (Stigberg & Stigberg, 2020). Stigberg og Stigberg (2020) fant også at til tross for at lærere brukte internett for å finne inspirasjon, så var det ingen av lærerne i deres studie som brukte et kurs som var utviklet av Skolverket (en del av det svenske utdanningsdirektoratet) og spesifikt rettet mot å undervise programmering i skolene. Av respondentene i undersøkelsen til denne oppgaven som skriver at de har tatt i bruk ressurser fra internett, så skriver en at hen har tatt i bruk code.org og to skriver at de har tatt i bruk Lær Kidsa Koding. Ellers er det usikkert hvor respondentene henter inspirasjon og ressurser fra. En positiv side ved å ha ressurser lett tilgjengelig vil naturligvis være at selv lærere som selv ikke føler de har nok kompetanse til å utvikle et undervisningsopplegg på egenhånd kan få hjelp til det. Det er også, slik som noen respondenter selv også trekker frem, en enkel måte å finne alternative oppgaver for elever som for eksempel har behov for større utfordringer.

En tenkt ulempe ved å bruke ressurser fra internett hyppig er at lærerne kan bli veldig låste til undervisningsoppleggene de finner på internett, spesielt hvis de i liten grad klarer å videreutvikle dem selv. Vannatta og Nancy (2004) påpeker at det er viktig å ta seg tid til å leke med og utforske teknologi på egenhånd om man skal utvikle kompetansen sin og ta i bruk mer teknologi i klasserommet. Spørsmålet er hvor mye av sin egen tid respondentene bruker på å utforske programmering på egenhånd om de kun bruker eksterne ressurser. På en annen side skrev også flere respondenter at de bruker tid på å utforske, finne løsninger og lære sammen med elevene sine. Vannatta og Nancy (2004) nevnte at villigheten til å lære mer om teknologi på egenhånd kan komme til syne ved at lærerne er villige til å bruke tid utenfor sin vanlige arbeidstid til å forberede undervisning. Hvor mye tid respondentene til denne oppgaven bruker utenfor sin arbeidstid til å planlegge undervisning er uvisst. Men det kan argumenteres for at det er like greit at de bruker tiden på å utforske og lære sammen med elevene. På den måten er de flere som tenker sammen, og elevene får også se at læreren ikke alltid sitter med svarene. Det siste var noe som det også virket som at var viktig for flere av respondentene. De påpekte at elevene ikke skulle se på dem som en som sitter på fasiten, og at det heller skulle være fokus på å lære sammen.

Kilder til mestringstro som ble funnet i analysen

Mestringserfaringer er den største kilden til mestringstro blant respondentene. Flertallet av respondentene nevner positive mestringserfaringer de har opplevd i undervisningssituasjoner. At mestringserfaringer er noe respondentene selv trekker frem som viktig stemmer godt overens med Bandura (1997) sin teori om mestringstro og kilder til mestringstro, da han mener at mestringserfaringer er den mest innflytelsesrike kilden til mestringstro. Noe som er litt interessant, er at noen av respondentene nevner erfaringer de ikke har mestret, men at noe har påvirket situasjonen positivt, slik at de mestrer den allikevel. Et eksempel er respondentene som skriver at det ikke nødvendigvis er et problem om det er spørsmål de ikke kan svare på, siden det ofte er andre elever som kan svare. Med andre ord var det en situasjon de i utgangspunktet ikke mestret, men på grunn av tilgjengelighet til hjelp, i dette tilfellet fra elever så ble det allikevel en positiv erfaring. Dette viser hvordan andre kilder til mestringstro, i dette tilfellet tilgjengelighet til hjelp, også kan påvirke mestringstroen positivt selv om en annen av kildene til mestringstro står i fare for å bli påvirket negativt.

Den andre kilden til mestringstro som er tydelig blant respondentene er forestilt faglig mestring. Flertallet av svarene som er kodet til forestilt faglig mestring er kodet til negativ, og av de som er kodet til positiv forestilt faglig mestring, er det mange av som ikke indikerer spesielt høy vurdering av egne faglige evner. Med andre ord viser ikke disse resultatene at mange av respondentene har høyere mestringstro *fordi* de har høy forestilt faglig mestring, men det at de selv nevner det såpass hyppig viser at det er viktig for dem. Altså kan man anta at de hadde hatt høyere mestringstro hvis de også hadde hatt høyere forestilt faglig mestring. Det var Palmer (2006) som argumenterte for at forestilt faglig mestring og forestilt didaktisk mestring er kilder til mestringstro i tillegg til de fire kildene Bandura (1997) definerte. Han fant at det var forestilt didaktisk mestring som var en av de viktigste kildene til mestringstro hos sine respondenter. En av grunnene til at kildene til mestringstro er annerledes hos respondentene i denne oppgaven kan være hvem respondentene er. Palmer (2006) undersøkte mestringstro hos lærerstudenter. Derfor kan det igjen tenkes at forskjellen ligger i hvilke erfaringer lærere har i forhold til lærerstudenter. Igjen vil jeg trekke frem eksempelet med respondentene til Nilsson (2008) og Thorsnes et al. (2020) som viser at lærerstudenter og lærere hadde forskjellige måter å håndtere spørsmål fra elever som de ikke klarte å svare på. Det kan igjen tenkes at forskjellen ligger i at lærerne har mer trening i det didaktiske enn det lærerstudentene har. Mens lærerstudentene kanskje er mest bekymret for den forestilte didaktiske mestringen, om de føler de har forstått *hvordan* de skal lære bort fagstoffet på best

mulig måte, så har erfarne lærere dette allerede litt mer under huden. Dermed er det heller den forestilte faglige mestringen som kommer tydelig frem hos dem, og spørsmålet om de har forstått det faglige innholdet godt nok til å lære det bort. Dette kan peke på at kompetansehevende kurs i programmering for lærere burde fokusere mest på det faglige, og å øke generell kunnskap om og ferdigheter i programmering, og litt mindre på hvordan man burde lære det bort.

Selv om respondentene gav mer uttrykk for at de manglet faglig mestring enn didaktisk mestring, så var det noe som gikk igjen blant de som følte på negativ forestilt didaktisk mestring. Det som gikk igjen her, var en følelse av at de var vanskelig å knytte programmering til naturfag. Det er anbefalt at lærere får en grundig opplæring for å lære hvordan de skal integrere programmering i undervisningen sin når programmering legges til i en ny læreplan (Bocconi et al., 2018), og andre studier har vist at lærere i andre land ikke har fått denne opplæringen etter at programmering har blitt implementert i de nye læreplanene der (Stigberg & Stigberg, 2020; Vinnervik, 2022). Funnene i denne oppgaven kan tyde på det samme, i hvert fall blant respondentene, siden flertallet av dem uttrykker at de mangler faglig kunnskaper om programmering, kunnskaper om hvordan de skal knytte det til naturfag, eller begge deler.

Det som ellers er tydelig fra resultatene, er at respondentene sin mestringstro blir påvirket av ulike ting. Selv om det er mestringstro som ifølge Bandura (1997) er den mest innflytelsesrike kilden til mestringstro, og også den som dukket opp oftest i analysen av datamaterialet, så viser resultatene også at det er flere ting som er viktig for respondentene for at de skal ha tro på at de klarer å gjennomføre undervisning i programmering i naturfag. Noe som er interessant å merke seg når det kommer til forestilt faglig mestring og forestilt didaktisk mestring, er at Palmer (2006) argumenterer for at de kan være forutsetninger for mestringserfaringer for lærere. Han begrunner det med at det er viktig for lærere å ha god kunnskap om faglig innhold og metoder for å undervise om man skal lykkes i en undervisningssituasjon. Han argumenterte også for at å *forstå* faglig innhold og didaktiske metoder i seg selv kunne regnes som mestringserfaringer, siden respondentene hans virket å få høyere mestringstro kun gjennom dette. Sånn sett kan man også si om respondentene i denne oppgaven at mer av det de skriver om egentlig er mestringserfaringer. Noen av svarene som for eksempel er kodet til selvregulering, kunne også ha blitt kodet til mestringserfaringer fordi respondentene skriver at det går greit å vurdere elevene, altså at de har hatt positive

mestringserfaringer når det kommer til å vurdere elevene. Jeg vil allikevel argumentere for at det har kommet frem andre sider av datamaterialet når det har blitt analysert med utgangspunkt i flere kilder enn de fire Bandura (1997) definerte. Hovedgrunnen til dette er at man får noe mer detaljert oversikt over hva som ligger bak respondentenes mestringstro. Det viktige er å prøve å finne ut av hva som skal til for å gi lærere høyere mestringstro. Mennesker har en tendens til å velge aktiviteter de tror de kommer til å mestre (Bandura, 1997). Dermed kan det antas at en lærer med høyere mestringstro i å undervise i programmering i naturfag også vil bruke mer tid på å gjøre nettopp dette. Og det kan igjen føre til flere positive mestringserfaringer, som igjen vil øke mestringstroen.

Kilder til mestringstro som ikke ble funnet i analysen

Det var tre overordnede koder som ikke ble brukt i analysen fordi det ikke ble funnet noe i datamaterialet som passet til disse kodene: fremtidsperspektiv, sosial påvirkning og vikarierende erfaringer. Vikarierende erfaringer ble funnet siden det ble funnet et eksempel på kognitiv selvmodellering, og dette er en form for vikarierende erfaring (Bandura, 1997). Allikevel var det ene svaret såpass kort, og det kom ikke frem andre former for vikarierende erfaringer i analysen. Derfor var det ingen eksempler på hvordan vikarierende erfaringer har vært med på å øke mestringstroen til respondentene. At flere overordnede koder ikke ble brukt i analysen, betyr ikke nødvendigvis at respondentene ikke får økt mestringstro fra kildene som kodene er laget ut ifra. Det kan heller bety at det ikke var det respondentene anså som mest relevant da de svarte på spørreundersøkelsen. Hadde man hatt mulighet til å stille oppfølgingsspørsmål eller stilt flere spørsmål som mer konkret ba om eksempler fra de ulike kildene til mestringstro, kunne man kanskje ha funnet mer detaljer om alle kildene til mestringstro.

Begrensninger

En begrensning med studien til denne oppgaven, er at det ikke var mulig å be respondentene om mer utdypning av svarene sine i spørreundersøkelsen, og dermed var svarene ikke like detaljerte som de mulig ville ha vært om man hadde hatt mulighet for dette. Dette var en konsekvens av ønsket om å ha et større antall respondenter og se om det var svar som gikk igjen hos flere heller enn å gå i dybden hos noen få respondenter. Allikevel har ikke studien nok respondenter til å kunne generalisere svarene. Dette blir en annen begrensning med studien, for resultatene sier kun noe om hva akkurat respondentene som svarte på

undersøkelsen mener. Tid er også en tydelig begrensning for studien, fordi det er en masteroppgave som kun strekker seg over ett semester. Det er begrenset hvor mye data man kan samle og behandle i løpet av tiden satt av til arbeidet med oppgaven.

Oppsummering og konklusjon

Hensikten med denne masterstudien var å undersøke læreres mestringstro i programmeringsundervisning i naturfag. For å undersøke dette utarbeidet jeg to forskningsspørsmål:

- Hvordan beskriver respondentene sine opplevelser av mestrings i programmeringsundervisning i naturfag?
- Hvilke strategier oppgir respondentene at de bruker på tross av lav forestilt mestring?

Resultatene viser at det er varierende grad av mestringstro i programmeringsundervisning i naturfag blant respondentene, noe som også har blitt funnet i andre studier (Hartell et al., 2019). Et par respondenter skiller seg ut ved å enten vurdere evnene sine som lave innenfor alle delene av programmeringsundervisning, eller ved å vurdere evnene sine som høye innenfor alle delene av programmeringsundervisning. Ellers har respondentene både deler av programmeringsundervisningen de føler seg sterke i, og deler de ikke føler seg så sterke i.

Noe som går igjen som en negativ påvirkning på respondentenes mestringstro er mangel på kunnskaper om og kompetanse i programmering. At kunnskaper om faget man skal undervise i er viktig for å ha større tro på at man klarer å undervise i faget, er også noe andre studier har funnet, både hos lærere (Vinnervik, 2022) og lærerstudenter (Nilsson, 2008). Mangelen på kunnskaper påvirker blant annet respondentenes vurderinger av egne evner til å svare på spørsmål fra elever og tilrettelegge undervisningen, spesielt for elevene med høy faglig mestring. Det tyder på at respondentene har behov for mer kursing og oppfølging i programmering. Dette behovet for en mer gjennomgående kompetanseheving av lærere som skal undervise i programmering har også andre studier funnet (Kravik et al., 2022; Mannila et al., 2018; Vinnervik, 2022).

Jeg tenker at en negativ konsekvens av at det er varierende mestringstro blant respondentene kan være at det blir varierende kvalitet på undervisningen, som igjen kan føre til varierende kompetanse blant elevene. Når man i tillegg ser antydningene til at forestilt faglig mestring er viktig for mestringstro blant lærere, er det bekymringsfullt at det virker å være mangel på omfattende kompetanseheving og oppfølging av lærere som skal undervise i programmering. Om alle elever skal få like muligheter, trenger også lærere kompetanseheving slik at de har kunnskapene de trenger for å kunne undervise i alle temaene på læreplanen.

Til tross for at en del respondenter har lav forestilt mestring innenfor noen deler av programmeringsundervisningen, har flere av dem strategier de tar i bruk for å løse situasjoner de i utgangspunktet ikke mestrer. Strategiene flest respondenter nevnte, er å bruke andre elever som hjelp når de selv ikke kan hjelpe, og å bruke ressurser fra internett. Spesielt det å bruke andre elever som hjelp virket å være noe flere respondenter var positive til, og de virket generelt ikke bekymret om det var spørsmål fra elever de selv ikke kunne svare på. Også lærerne i Thorsnes et al. (2020) sin studie var lite bekymret for at de ikke klarte å svare på spørsmål fra elevene med en gang. Studien til Kravik et al. (2022) peker på at lærere som er avhengig av hjelp fra elever når de skal undervise i programmering er et tegn på at de trenger kompetanseheving, og dette kan også sies om respondentene i denne masterstudien. Allikevel virker det som at respondentene ofte ser på elever som viktige ressurser i undervisningen, og legger heller vekt på viktigheten av å utforske og lære sammen.

En tanke om hvorfor respondentene virker å være positive til å se på elever som ressurser og lære sammen med elevene, er fokuset på *utforskning* i skolen. I læreplanen i naturfag (Utdanningsdirektoratet, 2020), under fagets relevans og sentrale verdier, står det at elevene gjennom naturfag skal bli forberedt på et arbeids- og samfunnsniv som vil stille krav til en utforskende tilnærming og teknologisk kompetanse. Under kjerneelementer teknologi står det også at elevene skal tenke kreativt og nyskapende gjennom å bruke og skape teknologi. Jeg tenker at man ikke klarer å oppnå målene om utforskning, kreativitet og nyskaping med mindre man oppfordrer elever til å samarbeide og finne svar sammen, heller enn at det står en lærer med alle svarene. Og ut ifra svarene til respondentene, kan det tenkes at de tenker det samme, siden de virker positive til å lære sammen og ikke se på seg selv som en som sitter med alle svarene på alt.

Avslutningsvis ønsker jeg å vise til studien til Thorsnes et al. (2020), som skriver at resultatene til studien deres kan hjelpe med å lette sinnet til fremtidige lærere som er bekymret for programmeringsferdighetene sine. De fant nemlig at selv om mange lærere i studien deres vurderte programmeringsferdighetene sine som relativt lave, så var de positive til at de kom til å klare og undervise i programmering. Selv om resultatene av denne masterstudien bare kan si noe om respondentene som bidro til studien, så kan det virke som at de peker i samme retning. For selv om flere av respondentene følte at de manglet kompetanse i programmering, og ønsket å øke kompetansen sin, så hadde de fleste noen områder av

programmeringsundervisning der de vurderte sine evner som gode. Og om det var områder der de ikke vurderte sine evner som gode, så hadde flere av dem strategier de tok i bruk som virket å gjøre det enklere for dem å mestre programmeringsundervisningen.

Forslag til videre forskning

Resultatene av denne studien viser at det er behov for mer forskning om hvilke kompetansehevende kurs i programmering som lærere har deltatt på, og hva langtidseffektene av disse kursene er. Det kan også være nyttig å undersøke læreres forståelse av programmering i naturfaget, og gå mer i dybden på hva lærerne selv tenker de må kunne for å undervise i programmering i naturfag. Da vil man også få et innblikk i hva de tenker elevene må lære i programmering i naturfag for å oppfylle kompetansemålene i læreplanen. Flere studier kan også gjøres for å undersøke kilder til mestringstro hos lærere, og hvilke kilder som er viktigst for å øke læreres mestringstro.

Litteraturliste

- Bakken, J. & Andersson-Bakken, E. (2021). Innholdsanalyse. I E. Andersson-Bakken & C. P. Dalland (Red.), *Metoder i klasseromsforskning* (s. 305-326). Universitetsforlaget.
- Bandura, A. (1977). Self-efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, 84(2), 191-215. <https://doi.org/10.1037/0033-295X.84.2.191>
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. W. H. Freeman.
- Bandura, A. (2006). Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents*, 5(1), 307-337.
https://books.google.no/books?hl=no&lr=&id=P_onDwAAQBAJ&oi=fnd&pg=PA307&dq=bandura+1997&ots=rjPJp-GiAM&sig=CXoa2l3qdGqnpjADCEcv38iLlbs&redir_esc=y#v=onepage&q=bandura%201997&f=false
- Bocconi, S., Chiocciariello, A., Earp, J. & Group, N. B. S. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education: Report prepared for the Nordic@ BETT2018 Steering Group: Nordic@ BETT2018 Steering Group. <https://doi.org/10.17471/54007>
- Butz, A. R. & Usher, E. L. (2015). Salient sources of early adolescents' self-efficacy in two domains. *Contemporary Educational Psychology*, 42, 49-61.
<https://doi.org/10.1016/j.cedpsych.2015.04.001>
- Cohen, L., Manion, L. & Morrison, K. (2018). *Research methods in education* (8. utg.). routledge.
- Fagerlund, J., Leino, K., Kiuru, N. & Niilo-Rämä, M. (2022). Finnish teachers' and students' programming motivation and their role in teaching and learning computational thinking. *Frontiers in Education*, 7. <https://doi.org/10.3389/feduc.2022.948783>
- Finger, G. & Houguet, B. (2009). Insights into the intrinsic and extrinsic challenges for implementing technology education: Case studies of Queensland teachers. *International journal of technology and design education*, 19, 309-334.
<https://doi.org/10.1007/s10798-007-9044-2>
- Goddard, R. D., Hoy, W. K. & Hoy, A. W. (2000). Collective teacher efficacy: Its meaning, measure, and impact on student achievement. *American educational research journal*, 37(2), 479-507. <https://doi.org/10.3102/00028312037002479>
- Hartell, E. (2017). Teachers' self-efficacy in assessment in technology education. *Springer international handbook of education: Handbook of technology education*, 785-800.
https://doi.org/10.1007/978-3-319-44687-5_56
- Hartell, E., Doyle, A. & Gumaelius, L. (2019). Teachers' attitudes towards teaching programming in Swedish Technology education. *PATT* 37, 195-202.
https://research.edgehill.ac.uk/ws/portalfiles/portal/20775897/PATT_37_Malta2019_Proceedings.pdf#page=195
- Hartell, E., Gumaelius, L. & Svärth, J. (2015). Investigating technology teachers' self-efficacy on assessment. *International journal of technology and design education*, 25, 321-337. <https://doi.org/10.1007/s10798-014-9285-9>
- Heintz, F., Mannila, L. & Farnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. 2016

- IEEE Frontiers in Education conference (FIE).
<https://ieeexplore.ieee.org/document/7757410>
- Hsieh, H.-F. & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative health research*, 15(9), 1277-1288.
<https://doi.org/10.1177/1049732305276687>
- Johannessen, A., Tufte, P. A. & Kristoffersen, L. (2005). *Introduksjon til samfunnsvitenskapelig metode* (3. utg.). Abstrakt forlag.
- Kert, S. B. (2019). A Proposal of In-Service Teacher Training Approach for Computer Science Teachers. *European Journal of Educational Research*, 8(2), 477-489.
<https://doi.org/10.12973/eu-jer.8.2.477>
- Kravik, R., Berg, T. K. & Siddiq, F. (2022). Teachers' understanding of programming and computational thinking in primary education—A critical need for professional development. *Acta Didactica Norden*, 16(4), 23 sider.
<https://doi.org/10.5617/adno.9194>
- Kunnskapsdepartementet. (2017). *Digitaliseringstrategi for grunnsopplæringen 2017-2021*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/framtid-fornyelse-og-digitalisering/id2568347/>
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg.). Gyldendal.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32-37.
<https://doi.org/10.1145/1929887.1929902>
- Mannila, L., Nordén, L.-Å. & Pears, A. (2018). Digital competence, teacher self-efficacy and training needs. *Proceedings of the 2018 ACM Conference on International Computing Education Research*, 78-85. <https://doi.org/10.1145/3230977.3230993>
- Mason, S. L. & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 19(4), 790-824. <https://www.learntechlib.org/p/184723/>
- Nardi, P. M. (2018). *Doing survey research: A guide to quantitative methods*. Routledge.
- Nilsson, P. (2008). *Learning to teach and teaching to learn: Primary science student teachers complex journey from learners to teachers* [Linköping University Electronic Press].
<https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A223795&dswid=8262>
- NOU 2013: 2. (2013). *Hindre for digital verdiskaping* K.-o. distriktsdepartementet.
<https://www.regjeringen.no/no/dokumenter/nou-2013-2/id711002/?ch=2>
- NOU 2015: 8. (2015). *Fremtidens skole - Fornyelse av fag og kompetanser*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001/>
- Opplæringslova. (1998). *Lov om grunnskolen og den videregående opplæringa* (LOV-1998-07-17-61). Lovdata. <https://lovdata.no/lov/1998-07-17-61>
- Palmer, D. H. (2006). Sources of self-efficacy in a science methods course for primary teacher education students. *Research in Science Education*, 36(4), 337-353.
<https://doi.org/10.1007/s11165-005-9007-0>
- Riggs, I. M. & Enochs, L. G. (1990). Toward the development of an elementary teacher's science teaching efficacy beliefs instrument. *Science education*, 74(6), 625-637.
<https://eric.ed.gov/?id=ED308068>

- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G. & Voll, L. O. (2016). *Teknologi og programmering for alle: En faggjennomgang med forslag til endringer i grunnopplæringen*. Utdanningsdirektoratet. <https://www.udir.no/globalassets/filer/tall-og-forskning/forskningsrapporter/teknologi-og-programmering-for-alle.pdf>
- 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>
- Stigberg, H. & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy futures in education*, 18(4), 483-496. <https://doi.org/10.1177/1478210319894785>
- super:bit. (u.å.). *Over 13.000 har sagt sin mening - elever og lærere mener super:bit er utrolig bra*. super:bit. <https://www.superbit.no/hva-er-superbit/superbit-fungerer/>
- Thorsnes, J., Rouhani, M. & Divitini, M. (2020). In-service teacher training and self-efficacy. *Informatics in Schools. Engaging Learners in Computational Thinking: 13th International Conference, ISSEP 2020, Tallinn, Estonia, November 16–18, 2020, Proceedings 13*, 158-169. https://doi.org/10.1007/978-3-030-63212-0_13
- Tschannen-Moran, M., Hoy, A. W. & Hoy, W. K. (1998). Teacher efficacy: Its meaning and measure. *Review of educational research*, 68(2), 202-248. <https://doi.org/10.3102/00346543068002202>
- Utdanningsdirektoratet. (2017). *Overordnet del - verdier og prinsipper for grunnopplæringen*. <https://www.udir.no/lk20/overordnet-del/?lang=nob>
- Utdanningsdirektoratet. (2019, 27.03.2019). *Algoritmisk tenkning*. Utdanningsdirektoratet. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2020). *Læreplan i naturfag (NAT01-04)*. <https://www.udir.no/lk20/nat01-04?lang=nob>
- Utdanningsdirektoratet. (2021, 01.03.2021). *Kompetansepakke for programmering og algoritmisk tenkning*. Utdanningsdirektoratet. <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/kompetansepakke-for-programmering-og-algoritmisk-tenkning/>
- Vannatta, R. A. & Nancy, F. (2004). Teacher dispositions as predictors of classroom technology use. *Journal of Research on Technology in Education*, 36(3), 253-271. <https://doi.org/10.1080/15391523.2004.10782415>
- Vinnervik, P. (2022). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *International journal of technology and design education*, 32(1), 213-242. <https://doi.org/10.1007/s10798-020-09602-0>
- Aalderen-Smeets, S. v. & Molen, J. W. v. d. (2013). Measuring Primary Teachers' Attitudes Toward Teaching Science: Development of the Dimensions of Attitude Toward Science (DAS) Instrument. *International Journal of Science Education*, 35(4), 577-600. <https://doi.org/10.1080/09500693.2012.755576>

Vedlegg

Vedlegg 1 – spørreundersøkelse til masterprosjektet



Undersøkelse til masterprosjekt

Dette er en undersøkelse til et masterprosjekt på grunnskolelærerutdanningen 1-7.

I undersøkelsen vil du bli bedt om å vurdere egne evner knyttet til ulike områder innenfor programmeringsundervisning i naturfag. Det ønskes svar fra alle lærere som underviser i naturfag på 1.-7. trinn, uavhengig om de har erfaring med å undervise i programmering eller ikke.

Du velger selv hvor utfyllende du ønsker å svare, men svar gjerne med hele setninger. Kom gjerne med eksempler hvis du føler at det er hensiktsmessig for å utdype svaret ditt, men husk å ikke oppgi personidentifiserende informasjon eller taushetsbelagte opplysninger.

Ved å sende inn svarene dine på dette spørreskjemaet samtykker du til at jeg kan bruke svarene i forskning. Svarene dine er anonyme med mindre du selv oppgir personspeifikke opplysninger. Jeg kommer uansett ikke til å offentliggjøre personidentifiserende resultater fra undersøkelsen.

Dette prosjektet har blitt vurdert og godkjent av Sikt (tidligere NSD).

Er du lærer på 1.-7. trinn?

Jeg er lærer på 1.-7. trinn

Har du lest og forstått informasjonen over?

Jeg har lest og forstått informasjonen over

Undersøkelse til masterprosjekt

På hvilke trinn har du erfaring med å undervise i naturfag?

(Flere valg er mulig)

1.-4. trinn

5.-7. trinn

Har du undervist i programmering?

Ja

Nei

I hvilke fag har du undervist i programmering?

Dette elementet vises kun dersom alternativet «Ja» er valgt i spørsmålet «Har du undervist i programmering?»

(Flere valg er mulig)

Naturfag

Matematikk

Annet

I hvilke andre fag har du undervist i programmering?

Dette elementet vises kun dersom alternativet «Annet» er valgt i spørsmålet «I hvilke fag har du undervist i programmering?»

På hvilke trinn har du undervist i programmering?

Dette elementet vises kun dersom alternativet «Ja» er valgt i spørsmålet «Har du undervist i programmering?»

(Flere valg er mulig)

1.-4. trinn

5.-7. trinn

Kunnskaper om programmering og programmeringsferdigheter

Her ønsker jeg at du skal vurdere dine egne kunnskaper om programmering og dine programmeringsferdigheter. Her tenker jeg både på det du føler at du mestrer og det du føler at du ikke mestrer.

Undervisning i programmering i naturfag

Her ønsker jeg at du skal vurdere din evne til å undervise i programmering i naturfag. Her tenker jeg både på det du føler at du mestrer og det du føler at du ikke mestrer.

Planlegging og videreutvikling av undervisningsopplegg

Her ønsker jeg at du skal vurdere din evne til å planlegge og videreutvikle et undervisningsopplegg i programmering i naturfag. Her tenker jeg både på undervisningsopplegg du har laget selv, og undervisningsopplegg du har lånt fra andre.

Elevspørsmål

Her ønsker jeg at du skal vurdere din evne til å svare på spørsmål elevene kommer med under programmeringsundervisningen i naturfag.

Tilrettelegging av programmeringsundervisningen

Her ønsker jeg at du skal vurdere din evne til å tilrettelegge programmeringsundervisningen i naturfag, både for elever som synes programmering er vanskelig og for elever som mestrer programmering godt.

Vurdering av elever

Her ønsker jeg at du skal vurdere din evne til å vurdere elevene i programmering i naturfag. Her tenker jeg både på undervisningsvurdering og sluttvurdering

Ekstra

Er det noe mer du ønsker å legge til om programmeringsundervisning i naturfag, som du ikke syntes passet inn i de foregående spørsmålene?

Vedlegg 2 – Vurdering fra Sikt



[Meldeskjema](#) / [Læreres mestringstro i programmeringsundervisning i naturfag](#) / Vurdering

Vurdering av behandling av personopplysninger

Referansenummer 968101	Vurderingstype Standard	Dato 09.02.2023
----------------------------------	-----------------------------------	---------------------------

Prosjekttittel

Læreres mestringstro i programmeringsundervisning i naturfag

Behandlingsansvarlig institusjon

OsloMet – storbyuniversitetet / Fakultet for lærerutdanning og internasjonale studier / Institutt for grunnskole- og faglærerutdanning

Prosjektansvarlig

Per Øyvind Sollid

Student

Mai Sigrid Willoch

Prosjektperiode

01.01.2023 – 01.08.2023

Kategorier personopplysninger

Alminnelige

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 01.08.2023.

[Meldeskjema](#)

Kommentar

OM VURDERINGEN

Sikt 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.

KOMMENTARER TIL INFORMASJONSSKRIVET

Informasjonsskrivet ditt mangler noen punkter loven krever er med. Du må derfor legge til disse punktene i informasjonsskrivet før du gir dette til forskningsdeltakerne dine. Du trenger ikke å laste opp den oppdaterte versjonen i meldeskjemaet:

- Opplys forskningsdeltakerne at prosjektslutt er 01.08.2023 og at alle personopplysninger vil bli slettet innen den datoen.

DATABEHANDLER

Vi legger til grunn at behandlingen oppfyller kravene til bruk av databehandler, jf. personvernforordningen art. 28 og 29.

TAUSHETSPLIKT

Forskningsdeltagerne har yrkesmessig taushetsplikt. De kan ikke dele taushetsbelagte opplysninger med forskningsprosjektet. Vi anbefaler at du minner dem på taushetsplikten. Merk at det ikke er nok å utelate navn ved omtale av elever, pasienter el. Vær forsiktig med bruk av eksempler og bakgrunnsopplysninger som tid, sted, kjønn og alder.

FØLG DIN INSTITUSJONS RETNINGSLINJER

Vi har vurdert at du har lovlig grunnlag til å behandle personopplysningene, men husk at det er institusjonen du er ansatt/student ved som avgjør hvilke databehandlere du kan bruke og hvordan du må lagre og sikre data i ditt prosjekt. Husk å bruke leverandører som din institusjon har avtale med (f.eks. ved skylagring, nettspørreskjema, videosamtale el.)

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).

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. Se våre nettsider om hvilke endringer du må melde: <https://sikt.no/melde-endringer-i-meldeskjema>

1/2

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!