

MASTEROPPGAVE

Masterstudium i skolerettet utdanningsvitenskap med fordypning i naturfag Mai 2022

Tittel:

En kvalitativ studie for å få frem hvordan
programmering kan tilrettelegge for læring i naturfag

A qualitative study to find out how
programming can facilitate learning in science

Navn:

Kristin Ausland



OsloMet – storbyuniversitetet

Fakultet for lærerutdanning og internasjonale studier

Institutt for grunnskole- og faglærerutdanning

Sammendrag

Hensikten med denne studien er å få innblikk i hvordan programmering kan tilrettelegge for elevers læring av teknologi. Forskningsspørsmålene er: «*Hvilke kunnskaper og ferdigheter tilknyttet kjerneelementet teknologi uttrykker elevene?*» Og «*Hvordan arbeider elevene med programmeringsoppgaven, og hvordan tilnærmer de seg eventuelle utfordringer som oppstår?*».

Utgangspunktet for undersøkelsen er et undervisningsopplegg av Skaperskolen. Det består av programmering av micro:bit som en modell for en hjemmeruter. Undervisningsopplegget ble gjennomført i tre ungdomsskoleklasser. Undersøkelsen består av fokusgruppeintervjuer og observasjoner av elever som deltok i gjennomføringene. Analyseverktøyet består av kjerneelementet *teknologi* og de kompetansemålene undervisningsopplegget dreier seg om. I tillegg ble to rammeverk for algoritmisk tenkning brukt. Disse er *den algoritmiske tenkeren* (Utdanningsdirektoratet, 2019a) og *fasetter av algoritmisk tekning* (Shute, Sun, & Asbell-Clarke, 2017).

Hovedfunnene fra undersøkelsen er at et undervisningsopplegg hvor programmering inngår, ser ut til å kunne tilrettelegge for læring tilknyttet teknologi. Dette baserer seg på at elevene blant annet får arbeide med det teknologiske systemet og modellering ved hjelp av programmering. Elevene fikk også øvd på problemløsning, særlig feilsøking og koding. Et annet funn er at elevenes forkunnskaper ser ut til å ha stor betydning for hva elevene får arbeidet med, og hvordan de går frem i problemløsningen. Forkunnskapene er hovedsakelig relatert til det teknologiske systemet og grunnleggende programmeringskompetanse. For å få frem potensialet som ligger i bruk av programmering, er det noen elementer som kan støtte elevenes læring. Samarbeid er en form støtten kan bli gitt gjennom. Programmering kan på en enkel måte organiseres som pararbeid. For at samarbeidet skal fungere på en god måte, er det tydelig at parsammensetningen gjøres slik at elevene har kompetanse som kan brukes i arbeidet. I utvalget var det elever med ulik erfaring i programmering. Både de med lite og de med mer erfaring så ut til å ha utbytte av undervisningen. Læreren kan også støtte elevene, men fordrer at læreren har tilstrekkelig programmeringskompetanse.

Ifølge kjerneelementet *Teknologi* skal elevene *bruke* programmering i arbeidet med naturfag. For at programmering skal bli et slikt verktøy til å lære naturfag, er det nødvendig at elevene får utviklet grunnleggende ferdigheter i programmering. I tillegg vil elevene ha behov for tilstrekkelig støtte.

Abstract

The aim for this Master's thesis is to obtain knowledge regarding how programming can facilitate students' learning of technology. This is performed by answering the following research questions: “What knowledge and skills related to the core element named *Technology* in science do the students express?” and “How do the students work with programming, and how do they approach challenges that may appear?”.

The starting point for this research is the teaching arrangement prepared by “Skaperskolen”. It contains programming of micro:bit as a model for an internet router. This teaching arrangement was carried out at three different middle school classes in the eastern side of Norway. The survey consists of interviews in groups and observations of the participating students. The analysis tool applied is the core element named *Technology* in science and the competence objectives for the teaching arrangement. In addition, two frameworks for algorithmic thinking were used. These are *Demystifying computational thinking* (Shute, Sun, & Asbell-Clarke, 2017) and *den algoritmiske tenkeren* (Utdanningsdirektoratet, 2019a).

The main findings from the research work made in the process of writing this Master's thesis is that programming in science education has a great potential for learning. The students got to work with both the technological system and modeling through the programming arrangement. Also troubleshooting and coding were skills the student got occasions to try. Although it appeared from the analysis that the preliminary knowledge of the students has a great significance. This had an impact on what tasks the students were able to solve and how they did approach the problems that occur during the process.

Simultaneously, the effect of prior knowledge related to the technological system and the programming concepts seem to support the students' learning process. As the students are co-operating in pairs, it is found that the competence of the students affects whether the process of learning programming is working well. It is also possible for the teacher to support the students, but doing so depends on an adequate level of programming competence of the teacher. To make programming a tool to learn science, the students need sufficient prior knowledge, time, clear structure, support and instructions.

Forord

I mål med denne masteroppgaven, er det seks år siden jeg startet på grunnskolelærerutdanningen på OsloMet. En passende overskrift for denne perioden av mitt liv er «Den store reisen», for trolig vil dette være en av de periodene som jeg har utviklet meg mest gjennom livet. Jeg sitter igjen med kunnskap og erfaringer jeg vil ta med meg inn i læreryrket. Noe av motivasjonen med å skrive denne masteroppgaven var å bli bedre forberedt til å møte lærerhverdagen, og å bli en lærer med god faglig tyngde. Disse målene vil jeg si at jeg har nådd. Jeg sitter igjen med innsikt jeg ikke hadde fra før, og jeg føler meg tryggere på å kunne undervise i programmering og ellers i naturfag. Jeg vil påstå at jeg i løpet av studietiden både har blitt utdannet og dannet, og det er tydelig at denne perioden har hatt stor innvirkning både for min profesjonelle og personlige identitet. Min store reise blir nå kronet med å fullføre denne masteroppgaven. Det er jeg både takknemlig og stolt over å ha klart!

Noe jeg har fått erfare, er at Vygotskys (1987) læringsteori ser ut til å stemme. Det er mange ting som blir lettere i fellesskap med andre. I dette fellesskapet må noen spesielle trekkes frem:

Tusen takk til min masterveileder, Katarina Pajchel. Samarbeidet med deg har vært både lærerikt, hyggelig og inspirerende. Takk for at du har vært så tilgjengelig og hjelpelig underveis i arbeidet. Din kompetanse og ditt engasjement hadde jeg definitivt ikke klart meg uten.

Takk til dere i forskningsprosjektet TRELIS, for at jeg fikk være en del av deres arbeid.

Takk til lærerne og elevene som bidro i datainnsamlingen. Det har gitt innsikt og erfaringer som både var vesentlig for denne oppgaven, og nyttig for min fremtidige praksis.

Tusen takk til min medstudent og gode venn, Martin Løvdahl. Det har vært veldig hyggelig å få dele masterhverdagen med deg. Du har gjort dette til minneverdige og gode dager!

Ellers ønsker jeg å takke venner og familie for alle kaffekopper, sjokolade, gode klemmer og annen ytre motivasjon og omsorg som dere har gitt meg underveis i arbeidet med oppgaven.

Takket være alle disse har jeg nå til slutt kommet i mål med denne store reisen, noe som underveis har virket helt umulig. Her passer det å sitere en av mine forbilder, Dag Otto Lauritzen: «Ingenting er umulig. Det umulige tar bare litt lenger tid» (Eide, Lauritzen, & Bernitz, 2022).

Innholdsfortegnelse

1. Introduksjon	9
1.1 Hensikt og forskningsspørsmål	10
2. Teori og tidligere forskning	13
2.1 Algoritmisk tenkning i grunnskolen	13
2.2 Programmering i læreplanen.....	14
2.2.1 Kompetanse	14
2.2.2 Kjerneelementet teknologi	15
2.2.3 Modeller og modellering	16
2.2.4 Programmering	17
2.2.5 Fysisk programmering og micro:bit	19
2.2.6 Algoritmisk tenkning.....	20
2.3 Læring i teknologi	23
2.3.1 Støttestrukturer	23
2.3.2 Kunnskapsintegrasjon.....	25
2.3.3 Begrepslæring og språklige ferdigheter	25
2.4 Oppsummering teori.....	27
2.5 Tidligere forskning	28
2.5.1 Forskning på naturfag og programmering	29
2.5.2 Forskning relatert til fysisk programmering.....	31
2.5.3 Forskning hvor problemløsning tilknyttet programmering er undersøkt	32
2.5.4 Forskning på lærerens og medelevers betydning for læring	34
2.5.5 Oppsummering tidligere forskning	37
3. Metode.....	38
3.1 Undervisningopplegget Internett fanger	38
3.2 Utvalg	41
3.3 Gjennomføring av datainnsamling	43
3.3.1 Pilotering.....	43
3.3.2 Fokusgruppeintervjuer	44
3.3.3 Observasjon av elevgrupper	46
3.4 Bearbeiding av datamaterialet.....	49
3.4.1 Transkribering.....	49
3.4.2 Operasjonalisering av rammeverk	50

3.4.3	Analyse.....	53
3.5	<i>Etiske vurderinger</i>	56
3.6	<i>Vurderinger av studien</i>	57
4.	Resultater	61
4.1	<i>Observasjon av gjennomføringene</i>	61
4.1.1	Klasse A.....	61
4.1.2	Klasse B.....	62
4.1.3	Klasse C.....	63
4.2	<i>Elevenes kunnskaper og ferdigheter</i>	65
4.2.1	Kunnskaper tilknyttet kjerneelementet teknologi.....	65
4.2.2	Kunnskaper tilknyttet programmering.....	67
4.2.3	Elevenes problemløsning.....	72
4.3	<i>Elevenes arbeid med programmeringsoppgaven og utfordringer</i>	75
5.	Diskusjon	78
5.1	<i>Forkunnskapenes betydning</i>	78
5.1.1	Forkunnskaper modeller.....	79
5.1.2	Forkunnskaper programmering.....	83
5.2	<i>Elevenes problemløsning</i>	87
5.2.1	Elevenes tilnærming til feilsøking.....	87
5.2.2	Koding.....	89
5.3	<i>Støtte i elevers læring</i>	91
5.3.1	Samarbeid mellom elever.....	91
5.3.2	Undervisningsmateriell.....	94
5.3.3	Støtte fra læreren.....	95
5.4	<i>Metodediskusjon og videre forskning</i>	96
6.	Konklusjon	99
7.	Litteraturliste	101
8.	Vedlegg	107
	<i>Vedlegg 1 Godkjenning NSD</i>	107
	<i>Vedlegg 2 Informasjonsskriv og samtykkeskjema</i>	110
	<i>Vedlegg 3 Intervjuguide</i>	115

<i>Vedlegg 4 Shutes fasetter</i>	121
<i>Vedlegg 5 Resultat tabeller</i>	122

1. Introduksjon

Det samfunnet vi lever i er komplekst og i stadig utvikling, noe som gjør at vi i fremtiden vil kunne møte på ukjente og komplekse problemstillinger. Dette kan løses ved bruk av såkalt algoritmisk tenkning. Dette er et komplekst begrep, og består av blant annet ferdigheter som evne til å evaluere, logisk tenkning og strategivalg. Dette er ifølge Sevik (2016) en sentral kompetanse for det 21. århundret. En del av dette begrepet er også å forstå hvordan digital teknologi fungerer, og å bruke det til å løse problemer. Dette vil elevene trenge i fremtidig arbeids- og hverdagsliv (Buitrago Flórez et al., 2017; Nygård, 2018). Digital teknologi baserer seg på programmering, og det å forstå hvordan denne teknologien fungerer og påvirker oss er også en del av algoritmisk tenkning (Wing, 2006). Fremtidens problemløserer er de som sitter på skolebenken i dag, og det er derfor nyttig å starte opplæringen av algoritmisk tenkning allerede her. Det er derfor ikke uten grunn at det er blitt en internasjonal trend å innføre dette i skolen, også i Skandinavia (Bocconi et al., 2022). Måten begrepet forstås i LK20 er en form for problemløsning. Den norske skolen har valgt å tilrettelegge for opplæring i algoritmisk tenkning ved å la det inngå i noen av de allerede eksisterende fagene (Bocconi et al., 2022).

I den nye læreplanen, Fagfornyelsen, er ett av kjerneelementene i naturfag *Teknologi*. Noe av det sentrale som inngår i dette kjerneelementet, er at elevene skal *bruke* teknologi når de arbeider i faget. Dette omhandler både programmering og modellering (Utdanningsdirektoratet, 2020a). Det er matematikkfaget som har fått hovedansvaret for å lære elevene kunnskaper og ferdigheter i programmering. Av kompetansemålene innebærer dette grunnleggende programmeringskonsepter, slik som variabler, løkker og funksjoner. I løpet av opplæringen i faget, skal elevene bruke dette til å lære i faget (Utdanningsdirektoratet, 2020b). Denne kompetansen er det videre ment at elevene skal bruke i arbeid i andre fag. Læreplanen legger altså opp til at matematikk og naturfag skal samarbeide. Derfor vil det være nødvendig å undersøke hvordan dette kan gjøres, og hvordan programmering kan tilrettelegge for læring i naturfag.

Kompetansemålene for fagene er satt opp med en tydelig progresjon. Kompetansemålene på de ulike trinnene bygger på hverandre, og baserer seg på at eleven har nådd kompetansemålene i løpet av grunnskoleopplæringen. Programmering er et forholdsvis nytt tema i grunnskolen, noe som gjør situasjonen spesiell både for elever og lærere. Alle elever skal lære og bruke programmering, men nå i starten av innføringen i skolen vil elever bli

introdusert for programmering på ulike trinn. De som nettopp har startet på skolen vil ha hele grunnskoleopplæringen på seg til å utvikle programmeringskompetanse, mens elever på høyere trinn vil både få en introduksjon som de nesten samtidig skal bruke det i arbeid med annet naturfaginnhold. Elevene vil derfor ha ulike utgangspunkt og forutsetninger for å lære gjennom programmering. Dette tilsier at fagenes måte å arbeide med programmering vil spille inn på hverandre. Selv om det er matematikk som egentlig skal stå for opplæringen av programmering, vil det være en fordel å støtte opp om opplæringen av dette også i naturfag.

Av lærerne som arbeider i skolen i dag, er det mange som ikke kan programmering, selv om dette er noe de skal undervise i (Moreau, 2021). Mange lærere har ikke nok kompetanse. Lærere trenger å utvikle kompetanse til å kunne undervise i programmering (Bocconi et al., 2022; Mishra & Koehler, 2006; Moreau, 2021). Når elever og læreres programmeringsferdigheter i fremtiden er mer innarbeidet, vil situasjonen sannsynligvis være annerledes. Da vil innkjøringsfasen være over, og da vil fagene i større grad fokusere på nettopp å *forstå, skape og bruke* programmering slik det er formulert i kjerneelementet *Teknologi* (Utdanningsdirektoratet, 2020a). Ut ifra at vi i dag er i en situasjon hvor det er mange elever og lærere som ikke har kompetanse i programmering, vil det være relevant å undersøke hvordan elevene bruker programmering og hva som skal til for å kunne bruke det sli det er tenkt i læreplanen. Det er her relevansen for denne masteroppgaven kommer inn.

1.1 Hensikt og forskningsspørsmål

Programmering er nylig innført og er ment å være noe elevene skal bruke i arbeidet med naturfaglig innhold. Derfor er det relevant å undersøke hvordan dette kan gjøres med tanke på elevens læring, det kan svare på et behov i skolen. Dette er konkretisert ved å ha et særlig fokus på et undervisningsopplegg med tema teknologi.

Hensikten med min studie er derfor å undersøke hvordan et undervisningsopplegg med programmering tilrettelegger for læring om teknologi i naturfag.

Det er formulert to forskningsspørsmål som studien forsøker å besvare:

RQ1: Hvilke kunnskaper og ferdigheter tilknyttet kjerneelementet *teknologi* uttrykker elevene?

RQ2: Hvordan arbeider elevene med programmeringsoppgaven, og hvordan tilnærmer de seg eventuelle utfordringer som oppstår?

Dette blir en empirisk studie, og det kan bidra til å dekke noen av behovene for forskning på feltet. Ifølge Grover og Pea (2013b) er det aktuelt å undersøke hvordan undervisning i realfagene bør mot hvordan det kan gjøres for å tilrettelegge for læring. Som nevnt er programmering nylig innført i skolen. Derfor er det nødvendig med ytterligere forskning på utfordringer tilknyttet undervisning av naturfaginnhold gjennom programmering. Programmering er noe elevene skal arbeide med iblant annet naturfag. Lærere har generelt lite kompetanse i programmering og hvordan det kan undervises i (Moreau, 2021).

Tidsaspektet i lærerens arbeidshverdag gjør at det er relevant å undersøke et utarbeidet undervisningsopplegg. Det kan videre revideres og brukes som et eksempel på hvordan programmering kan implementeres. I tillegg kan utfordringer som generelt kan oppstå i undervisning i programmering komme frem. Dermed kan lærere få innspill til hva de bør være oppmerksom på i planlegging og gjennomføring av undervisning. Derfor er min studie relevant. Det er nødvendig å se på elever på ulike alderstrinn og komme med måter å tilnærme seg disse utfordringene. Skaperskolen har grunnet pandemien ikke fått testet ut opplegget selv. Resultater fra studien kan være nyttige i arbeidet med å videreutvikle undervisningsopplegget. Studien kan med andre ord være et verdifullt bidrag i den helhetlige forskningen på programmering i skolen, og med tanke på min egen fremtidige lærerpraksis.

Studien inngår i forskningsprosjektet TRELIS. Undervisningsopplegget som er undersøkt i forbindelse med masteroppgaven, inngår i et større undervisningsopplegg utarbeidet av Skaperskolen. Denne organisasjonen er en av TRELIS' samarbeidspartnere.

Undervisningsopplegget til Skaperskolen heter *Internett fanger*, og tar for seg det teknologiske systemet Internett. En del av dette er en programmeringsaktivitet med micro:bit, og det er denne delen av *Internett fanger* som er blitt undersøkt i denne studien. I undervisningsopplegget skal elevene skal bruke programmering til å lage en modell for hvordan hjemmerutere fungerer. I kap. 3.1 beskrives undervisningsopplegget mer inngående. Undervisningsopplegget er valgt ut ifra hensikten med studien, hvor både teknologiske system og programmering er sentralt.

Metodene som er brukt for å undersøke hvordan undervisningsopplegget tilrettelegger for læring, er observasjoner og fokusgruppeintervjuer med totalt 24 elever. Utvalget er hentet fra tre klasser på to ungdomsskoler på Østlandet. Måten datamaterialet er analysert og bearbeidet

er inspirert av både innholdsanalyse og tematisk analyse. Forskningsspørsmålene besvares gjennom en sammenstilling av observasjons- og intervjudata.

Struktur på oppgaven

Videre følger en gjennomgang av relevant teori og tidligere forskning. Deretter et metodekapittel hvor undersøkelsen og utvalget beskrives og vurderes, samt hvordan analysen av datamaterialet er blitt gjort. I metodekapittelets avsluttende del, blir etiske vurderinger og vurdering av studien redegjort for. I resultatkapittelet presenteres de viktigste funnene fra studien, før disse diskuteres opp mot relevant teori og tidligere forskning. Diskusjonen inneholder også en drøfting av måten studien er blitt gjennomført. Avslutningsvis gis det noen refleksjoner omkring implikasjoner, og forslag til videre forskning.

2. Teori og tidligere forskning

2.1 Algoritmisk tenkning i grunnskolen

Det moderne samfunnet vi lever i er i stor grad preget av digital teknologi, som er bygd opp ved bruk av programmering. Det er derfor relevant å tilrettelegge for at elevene forstår hvordan denne teknologien fungerer og påvirker oss. Prinsipper som inngår i den digitale teknologien kan bli utnyttet i kreativ problemløsning, og inngår i begrepet *algoritmisk tenkning* (Israel, Pearson, Tapia, Wherfel, & Reese, 2015). Programmering legger til rette for at elevene får utviklet problemløsning, analytisk og logisk tenkning, som ifølge Balanskat mfl. (2017) inngår i ferdigheter for det 21. århundret. Med dette menes det at tenkemåten som brukes i programmering, vil være til nytte i møte med de komplekse utfordringene som verden vil stå overfor i fremtiden (Lodi & Martini, 2021; NOU 2020:2, 2020; Wing, 2006).

Algoritmisk tenkning og programmering er innført læreplaner internasjonalt, også i Fagfornyelsen som gjelder for skolene i Norge (Bocconi et al., 2022). Noe av hensikten med å implementere teknologi i skolen, er å gi elevene verktøy slik at de skal kunne forstå og være aktive medborgere i det teknologiske samfunnet de er en del av (Wing, 2006). I forkant av Fagfornyelsen ble det publisert en forskningsrapport, *Teknologi og programmering for alle* (Sanne et al., 2016). Den består av anbefalinger til hvordan dette temaet bør implementeres i skolen. De påpeker nødvendigheten av å gjøre elevene i stand til å forstå og utnytte teknologien de omgir seg med. Dette inngår i teknologisk allmenndannelse. Rapporten får frem programmeringens overføringsverdi. Algoritmisk tenkning er noe elever kan få med seg fra arbeidet med programmering. Det blir gitt en grundigere beskrivelse av begrepet algoritmisk tenkning i delkapittel 2.2.6.

I den norske læreplanen, LK20, inngår programmering i flere av fagene. Det er som nevnt i introduksjonen matematikkfaget som har fått hovedansvaret for å lære elevene programmering. Dette skal de videre bruke i arbeidet i andre fag (Utdanningsdirektoratet, 2020b). Ett av de fagene dette gjelder, er naturfag. I læreplanen for naturfag er det fem hovedområder, såkalte kjerneelementer, som undervisningen skal dreie seg om.

Programmering inngår i et av disse, nemlig *Teknologi*. Dette blir uttrykt på følgende måte: «*Elevene skal forstå, skape og bruke teknologi, inkludert programmering og modellering, i arbeid med naturfag. Gjennom å bruke og skape teknologi kan elevene kombinere erfaring og faglig kunnskap med å tenke kreativt og nyskapende*» (Utdanningsdirektoratet, 2020a). Det er

blitt lagt merke til at i både dette kjerneelementet og kjerneelementet *naturvitenskapelige praksiser og tenkemåter*, står det eksplisitt at skal brukes i kombinasjon med de andre kjerneelementene. Dette får frem at programmeringen blir ansett som noe som har overføringsverdi til andre naturfaglige temaer.

Til hvert av de ulike kjerneelementene, er det kompetansemål som er ment til å støtte disse hovedområdene av faget. De kompetansemålene for 10.årstrinn, som gjelder for kjerneelementet *Teknologi* er: «*elevene skal kunne utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker*» og «*bruke programmering til å utforske naturfaglige fenomener*» (Utdanningsdirektoratet, 2020a).

Det er i dette delkapittelet blitt redegjort for hvordan algoritmisk tekning er blitt implementert i læreplanen som styrer innholdet av opplæringen skal foregå i grunnskolen. Videre blir temaer som er relatert til programmering i læreplanen beskrevet.

2.2 Programmering i læreplanen

Hensikten med denne studien er å undersøke hvordan et undervisningsopplegg hvor programmering legger til rette for læring om teknologi. Med utgangspunkt i oppgavens forskningsspørsmål skal det nå redegjøres for sentrale begreper som inngår.

2.2.1 Kompetanse

I det første forskningsspørsmålet dreier det seg om elevenes kunnskaper og ferdigheter. I Fagfornyelsen er dette en del av begrepet kompetanse. Kompetanse er et samlebegrep bestående av komponentene *kunnskap, forståelse, ferdigheter, egenskaper, holdninger og verdier* (NOU 2020:2, 2020).

I LK20 deles begrepet kompetanse inn i *kunnskap* og *ferdigheter*. Kunnskap defineres som «*å kjenne til og forstå fakta, begreper, teorier, ideer og sammenhenger innenfor ulike fagområder og temaer*». Ferdigheter innebærer å «*beherske handlinger eller prosedyrer for å utføre oppgaver eller løse problemer, og omfatter blant annet motoriske, praktiske, kognitive, sosiale, kreative og språklige ferdigheter*» (Kunnskapsdepartementet, 2017).

Den kompetansen elevene er ment å utvikle i løpet av grunnskoleopplæringen, er uttrykt i fagenes kompetansemål. Disse er ment å støtte fagenes kjerneelementer, som utgjør det mest sentrale innholdet i fagene (Utdanningsdirektoratet, 2019b). Videre vil ett av kjerneelementene i naturfag, nemlig *Teknologi* beskrives. Det er dette kjerneelementet undervisningsopplegget denne studien tar utgangspunkt i dreier seg om.

2.2.2 Kjerneelementet teknologi

Teknologi utgjør ett av kjerneelementene i læreplanen for naturfag. I dette hovedområdet skal elevene forstå, skape og bruke teknologi i naturfagsarbeidet (Utdanningsdirektoratet, 2020a). Kjerneelementet *Teknologi* skal som nevnt i introduksjonen, skal brukes i arbeidet med de andre kjerneelementene. Ett av kjerneelementene er *naturvitenskapelige praksiser og tenkemåter*. Ifølge Voll og Sollid (2022) kan naturvitenskapelig arbeid kombinert med teknologi gjøre at arbeidet blir utforskende. Teknologi er gjenstander og systemer som er skapt av mennesker og kunnskapen om disse (Sanne et al., 2016). Hovedhensikten med å lage denne teknologien er å løse problemer og produsere noe nytt. I utformingen av teknologien, utnyttes ulike naturvitenskapelige prinsipper, og er basert på naturvitenskapelig kunnskap (Voll, 2019). Det finnes ulike typer teknologi, og digital teknologi er en av dem. Grunnlaget for digital teknologi er programkode som utføres av en digital enhet (Voll, 2022). Dette inngår i programmering. Bruk av både programmering og modellering inngår i kjerneelementet teknologi. Modellering blir ytterligere redegjort for i neste kapittel.

Ut ifra ett av kompetansemålene som er tilknyttet kjerneelementet teknologi, skal elevene arbeide med teknologiske systemer (Utdanningsdirektoratet, 2020a). Det er derfor naturlig før vi går videre, å kort forklare hva begrepet system vil si. Et system består av mindre deler som henger sammen på en måte som gjør at systemet fungerer og får den funksjonen den har. De ulike delene er vesentlige. Det er ulike måter å tilnærme seg systemer når disse skal arbeides med. Dersom systemet undersøkes ved å se på de ulike delene hver for seg, er det en form for tradisjonell analyse. Ved en systemtenkningstilnærming, betraktes systemet som en helhet. Her er det sentralt å se på sammenhengene mellom de ulike delkomponentene. Her blir det tydelig at endringer i en del av systemet, vil ha innvirkning på de andre delene og helheten. Da vil kompleksiteten bli tydeliggjort. Systemtenkning er en tilnærming som kan være nyttig i arbeid med komplekse problemer (Aronson, 1996).

Dette kan blant annet gjøre ved å arbeide med konkrete teknologiske produkter (Svensson, 2011). Et eksempel på et slikt produkt er en hjemmeruter. Den brukes for å skape forbindelser mellom ulike digitale enheter, som datamaskiner og smarttelefoner. Forbindelsene danner et lokalt nettverk (Store norske leksikon, 2020). Flere slike nettverk er koblet sammen med hverandre, både trådløst og gjennom kabler, til større nettverk. Alle er sammenkoblet til et verdensomspennende nett, som kalles Internett (Øverby & Dvergsdal, 2021).

Teknologi kan blitt tatt i bruk og skapes gjennom kreativ og nyskapende tenkning. Her vil å erfaring og faglig kunnskap også være nyttig å implementere (Utdanningsdirektoratet, 2020a).

2.2.3 Modeller og modellering

Modeller inngår som en sentral del av naturfag, og er spesifikt nevnt i både kjerneelementet *teknologi og naturvitenskapelige praksiser og tenkemåter* (NPT). Modeller og arbeid med å lage dem skal bidra til at elevenes naturfaglige forståelse utvikles (Utdanningsdirektoratet, 2020a). Å utvikle og bruke modeller inngår som en av åtte praksiser for naturfagene (National Research Council, 2012). NPT innebærer blant annet *å lage egne modeller for å løse faglige utfordringer* for å utvikle *forståelse av naturfaglig teori* (Utdanningsdirektoratet, 2020a). Ifølge NRC (2012), et amerikansk rammeverk, inngår det å utvikle og bruke modeller som en av åtte praksiser for naturfagene. Naturvitenskapelige praksiser og tenkemåter erstatter det mer generelle konseptet om *utforsking*. Grunnen til denne endringen, var for å konkretisere hva utforsking innebærer (Haug, Sørborg, Mork, & Frøyland, 2021).

Modeller kan gi gode beskrivelser av virkeligheten og de fenomenene elevene skal lære om (Angell et al., 2011). Modellene er forenklinger av virkeligheten. Det kommer av at noen deler av fenomenet eller begrepet fremheves, mens andre kan bli fremstilt på en måte som ikke samsvarer helt med virkeligheten (Hannisdal & Ringnes, 2015). Det finnes ulike typer modeller, både visuelle, fysiske og kognitive (Pajchel, Ramton, & Sollid, 2019). Atommodell er mye brukt i naturfagundervisning, og er et eksempel på en visuell modell.

Det å lage en modell, kalles i naturfaglig sammenheng modellering. Utforming av en modell, skjer i en modelleringsprosess, og dette inngår i teknologiundervisningen (Pajchel et al., 2019). Ifølge Shute (2017) inngår modellering i komponenten abstraksjon i AT.

Programmering kan bidra til at deler av et fenomen som ikke kan observeres direkte, blir undersøkt på et observerbart nivå. Derfor er det naturlig å redegjøre for hva programmering vil si. Datasimuleringer kan brukes til å modellere fenomener (German, 2019).

Det er ulike måter å arbeide med modeller og modellering i naturfagundervisning (Pajchel et al., 2019). Slik det kommer frem i ordlyden av kjerneelementene, skal modeller brukes i arbeid med de andre kjerneelementene. Ett av disse er naturvitenskapelige praksiser og tenkemåter (Utdanningsdirektoratet, 2020a). Modeller kan brukes når elever skal lage prediksjoner eller hypoteser. Dette vil si å forestille seg hva som vil komme til å skje og formulering av mulige forklaringer. Dette er sentrale praksiser i det som inngår i utforskende arbeid i naturfag (Sørvik, 2016). Elevenes læring i modelleringsprosessen avhenger av at de får støtte, dette kan skje gjennom aktivisering av forkunnskaper og at modellen gir mening for eleven. Det å samarbeide og sette ord på forståelsen av modellen kan støtte læringen (Kluge, 2021).

2.2.4 Programmering

Det er matematikkfaget som har fått hovedansvaret for å lære elevene å programmere. Dette vil si å lære dem de grunnleggende konseptene og ferdighetene som brukes i programmering. Dette skal elevene videre ta i bruk i blant annet naturfag (Utdanningsdirektoratet, 2020b). Det er ikke det programmeringstekniske som er i hovedfokus for denne oppgaven. Fokuset for denne studien er hvordan programmering kan tilrettelegge for læring om teknologi gjennom programmering. Derfor blir det ikke gjort en omfattende redegjørelse for begrepet programmering og alle konseptene som inngår. Dette er andre som har gjort, blant annet Nygård (2018). Leseren kan se i denne referansen, dersom det er ønskelig med en grundigere innføring i begrepet programmering enn det som blir gitt i denne oppgaven.

Programmering kan defineres på ulike måter. Ifølge Papert (1980) er essensen i begrepet å kommunisere med en datamaskin ved å bruke et språk både den som programmerer og datamaskinen forstår (Se senere i dette delkapittelet, *ulike programmeringsspråk*).

Instruksjonene datamaskinen skal gjøre og som derfor styrer måten den fungerer, kalles en programkode (Sevik, 2016). Ifølge Nygård (2018) dreier programmering seg om en prosess med problemanalyse, abstraksjon og planlegging av hvordan en programkode skal legges opp, for å få en datamaskin til å løse et bestemt problem. Altså fra man får oversikt over problemets ulike deler, til utformingen av instruksjonene til datamaskinen som skal løse problemet (Nygård, 2018). Programmering vil dermed si alt som inngår i det å løse et problem ved hjelp av en algoritme eller et program, fra planlegging til revurdering og generalisering av programkoden (Gjøvik & Torkildsen, 2019). Programmeringsbegrepet

inkluderer også systematisk feilsøking i og forbedring av programkoden (Sevik, 2016). Programmering og koding blir i en del sammenhenger brukt som synonymer. I denne oppgaven blir disse begrepene forstått som to ulike ting. Programmering kan sies å være et mer omfattende begrep hvor blant annet koding inngår. Koding blir her forstått som en fase av programmeringsprosessen, og innebærer å gjøre om en løsning av et problem om til programkode i et bestemt programmeringsspråk (Gjøvik & Torkildsen, 2019). Utvikling av programkoden gjennom testing og feilsøking, inngår også i begrepet koding (Lodi & Martini, 2021).

Som nevnt kan programkoden skrives i ulike programmeringsspråk. Disse er enten tekst eller blokk-baserte. Språkene følger bestemte regler, såkalt syntaks. For at programkoden skal kunne bli utført av den digitale enheten, må reglene følges. Derfor utgjør feilsøking, også kjent som debugging, en viktig del av programmeringsarbeidet (Sevik, 2016). Blokkbasert programmering er en vanlig måte å introdusere elever for programmeringskonsepter (Bocconi et al., 2022). Som navnet tilsier, består dette programmeringsspråket av blokker. Dette vil si grunnstrukturer som inneholder ferdiglagde programkoder. Dermed kan programmeringsprosessen forenkles, siden elevene ikke må skrive avansert programkode selv, slik som i tekstprogrammering (Louca & Zacharia, 2008). Det er ulike blokkgrupper med ulike funksjoner å velge ut ifra, og faste rammer for hvordan disse blokkene kan systematiseres. Programkodene lages med en klikk-og-dra-teknikk (Duckworth, 2019). Formen på blokkene indikerer på hvem som kan settes sammen, slik at systematiske feil hindres i stor grad. Det er mye som er forhåndsdefinert, som gir styrte handlingsrom . Programmet Scratch er den mest kjente og brukte innen denne typen programmeringsspråk (Kluge, 2021). Denne tilnærmingen til programmering kan hjelpe elevene til å tenke kreativt, resonnere systematisk, arbeide sammen, legge til rette for algoritmisk tenkning. Det kan med andre ord tilrettelegge for et godt læringsmiljø, hvor elevene kan engasjere seg i spesifikke kontekster (Shute et al., 2017).

En sentral del av det eleven lærer, er ulike programmeringskonsepter (Hromkovic, Kohn, Komm, & Serafini, 2016). De mest relevante for denne oppgaven er løkker, betingelse, funksjon og algoritme. En løkke vil si en grunnstruktur hvor en programkode kan gjentas istedenfor å legge inn den bestemte programkoden manuelt tilsvarende antall ganger. Den er ment til å effektivisere programmet. En betingelse vil si at det er et kriterium for at programkoden skal bli utført av datamaskinen. En del av et program kalles en funksjon.

Hensikten med funksjoner, er å kunne gjenbruke hele eller deler av programmet i andre løsninger (Nygård, 2018). I undervisningsopplegget hvor elevene skal programmere micro:bit, bruker elevene blant annet funksjonen Radio. Denne gjør at det er mulig å sende og motta informasjon mellom de ulike micro:bitene. (Micro:bit Educational Foundation, 2022) En algoritme vil si en bestemt stegvis instruksjon bestående av programkode, og denne blir utført av en datamaskin. En algoritme kan bestå av de tidligere nevnte konseptene (Denning, 2017).

Det er noen utfordringer med å lære seg programmering. Presisjonsnivået som kreves er høyt, og det er lite som skal til for at det oppstår feil og dermed at programmet ikke fungerer. I tillegg er måten å arbeide på, ganske annerledes enn mange er vant med (Kluge, 2021). Dette leder oss over til neste delkapittel, fysisk programmering og micro:bit. Det er en måte å arbeide med programmering som er mye brukt i skolen (Eidsvig, 2021).

2.2.5 Fysisk programmering og micro:bit

Læringen bør skje i en setting hvor man holder på med noe som er fysisk og synlig (Papert, 1980). Fysisk programmering kan beskrives som resultatet man får ved å utforme interaktive objekter eller systemer gjennom programmerbar *hardware*. De gir en hybrid og utvidbar opplevelse som kobler sammen hardware og software (Halfacree, 2017). Forskjell mellom software og hardware. Software kalles også programvare, og vil si de programkodene som lages for å styre digitale enheter (Store norske leksikon, 2009).

Micro:bit er *små, enkle og ofte billige digitale komponenter* som kan programmeres. De er utviklet av BBC og flere teknologifirmaer (Ball et al., 2016). Hensikten med dem er å åpne den abstrakte og lukkede datamaskinen, engasjere elever ved å koble til ledninger og se effekten av det (Kluge, 2021). Inspirere til å bruke digital teknologi på en kreativ måte og utvikle grunnleggende ferdigheter innen naturfag, teknologi og ingeniør-fag (Sentance, Waite, Hodges, MacLeod, & Yeomans, 2017). Vekke og opprettholde interessen for datavitenskap/informatikk hos de unge som vokser opp i dag. De er tilrettelagt for at elever kan bruke teknologi på en kreativ måte. De kan enkelt programmeres, samt har sensorer og tilkoblingsmuligheter slik at Micro:Bit får ulike funksjoner (Ball et al., 2016). De kan både programmeres med tekst- og blokkbasert programmering (Kluge, 2021). Programkoden kan skrives og testes i et programmeringsprogram på datamaskin eller iPad, og enkelt overføres til Micro:Bit (Ball et al., 2016).

Utviklerne har tilpasset brukervennligheten, og sørget for at den er intuitiv, sier seg selv. Dermed kan f.eks. lærere ta i bruk Micro:Bit i sin undervisning uten at de eller elevene trenger spesifikk kompetanse i forkant. De har også utformet den slik at elevenes læring og progresjon tas hensyn til ved at den kan brukes og justeres opp eller ned ut ifra nivået til elevene (Ball et al., 2016). Det at programmeringen gjennom enheter som micro:bit blir overført til noe som er fysisk, åpner opp mulighetene. Ved å ta programmeringen ut av skjerm og til fysiske enheter, gjør det for eksempel mulig å lage prototyper av ulike ideer. Dette er veldig relevant for kjerneelementet teknologi, siden det blant annet omhandler modeller. De ulike sensorene som micro:bitene er utrustet med, kan brukes i naturfag i utforskning og arbeid i naturfag (Eidsvig, 2021).

2.2.6 Algoritmisk tenkning

Begrepet algoritmisk tenkning er som tidligere nevnt, brukt i den nye læreplanen. Dette er den norske oversettelsen av det engelske begrepet *Computational thinking* (Utdanningsdirektoratet, 2019a). Denne tenkemåten kan bidra til å forstå og løse komplekse problemer, både teoretiske og praktiske. Det er en mental prosess som har innvirkning på problemløsningsferdigheter (Shute et al., 2017). Algoritmisk tenkning er relatert til teknologi. Ifølge Papert (1980) kan forståelse om hvordan datamaskiner fungerer, påvirkes hvordan man lærer alle andre ting. Utvikling av evne til slik problemløsning som inngår i algoritmisk tenkning kan støttes gjennom å arbeide med programmering (Grover & Pea, 2013a). Programmering er derfor en naturlig måte å tilnærme seg AT på i skolen (Gjøvik & Torkildsen, 2019). Samtidig er algoritmisk tenkning mer enn bare å vite hvordan man programmerer. Det er en universell og anvendelig holdning, og et ferdighetssett som alle bør ivre etter å lære og bruke (Wing, 2006). I AT inngår prosessen å utvikle programmeringskompetanse, som trengs for å bruke datamaskiner (Israel et al., 2015). Denne måten å drive problemløsning på kjennetegnes ved at den er *systematisk og analytisk*.

Det er ulike måter å definere algoritmisk tenkning. I denne oppgaven, er det hovedsakelig Shute mfl. (2017) og Udirs definisjoner som blir brukt. I førstnevnte, blir algoritmisk tenkning definert slik: «*det konseptuelle grunnlaget som trengs for å løse et problem effektivt og på en måte som kan løses enten med eller uten hjelp fra en datamaskin, med løsninger som er mulige å bruke i andre kontekster*» (min oversettelse). Læreplanen som gjelder for norsk skole baserer seg på følgende definisjon av algoritmisk tenkning: «*... innebærer å tilnærme seg problemer på en systematisk måte, både når vi formulerer hva det er vi ønsker å løse og*

når vi foreslår mulige løsninger. Litt forenklet kan vi si at det er 'å tenke som en informatiker' når vi skal løse problemer eller oppgaver» (Utdanningsdirektoratet, 2019a).

I begge disse definisjonene, blir algoritmisk tenkning forstått som en form for problemløsning. De har også til felles at de er ment å bli brukt i undervisning, og er uavhengige av fag. Shute mfl. (2017) har kommet frem til definisjonen på bakgrunn av en litteraturstudie av tidligere forskning på algoritmisk tenkning. I litteraturstudien presenteres ulike perspektiver på dette begrepet, og forståelsen av algoritmisk tenkning er detaljert beskrevet.

En annen likhet er at det ut ifra begge definisjonene tilhører det et rammeverk hvor begrepet er operasjonalisert i «undergrupper» for hva som inngår i denne problemløsingstilnærmingen. Det er ikke slik at undergruppene brukes i en bestemt rekkefølge eller at alle må brukes i problemløsningen (Shute et al., 2017). Udir (2019a) har dekomponert begrepet gjennom modellen *den algoritmiske tenkeren*. (Se figur 7). Her er undergruppene omtalt som *nøkkelbegreper* og *arbeidsmåter*. Shute mfl. (2017) kaller dem *fasetter*, og har kommet frem til følgende fasetter: *dekomposisjon*, *abstraksjon*, *algoritmer*, *debugging (feilsøking)*, *iterasjon* og *generalisering*. Se vedlegg 1 for detaljer om fasettene.



Figur 1: Barefoot comp./Udir., <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>. Hentet 27.04.22

Etter å nå ha beskrevet ulike definisjoner av algoritmisk tenkning, blir de komponentene fra rammeverkene som er mest relevante for resten av oppgaven beskrevet. En av fasettene som inngår i begge rammeverkene, er *feilsøking*. I rammeverket til Shute mfl. tilsvarer dette *debugging*, som innebærer måter å oppdage og rette opp feil som oppstår i problemløsningen (Utdanningsdirektoratet, 2019a). Feilsøking kan relateres til en av arbeidsmåtene som inngår i Udirs (2019a) modell, nemlig fikling. Det vil si å utforske og eksperimentere, altså å prøve

seg frem til en løsning. Abstraksjon vil si å fokusere på de delene av problemet som er de mest essensielle, noe som kan gjøre det lettere å få oversikt over hva som skal løses. Videre kan dette brukes for å generalisere løsninger slik at de kan brukes i andre sammenhenger. Denne delen av algoritmisk tekning er essensielt i utformingen av passende modeller og implementering i problemløsningsprosess (Shute et al., 2017). Det er samtidig noe som kan være utfordrende å utvikle, og ifølge Çakiroğlu mfl. (2021) er dette en kompleks ferdighet.

Noen av undergruppene inngår i begge rammeverkene. For å få frem hvordan undergruppene overlapper, har jeg utformet en tabell (se Tabell 2). Måten Udir (2019a) forstår begrepet algoritmisk tenkning, føyer seg derfor inn i Shute mfl. (2017), som er et generelt rammeverk og legger til noen arbeidsmåter. Udir er ment å være lett å bruke i arbeidet med algoritmisk tenkning i undervisning, og får frem de sentrale delene som inngår i denne måten å tilnærme seg problemer.

Tabell 1: Overlapp mellom rammeverk for algoritmisk tenkning

Shute et al.	Udir/Barefoot comp.
Dekomponering	Dekomposisjon
Algoritmisk design	Algoritmer
Debugging	Feilsøke
Abstraksjon	Abstraksjon
Generalisering	Logikk
Iterasjon	Mønstre
	Evaluerings
	Fikle
	Skape
	Holde ut
	Samarbeid

Algoritmisk tekning kan også bli forstått på en måte som er mer naturvitenskapelig enn disse to rammeverkene gjør (Shute et al., 2017; Utdanningsdirektoratet, 2019a). Et rammeverk som er det, er den som er laget av Weintrop mfl. (2016). Dette er ment å hovedsakelig dreie seg om arbeid som er relatert til naturfag- og matematikkundervisning, og vil være passende å bruke ved utforskning av naturfenomener. Her algoritmisk tenkning inndelt i fire hovedkategorier: *datapraksiser, modellering og simulering, praksiser for å løse dataproblemer* (min oversettelse) og *systemtenkning*. Det er begreper i disse kategoriene som overlapper med de to tidligere presenterte rammeverkene, blant annet *feilsøking* og

abstraksjon. Weintrop (2016) sitt rammeverk er mer teknisk enn de to andre rammeverkene, ved at det sentreres omkring digitale momenter. Noe av det som gjør det relevant å presentere dette rammeverket i sammenheng med denne oppgaven, er at Weintrop har inkludert modeller. Det kommer frem i dette rammeverket at å lage nye eller utvide allerede eksisterende modeller er en sentral naturfaglig praksis. I tillegg er ifølge Weintrop (2016) ferdigheter tilknyttet systemtenkning en viktig del av algoritmisk tenkning. Dette begrunnes med at det kan gjøre det lettere å holde oversikt over systemet som undersøkes, som videre kan gjøre det lettere å løse problemer.

2.3 Læring i teknologi

2.3.1 Støttestrukturer

Dette delkapittelet består av en redegjørelse av begrepet *støttestrukturer*. Dette er noe Skaperskolen har som et eget pedagogisk verktøy i sine undervisningsopplegg (2020c). Det er ulike måter å se på læring og hva som skal til for at læring skal forekomme. I perspektivet som inngår i *sosiokulturell læringsteori* skjer læring i fellesskap med andre. Ved å utveksle ideer mellom personer, kan kunnskap utvikles (Vygotsky, 1987). Fellesskapet tilrettelegger for at personer kan hjelpe hverandre til å mestre en oppgave eller forstå noe bedre. Det leder oss inn på begrepet *den nærmeste utviklingssonen*. I dette menes det at når man skal lære seg noe nytt, vil det være forskjell på hvor mye man kan klare alene og sammen med en person med høyere kompetanse. Derfor vil det å få tilstrekkelig støtte spille inn på læringen (Woolfolk, 2016).

Skaperskolen trekker frem at selv godt utarbeidede opplegg ikke nødvendigvis fører til ønsket resultat. De presenterer ulike faktorer som kan gjøre at elevene ikke kommer i mål med aktiviteten. Eksempler på dette er at elevene ikke forstår oppgaven eller at de mangler kunnskap og ferdigheter til å lykkes. Noe av hensikten med støttestrukturene, er å redusere mulighetene for at slike faktorer inntreffer (Skaperskolen, 2020c). Faktorene vil oppta mye av elevens hjernekapasitet, altså bli en kognitiv belastning. Dette vil kunne gå ut over kapasiteten eleven har til å løse oppgaven. Med tanke på å tilrettelegge for læring, er det viktig å redusere den kognitive belastningen. Det å øve på ferdigheter slik at de går mer og mer av seg selv, såkalt automatisering, ser ut til å være med på å frigjøre kognitiv kapasitet (Sweller, van Merriënboer, & Paas, 1998). Tilstrekkelig opplæring i et programmeringsprogram som eksempel på en støttestruktur som kan sikre at elevene kan bruke programmeringen som

verktøy i problemløsingen (Skaperskolen, 2020c). Forståelse for programmering og kompetanse i hvordan dette gjøres, er nøkkelpunkter for å kunne skape og produsere digitale produkter (Sevik, 2016).

Læreren er en sentral person i elevenes læring og kan fungere som støtte for elevene. Ved å organisere elevene i par, er et grep læreren kan bruke (Woolfolk, 2016). Støtten kan også bli gitt i form av modellering, som innebærer å vise eleven hvordan den selv ville gått frem for å løse oppgaven (Holt, Øyehaug, & Voll, 2019). Ifølge Rehmat, Ehsan, and Cardella (2020) kan dette hjelpe eleven til å forstå hva oppgaven spør etter, samt få elevene til å oppdage og rette opp feil. Det å komme med hint, gjentakelse av pensum som er relatert til oppgaven. Ifølge Woolfolk kan elevene bli støttet ved å få hjelp til å bryte ned problemet i mindre deler (2016). Bruk av spørsmål i samtalene med elevene kan gjøre at elevene forstår oppgaven og dens kriterier lettere (Skaperskolen, 2020c). Målet er at eleven skal bli selvstendig i arbeidet. Derfor bør støtten justeres gradvis ettersom eleven utvikler sin kompetanse. Med andre ord er det viktig å gi støtten når eleven trenger det og i passe form og mengde (Woolfolk, 2016, p. ”).

2.3.2 Kunnskapsintegrasjon

Denne tilnærmingen (kunnskapsintegrering) til undervisning baserer seg på at elever kobler ny kunnskap til de antakelsene de allerede har gjort seg ved å observere verden de omgir seg i. Her står det sentralt at elevene selv får vurdere egne ideer, bruke det de allerede har av ideer, sammenlikne og analysere bevis for ideene, samt skille ideene fra hverandre.

Kunnskapsintegrering består av blant annet å revurdere egen kunnskap (Linn & Eylon, 2011).

Dette gjør de gjennom en prosess med fire faser:

Tabell 2: Operasjonalisering av prosesser i kunnskapsintegrasjon. Hentet fra s.7 i Waters(2020) masteroppgave

KI-prosess	Definisjon	Operasjonalisering
Elisitere ideer	Elever blir gjort oppmerksomme på kunnskap de har fra før	Elever uttrykker forkunnskaper ved å f.eks. huske fakta og formler
Legge til nye ideer	Elever blir introdusert for ideer som er nye for dem	Elever blir introdusert for nye ideer fra programmet, medelev eller lærer.
Utvikle kriterier for å evaluere ideer	Elever utvikler og bruker spørsmål og tester for å vurdere om nye ideer er akseptable eller ikke	Elever vurderer informasjon og resultater som gyldige eller ikke gyldige.
Sortere og reflektere	Elever reflekterer over og sorterer forskjeller og likheter mellom ny og gammel kunnskap, baser på de kriteriene de har utviklet	Elever uttrykker fysikkforståelse ved å organisere, tolke og forklare nye og gamle ideer.

Dersom undervisning har denne tilnærmingen, kan elevene lettere huske og bruke den nye kunnskapen i nye situasjoner. Kunnskapsintegrasjon kan bidra til at elevene utvikler synspunkter som er sammenhengende. Den legger til rette for at alle elever kan lære ut ifra på sine egne ideer (Linn & Eylon, 2011).

2.3.3 Begrepslæring og språklige ferdigheter

Ifølge sosiokulturell læringsteori foregår læring i fellesskap med andre, og her spiller språket en viktig rolle. Elevers tenkning uttrykkes gjennom språket (Vygotsky, 1987). Det brukes ulike måter å uttrykke seg på avhengig av sammenhengen. Det hverdagslige språket er ofte mer uformelt og preget av muntlige formuleringer. I naturfag brukes det en annen måte å ordlegge seg på for å få frem et bestemt faginnhold. Denne er bestående av særegne begreper, som vil si ord for ideer, prinsipper og forestillinger (Mork & Erlien, 2017). En stor del av læringen av innholdet i naturfag, handler om å tilegne seg kunnskap om begrepene. Derfor er

begrepslæring en sentral del av dette faget (Haug, 2016). I tillegg til å lære om de ulike begrepene, innebærer begrepslæring også å kunne implementere begrepene i språket elevene bruker. Hensikten med dette er blant annet å kunne uttrykke innholdet på en forståelig, men samtidig presis måte. Jo mer bruk av begreper, desto mer presise blir formuleringene (Mork & Erlien, 2017). Dette kan videre sees i sammenheng med utforskende arbeid. Fokus på språk og begrepslæring er også relevant å trekke inn med tanke på denne måten å arbeide med naturfaginnholdet. En sentral del av utforskende arbeid i naturfag, er ifølge Mork (Mork, 2016) å bruke språket i faglige diskusjoner. Denne måten å bruke språket på kan tilrettelegge for at eleven oppdager sammenhenger, og får testet ut egne eller andres antakelser omkring et naturfaglig fenomen. Dette leder oss videre til utvikling av elevers begrepslæring.

Elevene trenger mer enn å huske mange ord og deres definisjoner. Læring i naturfag består i å gradvis forstå begrepene bedre, samt å sette dem i sammenheng med hverandre (Haug, 2016). Det er utformet et rammeverk «*Fra ord til begrep*» (se Tabell 4) som kan brukes for å vurdere elevenes begrepsforståelse og tilrettelegge for hvordan elevene kan utvikle den videre. Rammeverket presenterer en progresjon i begrepsforståelsen, fra passiv til mer aktiv begrepsbruk. En passiv begrepsbruk vil si at man kan gjenkjenne et ord og gi en definisjon av ordet. Ifølge Haug (2016) er dette starten på elevers begrepslæring. Når elevene klarer å bruke ordet på mer komplekse måter, vil begrepsforståelsen bli beskrevet som aktiv. Eksempler på dette er å bruke begrepene i sammenheng med andre begreper og i andre meningsfulle kontekster, samt å bruke dem mer generelt. Når dette relateres til læring om teknologi kan eksempel være å utvikle forståelse for hjemmeruteren. Ut ifra rammeverket som Haug (2016) presenterer, vil eleven da kunne starte med å lære seg definisjonen på begrepet ruter. En del av begrepslæringen kan videre være å sette dette begrepet i sammenheng med Internett som system, og å kunne si noe mer generelt om hvordan det innvirker på annen digital teknologi vi omgir oss med.

Tabell 3: "Fra ord til begrep" – rammeverk Haug (2016) s. 148

	Kunnskap om ordets betydning	Nivå	Beskrivelse
	Lav	Gjenkjenning	Kjenner igjen ordet i tekst og tale og kan uttale det.
	Passiv	Definisjon	Kan gjengi definisjonen til et ord, men har liten forståelse for hva ordet betyr.
Begrepsforståelse	Aktiv	Nettverk	Vet hvordan ordet kan knyttes til andre ord og begreper.
		Kontekst	Kan bruke ordet i flere setninger og i en sammenheng som gir mening.
		Anvendelse	Kan bruke ordet i tilknytning til sin egen utforskning, både under innsamling og diskusjon av egne data.
		Syntese	Vet hvordan ordet kan anvendes for å kommunisere egen forståelse av fenomenet som utforskes. Kan anvende ordet mer generelt, på tvers av og i nye situasjoner.

2.4 Oppsummering teori

Det er nå blitt trukket frem teoretiske perspektiver som sier noe om momenter som undersøkelsen dreier seg om. Dette er hvordan algoritmisk tenkning kommer til uttrykk i opplæringen og deretter ulike temaer som omhandler programmering i skolen. Ulike læringsperspektiver som kan knyttes til læring i kjerneelementet teknologi er også blitt beskrevet. I det følgende beskrives noe forskning som allerede er gjort omkring bruk av teknologi og programmering. Dette blir gjort for å kunne sette studien som denne masteroppgaven tar utgangspunkt i inn i en større sammenheng.

2.5 Tidligere forskning

I dette kapitlet presenteres noe tidligere forskning som anses som relevant for diskusjonen. Da min oppgave ikke er en litteraturside, er det blitt gjort et utvalg av forskningen som er blitt gjort på området. Den forskningen som er utvalgt, er motivert av at de skriver om begreper som sentrale i denne oppgaven. De har forsket på og/eller har liknende tematikk.

Det er blitt gjort mye forskning på ulike sider ved programmering. Forskning på innføring og bruk av programmering har blitt gjort i større grad på videregående- og universitetsnivå enn på barne- og ungdomsskole. [Kunnskapshull]. På høyere undervisningsnivå ofte vil det ofte være mer tid til å kunne sette av til programmeringsaktiviteter enn i grunnskolen. Det kommer av at det er mange temaer som elevene i grunnskolen skal bruke tid og fokus på. Derfor kan forskningen på de ulike undervisningsnivåene gi ulike resultater. Det er likevel relevant å inkludere noen slike studier, da resultater herfra sannsynligvis vil kunne være nyttig og overførbare også for undervisning på lavere klassetrinn.

Litteratursøk ble gjort ved å søke i ulike arkiver, slik som Google Scholar, Oria. Blant søkeordene som ble brukt var *technology*, *computational thinking*, *programming*, *science education*, *teaching* og *micro:bit*. Søkeordene ble kombinert på litt ulike måter for å få frem flere studier, og referanser knyttet til disse artiklene ble også undersøkt.

Forskningen som presenteres omhandler ulike aspekter ved tema programmering. Første del får frem noen interessante funn tilknyttet bruk av programmering for å lære naturfaglige temaer. I det følgende blir noen studier hvor undervisning som inkluderer bruk av fysisk programmering og micro:bit er blitt undersøkt. Neste del dreier seg om elevers måte å tilnærme seg problemer i programmeringsaktiviteter. Avslutningsvis presenteres forskning relatert til ulike støttestrukturer som kan tas i bruk i undervisning med programmering. Her er det lærerens betydning, samt organisering av elevene i par/grupper det hovedsakelig dreier seg om.

2.5.1 Forskning på naturfag og programmering

Denne masteroppgaven omhandler bruk av programmering til å lære om teknologi. Derfor var det ønskelig å finne studier som tar for seg hvordan dette kan gjøres, og hva andre har funnet når teknologiprojekter gjennomføres. I 2014 ble det publisert en artikkel som omhandlet en studie omkring *teknologi og design* (ToD). Hensikten til studien av Bungum mfl. (2014), var å se på hvordan prosjekter i dette faget ser ut til å fungere med tanke på å inkludere kunnskap fra matematikk og naturfag. Utvalget var hentet fra seks klasser på ulike skoler i Norge. Disse klassene fikk tildelt hvert sitt prosjekt som de skulle gjennomføre. Datainnsamlingen tok utgangspunkt i gjennomføringene av disse prosjektene, og ble gjort ved å ta videoopptak av elevene. Blant hovedfunnene i studien var at fagkunnskap fra disse to fagene i liten grad ble inkludert i prosjektene. Dette til tross for at slik kunnskap i utgangspunktet ble ansett som relevant av læreren og forskergruppen. I diskusjonen blir det trukket frem fire underliggende årsaker. *Begreper og prosedyrer er ikke nødvendig, problemløsingen kan bli bedre utført med andre metoder, kvaliteten på produktet er i hovedfokus og fagkunnskapen er ikke den kunnskapen elevene ser ut til å trenge.* De foreslår at teknologi som kunnskapsområde i skolen, bør representeres i pensum for seg selv istedenfor en arena for å lære naturfag og matematikk (Bungum et al., 2014).

Det er laget rapport omkring hvordan algoritmisk tenkning kan integreres i grunnskolen, og omhandler ulike land i Europa. I rapporten blir begrepet AT undersøkt på ulike nivåer ved å bruke fler metoder, blant annet litteratur-studie, spørreundersøkelser og samtaler med lærere, politikere og forskere. Rapporten sammenfatter omfattende forskning på hvordan algoritmisk tenkning kan implementeres i grunnskolen. Blant det de presenterer er grunner til å bruke programmering for læring tilknyttet teknologi. Programmering ved bruk av instruksjoner kan føre til at de går fra å være passive brukere av teknologi, til å kunne skape digitale objekter. De har funnet at det i mange land blir nedprioritert å bruke tid på at elevene får utvikle ferdigheter innen algoritmisk tenkning. Lærere rapporterer at det er utfordrende å implementere AT i fagene. De konkluderer blant annet med å sette av tid til å lære grunnleggende programmering og ferdigheter innen algoritmisk tenkning (Bocconi et al., 2022).

Hovedpoenget i studien til Haug (2021) er at det å bruke naturfaglige praksiser og tenkemåter, kan forstås på ulike måter. Lærere ser ut til å stå alene i å vite hvordan NPT skal tolkes, implementeres og vurderes. Artikkelen består av forslag på en felles forståelse for hva NPT

innebærer. Det er nyttig i læreres arbeid. De har kommet frem til en konkretisering av de NPT og forslag til hvordan de kan implementeres i fagene. Kunnskap om praksisene kommer frem som sentralt, både hos lærere og elevene. De foreslår å arbeide med egne modeller og modeller som allerede eksisterer i naturfag. Arbeidet med praksisen må kobles nært til det naturfaglige innholdet. På den måten kan NPT brukes til utforskning, som er sentralt i naturfag (Haug et al., 2021).

En av arbeidsmåtene i Udirs rammeverk for algoritmisk tenkning, er å skape. Det er blitt gjort noen studier på hvordan dette kan bidra i utviklingen av nettopp algoritmisk tenkning. Hoppe og Werneburg (2019) har undersøkt hvordan kreative aktiviteter ser ut til å innvirke på utviklingen av denne problemløsingstilnærmingen. Noe av det de beskrev i resultatene, var at det avgjørende for elevers tilegnelse av nye AT kompetanse at de får anledning til å skape. Programmering kan tilrettelegge for at elevene får uttrykke seg og være kreative. Kreative aktiviteter hvor modellering, spesifisering og representasjon av problemer inngår, er sentralt i utviklingen av algoritmisk tenkning (Hoppe & Werneburg, 2019).

I de neste avsnittene blir det beskrevet noen studier som er relatert til måten elevene får arbeidet med programmeringen. Taub mfl. (2015) gjorde en undersøkelse for å få frem om og hvordan elementer i et læringsmiljø bestående av programmering fungerte. Utvalget bestod av videregående elever som deltok i et kurs i fysikk. Disse ble observert, og det ble fokusert på hvordan bruk av modeller uttrykt i matematiske formler påvirket elevenes forståelse. Studien ble analysert hovedsakelig kvalitativt, hvor *kunnskapsintegrasjon* (Se teori, kap.2.2.8) ble benyttet som analyseverktøy. I en observert programmeringssituasjon, var det flere kunnskapsområder eller faser, det så ut til at elevene anvendte. Blant funnene fra analysen, var at denne måten å bruke programmering kan styrke elevenes naturfaglige forståelse. Det å la elevene ta i bruk en representasjonsform, programmering, så ut til å gjøre at elevene vil kunne i ledes til å anvende sin naturfaglige kunnskap (Taub et al., 2015).

Noen andre forskere som har undersøkt fysikkundervisning fokus på programmering, er Caballero mfl. (2014). I likhet med Taub (2015), var det også her modellering gjennom programmering undervisningen bestod av. Utvalget deres bestod av 9.klasseelever uten erfaring med programmering. Den brukes til å sammenlikne ulike representasjoner av naturfaglige fenomener. Blant hovedfunnene i denne studien er at elevene synes det var vanskelig å feilsøke, og trengte hjelp til å komme frem til hva som var årsaken til feil som

oppstod. De hadde utfordringer med å skjelve om årsaken lå i programkoden eller i deres naturfaglige forståelse. Videre kom det frem at elevene slet med å vite hva de skulle gjøre for å løse problemet. Det ble også trukket frem at programmeringen ikke ble koblet tydelig nok til det naturfaglige innholdet, og at mye tid gikk til at elevene måtte friske opp programmeringsferdighetene sine. De foreslår å prioritere programmering i undervisningen, siden elevenes ferdigheter i programmering er noe som må holdes vedlike. Dette kom av at det var lenge mellom hver gang elevene programmerte.

Blokkprogrammering er en måte å arbeide med programmering som er mye brukt i grunnskolen. Lai og Lai (2012) har undersøkt bruk av Scratch-programmering i naturfag. Dette er en langvarig studie gjort med 96 elever som gikk i 5.klasse. De oppdaget at det var en del av elevene som opplevde at det var litt vanskelig å jobbe med programmeringsspråket Scratch. Dette var relatert til grunnstrukturene og at det var en uvant måte å arbeide på for elevene. Samtidig kom det frem at denne typen programmering forbedret elevenes logiske tenkeferdigheter. Data tyder på at de fikk forbedret problemløsningsevne, og ble bedre til å lage modeller av naturfaglige fenomener. Over 60% av elevene svarte at de forstod naturfagtemaet bedre etter å ha deltatt i undervisningen. Dette tyder på en positiv effekt på elevenes naturfaglige forståelse ved å drive med denne programmeringstilnærmingen.

2.5.2 Forskning relatert til fysisk programmering

Det er flere måter man kan tilrettelegge for å undervise i programmering. Fysisk programmering er en tilnærming. Videre vil noen studier som har undersøkt hvordan dette kan gjøres bli presentert.

Det er blitt gjennomført en studie for å undersøke ulike aspekter ved bruk av micro:bit for elevenes algoritmiske tenkning. De så på hvordan elever arbeider, hvilke behov elevene har og hvordan progresjonen deres er. Det at micro:bitene bruker blokker, gjør at feilene som oppstår oftest dreier seg om logiske feil fremfor syntaksfeil. Elevenes forståelse for programmering kan forbedres ved å arbeide med slike feil ifølge denne studien. Blant funnene var at elevene trenger tilstrekkelig forkunnskaper, samt hjelp til å løse tekniske utfordringer i utstyret som oppstår. De foreslo jevnlig bruk av oppmuntringer fra læreren for å motvirke at elevene stopper opp i arbeidet (Tyrén, Carlborg, Heath, & Eriksson, 2018).

Fysiske roboter er en annen form for fysisk programmering som kan tas i bruk i undervisning i skolefagene. Det er blitt gjort en studie hvor undervisningen med fysiske roboter ble sammenliknet med virtuelle roboter. Sistnevnte var en type datasimulator. Dette ble gjort for å få innsikt i hvordan undervisning i programmering er hensiktsmessig å gjennomføre. Dette var med tanke på å støtte elevene i sin systemtenkning og ferdigheter i algoritmisk tenkning. Elever på 8.trinn deltok i undervisningen, og undervisningsperioden hadde en varighet på to uker. De samlet inn data ved å filme elevene, en aktivitetslogg over elevenes arbeid i blokk-editoren i tillegg til en før-etter-test. Blant funnene var at de fysiske robotene så ut til å være litt mer utfordringer med å bruke, siden de var ikke like forutsigbare som de virtuelle. I begge undervisningsoppleggene ble det godt tilrettelagt for at elevene fikk lære programmering. Studien konkluderer med at både virtuell og fysisk programmering kan virke positivt på elevenes læring og utvikling av elevenes systemtenkning (Berland & Wilensky, 2015).

2.5.3 Forskning hvor problemløsning tilknyttet programmering er undersøkt

Det er flere som har undersøkt problemløsning i programmering. Videre gis det noen beskrivelser av noen som anses som relevante for diskusjonen i denne oppgaven.

Det er flere som har gjort undersøkelser på bruk av programmering i fysikk, og hvordan elevene arbeider med dette. Sørby og Angell (2012) ønsket å belyse hvilke utfordringer fysikkstudenter møter i læringssituasjoner, og hvordan elevene kunne imøtekomme disse. I sine resultater kommer det frem at studentene tilnærmet seg selv de enkleste oppgavene med en prøve-og-feile-strategi. I situasjoner hvor programkoden ikke fungerte, ser det ut til at tilnærmet studentene seg det ved å prøve ut ulike justeringer i koden. Fysikken bak ble i mindre grad brukt som måte å komme frem til riktig løsning. De så ut til å trenge hjelp til å gå frem mer systematisk og ta i bruk fagkunnskapene sine (Sørby & Angell, 2012).

Problemløsning i programmering fysikk har også Kohl og Finkelstein (2008) undersøkt. De ønsket å undersøke hvilken betydning forkunnskaper i programmering har på måten elevene går frem i oppgaven. Resultatene fra denne studien tilsier, at elever med mye erfaring bruker tid på å analysere og går frem på en målrettet måte. I tillegg klarer disse å arbeide med et problem fra ulike «startpunkter». De med mindre erfaring går frem med å teste det første og beste, og ved å prøve seg frem (Kohl & Finkelstein, 2008).

En av ferdighetene i algoritmisk tenkning er abstraksjon. Çakıroğlu mfl. (2021) ønsket å undersøke hvilken effekt blokkbasert programmering har på elevers abstraksjon. De kom frem til at denne måten å programmere kan bidra til at elever oppøver evne til abstraksjon. De har kommet frem til nødvendige tiltak for at abstraksjon skal utvikles. Elevene bør være kjent med og ha erfaring med egenskapene i blokkstrukturene, vite hvilke blokkgrupper som finnes og hvordan de kan brukes sammen. At de blir klar over at deler av programmene kan brukes videre og forenkles.

Programmering er en måte å arbeide på som kan tilnærmes på ulike måter. Dette er blitt undersøkt i et av våre naboland, Sverige. Funnene vil derfor kunne ha overføringsverdi til undersøkelsen tilknyttet denne masteroppgaven. Studien er utført av Sparf mfl. (2022) for å få innblikk i hvilke tilnærminger elever som er nybegynnere innen programmering bruker. Utvalget var elever som deltok i programmeringsundervisning på tre svenske vitensentre. Undersøkelsesmetodene var observasjon og uformelle samtaler med elevene. Det ble trukket frem hvordan undervisningen ble tilrettelagt gjennom de fysiske omgivelsene, programvaren som ble brukt og instruksjoner fra personen som underviste. De har funnet at elevene tilnærmet seg oppgaven på fem ulike måter: matematisk, prøving-og-feiling, steg for steg, rutiner og estetisk. Uavhengig av hvordan elevene gikk frem, kom Sparf (2022) frem til at elevene fikk anledning til å utvikle sentrale programmeringsferdigheter, slik som problemløsning og abstraksjon.

Problemløsning og programmering er temaer som elever skal arbeide med og bruke i naturfag. I rapporten som Bocconi mfl. (2022) presenteres blant annet en studie tilknyttet hvordan dette kan gjøres og hvilke fordeler det kan ha. De har undersøkt hvordan problemløsning og programmering blir arbeidet med i undervisning i naturfag. Her understrekes det at feilsøking er en helt vesentlig komponent innen programmering. Det kommer frem at det å lære av feil, og å jobbe i mindre grupper ser ut til å fungere bra med tanke på elevenes algoritmiske tenkning. Det å verdsette og få frem feilsøking som en strategi kan utnyttes til å skape en kultur hvor elevene kan få et nytt syn på det å gjøre feil (Bocconi et al., 2022).

2.5.4 Forskning på lærerens og medelevers betydning for læring

Denne delen av dette delkapittelet tar for seg noe forskning som dreier seg om aspekter tilknyttet hvordan elevene kan støttes i arbeidet. Det disse har til felles, er at støttestrukturer har vist seg å være en sentral faktor når elevene programmerer. I Repenning mfl. (2010) sin studie ble det synlig at elever vil ha utfordring i sin utvikling av algoritmisk tenkning dersom det oppstår for mye frustrasjon og lite progresjon i oppgaveløsingen. *Lav reskel, stor takhøyde* (LTST) er noe de trekker frem som viktig i elevenes problemløsning (Repenning et al., 2010). Det er flere studier som omhandler bruk av støttestrukturer i undervisning. Basu (2016) har i sin studie undersøkt hvordan algoritmisk tenkning kan tilrettelegges for i naturfaglig sammenheng. I tillegg ville de se etter hvilke utfordringer elever møter i et undervisningsopplegg hvor de skal lage modeller ved bruk av programmering. Undervisningsopplegget ble gjennomført i fire klasser på 6.trinn i en delstat i USA, og foregikk over en periode på tre uker. Et hovedfokus med studien er å få frem aspekter omkring bruk av støttestrukturer. Målet med støttestrukturene var å hjelpe elevene med å forstå hvorfor modellen ikke fungerer slik de hadde forutsett og hvordan de kan gå frem systematisk for å rette opp i feilen(e). Dette kan gjøre at de blir i stand til å utvikle generelle ferdigheter tilknyttet modellering, og ta dem i bruk i andre settinger. Utfordringene som ble synlige i deres undersøkelse, var både relatert til programmeringen, modellering og det naturfaglige. Blant hovedfunnene i studien er at støttestrukturene hjelper elevene til å lage presise modeller og løse problemet på en systematisk måte.

Det er ulike måter å tilrettelegge for støtte i elevers læring. Basu (2017) undersøkte et undervisningsopplegg hvor støtten ble gitt i form av en virtuell veileder som var en del av programmeringsplattformen elevene brukte. Oppgaven elevene skulle løse bestod av å forutsi hva resultatet av en utdelt programkode, samt sette sammen en programkode med ulike programmeringskonsepter for å modellere ulike scenarioer. Veilederen fungerte ved å gi en respons når eleven svarte feil på oppgaven tre til fem ganger, samt å stille spørsmål som hjelper eleven videre i oppgaven. Blant hovedfunnene i denne studien var at støttestrukturene i dette opplegget bidro til at elevenes modeller ble mer presise. Måten elevene modellerte på var mer effektive, kunne overføre modelleringsferdighetene i nye situasjoner og elevenes forståelse for naturfaglige og AT-konsepter ble forbedret. Dette programmeringsverktøyet virket positivt inn på elevenes læring i naturfaglig innhold (Basu et al., 2017).

Læreren er også en måte å gi støtte til elevene. Rehmat mfl. (2020) ønsket å relatere lærerens støtte av algoritmisk tenkning (AT) hos elevene i et undervisningsopplegg. Blant funnene de legger frem, er at lærerens bruk av støttestrukturene spørsmål og modellering virker positivt på problemløsingen til elevene. Det hjalp elevene til å oppdage mønstre og støtte elevenes måte å feilsøke i programmet. Dekomponering av problemer, abstraksjon, algoritmer og prosedyrer er andre AT-ferdigheter som kom til syne (Rehmat et al., 2020). Det var altså fem av kompetansene innen AT som ble kom til syne ved bruk av disse støttestrukturene i undervisningen (Rehmat et al., 2020).

Den neste delen av dette delkapittelet tar for seg forskning som ser på støtte i form av samarbeid. Samarbeid er en av arbeidsmåtene i *den algoritmiske tenkeren*. (Utdanningsdirektoratet, 2019a). Parprogrammering vil si at to personer programmerer sammen for å løse en oppgave (Cao & Xu, 2005). Nå som det er utviklet gode og flyttbare hjelpemidler, som laptop og iPad, ligger mye til rette for at elevene kan organiseres i par og lære gjennom samarbeid (Harlow & Leak, 2014). Undervisning i programmering hvor elevene får brukt språket, kan virke positivt på læringen (Kluge, 2021).

I denne studien Harlow og Leak (2014) er måten elever kommuniserer når de programmerer undersøkt. Videoopptak av elever som har programmert med blokkprogrammering analysert. De så på hvordan Ideene elevene kom med utviklet seg i løpet av undervisningen. De har kommet frem til at måten dette gjøres på har innvirkning på elevenes læring av programmering. Verbaliseringen bidro til at elevene kom frem til løsninger som var avanserte. En studie utført av Denner, Werner, Campe, and Ortiz (2014) har hvordan samarbeid fungerer tilknyttet programmering. Deres utvalg bestod av 325 ungdomsskoleelever som brukte Alice programmerings-miljø til å lage dataspill. Elevene arbeidet i grupper. Denne arbeidsmåten så ut til å gi økte ferdigheter innen algoritmisk tenkning enn om elevene jobber alene. Det ser ut til at samarbeid er fordelaktig uavhengig av hvor mye erfaring elevene har (Denning, 2017).

Elevers måte å samarbeide på, er blitt undersøkt i en studie av Cao og Xu (2005) er blitt gjort for å se på mønstre i måten elever arbeider når de programmerer i par. Et av funnene fra studien var at parsammensetningen ga ulike mønstre. Om det er to elever med mye erfaring (ekspert), to elever med middels erfaring, én med mye (ekspert) og én med mindre erfaring (nybegynner). I den siste var det minst samhandling mellom elevene. Det så ut til at den eleven med mye erfaring ikke likte øvelsen, mens den med lite erfaring hadde stor nytte av å

jobbe slik. Samtidig fikk den mer erfarne mange anledninger til å reflektere og forklare egen tankegang. Dermed indikerer det på at begge fikk utviklet seg (Cao & Xu, 2005).

En annen forskning tilknyttet samarbeid er utført av Webb og Mastergeorge (2003). Metoden de har brukt er litteratur-review. I artikkelen får de frem aspekter ved samarbeid. De har undersøkt hva som skal til for at samarbeid skal fungere på en god måte. Det ser ut til at det er en del ting som har innvirkning på hvor mye utbytte elevene har av samarbeidet. Blant funnene var kvaliteten på måten elevene kommuniserer.

2.5.5 Oppsummering tidligere forskning

Det er her gitt en beskrivelse av et utvalg studier som er gjort tilknyttet teknologi og programmering i skolen. Programmering ser ut til å ha et potensiale med tanke på å styrke elevenes læring av naturfaglig innhold. Flere av studiene har funn hvor elevene ser ut til å få forbedret faglig forståelse etter å ha drevet med programmeringsaktiviteter. Det at programmeringen blir en måte å få frem vesentlige deler ved faget, altså fungere som en modell, kan gjøre at de forstår innholdet bedre. Det å konkretisere det gjennom steg som gir et resultat, kan styrke elevenes systemtenkning. Samtidig kommer det frem at det naturfaglige innholdet kan komme i skyggen i programmeringsprosjekter. Det ser ut til å være flere grunner til dette. Kognitiv belastning, altså hva som kreves av elevenes fokus, kan forstyrres dersom det oppstår mange tekniske feil. I fysiske programmeringsverktøy, slik som mibto:bit, er dette en gjennomgående utfordring. Teknikken spiller ikke alltid på lag med de som skal bruke dem. Dette er ikke noe elevene kan noe for. For å demme opp om slike tekniske utfordringer vil læreren kunne støtte elevene. Det kan gjøres ved å gi hint og veiledning underveis.

Når elevene skal bruke programmering som modell, vil det kunne gjøre at elevene løser oppgaven uten nødvendigvis å trekke inn sin naturfaglige kunnskap. Det kan komme av at det rett og slett ikke alltid ser ut til å være nødvendig for å kunne klare oppgaven. Andre ganger er det fordi eleven ikke selv ser at naturfaglig kunnskap kan hjelpe dem videre. Her kan læreren lage koblinger mellom programmeringen og det naturfaglige programmeringen skal brukes til.

Det er også trukket frem studier som har undersøkt hvordan elevene går frem for å programmere. Her er en tendens at uerfarne elever går frem på en mindre systematisk måte enn elever som har jobbet med det tidligere. Prøving-og-feiling er en mye brukt strategi. Derfor vil elevene ha behov for veiledning til å komme frem til hvordan de målrettet kan finne ut hvordan de skal gå frem. Det å effektivt sette sammen programkoden og bruk av abstraksjon, altså å trekke ut det vesentligste, er også noe elever kan ha utfordringer med. Det har også kommet frem fra studier at samarbeid om programmeringen kan virke positivt på elevenes progresjon i arbeidet. Her vil måten gruppene settes sammen på, i forhold til forkunnskaper, være avgjørende for hvor godt samarbeidet fungerer. Det er tydelig at det å sette ord på egne tanker vil kunne virke oppklarende.

3. Metode

I dette kapitlet blir først undervisningsopplegget som masteroppgaven tar utgangspunkt i beskrevet. Videre begrunnes metode og utvalg. Deretter beskrives hvordan forberedelsene og selve gjennomføringen av datainnsamlingen ble gjort. Analyse av datamateriale redegjøres for, før kvaliteten på studien avslutningsvis diskuteres.

Denne masteroppgaven baserer seg på en undersøkelse hvor elever er blitt intervjuet og observert. De deltok i et undervisningsopplegg hvor programmering inngår. Undersøkelsen hadde til hensikt å belyse hvordan programmering kan legge til rette for læring av teknologi i naturfag. Dette var for å kunne få innsikt som får frem perspektiver om bruk av programmering i naturfagundervisning. Masteroppgaven er en del av forskingsprosjektet TRELIS. En av deres samarbeidspartnere er Skaperskolen. De har blant annet utformet et undervisningsopplegg som heter *Internett fanger*. Programmering er en del av dette, og derfor ble det ansett som et aktuelt undervisningsopplegg å bruke i denne studien. I følgende avsnitt blir undervisningsopplegget beskrevet.

3.1 Undervisningsopplegget Internett fanger

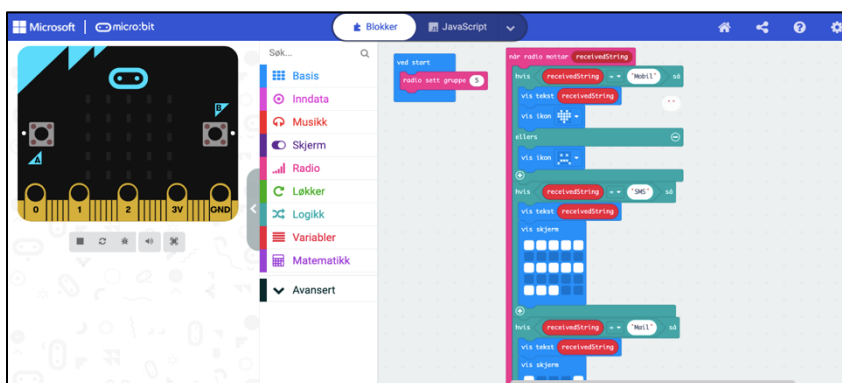
Det undervisningsopplegget som undersøkelsen tok utgangspunkt i, heter *Internett fanger*. Dette er utarbeidet av Skaperskolen (2020a). Det er følgende kompetansemål opplegget dreier seg om: «elevene skal kunne utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker» og «bruke programmering til å utforske naturfaglige fenomener» (Utdanningsdirektoratet, 2020a). Som navnet tilsier, er Internett det teknologiske systemet elevene skal arbeide med (Skaperskolen, 2020a).

Undervisningsopplegget består av totalt åtte deler, men det er den fjerde delen som er blitt undersøkt i forbindelse med denne masteroppgaven. Hele undervisningsopplegget er utformet med undervisningsverktøyet «boblemodellen», som Skaperskolen har utformet. Gjennom bruk av dette undervisningsverktøyet, kan læreren støtte elevene i en skapende arbeidsprosess. Boblene i modellen står for ulike typer undervisning. Disse skiller seg fra hverandre blant annet ved at det består av ulike måter elevene arbeider på, samt grad av lærerstyring. Hovedfokuset for denne studien, den fjerde delen av *Internett fanger*, er utformet som en liten og gul boble. At den er liten vil si at tidsrammen på opplegget er relativt kort. Fargen gul indikerer at elevene skal arbeide innenfor tydelige rammer gitt av læreren.

Hensikten med å organisere undervisningen på denne måten, er at det kan bidra til at elevene får tilstrekkelig støtte til å arbeide med oppgaven (Skaperskolen, 2020b). I neste avsnitt blir den fjerde delen av undervisningsopplegget beskrevet.

Elevene skal i denne delen av *Internett fanger* lage en modell av en hjemmeruter ved å programmere flere micro:bit til å kommunisere. Læringsmålet for undervisningsopplegget er «Programmere micro:bit til å utveksle informasjon ved bruk av radiosending». Målet at elevene skal få øvd på å kommunisere med en digital enhet (programmere), samt modellere et naturfaglig fenomen (del av internett). Elevene følger en utdelt instruksjon. Dette er en PowerPoint-presentasjon som inneholder instruksjoner for hvordan elevene skal gå frem for å løse oppgaven. De arbeider i par som skal samarbeide om å programmere på én datamaskin. Det er tenkt at elevene skal bytte på hvem som programmerer og hvem som assisterer og leser i instruksjonene. Programmeringen av micro:bit foregår på en egen nettside (Micro:bit Educational Foundation, 2022). En del av nettsiden er en blokk-editor hvor elevene lager programmene sine. Disse kan lages i enten et scratch-liknende blokkprogrammeringsspråk eller tekstprogrammeringsspråket JavaScript. Elevene kan underveis i arbeidet bytte programmeringsspråk, slik at de kan se hvordan de ulike språkene er i forhold til hverandre. Det er blokkprogrammering elevene i denne studien programmerer med.

Blokkeditoren består av flere deler (se figur 2). Til venstre er det en simulator av micro:biten. Simulatoren brukes til å teste ut hvilket resultat programkoden vil gi på micro:biten. Dermed kan programkoden testes før den overføres til micro:biten. Ved siden av simulatoren er det en kolonne hvor de ulike blokkgruppene står listet opp med tilhørende blokker elevene kan velge mellom. Til høyre i blokkeditoren er det et område hvor blokkene settes sammen. Når programkoden er satt sammen, overføres den til micro:biten ved å bruke en USB-kabel. Elevene kan lage flere prosjekter, samt lagre og hente opp andres prosjekter.



Figur 2: Blokk-editor, skjermbilde fra <https://makecode.microbit.org/>. Hentet 11.04.22

Undervisningsopplegget fulgte en plan (se Tabell 5). I starten av leksjonen får elevene utdelt instruksjonen. Denne kan de få enten ved å følge instruksjonen på en annen datamaskin, eller få en utskrevet versjon. I denne er det instruksjoner som veileder elevene i arbeidet med å løse oppgaven. Aktiviteten er elevstyrt, og læreren skal innta en rolle som veileder. Elevene starter med det samme utgangspunktet, samme oppgave, men de vil nok komme seg gjennom de ulike delene i varierende hastighet.

Tabell 4: Plan for undervisning, 4.del av Internett fanger

Tid	Innhold
5-10 min	Oppstart. Mål for timen. Utstyr. Nettsiden. Arbeidsmåte.
40 min	Arbeid med PowerPoint-instruksjoner i par/firer-grupper
10 min	Oppsummering

Etter aktiviteten med micro:bit følger det undervisningsopplegg hvor programmeringsaktiviteten skal settes i en større sammenheng. Det innebærer å lage en modell hvor micro:bit inngår. Her skal elevene presentere for hverandre hvordan de har planlagt å lage en modell med fokus på aspekter innen teknologi. Her er form og materialvalg inkludert. Beskrivelsen av dette kan leseren finne på Skaperskolens nettside (2020a). I følgende avsnitt blir det gitt en beskrivelse av hvordan lærerne forberedte seg i forkant av datainnsamlingen. Dette kan fungere som et forslag til hvordan den fjerde økten kan implementeres i undervisningen uten å gjennomføre hele undervisningsopplegget *Internett fanger*.

I forkant av gjennomføringen av dette undervisningsopplegget er det tenkt at elevene skal ha et *forarbeid*. Hensikten med forarbeidet er å forberede elevene på det de skal jobbe med. Lærerne i studien planla og gjennomførte ulike forarbeid. I klasse A og B bestod forarbeidet i en undervisningsleksjon hvor elevene skulle få en smakebit på hvordan en programkode fungerer og lære om Internett som system. Her ble noen av aktivitetene som inngår i de tidligere delene av *Internett fanger* gjennomført. Dette var blant annet en starter-aktivitet STOPP. Det er en analog programmeringsaktivitet, hvor elevene skal settes på sporet av tenkningen i programmering. Noe annet disse klassene skulle gjøre var å arbeide med en tekst om rutere. Klasse C hadde gjennomført deler av *internett fanger* forrige skoleår. I denne klassen skulle elevene bruke tid i forarbeidet på å bli kjent med micro:bit-utstyret og blokk-

editoren ved å lage enkle programmer og teste dem på micro:bit. Det er nå blitt gitt en beskrivelse av undervisningsopplegget som denne studien tar utgangspunkt i. Videre følger det en begrunnelse for metodene som ble brukt til datainnsamling.

Undervisningsopplegget er blitt undersøkt ved å gjennomføre fokusgruppeintervjuer og observasjon av elever. Disse metodene kan gi datamateriale som går i dybden, samt detaljerte beskrivelser (Johannessen, Christoffersen, & Tufte, 2016). Algoritmisk tenkning og programmering består både av praktiske handlinger (dekomponering, feilsøking..), og kognitive handlinger (vurdering, analysere..). Begge aspektene kan jeg få informasjon om ved hjelp av de valgte metodene. Ved å intervjuere elevene, kan deres refleksjoner og tenkning om det de arbeidet med i undervisningsopplegget komme tydelig frem. I tillegg til intervjuer ble det gjort observasjoner for å få data om elevenes arbeidsmåter og samhandling (Kvale & Brinkmann, 2015). Jeg erfarte at observasjonene kunne underbygge og gi gode elementer til fokusgruppeintervjuene. På denne måten kan metodene gi data som utfyller hverandre (Cohen, Manion, & Morrison, 2018). Dette blir en form for triangulering, som får frem flere perspektiver og et bedre datagrunnlag (Yin, 2018).

Datainnsamlingen ble gjort i samarbeid med en medstudent, noe som vil si at vi var ute på skolene og samlet inn data sammen. Vi har transkribert halvparten av intervjuene hver, og disse utgjør begge datamateriale. Hoveddelen av datamaterialet er hentet fra seks fokusgruppeintervjuer med elever, som ble gjennomført kort tid etter undervisningen. Det ble i tillegg skrevet observasjonsnotater både underveis og i etterkant av gjennomføringene. Hensikten med dette var å få med interessante elementer fra elevenes arbeid som var relevante å trekke inn i intervjuene. Videre beskrives utvalget.

3.2 Utvalg

Informantene går i tre ulike klasser på to forskjellige ungdomsskoler på Østlandet. Alle klassene hadde gjennomført undervisningsopplegget. Klasse A og B var 9.trinn på skole 1, mens klasse C var 10.trinn på skole 2. Det var totalt 24 elever som deltok i fokusgruppeintervjuer. I resultatene (se kapittel 5) brukes følgende pseudonymer om elevene i utvalget (Se tabell 4). Det var klassenes lærere som gjennomførte undervisningen. Under observasjonene var vi *deltakende observatører*. Det vil si at vi hjalp til ved behov, for eksempel når elevene hadde spørsmål. Mellom gjennomføringene i klasse A og B ble det

gjort noen endringer etter å ha lagt merke til noen utfordringer i klasse A. Dette gikk særlig på tilkobling og bruk av utstyret. Læreren viste elevene hvordan de kom seg inn og i gang med programmeringen på nettsiden.

Tabell 5: Utvalg med pseudonymer

	Klasse A		Klasse B		Klasse C	
	Fokusgruppe 1	Fokusgruppe 2	Fokusgruppe 3	Fokusgruppe 4	Fokusgruppe 5	Fokusgruppe 6
Par	Ane	Alice	Benjamin	Brage	Christin	Catharina
	Andrine	Astrid	Bjørn	Billy	Celine	Christian
Par	Anton	Alette	Benedicte	Bodil	Calle	Cenneth
	Alfred	Albert	Bianca	Borgny	Charles	Christoffer

Utvalgsstrategien ble gjort strategisk, fordi lærere som ønsket å satse litt på programmering ble kontaktet. Disse hadde deltatt på kurset Skaperskolen hadde, hvor det ble gitt en innføring i undervisningsopplegget. Lærerne satser litt på programmering ved at de ønsket å teste ut opplegget i sine egne klasser. Samtlige elever i utvalget har meldt seg frivillig (Se vedlegg 2). De som ønsket å delta ble videre inndelt i grupper i samråd med læreren. Her var det ønskelig å ha en blanding av jenter og gutter, elever med litt ulik erfaring med programmering og at de vanligvis pleier å være ganske muntlig aktive i undervisningen. Dette ble gjort for å få innblikk i ulike typer elevers måte å arbeide på, samt ha med elever som kunne bidra til å få i gang samtalen. I klasse C ble to elever som hadde programmering som valgfag satt sammen. Dette var for å få frem hvordan elever med mer erfaring jobber sammenliknet med dem som ikke har like mye erfaring. To og to par utgjorde fokusgruppene elevene senere ble intervjuet i. Inndelingen av disse parene er forsøkt vist ved kolonnen til venstre i tabellen hvor det står «par».

Lærerne i de valgte klassene hadde varierende bakgrunn innen programmering. Lærer på skole 1 hadde svært lite forkunnskap i programmering. Likevel takket han ja til å delta i vår studie, da han så på det som en god mulighet til å «kaste seg ut i det». På den andre skolen hadde læreren både deltatt på programmeringskurs og lært seg programmering på eget initiativ. Hun var kjent med både blokk- og tekstprogrammering, og hadde undervist en del i programmering i ulike klasser.

Læringsmiljøet i samtlige klasser virket til å være positivt. I dette legges det at klassene hadde lav terskel for å prøve ut nye ting og gjøre feil. Flesteparten av elevene var arbeidsomme og fokuserte på arbeidsoppgaven. Det var faglige samtaler som dominerte i samtlige klasser. Informantene virket til å være elever som både fikk gode og mindre gode resultater i naturfag. Utvalgets forkunnskaper i programmering var ulikt. Klasse A og B hadde ikke hatt noe særlig om programmering eller micro:bit tidligere, men noen av elevene hadde hatt programmering som valgfag. Klasse C hadde gjennomført deler av *Internett fanger* forrige skoleår, men ikke den fjerde delen som de skulle gjøre i forbindelse med denne studien.

Måten utvalget settes sammen har innvirkning på studiens troverdighet. Dette er en begrenset del av populasjonen «norske ungdomsskoleelever». Funnene/resultatene i denne studien er derfor er ikke generaliserbare, men kan si noe om hvordan det kunne blitt i liknende tilfeller. Jeg anser dette som et hensiktsmessig utvalg siden det har trekk som er overførbare til det typiske norske klasserom. I de fleste klasser vil programmering være noe elever har ulik erfaring med og interesse av, og en del elever har det som valgfag. Lærere som leser denne studien vil kunne kjenne seg igjen i lærerne som gjennomførte undervisningen. Som lærer 1 sa «Det finnes nok mange som meg [med svært lite erfaring med programmering] rundt på norske skoler». Selv om det vil variere fra klasse til klasse hvordan et slikt undervisningsopplegg med programmering vil se ut, er sannsynligheten stor for at momenter ved denne undersøkelsen kan tilføye noe til diskusjonen omkring programmering i naturfag.

3.3 Gjennomføring av datainnsamling

Datamaterialet består av data fra fokusgruppeintervjuer og observasjoner. For å teste metodene i forkant, ble det utført en pilotering. Hensikten med det var å få øvd, samt å sjekke om det vi hadde planlagt fungerte. Dette kan bidra til å styrke kvaliteten på datamaterialet. Videre vil piloteringen bli beskrevet, og deretter blir gjennomføringene av intervju og observasjon beskrevet. Intervjuene blir beskrevet først fordi dette er hovedkilden til datamaterialet.

3.3.1 Pilotering

Det ble gjennomført en pilotering i en 7.klasse på Østlandet. Det var en naturfagsøkt som varte i en time. Elevene utførte programmering i *MineCraft* på følgende nettside: <https://code.org/minecraft>. Elevene arbeidet individuelt med hver sin iPad. I etterkant av

undervisningen ble fem av elevene med i et fokusgruppeintervju. Disse hadde meldt seg frivillig. Det var et annet innhold enn den undervisningen jeg senere skulle undersøke, og elevene var også yngre enn utvalget. Informasjonen fra piloteringen var likevel overførbar. I likhet med utvalget hadde ikke «pilot-klassen» arbeidet så mye med programmering tidligere. Undervisningsopplegget bestod av en form for blokkprogrammering. De arbeidet på liknende måter som i *Internett fanger*, og derfor fikk vi testet hvordan det var å observere og intervjuere elever som programmerte.

På bakgrunn av erfaringene fra piloteringen, ble det gjort noen endringer i forhold til hvordan observasjonen ble gjennomført. Vi erfarte at fokusområdet for observasjonene ikke var konkret nok. Det ble derfor besluttet å lage et dokument med en operasjonalisering av rammeverkene som omhandler algoritmisk tenking (Shute og Udir). Operasjonaliseringen gjorde det lettere å vite hva vi skulle se etter under observasjonen, og det kunne dermed bli en mer systematisk datainnsamling. Under piloteringen deltok vi som deltakende observatører, og hadde ikke valgt ut noen konkrete elever vi skulle fokusere på. Det ble erfart at det var utfordrende å faktisk fange opp interessante observasjoner. Derfor endret vi observatørrollen ved at den skulle gjennomføres mindre deltakende, og heller mer statisk. Dette innebærer at vi hadde samme posisjon under hele observasjonen. I tillegg til dette bestemte vi oss for å fokusere på enkeltelever fremfor alle samtidig.

Piloteringen av intervjuet var også nyttig. Vi fikk øvd oss på å samtale med elever på denne måten, samtidig som at intervjuguiden ble testet og evaluert i etterkant. Formuleringene ble justert og oppfølgingsspørsmål ble lagt til for at de skulle være tydeligere og mer hensiktsmessige. Strukturen på intervjuguiden ble også endret noe. Vi erfarte at intervjuguiden bestod av mange spørsmål, og at det var utfordrende å få tid til å gjennomgå alle. Vi besluttet derfor å ha fokus på de mest sentrale temaene og reduserte antall forhåndsbestemte spørsmål.

3.3.2 Fokusgruppeintervjuer

Elevene arbeidet i par i undervisningen. Intervjuene ble derfor organiserte slik at elevene som hadde arbeidet sammen ble intervjuet i grupper. Jeg ønsket å få frem deres ulike synspunkter om det de hadde arbeidet med (Kvale & Brinkmann, 2015). Intervjuene dreide seg om undervisningsopplegget, og hovedtemaene var programmering og elevenes læring. Et slikt tematisk intervju kan gi enda mer utfyllende svar enn f.eks. et spørreskjema (Cohen et al.,

2018). Fokusgruppeintervjuer hvor elevene får delta sammen med medelever de har jobbet med i undervisningen, kan gjøre det til en tryggere setting (Walford, 2001). Det å samtale med elevene i grupper kan bidra til at flere elever tør å åpne seg, og dermed få frem flere synspunkter i datamaterialet. Elevenes innspill kan også føre til at de andre kan komme på ting de ikke hadde gjort hvis de snakket med meg alene (Johannessen et al., 2016). Samtidig er det relevant å nevne at elevene kan bli påvirket av hverandres svar, og gjøre at elever ikke nødvendigvis tør å dele synspunktene sine.

Intervjuene ble gjort kort tid etter undervisningen. Vi opplevde at dette bidro til at det var lavere terskel for å delta, samt at det var lettere å samle elevene. Alle intervjuene foregikk i et grupperom på skolen slik at elevene skulle føle seg komfortable (Walford, 2001). Varigheten på intervjuene var på mellom 45 og 60 minutter. Det var gjennomgående at elevene var ulike på hvor delaktige de var og hvor utfyllende de svarte. Min medstudent og jeg gjennomførte intervjuene hver for oss. Intervjuene ble dokumentert med lydopptak på egne lydopptakere (Kvale & Brinkmann, 2015). Disse ble testet i forkant og plassert med så kort avstand til samtlige elever i gruppene for å sikre god lyd (Johannessen et al., 2016). Ved å bruke lydopptak fremfor å måtte notere elevenes svar underveis var fordelaktig. Det ga det oss anledning til å være til stede i samtalen og stille gode oppfølgingsspørsmål. For å få engasjert samtlige elever i samtalen, fungerte jeg som ordstyrer (Kvale & Brinkmann, 2015).

Intervjuene var en form for semistrukturert intervju. Samtalen ble i stor grad styrt av deltakerne for å sørge for en naturlig flyt i samtalen (Johannessen et al., 2016). Denne formen ble også benyttet fordi den gir mulighet til å gå dypere inn i noe og få tydeligere frem hva intervjupersonen faktisk mener. Misforståelser kan oppklares, i tillegg til at det er rom for uventede svar (Cohen et al., 2018). Spørsmålene ble formulert med mål om å være minst mulig ledende, samtidig som at de skulle få samtalen til å gå i «riktig retning» av relevante aspekter tilknyttet temaene for samtalen. Derfor er de i stor grad åpne fremfor lukkede, med mål om å få frem informantenes synspunkt (Johannessen et al., 2016) Spørsmålene ble formulert med et hverdagslig språk, og med færrest mulig avanserte ord (Kvale & Brinkmann, 2015). Spørsmålene som ble stilt bestod blant annet av oppklarende spørsmål som for eksempel: «Har jeg forstått deg rett når du sier at ...?». Det var for å sørge for at elevenes tanker og refleksjoner ble forstått riktig (Drageset & Ellingsen, 2010). Det ble gjort noen refleksjoner omkring intervjuet etter første gjennomføring. Selv om vi hadde laget noen korte spørsmål i intervjuguiden, erfarte både jeg og min medstudent at vi som intervjuere snakket

mest. Vi ble derfor oppmerksomme på å holde oss så korte som mulig, og få elevene i større grad til å prate i de to gjenstående intervjuene.

I forkant av intervjuene ble det utformet en intervjuguide (Se vedlegg, intervjuguide). Den ble brukt under intervjuene for å være sikker på alle temaene ble snakket om (Johannessen et al., 2016). Det å ha en viss struktur på intervjuet kan gjøre det enklere å både gjennomføre intervjuet, samt å analysere det. Intervjuet bestod av tre hoveddeler (Se tabell 7).

Tabell 6 - intervjuets deler

Innledning:	<ul style="list-style-type: none"> - <i>Forventninger til intervjuet</i> - <i>Deltakernes tidligere erfaringer med og kunnskaper om programmering</i> - <i>Kort om intervjuobjektets forventinger</i>
Hoveddel:	<ul style="list-style-type: none"> - <i>Deltakernes tanker om selve undervisningen</i> - <i>Deltakernes refleksjon omkring læringsutbytte/ videre arbeid/ problemløsning</i>
Avslutning:	<ul style="list-style-type: none"> - <i>Eventuelt</i>

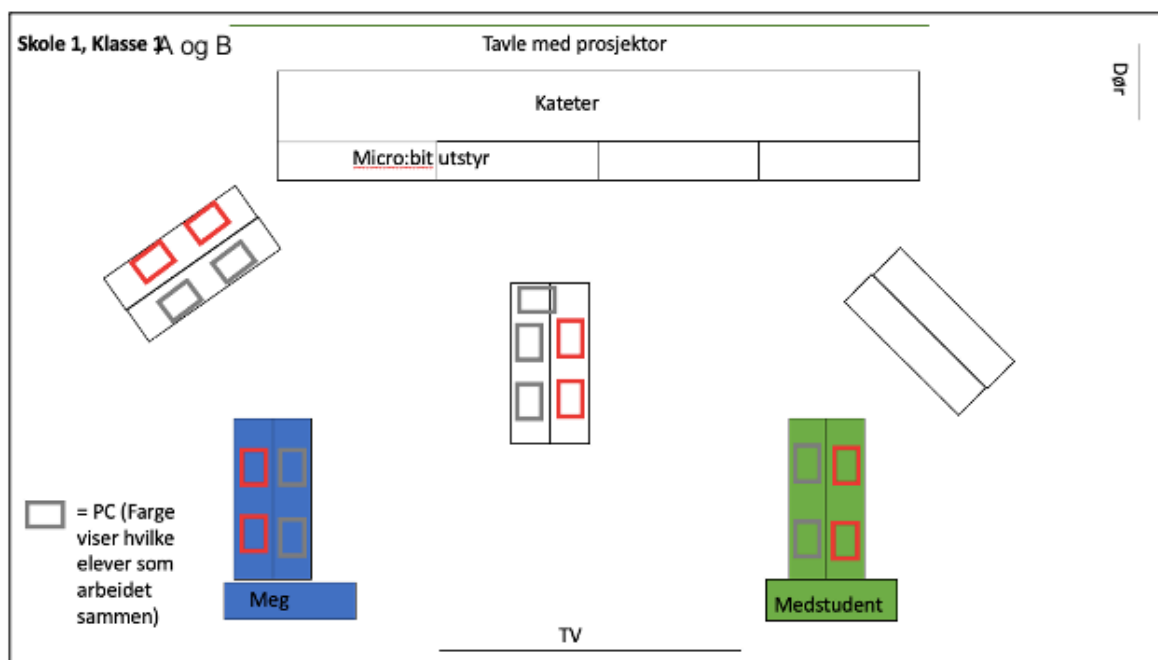
Innledningsvis ønsket jeg å ufarliggjøre situasjonen ved å presentere mine tanker om intervjuet og hvordan det ville brukes i studien (Walford, 2001). For å redusere belastningen og nervøsiteten for elevene, ble det også sagt at det ikke var noen riktige eller gale svar. Målet i innledningen var å engasjere elevene og lede dem inn mot temaene for intervjuet (Johannessen et al., 2016). I neste del var det planlagt noen hovedspørsmål knyttet til programmeringsaktiviteten og elevenes problemløsning. For å få mer utdypende svar fra elevene ble de oppfølgingsspørsmålene som var laget i forkant stilt (Cohen et al., 2018). Det ble trukket frem konkrete eksempler fra observasjonene og elevarbeidene (modellen). Dette metodiske grepet ble benyttet for å *oppmuntre til refleksjon* (Johannessen et al., 2016), og er en form for stimulert samtale (Fox-Turnbull, 2009). Avslutningsvis fikk elevene anledning til å komme med innspill de eventuelt hadde på hjertet, og ble takket for at de deltok.

3.3.3 Observasjon av elevgrupper

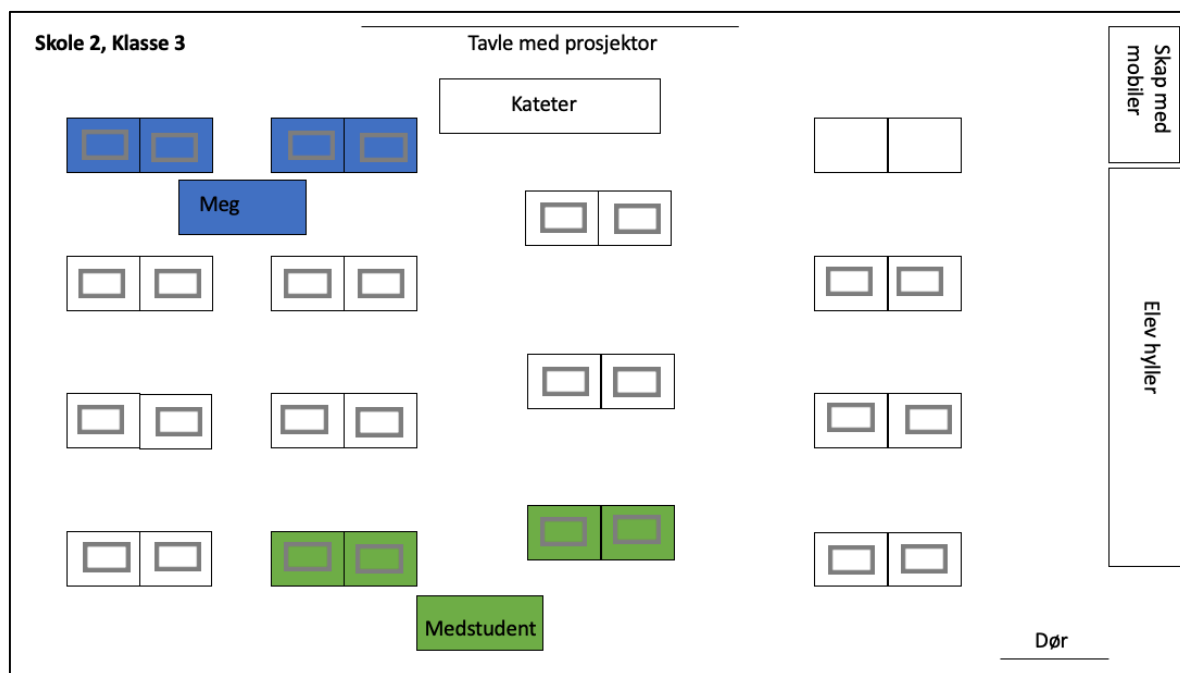
Datamaterialet ble også samlet inn ved å observere elevene som deltok i undervisningsopplegget. Det var lærerne som gjennomførte undervisningen. Vi ble i starten av undervisningen presentert av læreren, og elevene ble opplyst om hvorfor vi skulle være der. Observasjonene i klasse 1 og 2 varte i 60 minutt, mens den varte i 90 minutter i klasse 3. Til stede i rommet var det en medstudent og jeg, og vi satt i nærheten av to elevgrupper hver.

Vi intervjuet de gruppene vi hadde fokusert på i observasjonene. Som observatører var vi først og fremst passive. Samtidig ønsket vi også å ha mulighet til å stille spørsmål og snakke med elevene i situasjoner hvor vi ønsket å få mer innblikk i deres tenkning. Dette førte imidlertid til at elever blant annet stilte meg spørsmål. Jeg valgte da å få en av de elevene som hadde en del erfaring med programmering til å svare på spørsmålene. Dette bidro blant annet til å få observasjonsdata på hvordan de samarbeidet, forklarte og ordla seg.

Underveis i datainnsamlingen diskuterte vi hvordan vi bør være plassert i forhold til elevene under observasjonene. På skole 1 satt parene på gruppebord og vi observatører satt på siden av pultene til elevene (Se figur 3). Vi fikk da lett øye på elevene og måten de samhandlet på. Samtidig var det litt utfordrende å få oversikt over elevenes programmering. Dette løste vi med å bytte på hvor vi stod underveis i observasjonene. På skole 2 organiserte vi parene slik at de satt ved siden av hverandre. I tillegg plasserte vi oss bak elevene (se figur 4), noe som bidro til at vi fikk bedre oversikt over elevenes dataskjermer. Dette gjorde at jeg fikk fulgt mer med på selve programmeringen elevene gjorde. Elevene ville på denne måten også være nærmere hverandre og lettere for dem å se hverandres arbeid.



Figur 3: Oppsett klasse A og B



Figur 4: Oppsett klasse C

Det ble skrevet feltnotater i et observasjonsskjema som vi hadde utarbeidet i forkant (Se tabell 8). Dette ble brukt ved å fortløpende under observasjonene fylle ut i kolonne 1. Ting som ble tatt skrevet ned her var elevutsagn, interessante diskusjoner og samhandling mellom elevene, osv. Det som styrte disse notatene, var punkter fra valgte rammeverk. Det å få tid til å skrive ned veldig detaljerte observasjoner viste seg å være krevende, da elevene trengte en del hjelp fra oss. Vi endte opp med å hjelpe elevene en del mer enn planlagt. Samtidig ga det mye informasjon som belyste det vi var ute etter å finne ut om. Det ble lagt til utfyllende notater i etterkant av gjennomføringen.

Tabell 7: Utsnitt av observasjonsskjema

Observasjon – beskrivelse (Elevutsagn, atferd)	Tolkning/vurderinger	Begreper fra rammeverk	Viktig for intervju (avkrysning)

I etterkant av observasjonene snakket min medstudent og jeg sammen. Vi delte umiddelbare tanker og gikk gjennom feltnotatene. Ut ifra dette valgte vi ut notater og situasjoner som vi kunne ta med inn i intervjuene for å få belyst dem ytterligere. Dette var eksempelvis situasjoner hvor elevene trengte mye hjelp til å få løst oppgaven, eller hvor de gikk frem på en

spesiell måte. Ved å snakke med elevene om undervisningen i etterkant, kunne vi få høre elevenes refleksjoner om oppgaven de arbeidet med. Med dette var det tenkt at tegn på algoritmisk tenkning kunne komme til syne.

3.4 Bearbeiding av datamaterialet

Datainnsamlingen ble fullført på starten av 2022, og arbeidet med å transkribere intervjuene startet kort tid etter dette. I det følgende blir både transkriberingen og analysen av data beskrevet.

3.4.1 Transkribering

Det ble i etterkant av datainnsamlingen utført transkribering. Det vil si at lydopptakene fra fokusgruppeintervjuene ble omgjort fra lydformat til tekst (Johannessen et al., 2016).

Transkriberingen ble gjort i samarbeid med min medstudent. Jeg transkriberte de intervjuene jeg gjennomførte, og han transkriberte sine. For at dette skulle bli gjort så likt som mulig, kom vi frem til hvilke felles retningslinjer vi skulle følge før vi begynte dette arbeidet.

Intervjuene ble transkribert ord for ord slik at de skulle være så virkelighetsnære som mulig (Kvale & Brinkmann, 2015). Vi brukte komma og punktum for å markere pauser, og inkluderte ordlyder som «ehm» og påbegynte setninger. Uttalelsene ble ikke skrevet om til et mer formelt språk. Det gjorde vi for å unngå at meningen ble endret underveis i analysen.

Selv om dette kan gjøre transkriberingen noe vanskeligere å lese, vil utsagnene i større grad være virkelighetsnære. Det gir leseren anledning til å gjøre seg opp en mening om hva som legges i det elevene sier (Kvale & Brinkmann, 2015). For å unngå at transkripsjonene skulle inneholde feil, ble de korrigert ved å lese grundig gjennom dem flere ganger samtidig som det ble lyttet til lydopptakene. Vi sa ifra til hverandre om vi oppdaget mangler i hverandres transkripsjoner. Dette gjorde vi for å sikre nøyaktigheten/reliabiliteten på transkripsjonene (Postholm, 2005).

Transkripsjonsarbeidet er begynnelsen av analysen, siden jeg blir kjent med datamaterialet (Kvale & Brinkmann, 2015). Underveis i dette arbeidet, oppstod det tanker fra intervjuene og aspekter ved situasjonene som jeg noterte. Dette ble skrevet ned på et eget dokument, til videre bruk i drøftingen av funnene. Dermed er det starten på å skape mening ut av det elevene har sagt (Kvale & Brinkmann, 2015).

3.4.2 Operasjonalisering av rammeverk

Videre redegjøres det for prosessen med å operasjonalisere rammeverkene. Hensikten med dette er å beskrive hvilke briller datamaterialet ble analysert med.

Hensikten med studien er å undersøke *hva* undervisningsopplegget tilrettelegger for at elever lærer. Undervisningsopplegget er tilknyttet kjerneelementet Teknologi, samt følgende kompetansemål i naturfag: «*elevene skal kunne utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker*» og «*bruke programmering til å utforske naturfaglige fenomener*» (Utdanningsdirektoratet, 2020a). Dette er altså en del av kompetansen elevene skal utvikle, og dreier seg om både *kunnskap* og *ferdigheter* som elevene skal tilegne seg og anvende. Hva som legges i begrepene kunnskap og ferdigheter er beskrevet i kap. 2.2.1. Som en del av analyseverktøyet ble disse delene av LK20 inkludert. Det var for å kunne si noe om elevene får tilegnet seg og anvendt det som var tiltenkt i planleggingen av undervisningsopplegget.

I analysen er det forsøkt å trekke ut atferd og utsagn som kan si gi informasjon omkring elevenes kunnskap. Et eksempel på elevutsagn som inkluderes innen kunnskap er: «*Rutere og sånne sendere. Sånne som sender for eksempel wifi da, eller dekning til forskjellige små rutere... Og lagring, base og sånt*». Her blir flere begreper som inngår i det teknologiske systemet *internett* brukt. Fakta forstås her som sammenhenger og forklaringer. Elevene trenger begreper for å beskrive et «*faktum*». «*Ruter videresender signaler*» er et eksempel på faktakunnskap hvor kunnskapen uttrykkes ved hjelp av flere begreper. Naturfaglig kunnskap som modeller og hvilke deler som inngår i systemet *internett* plasseres også innunder kunnskap.

Det ble også analysert for å få frem tegn på elevenes algoritmiske tenkning (AT). Den forståelsen det hovedsakelig brukes her, er rammeverket til Shute mfl (2017). Dette rammeverket er basert på en studie av ulike måter å definere algoritmisk tenkning. De kom frem til noen hovedkomponenter, såkalte fasetter. Fasettene er beskrevet detaljert, noe som kan gjøre det lettere å få frem atferd og utsagn som inngår i de ulike. Rammeverket er ment å bli brukt i utviklingen av pedagogikk og vurdering tilknyttet AT uavhengig av alder og fag (Shute et al., 2017). Den er ikke like fagspesifikk som rammeverket til Weintrop (2016). Det vil si at Shutes fasetter ikke kun er fokusert på realfag og programmering. I likhet med opplegget jeg undersøkte, har Shute med eksempler fra grafisk programmeringsspråk, Scratch

(Shute et al., 2017). Shutes rammeverk er passende for programmeringsaktiviteter, samt kreative problemløsningsoppgaver. Modellering inngår det i komponenten abstraksjon i algoritmisk tenkning. Både programmering og modellering er det elevene skal arbeide med i undervisningsopplegget. Derfor vil Shute mfl. sitt rammeverk være en passende måte å få frem elevenes problemløsingstilnærming.

Det ble i forkant av observasjonen laget en tabell (Se tabell 12) med handlinger og utsagn som vil inngå i de ulike fasettene. Disse konkrete eksemplene ble utformet ut ifra samtale med veileder og ved å undersøke hvordan andre hadde gått frem for å vite hvilke eksempler som inngår.

I tillegg til dette rammeverket, ble også *den algoritmiske tenkeren* (Utdanningsdirektoratet, 2019a) brukt. Den består av nøkkelbegreper og arbeidsmåter, men rammeverket er ikke like detaljert beskrevet som i rammeverket til Shute mfl. Det at dette rammeverket inkluderer arbeidsmåter, gjør imidlertid at data fra observasjonene ble lettere å beskrive. Dermed utfylte disse rammeverkene hverandre. I kapittel 2.2.6 er det en ytterligere beskrivelse av algoritmisk tenkning. Da intervjuguiden ble utformet, ble disse rammeverkene brukt. Det var altså med på å styre innholdet i intervjuene. Ifølge Kvale og Brinkmann (2015) vil det derfor være hensiktsmessig å også bruke dem i analyseringen av transkripsjonene. Det var altså rammeverkene som var utgangspunktet for hvordan bearbeidingen av datamaterialet ble gjort. Videre skal den analytiske fremgangsmåten beskrives.

Tabell 8: Operasjonalisering av rammeverk

Fasett	Dette kan se ut som dette i klasserommet
<p><i>Dekomponering av problemer</i> bryte ned et komplekst problem i mindre deler og bruke systematiske prosesser for å håndtere hver av de mindre problemene.</p>	<ul style="list-style-type: none"> • Kategorisere hvilke deler hjemmeruteren består av • Hvordan elevene går frem for å få oversikt over «problemet». • Sørgе for at hver del fungerer og kan løses individuelt • Forstår elevene hva oppgaven spør om? • Oppskrift • Flytskjema
<p><i>Abstraksjon</i> å finne mønstre i ulike problemer og løsninger.</p>	<ul style="list-style-type: none"> • Få frem det mest sentrale, ignorerer irrelevant/overfladisk informasjon • Bruke elementer på nytt
<p><i>Generalisering</i> ut ifra de mønstrene man oppdager skal man lage mer overordnede løsninger for grupper med liknende problemer.</p>	<ul style="list-style-type: none"> • Bruker f.eks. looper istedenfor å ha mange like blokker etter hverandre • Gjenkjenne mønstre og sammenhenger • Se at deler av programsekvensene kan brukes på nytt, kan kopieres og justeres • Se hva andre har gjort, og trekker ut det som kan brukes i sin egen oppgave.
<p><i>Algoritmisk design</i> utvikling av gjenbrukbare verktøy eller prosedyrer for å løse grupper med problemer.</p>	<ul style="list-style-type: none"> • Lage en algoritme, og et program som kjører algoritmen
<p><i>Feilsøking/debugging</i> gå gjennom programkodene for å finne ut hvor det oppstår feil. Hva som får programmet til å gi feil utfall/resultat.</p>	<ul style="list-style-type: none"> • Teste ulike blokker, se hva som skjer i «simuleringen» • Ser de på helheten først før de retter opp feilen? • Går de rett inn i programmet, korrigerer feilen og kjører det på nytt? • Prøve seg frem? • Feil som skyldes feil plassering eller brukt av programstrukturer • Logiske feil: Får feil svar, men programmet virker
<p><i>Iterasjon</i> henger sammen med debugging. Er en systematisk feilsøking som sikrer at hver av de mindre delproblemene løses på en effektiv måte, og uten noen <i>loose ends</i>.</p>	<ul style="list-style-type: none"> • Trinnvis gjennomgang av programmet • Se for seg hva som kommer til å skje, og deretter «bygge».

3.4.3 Analyse

Etter fullført transkribering, gikk jeg systematisk gjennom dem. Jeg ønsket å få frem og gå i dybden på meningen i det elevene sa (Kvale & Brinkmann, 2015). En fordel med case studier, er at den åpner opp for en fleksibilitet i hvordan analysen foregår. Likevel er det nødvendig å beskrive den generelle analytiske strategien som ble benyttet (Yin, 2018) noe jeg skal gjøre i følgende avsnitt.

Det er ulike måter data kan analyseres. Her er det hovedsakelig tematisk analyse som analysen er inspirert av. Brown og Clarc (2006) har utarbeidet en trinnvis fremgangsmåte for analyse av data tematisk. Denne ble ikke fulgt til punkt å prikke, men bidro til å gjøre analysen systematisk. I likhet med tematisk analyse, startet jeg med å lese gjennom transkripsjonene flere ganger. De ble betraktet hver for seg og på tvers ved å sammenlikne hva de ulike elevene sa i de ulike fokusgruppeintervjuene. De ble undersøkt ved å se på de ulike intervjuene som en helhet (Kvale & Brinkmann, 2015). På denne måten ble jeg godt kjent med datamaterialet. Videre i neste fase ble datamaterialet kodet.

Kodingen ble gjort i programmet NVivo. Måten kodingen foregikk på, var ved å først gå gjennom transkripsjonene og se etter meningen i lengre utdrag fra intervjuene. Å markere mer enn enkeltsitater, ville gjøre at konteksten til uttalelsene ble bevart. Utdragene ble gitt en kode ut ifra hvilke overordnede temaer det dreide seg om. På denne måten brøt jeg teksten opp i mindre enheter (Kvale & Brinkmann, 2015). Denne prosessen med å markere utdrag var først og fremst *begrepsstyrt* (Braun & Clarke, 2006). Det var først og fremst aktuelt å se etter spesifikke tegn på elevenes kompetanse, og rammeverk, som tidligere beskrevet, ble derfor benyttet. Dette var til hjelp for å skjelne relevant data, og styrte hva og hvordan utdrag ble markert. I tematisk analyse går det også an å være mer datastyrt, som andre kaller induktiv koding. Dette ble benyttet i denne analysen for å kunne være åpen for hva som kom frem i data. Derfor ble det også utviklet noen koder i løpet av gjennomlesningen og gjennom tolkning av data (Kvale & Brinkmann, 2015). Kodingen var en systematisk måte å redusere datamaterialet på, og jeg kunne dermed konsentrere meg om data som omhandlet temaene for studien (Andersen, 2013).

Utdragene i de ulike kodene ble deretter lest gjennom. Jeg så etter trender i datamaterialet, og var samt åpen for å oppdage nye uventede begreper innen hver kode. Videre ble disse kodene redusert til noen få samlende kategorier. I kategoriseringen benyttet jeg etablerte begreper fra

rammeverkene. Samtidig ble også noen kategorier hentet ut ifra det elevene snakket om og som ikke passet i rammeverkskategoriene. Denne sorteringsmetoden var en god måte å få oversikt over datamaterialet (Kvale & Brinkmann, 2015). Etter kategoriseringen gikk jeg gjennom teoretiske begreper fra litteratur og funn fra studier som hadde likhetstrekk med min (Leseth & Tellmann, 2018). Dette var for få frem interessante sider ved data, og for å for eksempel undersøke om det var trender som skilte seg fra etablert teori og/eller tidligere forskning.

I tillegg til å kode og kategorisere på denne måten, gikk jeg gjennom datamaterialet på tvers. Spørsmål og svar fra de ulike intervjuene ble sortert i en tabell (Se tabell 8). Hensikten med det var å få mer oversikt over det som ble sagt i de ulike intervjuene. Utsagnene i tabellen ble kodet ut ifra rammeverkene og måten Udir definerer kunnskaper og ferdigheter. Denne prosessen ble gjort for å få frem innsikt som ikke lå direkte i intervju spørsmålene og elevutsagnene, altså en form for fortolkning av data (Kvale & Brinkmann, 2015). Det ble forsøkt å komme frem til interessante aspekter, slik som: «Ut fra teorien høres det ut som eleven snakker om [et tema/en kategori], det eleven sier tyder på at han/hun ...» Dette kan videre gi samlende kategorier som sier noe om elevenes algoritmiske tenkning og elevenes læring.

Tabell 9: Utsnitt av svar fokusgr.intervju

Fokusgr	1 (fg nr1, skole 1)	2 (fg nr2, skole 1)	3(fg nr3, skole 1)	4(fg nr4, skole 1)	5 (fg nr5, skole 2)	6 (fg nr6, skole 7)
Spørsmål	Har dere lært noe om programmering før?					
	<p>Nei</p> <p>Nei</p> <p>J2: Jo. Jeg hadde det litt på barneskolen.</p> <p>...</p> <p>J2: <u>eh</u>, vi hadde det litt i matematikk og sånn. så lærte vi det litt, <u>eh</u> og litt i naturfag</p> <p>...</p> <p>I: Hva lærte dere om da?</p> <p>J2: <u>ehh</u>.. vi fikk <u>i</u>.. vi lærte egentlig litt, vi gjorde egentlig litt, vi programmerte litt hvordan biler kan gå fremover og sånn.</p>	<p>J1: Vi har hatt litt.</p> <p>J1: Nei, men vi har hatt litt i timen, men jeg føler ikke jeg kan så mye.</p> <p>J1: Det er et litt nytt tema.</p> <p>J1: Da har det vært sånn én, ikke to som man kobler sammen.</p> <p>I: Så da har dere brukt <u>microbit</u> faktisk?</p> <p>J1: Ja, eller jeg tror vi gjorde det på barneskolen eller noe sånt eller her, <u>ihvertfall</u> vært borti det før.</p> <p>J2: Vi har bare hatt det i timen.</p>	<p>Elev 10: <u>Jeha</u> jeg hadde litt om det på barneskolen husker jeg, men det var noen få <u>timer</u> men ikke veldig mye.</p> <p>...</p> <p><u>Ehnia</u>, jeg hadde også det i barneskolen. Også lagde sånne dere små <u>roboter</u> som skulle da flytte sand.</p> <p>...</p> <p>Vi hadde vel bitte litt en mattetime her, men jeg har jo programmering på valgfag.</p> <p>...</p> <p>Jeg har ikke hatt noe programmering <u>før</u>... Så det er ganske nytt.</p>	<p>Elev 16: Jeg har aldri lært programmering jeg</p> <p>Elev 13: Jeg har <u>ehhm</u> gått på programmering i sommerferien og hatt litt programmering <u>pååå</u> ungdomsskolen som valgfag og programmering i skolens pensum på barneskolen</p> <p>Elev 14: Jeg hadde litt på barneskolen.</p> <p>15: <u>Ehh</u> jeg hadde litt programmering i fjor på <u>valgfag</u>...</p>	<p>Elev 2: Ja, altså jeg har jo programmering som valgfag her på skolen, og har så og si hatt det siden jeg fikk liksom, fra jeg kunne ta det</p> <p>Elev 1: <u>Ehh</u>.. Ja. Jeg har ikke drevet så mye med det, med det enn det vi har gjort på skolen, så jeg kan ikke så mye om det #00:02:45-0#</p> <p>I: Nei. Men hva er det dere har hatt om på skolen da? #00:02:48-7#</p> <p>Elev 1: <u>Ehm</u>, mest sånn <u>derre</u>, <u>micro</u>... bit ting #00:02:52-3#</p>	<p>Elev 21: <u>Ehhh</u> ja en gang. #00:00:31-8#</p> <p>Elev 22: <u>Ehja</u>. #00:00:33-3#</p> <p>Elev 24: Jeg har jo gått <u>ehh</u> i 8 klasse så hadde jeg teknologi i praksis så hadde vi veldig mye <u>ehh</u> programmering da. #00:00:41-3#</p> <p>Elev 23: <u>Mhm</u>, samme her. #00:00:42-5#</p> <p>Intervjuer: Men du også hadde hatt det tidligere..</p>

Feltnotatene som ble skrevet under observasjonene ble også analysert. Kunnskaper og ferdigheter, samt arbeidsmåter og utfordringer som oppstår ble trekt ut av disse notatene. Dette ble brukt i kombinasjon med transkripsjonene for å besvare forskningsspørsmålene. Underveis i analysen har kodene og kategoriene blitt diskutert med veileder og medstudent. I tillegg til flere samtaler med noen av deltakerne i TRELIS-forskningsprosjektet. Det oppstod uenigheter og noen uklarheter. Disse ble diskutert, og ledet frem til en justering vi ble enige om at var dekkende.

I analysen ble det utledet temaer og undertemaer, som er presentert i tabell 10 og 11. Måten disse temaene er utledet, er fra rammeverkene til Shute mfl. (2017) og Udir (Utdanningsdirektoratet, 2019a). I tillegg ble det kombinert med det som kom frem fra elevenes handlinger og utsagn både fra observasjonene og intervjuene. Måten temaene ble utarbeidet var et resultat av en form for både deduktiv og induktiv koding av datamaterialet. Elevutsagnene som er oppført i tabell viser eksempler på hvilke elevutsagn som ble tatt med.

Tabell 10: Temaer og undertemaer RQ1

Temaer	Undertemaer	Beskrivelse av undertema	Elevutsagn
Teknologi	Begrepsbruk	Hvilke og hvordan elevene bruker begreper	«Rutere og sånne sendere»
	Modeller	Hvordan dette kan brukes for å representere det naturfaglige innholdet	«Vi har hatt sånn atommodeller. At det er litt forenkla»
Programmering	Forklaring og relevans	Hva programmering er, hva det brukes til	«Lage en enhet som hører på deg»
	Teknisk	Omtale og bruk av programmeringstekniske konsepter og begreper (funksjoner, algoritme, løkker, betingelser..), programmeringsspråk	«Hvis du vil ha den hele tiden, så tar du den som sier «gjenta» hele tiden»
Problemløsning	Feilsøking/ debugging	Elevenes ulike tilnærminger til å finne feil. Prøve-og-feile.	«Du må nesten bare prøve deg fram helt til du klarer det da.»
	Effektivisering av kode	Effektivisering av programmeringsprosessen. Dekomponering, generalisering, automatisering.	«Vi tok mye av det som var på den forrige oppgaven inn på den nye og bare endra»

Tabell 10: Temaer og undertemaer RQ2 (Arbeidsprosess?)

Temaer	Undertema	Beskrivelse
Utfordringer	Undervisningsmateriell	Det kommer frem aspekter ved resursene elevene arbeider med.
	Elevenes forkunnskaper	Hva elevene kan eller kjenner til fra før.
Arbeidsmåter	Samarbeid	Måten elevene arbeider med hverandre på
Rammebetingelser	Tid	Lengden på opplegget
	Lærerens forkunnskaper	Lærerens erfaringer med temaet. Dette gjelder spesielt innen teknologi og programmering.

3.5 Etske vurderinger

Reliabilitet dreier seg om påliteligheten til datamaterialet. At man kan gjenta studien og få det samme resultatet, inngår i dette. Min studie vil ikke være mulig å gjennomføre helt likt av andre forskere. Den er situasjonsavhengig og verdiladet (Johannessen et al., 2016). Sentrale forskningsetiske prinsipper har jeg forholdt meg til i arbeidet med masteroppgaven. I forkant av gjennomføring av datainnsamling, ble det metodiske opplegget godkjent av NSD Norsk senter for forskningsdata. Godkjennelsen ligger i vedlegg 1.

Elevene fra de tre klassene som deltok i prosjektet, har jeg et særlig ansvar for. Studiens formål og bruk av datamaterialet fikk de informasjon om både muntlig og skriftlig, og de har skrevet under på et samtykkeskjema. I vedlegg 2 finner leseren informasjons- og samtykkeskriv. Bruk av datamaterialet, anonymisering og taushetsplikt var elementer elevene fikk vite her (Dalland, 2017). Det var lærerne som delte ut og samlet inn disse. Elevene fikk god tid på å lese gjennom og de kunne stille spørsmål til læreren og oss masterstudenter. De kunne trekke seg når som helst i undersøkelsen, for å sikre deres selvbestemmelse og autonomi (Christoffersen & Johannessen, 2012). I samtykkeskjemaet kunne deltakerne krysse av for ulike punkter ettersom hva de ønsket å delta på (Se vedlegg).

Anonymiteten til elevene er bevart ved at jeg ga dem pseudonymer og personidentifiserende elementer ble ikke tatt med i transkripsjonene. Lydopptakene er kun delt med min medstudent. Forskningsgruppen TRELIS har kun fått de anonymiserte transkripsjonene og

notater. De observasjonene som ble gjort og fokusgruppeintervjuene har ikke noe innvirkning på elevenes faglige vurdering i naturfag. Dermed vil ikke elevene kunne bli gjenkjent eller vurdert på bakgrunn av sin deltakelse og uttalelser om undervisningsopplegget. Dermed vil jeg si at elevene har gitt sitt samtykke informert og frivillig, uten tvang. At elevene kunne trekke seg på hvilket som helst tidspunkt i gjennomføringen av både observasjon og intervju ble tydelig formidlet i forkant.

For å dokumentere observasjonene i tillegg til observasjonsnotater, ble det av hensyn til personvern besluttet å kun benytte lydopptak. Det ble ansett som tilstrekkelig verktøy, og da kunne vi unngå å filme elevene. Det var også tidsbesparende. Opptakene og transkripsjonene av disse ble lagret trygt på en kryptert minnepenn. Informantenes identitet ble anonymisert allerede i transkripsjonsprosessen (Kvale & Brinkmann, 2015). Noe som er relatert til den etiske vurderingen av studien er dens troverdighet og kvalitet. Dette følger i neste delkapittel.

3.6 Vurderinger av studien

Utførelsen av forskningen må gjøres på en håndverksmessig god måte, og at den er til å stole på. Alle som holder på med forskning har et ansvar for at det som publiseres er sant. Videre presenteres noen av forskningsdesignets og gjennomføringens svakheter, og grep som er gjort for å styrke troverdigheten til resultatene.

Det er ulike måter å vurdere gyldigheten til forskning. Tatt antall informanter i betraktning, har jeg besluttet å forholde meg til begrepet «troverdighet», istedenfor «validitet» og «reliabilitet». Disse begrepene brukes om gyldighet og generaliserbarhet innen kvantitativ forskning. I denne studien er det et relativt lite utvalg, og er en form for kvalitativ forskning. Ifølge Rolfe (2006) er «troverdighet» det overordnede begrepet for gyldighet, pålitelighet og overførbarhet innen kvalitativ forskning. Troverdighet sier noe om hvor plausible (?), sannsynlige og forsvarlige resultatene er. Her er det naturlig å vurdere i hvilken grad fremstillingen av datamaterialet er blitt gjort korrekt og at analysen ikke får frem mening som ikke er der.

Som en del av forberedelsen av datainnsamlingen, ble det også brukt tid på å sette seg inn i temaet. Dette ble gjort på ulike måter. Aktuell litteratur og tidligere forskning (se kap. 2) ble undersøkt. Noen lærere i mitt nettverk som underviser i programmering, har delt av sin

kunnskap og erfaringer. De samtalene dette ble uttrykt, bidro til at jeg fikk flere tanker rundt hvilke temaer og vinklinger som kunne være hensiktsmessige å undersøke i studien. Det ble også brukt tid til å øve på å bli kjent med utstyret og blokk-editoren som brukes i undervisningsopplegget. Dette ble gjort for å lettere kunne gjenkjenne arbeidsmåter og utfordringer de vil kunne støte på. Dette vil kunne gjøre det lettere å samle inn data som faktisk belyser forskningsspørsmålene. Denne delen av forberedelsen av datainnsamlingen, bidro til å utvikle min forståelseshorisont. Denne er med på å prege hva som blir vektlagt og hvordan dette tolkes, samt gjøre datainnsamlingen mer målrettet (Johannessen et al., 2016).

Kvaliteten på transkripsjonene er av personvern hensyn ikke mulig for leseren å avgjøre, men det at både min medstudent og jeg har gått gjennom transkripsjonene nøye kan øke kvaliteten. Ved å samarbeide om dette, ønsket vi å fremstille informantenes refleksjoner på en så korrekt måte som mulig (Christoffersen & Johannessen, 2012). Det at vi som transkriberer følger de samme skriveprosedyrene er viktig for reliabiliteten til datamaterialet, og videre funnene i studien. Det vil nemlig ha innvirkning på hvordan informantenes utsagn tolkes (Kvale & Brinkmann, 2015). Dersom leseren er interessert i å kontrollere datamaterialet mitt og resultatene, kan transkripsjonene bli gjort tilgjengelige ved forespørsel. Det at jeg har gitt en detaljert beskrivelse av de tre ulike gjennomføringene, observasjonene, kan sette de inkluderte sitatene i en større sammenheng. Elevenes uttalelser kan få en dypere mening dersom man får informasjon om hvordan jeg observerte at de arbeidet i undervisningsopplegget.

Studiens antall informanter gjør at resultatene ikke direkte kan generaliseres til andre klasserom på andre skoler. Som hensikten til studien tilsier, er ikke dette er mål i seg selv. Det var derfor ønskelig å gå i dybden på ett opplegg, og få gjennomføringer. På denne måten kunne det komme frem detaljert innsikt om det enkelte tilfellet (Andersen, 2013). Resultatene vil kunne overføres til en viss grad til andre, lignende settinger. Det er her beskrivelse av utvalg og metoden kommer inn. Dette ble gjort ved å være så detaljert i beskrivelsen av utvalget som mulig. Dette gjelder også for beskrivelsen av datainnsamlingen fra observasjonene og intervjuene. Målet er å beskrive konteksten på en grundig måte, kan leseren selv vurdere i hvilken grad funnene kan anvendes i sin egen situasjon (Walford, 2001).

Det er flere grep som er blitt gjort for å øke studiens kvalitet. Undervisningsopplegget ble belyst ved å ta i bruk flere metoder, både intervju og observasjon. Det å triangulere på denne

måten, vil gi detaljerte beskrivelser og få frem flere sider av undersøkelsesobjektet (S. 126 (Yin, 2018)). Dette styrker studiens indre validitet, og er blant fordelene med case studier. Det valgte undervisningsopplegget, undersøkes mer detaljert. (Andersen, 2013) s.14

I kvalitative studier vil datamaterialet utledes ut ifra den spesifikke situasjonen studien foregår i. Derfor vil datamaterialet ikke kunne gjentas av andre forskere i etterkant. Undersøkelsen kan ikke bli gjentatt med akkurat samme omstendigheter og betingelser (Andersen, 2013) Det er her den detaljerte beskrivelsen som er gitt i metodekapittelet av hvordan datainnsamlingen foregikk, er avgjørende. Det gir leseren et innblikk i hva som ble gjort og begrunnelser. Dermed vil resultatene likevel kunne bli sammenliknet av andre forskere som har gjort liknende datainnsamling (Kvale & Brinkmann, 2015).

For å få tilstrekkelig informasjon, rikere data og mer dybde, ble undervisningsopplegget undersøkt i tre klasser med to ulike lærere (Yin, 2018). Begge lærerne gjennomførte opplegget ut ifra den samme planen. Det at det er bredde blant lærerne som gjennomførte undervisningsopplegget, gjør at sannsynligheten for at leseren kan overføre resultatene er større. Flere vil kunne kjenne seg igjen. Alle lærere står overfor programmering som nytt tema å inkludere i undervisning, og «alle» har tilgang til dette undervisningsopplegget. Innsiktene vil kunne være interessante for andre lærere, og kan ha overføringsverdi. Det at også utvalget bestod av elever med ulike forkunnskaper, bidrar også til denne bredden. Måten dette er blitt gjort håper jeg gir leseren nok informasjon til å selv kunne vurdere om forskningen har fellestrekk med sammenhengen den selv skal sammenlikne studien min med. Dermed styrkes studiens kvalitet ved at utvalget velges strategisk (Johannessen et al., 2016)

Kvaliteten på datainnsamlingen som ble gjort gjennom intervjuene er styrket ved å utarbeide intervjuguiden på en grundig måte. Den ble laget i samarbeid med min medstudent, og i dialog med veilederne våre. Den ble justert flere ganger. Det har vi gjort med tanke på at spørsmålene skal være så godt formulert som mulig, at rekkefølgen skal være hensiktsmessig, unngå å gjenta de samme spørsmålene, overlapp. Det gjør oss også godt forberedt til å stille dem, gjennomføre intervjuet. Har mentalt gått gjennom intervjuet flere ganger før vi gjennomfører datainnsamlingen. Har sett for oss hva elevene kan komme til å svare, hvordan besvare dem på en god måte, relevante oppfølgingsspørsmål. Oppfølgingsspørsmålene bidro også til å rette opp i eventuelle misforståelser eller feiltolkninger av elevenes utsagn (Kvale & Brinkmann, 2015). Elevenes uttalelser ble utdypet ved å være aktiv i samtalen og stille

oppfølgingsspørsmål. Dermed sikret jeg at elevene svarte på en måte som ga tilstrekkelig informasjon som kunne gi relevant data (Andersen, 2013).

Piloteringen av datainnsamlingen, test-intervjuer og -observasjoner i forkant av selve gjennomføringen, ble gjort for å styrke undersøkelsene. Ut ifra erfaringene fra piloteringen ble de faktiske undersøkelsene justert. Piloteringen bidro også til at jeg som forsker fikk anledning til å øve på rollene som observatør og intervjuer. Det øker sannsynligheten for at datainnsamlingen ble gjort på en systematisk måte, som videre øker troverdigheten (Newby, 2014). Videre vil tolkningen av svarene variere avhengig av intervjuerens «briller». For at det skal bli transparent, bør det tydelig skilles mellom hva intervjuobjektet har sagt, og hvordan det tolkes. En slik bevissthet er viktig tilknyttet analysen og drøftingen av studiens resultater (Dalland, 2017).

Intervjuene ble gjennomført i fokusgrupper, noe som gjør at den enkelte elev ikke settes i fokus. Målet er å få frem ulike elevers synspunkter. Delaktigheten på de ulike elevene var litt ulik, slik at det er mulig at noen synspunkter ikke kom med i datainnsamlingen.

I kvalitative studier vil forskerens forkunnskaper og forforståelse ha innvirkning på hvordan datainnsamlingen blir gjort og hvordan datamaterialet blir analysert i etterkant. Mine «briller» (de teoretiske rammeverkene jeg har lest meg opp på) vil ha innvirkning på hva jeg legger merke til, hvordan dette tolkes av meg. Det kan tenkes at interessante elevaktivitet eller - utsagn kan ha «gått tapt» som en konsekvens av at det var noe annet jeg hadde valgt å se etter. Min evne til å observere og fokus for observasjonene vil gjøre at noen hendelser vil bli notert og andre ikke (Kvale & Brinkmann, 2015).

4. Resultater

I dette kapitlet gis det en beskrivelse av resultatene fra datainnsamlingen. Analysen datamaterialet ledet til noen hovedtemaer og undertemaer for hvert forskningsspørsmål. I vedlegg 5 finner leseren en oversikt over hovedfunnene. Undervisningsopplegget ble gjennomført i tre ulike klasser, og disse gjennomføringene utartet seg litt forskjellig. Derfor består det første delkapitlet av en presentasjon av hovedtrekkene ved gjennomføringene. Dessuten er de ment til å gi kontekstualiserende data. Deretter trekkes det frem data som også omhandler intervjuene. Det er rekkefølgen på følgende forskningsspørsmål som styrer strukturen på denne delen av resultatkapitlet:

RQ1: Hvilke kunnskaper og ferdigheter tilknyttet kjerneelementet *teknologi* uttrykker elevene?

RQ2: Hvordan arbeider elevene med programmeringsoppgaven, og hvordan tilnærmer de seg eventuelle utfordringer som oppstår?

4.1 Observasjon av gjennomføringene

Hovedhensikten med observasjonene var å samle inn kontekstualiserende data til intervjuene. Det var også for å danne seg et helhetsbilde av undervisningsforløpet. De tre klassene hvor observasjonene ble gjennomført, hadde litt ulike trekk. I det følgende gis det et sammendrag av de tre gjennomføringene. Hensikten med dette delkapitlet er å gi leseren et innblikk i hvordan elevene i de ulike klassene arbeidet. Dette skal videre gi en kontekst for det datamaterialet fra fokusgruppeintervjuene som blir gitt senere. På denne måten ønsker jeg at det elevene sier i intervjuene kan forstås på en dypere måte. Denne form for triangulering kan bidra til å si noe mer om elevenes kompetanse.

4.1.1 Klasse A

Parene som ble observert i denne klassen var arbeidsomme og ivrige, og da de fikk instruksjonene ble de raskt fokuserte på å løse oppgaven. Selv om programmering var nytt for mange av elevene, gikk de i gang med friskt mot. Under observasjonen av måten elevene snakket sammen på og diskuterte, fikk jeg inntrykk av at de fleste par fungerte samarbeidet greit. Mesteparten av tiden var elevene fokusert på oppgaven, men mot slutten ble elevene gradvis opptatt av ikke-faglige ting.

Mye tid i starten av timen gikk til at elevene ble kjent med utstyret. Tekniske feil og bruk av utstyret var noe som preget det elevene arbeidet med. I arbeidet gikk elevene frem ved å gjøre det som stod på de ulike sidene av PowerPoint-presentasjonen. Denne var styrende for hva og hvordan elevene programmerte. Elevene brukte litt tid på å få oversikt over hvilke blokker som fantes i blokk-editoren, og på å finne de blokkene som var anvist i instruksjonen. Det var få situasjoner hvor elevene testet andre blokker enn det instruksjonen ba om. Elevene snakket sammen innad i parene og mellom parene for å dele tanker og ideer til hvordan de kunne løse oppgaven. Det var gjennomgående at mange av elevene var ganske avhengige av støtte. Kort tid etter at elevene hadde prøvd å løse oppgavene i instruksjonen, stoppet de opp og trengte hjelp til å komme seg videre. Dette fikk de ved å høre med læreren, observatører og/eller medelever. Det oppstod en del usikkerhet rundt hvordan programmeringen fungerte. Elevene jobbet i litt ulikt tempo. Det var ikke alle parene som ble ferdig med oppgaven. Mye av tiden gikk til de første deloppgavene.

4.1.2 Klasse B

Det var den samme læreren som ledet undervisningopplegget i denne klassen som i klasse A. Etter en evaluering av undervisningen i klasse A ble det gjort noen justeringer av deler av undervisningen med tanke på at elevene skulle komme raskest mulig i gang med selve programmeringen. Læreren valgte denne gangen å bruke SmartBoard til å vise hvordan utstyret skulle kobles opp og viste elevene micro:bit-nettsiden. Det ble observert at elevene i denne klassen kom raskere i gang med selve programmeringen enn klasse A. Likevel var det mange av elevene som slet med tekniske utfordringer underveis, slik som det å overføre programmene fra internettsiden til microbiten.

Elevene samarbeidet og kommuniserte greit sammen, men det gikk igjen at den personen som jobbet på pc-en var mer aktiv i oppgaveløsingen enn den som satt ved siden av. Også i denne klassen holdt elevene seg til instruksjonene. De virket til å være ganske avhengige av den. Det ble observert at elevene fiktet, utforsket og eksperimenterte noe. Selv om de ikke nødvendigvis fikk til oppgaven med en gang, fortsatte elevene og prøvde igjen.

Det virket som at det var utfordrende for elevene å få til oppgaven, siden de ikke har jobbet så mye med micro:bit og programmering tidligere. På nettsiden er det mange ulike blokk-typer å velge mellom. Det var tydelig at de ikke hadde oversikt over dem. De snakket sammen med partneren sin og lette seg frem til de fant blokker de trodde kunne gi riktig løsning. Både på

ansiktsuttrykk og ut ifra samtalene mellom elevene ble det tydelig at elevene var usikre på hvordan de skulle løse oppgaven. De fant noen blokker de trodde kunne passe, men var usikre på om måten de hadde tolket/forstått oppgaven var rett.

Det oppstod noen frustrasjoner hos elevene underveis i arbeidet. Det var noen av instruksene i PowerPoint hvor ikke alle blokkene man trengte for å få programmet til å fungere var tatt med i eksempelet. Elevene løste dette hovedsakelig med å prøve seg frem med ulike blokker. De ba også om hjelp fra andre par i klassen og læreren. Noen av parene fikk til dette ved å brukt likheter fra det de allerede hadde gjort til å finne en ny løsning. Det ble observert at de snakket sammen om hva som var likt og at de kopierte hele eller deler av programkoden.

4.1.3 Klasse C

Følgende sammendrag presenterer klasse C. De tilhører en annen skole og har en annen lærer enn klasse A og B. I naturfagstimen uken før observasjonen hadde klassen en «fikle-økt» hvor de fikk testet utstyret og øvd seg på å lage enkle programmer som de overførte til micro:bit. Innledningsvis ga læreren praktisk informasjon om timen, og fortalte hvorfor de skulle arbeide med programmering. Læreren fortalte at det de skulle jobbe med bygde videre på det de hadde om i forrige skoleår, Internett som teknologisk system. Læreren sa at hensikten med å programmere micro:bitene var å lage en modell/representasjon av hjemmeruteren.

Denne klassen kom raskt i gang med å løse oppgaven, men noen av parene trengte veiledning for å få koblet til utstyret riktig. Jeg fikk inntrykk av at omtrent samtlige elever får til å programmere. Noen av parene utvekslet ideer og her ble begreper i programmering, som betingelser, tydelig at de kjente godt til fra før. Det virket som at Charles og Calle hadde mye å jobbe ut ifra. Det at de raskt kom i gang og ville vise hverandre hva de tenkte ga inntrykk av at de var ivrige.

Mesteparten av tiden holdt elevene seg til faglig snakk relatert til oppgaven. Da undervisningen gikk mot slutten ble det gradvis mer uro. Dette bestod av ikke-faglig snakk, eller at elevene gjorde andre ting på PC-en. Det ble observert at noen elever forlot plassene sine og gikk til andre elevpar. Det var andre elevpar som også mot slutten holdt fokus på oppgaven. De virket til å være fokusert på å få til oppgaven før timen var over. Avslutningsvis fikk elevene beskjed om å pakke sammen utstyret før de hadde en plenumsamtale. Her stilte læreren spørsmål hvor elevene fikk dele tanker om hva de hadde lært og hva de hadde

opplevd som utfordrende med oppgaven. Hovedtrekkene fra denne samtalen var at elevene selv sa at de hadde lært mer om blokkprogrammering. Det ble nevnt at de hadde fått øvd seg på hvordan «radio»-blokken kunne brukes, og de sa at de forstod mer av hvordan hjemmerutere fungerer. Det ble også tatt opp at tekniske problemer som oppstod bidro til å gjøre feilsøkingen mer kompleks.

Elevene fikk beskjed om å jobbe sammen om programmeringen, og at de skulle bytte på hvem som satt med PC-en hvor de programmerte underveis i timen. Jeg la merke til at i noen av parene var det hovedsakelig en av elevene som bidro. Her skilte et av parene seg ut. Calle tok mest styring, men virket til å være bevisst på å la Charles slippe til. De diskuterte oppgaven seg imellom, og kom sammen frem til hva de burde gjøre. De trengte i liten grad hjelp fra læreren, som tyder på at de hadde tilstrekkelig støtte. Dette paret gikk raskt videre etter å ha løst en oppgave. Det var gjennomgående at elevene snakket sammen og diskuterte underveis, og at de prøvde seg frem en god del før de spurte om hjelp. Noen av elevene hjalp andre elevpar da de stod fast. Dette var ved å forklare hva oppgaven spurte om og hvordan programmene i instruksjonen var ment til å fungere. Det var flere situasjoner hvor elevene så på hverandres programmer og forklarte hvorfor de hadde gjort det på den måten.

Av de parene som ble observert i denne klassen stoppet noen fortere opp i oppgaveløsingen enn andre. Christin og Celine var et slikt elevpar. De så ut til å være usikre på hva de skulle gjøre og trengte de ofte støtte for å komme seg videre. Jeg fikk inntrykk av at PowerPointen elevene jobbet med var noe disse parene var nokså avhengige av. De satt sammen programmene ved å følge det som stod i denne instruksjonen. Det var noen av parene som var mer utholdende enn andre. Christin og Celine virket til å være ganske utholdende. De gangene programmet ikke ga riktig resultat, prøvde de å finne ut av problemet selv, jobber en god stund på egen hånd før de ba om hjelp.

4.2 Elevenes kunnskaper og ferdigheter

Den neste delen av resultatkapittelet beskriver data som omhandler elevenes kunnskaper og ferdigheter. Etter å ha analysert og bearbeidet datamaterialet har jeg kommet frem at elevene både uttrykker kunnskap som er tilknyttet det naturfaglige fenomenet Internett, og programmeringsteknisk kunnskap. Slik som tabell 6 viser, utgjør disse hovedtemaene for det første forskningsspørsmålet.

4.2.1 Kunnskaper tilknyttet kjerneelementet teknologi

Det naturfaglige innholdet blir som sagt uttrykt med blant annet begreper (Mork & Erlien, 2017). Begreper inngår i Utdanningsdirektoratets definisjon av *kunnskap*. Dette er noe elevene skal kjenne til og forstå, i tillegg til fakta, teorier, ideer og sammenhenger (Kunnskapsdepartementet, 2017). Det er ikke hvilke som helst begreper elevene skal lære. De det er interessant å trekke frem her, er de som er tilknyttet undervisningens kompetansemål: «Elevene skal kunne utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker» og «bruke programmering til å utforske naturfaglige fenomener» (Utdanningsdirektoratet, 2020a). Derfor er det disse som er ledende for begrepene som trekkes frem i resultatene. Det blir presentert hvordan disse kom til uttrykk i observasjonen og intervjuene. For å si noe om elevenes kunnskap, blir det i denne delen av resultatene beskrevet hvilke begreper elevene bruker.

Samtlige elever kjente til begreper som ruter, sender, mottaker, lagring og base. Disse er sentrale innen temaet undervisningsopplegget dreide seg om. Elevene arbeidet med dem i undervisningen. I instruksjonene ble begrepene brukt, i tillegg til at lærerne forklarte i starten av timen at programmeringen var relatert til det naturfaglige temaet Internett. I tillegg uttrykte elevene fakta og sammenhenger mellom systemets ulike deler. Dette gjorde de ved å bruke begrepene. Det kom frem at elevene kjente til måter begrepene var relatert til hverandre. Et eksempel er Antons beskrivelse av hva han har lært i timen: «*Rutere og sånne sendere. Sånne som sender for eksempel wi-fi da, eller dekning til forskjellige små rutere... Og lagring, base og sånt*». Videre kommer det frem at de fleste elevene ga uttrykk for at de forstod hva en hjemmeruter var, og kunne gi en forklaring ved å trekke frem vesentlige egenskaper. Dette kan indikere på en form for abstraksjon. Et utsagn som får frem hva flere elever uttrykte, var «*Den viser at man sender via noe annet. Som den ene... man gjorde noe på den ene, og det*

førte videre på den andre.» Dette er utdrag fra Astrids svar da ho skulle forklare modellen de hadde laget.

Det var også gjennomgående at selv om elevene var kjent med sentrale begreper, var språket til elevene preget av et hverdagslig språk. Elevene brukte egne ord og formuleringer fra andre sammenhenger til å få frem hva de mente. Da Anton skulle forklare hvordan de klarte å koble micro:bitene sammen sa han følgende: «vi legger to like formler inni». Det er fra matematikk og fysikk. I datamaterialet var det noen elever som sammenliknet konseptet algoritme med en kjedereaksjon. Det får frem at de kjenner til hvordan en sentral del av programmering fungerer.

Etter nå å ha beskrevet hvilke begreper og konsepter elevene fikk arbeidet med, skal jeg videre beskrive hvordan de fikk arbeidet med dem. Elevene skulle modellere en hjemmeruter. Både gjennom observasjonen og i påfølgende intervju kom det frem at elevene klarte å si noe om hva de ulike micro:bitene skulle være. Da elevene skulle fortelle om micro:biten sa Andrine «Det [micro:bit] skal være en sender og [den andre micro:bit] en mottaker, eller jeg vet ikke jeg». Anton brukte teknologisk begrep: «**En ruter**». Her ser det ut til at elevene har kjennskap til sammenhengen mellom programmeringen av micro:bit og de ulike enhetene som inngår i det teknologiske systemet. Det at Andrine svarte «eller jeg vet ikke» kan tyde på at hun var noe usikker. Flere av elevene uttrykte liknende usikkerhet i sine uttalelser. I observasjonene kom dette frem ved at flere av elevene stilte hverandre spørsmål angående modellen, og brukte tid på å løse oppgaven.

Et funn tilknyttet modellering er at det kan se ut til at flere av elevene gradvis blir mer sikre på modellen og hvordan den henger sammen med det teknologiske systemet i løpet av undervisningen. Det var nemlig flere av elevene som hadde spørsmål tilknyttet deler av instruksjonene, for eksempel en tekst hvor det stod «modellere en hjemmeruter med micro:bit». Det at de lurte på hva dette ville si, indikerer det at de trengte hjelp til å koble programmeringen til det naturfaglige innholdet. En av elevene som stilte slike spørsmål, viste mer forståelse senere i timen. I en samtale mellom Charles og Calle senere i undervisningen satt Calle ord på sin forståelse: «*Hjemmeruteren og mobil – det er det microbitene er... Når de er koblet på samme radiosignal, vil de kunne sende og motta koden.*» Ut ifra Calles utsagn så det nå ut til at han var klar over sammenhengen mellom det fysiske fenomenet, internett, og de ulike micro:bitene. Utsagnet tydeliggjør også at han har kunnskap om betydningen av

radio-gruppen, altså hva programmet vil gjøre. Dette kan tyde på at Calle forstår modeller og det naturfaglige fenomenet bedre i løpet av undervisningen. De ser ut til å ha en progresjon i sin modellforståelse. Noe annet Calle uttrykte i intervjuet var «... det prøver å vise oss er disse her, siden nå brukte vi «radio» som har samme konseptet som, eh, som nettverk.» Ut ifra disse situasjonene ser det ut til at han har en god forståelse av hva en modell er.

Etter nå å ha beskrevet hvordan elevene ser ut til å kjenne til programmeringen som måte å modellere internett, skal jeg beskrive hva som kom frem av elevenes mer generelle kunnskaper tilknyttet modeller i naturfaglig sammenheng. I intervjuene fortalte samtlige elever at modeller var noe de hadde jobbet med tidligere. Da noen av elevene skulle forklare hva en modell er ble det gjort på ulike måter. Noen gjorde det ved å komme med et eksempel på en modell. Andre fortalte at modeller kan brukes til å vise noe og hvordan noe er satt sammen. I tre av fokusgruppene ble atommodell nevnt som et eksempel på modell i naturfag. Slik som Arne: «*Hvordan det er satt sammen*». Elevene forklarte hva denne modellen brukes til. Flere av elevene trakk frem at modeller er forenklinger av virkeligheten. Astrid formulerte seg på følgende måte: «*At når man ser det så er det, det skal forestille da hvordan det ser ut i virkeligheten, men det er ikke akkurat sånn den ser ut.*». Dette er en vesentlig egenskap til modeller, og dermed uttrykker hun forståelse for naturfaglig innhold.

4.2.2 Kunnskaper tilknyttet programmering

Det neste temaet fra analysen dreier seg om kunnskaper elevene har tilknyttet programmering. Det var tydelig at et undertema som er hensiktsmessig å trekke frem er hvordan elevene forklarte begrepet programmering. Deretter hvilke kunnskaper de uttrykte om hva programmering brukes til.

Elevene uttrykte forståelse for hva det vil si å programmere på ulike måter. Bianca beskrev det som at det går på å «*lage en enhet som hører på deg*», og Calle sa «*Du sier hva PC-en skal gjøre, og hvor det skal ende opp, og du bare gir den instruksjoner på akkurat hva den skal gjøre*». Noen elever sammenlikner mennesker og datamaskiner, og får frem at datamaskiner ikke kan tenke. De forklarer videre at dette stiller krav til hvordan programmene man lager er bygd opp. Som nevnt tidligere, var ble programmering sammenliknet med en kjedereaksjon. I dette utdypet de dette med at programmene består av flere deler som følger hverandre. De sa at noe leder til noe annet og at de sammen får noe til å fungere. Anes utsagn er et godt eksempel på hvordan elevene ordla seg: «*At liksom, du gjør en ting også skjer det noe annet,*

ut ifra det som skjedde før det». Ut ifra min tolkning ser det ut til at dette er kjennskap til noe sentralt innen programmering, nemlig input og output. Også datamaterialet fra observasjonene viser at elevene kjente til dette. Han sa følgende «*Når denne micro:biten får programmet, vil beskjedden overføres herfra over til dit [den andre micro:biten]*». Calle forklarte til Charles hva resultatet (output) av radio-funksjonen i programmet vil bli (output). Calle viste på micro:bitene, samtidig som han forklarte.

Dette leder oss videre til neste funn. Det ser ut til at de fikk arbeidet med sammenhengen mellom software og hardware. Elevene arbeidet med micro:bitene, som er et eksempel på en fysisk programmerings-enhet. Et eksempel på dette er Antons formulering fra intervjuet: «*At det er noe du gjør, at du kan gjøre noe på data, også kan du fortsette å være på data, men du kan også være sann og så på data som kan forflytter seg til fysisk.*» Dette tolker jeg som at han kjenner til at det programmet han lager på datamaskinen styrer den fysiske enheten. Dette er sentralt innen programmering.

Elevene uttrykte også at de har kunnskap om hva programmering kan brukes til. En av elevene sa at de tidligere hadde brukt programmering til å lage modeller av naturfaglige fenomener. Calle sa følgende «*vi skulle lage vårt eget modell av, eh, Newtons lover... så, eh, da var jo ett av, eh, delene du kunne bruke til å lage en modell, var jo Python*». Hensikten med modellen, slik eleven så det, var at det gjorde det lettere og raskere å løse oppgaver. Det uttrykte han slik: «*Du på en måte lager et program som bruker en formel som du på en måte har lagt til ... Så den kan løse programmet for deg så du ikke bruker så og så mye tid på å regne det ut*». Andre elevene virket til å være mer usikre på hvorfor de skal bruke micro:bitene til å lære om det naturfaglige temaet, slik som Astrid sier: «*Jeg føler ikke at vi driver helt med naturfag. ... Jeg vet ikke helt hva det har med naturfag å gjøre.*».

Elevene kjente til ulike verktøy som er resultat av programmering eller består deler som er programmert. Det kom tydelig frem ved at elevene kom med mange eksempler og måtte ledes videre til neste tema når apper og slik ble tatt opp. De fikk frem at de kjente til nytteverdien av programmering. Borgny sa i intervjuet at programmering «*gjør ting enklere for oss å gjøre*», og forklarte hvorfor ved å nevne ulike typer verktøy og produkter som er programmert. Borgny og andre elever kjente til flere slike eksempler. Kalkulator, spill, nettsider og appen Vipps ble nevnt i flere av fokusgruppene. Charles kjente til en bedrift som brukte programmerte roboter til å sortere søppel. Denne kunnskapen var ikke noe som

undervisningsopplegget la opp til at elevene skulle lære om. Det inngår i elevenes forkunnskaper siden det var noe elevene kjente til fra før.

Elevene relaterte læring av programmering hovedsakelig til at det kan gi erfaring de kan ha bruk for hvis du skal jobbe med det senere. De kjente til at programmering er nyttig fordi det kan gi dem kunnskap og erfaring som vil være nyttig å kunne i fremtiden. Det kom frem at flesteparten av elevene hovedsakelig så på programmering som relevant med tanke på yrkesvalg. Dette er sentralt innen kjerneelementet teknologi. Astrid beskrev noe flere andre elever også uttrykte: *Jeg føler egentlig ikke jeg får så mye bruk for programmering... Hvis jeg ikke skal ha noen jobb som har noe med det å gjøre, så vet jeg ikke helt hva jeg kommer til å få bruk for det*». Dette understreker at i samtlige intervjuer var ikke programmering noe elevene så for seg at de vil ha bruk for.

En av kunnskapsområdene innen teknologi er begreper og konsepter innen programmering. Derfor blir det i følgende avsnitt beskrevet tegn på elevens forståelse av dette. I intervjuene kjente elevene til sentrale begreper innen programmering, slik som *enhet, kommando, blokk, kode* og *programmer*. De to sistnevnte ble brukt om hverandre, som synonymer. I fokusgruppe 1 ble begrepene *input, output* og *print* også nevnt. Elevene fortalte at de kjente til disse begrepene fra tidligere matematikkundervisning hvor de hadde jobbet med programmeringsspråket Python. Da hadde de brukt Python til å regne mattestykker.

Elevene hadde også kunnskaper om at det er ulike måter å programmere på. De kjente til både blokkprogrammering og tekstprogrammering, og at det innenfor disse er ulike programmeringsspråk. Programmeringsspråket Python ble nevnt i tre av fokusgruppene. Det var ulikt hvor mye kjennskap og hvor inngående de kunne forklare programmeringsspråkene. I fokusgruppe 5 beskrev Calle og Charles ganske detaljert noen forskjeller mellom de to typene programmering, blokk og tekst. Et eksempel er det Calle sa: «... *i de blokkene så er det innebygde koder, ... Det [kommandoene i tekstprogrammering] betyr det samme, bare at den blokken er liksom allerede en ferdiglagd kode, som du bare kan dra rett ut*». Det Calle får frem her, at blokkene er ferdiglagde programmeringskoder. Dette var det ikke alle elevene som kjente til.

Noen av elevene uttrykte forståelse for hvordan programmering gjøres med tekstprogrammering. De pekte på at i blokkprogrammering er det ferdiglagde programkoder som brukes istedenfor å skrive det inn selv. I sammenlikningen av disse typene programmeringsspråk sier elevene at blokkprogrammering er enklere å bruke. De forklarte også at tekstprogrammering stiller strengere krav til den som programmerer enn i blokkprogrammering. At det krever mer av den som programmerer i tekstprogrammering siden, som Charles sa: *«du må på en måte vite disse kommandoene da, for å klare å lage det programmet»*. Disse elevene gjenkjenner elementer på tvers av ulike programmeringsspråk. Calle sa følgende: *«Men blokkprogrammering er ikke akkurat, liksom det ideelle, når det kommer deg til et såpas høyt nivå. ... du kan ikke akkurat legge til noen fler, med mindre du, med mindre du lager den koden selv»*. Her trekker han frem at tekst- og blokkprogrammering vil kunne brukes forskjellig, og avhenger av hvilket nivå eller hvor mye erfaring man har med programmering. Ifølge Calle er blokkene i blokkprogrammering «spesielle» og noe begrensede, mens et språk som Python er noe generelt, og består av byggesteiner som kan benyttes til å lage «alt».

I samtlige intervjuer uttrykte elevene kunnskap om programmeringsspråkernes syntaks. Uavhengig av språk, er det bestemte regler. Disse må følges for å kunne kommunisere med den digitale enheten. Christian uttrykte dette slik: *«jeg har lært ... sånn generelt å programmere. At man må følge ehh nøyaktig det man får for at det skal kunne funke. ... Jeg gjorde en liten feil også ble alt ehh... Så funka ingenting på micro:biten.»*. Elevene sa at kommandoene må være presise for at datamaskinen skal forstå dem, og er klar over at utfallet blir at programmene ikke fungerer hvis man har gjort noe feil. Christian henviste i forklaringen sin til en situasjon som er en del av observasjonen. I denne situasjonen var det en syntaksfeil som var årsaken til at programmet ikke fungerte slik det skulle. Christian forstod at han hadde gjort en feil og hva som måtte til for at å rette det opp. Kunnskapen om syntaks henger sammen med elevenes ferdigheter til å feilsøke. Dette beskrives ytterligere under temaet *problemløsning*.

Noen få av elever kjenner til ulike kilder til hvor de kan finne veiledning og inspirasjon, og fortalte at de hadde forsøkt å lage egne spill og nettsider. Calle hadde utviklet en interesse for programmering gjennom *gaming* og hadde prøvd å programmere selv utenom skolen. Han fortalte at han lært om det ved å se på videoer på YouTube av personer som programmerer

YouTube, og fortalte at videoene inspirerte han til å lage egne spill. Det kan se ut til at disse elevene anvender en form for generalisering, ved at de bruker andres produkter til å lage egne.

Et annet funn er at elevene hadde forståelse for sentrale konsepter innen programmering, og dermed innen teknologi. Konseptene er en del av det elevene skal lære i matematikk som er i naturfag. Dette var noe de uttrykte uten å bruke fagordene. Alfred uttrykte seg slik: *«Hvis du vil ha den hele tiden, så tar du den som sier «gjenta» hele tiden, hvis du vil at den skal ta en gang, så tar du den som sier «ta en gang»»*. Det kan se ut til at han her snakker om konseptet løkker, som brukes mye i programmering. En annen elev, Alice, uttrykte kunnskap om hvorfor dette er hensiktsmessig å bruke: *«For å slippe å gjøre alt selv liksom, at det er, ja. At du trykker på en ting, også skjer masse ting istedenfor å trykke på masse ting»*. Det ble likevel observert situasjoner hvor elevene gjorde noe annet selv om løkker kunne blitt brukt. De plasserte inn flere like blokker i programmene sine istedenfor å legge inn programblokken i løkken.

Et annet konsept innen programmering elevene fikk anledning til å øve på, er betingelser. I undervisningsopplegget skulle elevene blant annet lage et program som fikk den ene micro:biten til å sende en tekst til den andre. Her brukte elevene konseptet betingelser. *«Hvis knapp A trykkes, så sendes teksten mobil. Hvis knapp B trykkes, så sendes teksten PC»*. Dette ble det observert at flesteparten av elevene fikk til uten store problemer. Dette samsvarer med det som kom frem i intervjuene. Et eksempel er Bjørns utsagn fra intervjuet: *«Hvertfall det vi holdt på med nå husker jeg vi hadde litt om i syvende på barneskolen, så det satt litt igjen i hodet og hvordan man gjorde det da. Jeg huska de derre brikkene som man vis start og trykk på A for å få tekst å sånt da»*. Betingelser var noe elevene kjente igjen fra tidligere erfaringer med programmering, og noe de brukte for å løse oppgaven i undervisningen.

En av de blokkgruppene elevene arbeider mye med i undervisningsopplegget er «Radio». Det er noe en del av elevene uttrykker kunnskap om, og det er varierende hvor mye støtte de trenger for å forstå denne blokkgruppen. Under observasjonen sa Calle følgende *«Overføre herfra [micro:bit 1] over til dit [micro:bit 2]... (litt senere i observasjonen). Den ene [micro:bit] vil ikke gjøre noe før den andre trykker.»* Disse forklaringene kom han med da han skulle snakke med Charles om hva som vil skje når man legger inn «radio gruppe». Utsagnet får frem at Charles forstår hvordan denne blokkgruppen har innvirkning på resultatet på micro:bitene.

Det var andre elever som trengte mer støtte for å få Radio-gruppen til å fungere. En situasjon som belyser dette, er fra en samtale jeg hadde med Celine og Christin under observasjonen. De trengte hjelp til å få Radio-blokken til å fungere. I løpet av samtalen ble det oppklarende å først få dem til å se på hva Radio-gruppen skal få frem av det teknologiske systemet. De resonnererte seg frem til at det fungerer slik som wi-fi-tilkoblingen. Det så ut til å hjelpe dem til å forstå hvordan de skulle bruke denne programmeringsgruppen. Deretter arbeidet de med det tekniske. De feilsøkte i programmene sine, og forsikret seg om at radio-blokken var lagt inn i begge prosjektene. Calle bistod underveis som veileder, og sa blant annet: «*Dere må lage et program som mottar, i tillegg til et som sender... På denne måten kobler dere sammen begge microbitene. .. Hvis du har lagt det inn [Radio] i microbiten så vil det komme*».

4.2.3 Elevenes problemløsning

Selv om ikke alle elevene fikk kommet gjennom alle sidene i instruksjonen, fikk samtlige anledning til å øve på problemløsning. Noen brukte mest tid på å få selve utstyret til å fungere slik det skulle. mens andre jobbet med å utvikle et program som ga ønsket resultat på micro:bitene. I datamaterialet fra både observasjon og intervju var feilsøking en stor del/dominerende arbeidsprosess. Elevene hadde ulike tilnærminger til det å finne og rette opp.

I situasjoner hvor elevene oppdaget og rettet feil, var tekniske justeringer en tilnærming. I samtlige klasser ble det observert at de sjekket om feilen lå i **utstyret de brukte**, og dette beskrev de også i intervjuene. Programmering med micro:bit krevde at elevene undersøkte at de hadde det rette oppsettet på utstyret. Anton formulerte seg slik i intervjuet: «*Vi prøvde mange forskjellige måter, å restarte enheten, ta ledningen ut og inn av pc-en og prøvde å endre det [koden] litt, men ingenting funka. Den stod akkurat på det samme*». De sjekket også om micro:biten var «connecta» på nettsiden underveis i arbeidet. Fra observasjonene fikk jeg inntrykk av at elevene virket til å være vant til å håndtere tekniske feil. De var raske til å snakke sammen og forklare for hverandre hva som kunne være årsaken til feilen. Elevene kjente til flere typiske komponenter som kan forårsake feil. Etter å ha feilsøkt på denne måten uten at feilen var fikset, konkluderte elevene med at feilen dreide seg om programkoden. Det leder oss over på elevenes feilsøkingsstrategier i programkoden.

I samtlige klasser var *prøv-og-feil* en tilnærming mange av elevene hadde i møte med oppgaven. Det ble observert situasjoner hvor elevene satt opp programmet ved å prøve seg frem.. Hvis resultatet ikke ble slik de hadde tenkt, altså oppstod en feil, prøvde elevene noe

nytt og testet det. I intervjuet ordla Calle seg på følgende måte: *«Det blir jo på en måte det konseptet med prøv og feil. Du må nesten bare prøve deg fram helt til du klarer det da.»*. De justerte altså måten programmene var satt sammen for rette opp feilen. Noen elever satt sammen hele programmet på nytt. Andre byttet ut deler av programmet med andre blokker. I intervjuene beskrev Bodil at i parene hennes valgte de *«den første og beste»*.

Det var andre elever som hadde en mer systematisk måte å feilsøke på. Det kom frem i intervjuene at elevene kjente til ulike måter å feilsøke på. Disse ble også tatt i bruk av noen av elevene. En strategi var å gå systematisk gjennom de ulike blokkene i programkoden. Det kunne si om programmet faktisk ville gjøre det de ønsket før det ble overført til micro:biten. Dette tolker jeg som tegn på *evaluering* av programmet. Ut ifra dette kunne de forutse hvor feilen lå, slik Astrid forklarer her: *«... prøvde å se etter hva som kunne vært feil»*. I observasjonene var det flere av elevene som sammenliknet sitt eget arbeid med programmene i instruksjonen. Dette utdypet elevene i intervjuene. De sa at det var fort gjort å ta feil blokk, og da refererte de til en konkret situasjon fra observasjonen. Elevene fortalte at de ble mer oppmerksomme på dette underveis. Noen elever fortalte at de hadde en feilsøkingsstrategi som gikk på å se gjennom programmet samtidig som programmet kjørte. Dette hjalp til å finne ut hvor *«det gikk galt»*. Altså at de på en systematisk måte fikk oversikt over hvilke deler av programmet som ga ønsket resultat og hvilke som ikke gjorde det. De fortalte at de deretter fjernet og endret på denne delen. Dette gjorde de istedenfor å starte programmeringsprosessen på nytt.

I datamaterialet kom det til syne at flere av elevene la inn mange blokker i prosjektet på en gang før det ble testet. Dermed hadde de mange blokker å gå gjennom hvis det ble feil. I intervjuet sa Charles dette: *«For at hvis man liksom gjør masse, og så tester man en eneste gang, og så blir det feil, da må du gå gjennom alt sammen for å sjekke hva som er feilen, ikke sant»*. Det var altså elever som var oppmerksomme på ulempene denne måten å gå frem i programmeringen kunne medføre. De kjente til prosedyren med å teste programmet ofte for å ha kontroll. I intervjuet kom det likevel frem at dette ikke var noe de bestandig praktiserte. De begrunnet det med at programmeringen av micro:bit ikke var så kompleks slik som for eksempel utvikling av apper. De fortalte også at de opplevde at det var tungvint.

Det var også elever som systematisk sjekket om feilen lå i syntaksen i programkoden. Noen av elevene snakket om i intervjuet at datamaskinen er *«streng»*, så hvis man gjør noen få feil i hvilke tegn man setter inn så stopper det opp. Det er bestemte *«regler»* datamaskinen følger,

noe som stiller krav til at elevene setter sammen programmet på bestemte måter, at de bruker blokker hvor de kan sette inn tekst når de ønsker det. Under observasjonen i klasse C ble Celine og Christin klar over en slik type feil. De skulle skrive inn en tekst, «mobil», men fikk ikke ønsket resultat. Elevene oppdaget at blokken de hadde satt inn var ment for tall/verdi. Derfor la de inn en annen blokk hvor det var ment å skrive inn tekst. I fokusgruppeintervjuet fortalte Christian følgende om en situasjon fra undervisningen: «*At jeg skrev et ord feil, så den ville ikke programmere riktig.*» Det at Christian oppdaget at årsaken var en syntaksfeil gjorde at feilen ble fikset.

Videre ser det ut til at undervisningen la til rette for at elevene kunne ta i bruk flere av nøkkelbegrepene i *den algoritmiske tenkeren*. *Dekomponering, abstraksjon, generalisering, automatisering*. Disse har til felles at det dreier seg om å løse problemet på en effektiv måte. I dette inngår å effektivisere måten programmet bygges opp og endres. gjenbruke istedenfor å måtte gjøre alt på nytt. Funn fra resultatene som inngår i dette blir presentert samlet i det neste avsnittet. Dette er blitt plassert i undertemaet *koding* i resultatene (se vedlegg 5).

Det var ulikt om og hvordan elevene benyttet seg av slik effektivisering. I datamaterialet ser det ut til at elevenes bruk av generalisering var noe varierende. Noen gjenkjente løsningene, og bruke hele eller deler av programmet de hadde laget i andre deler av problemløsningen. Billy forklarer her noe som er karakteristisk for disse elevene: «*Vi tok mye av det som var på den forrige oppgaven inn på den nye og bare endra.. sånn at den passa til den nye oppgaven*». Dette utdyper elevenes fremgangsmåte. De kopierte programkoden og justerte den slik at den ga et annet resultat på micro:biten. Det var andre elever som virket til å ikke være klar over denne tilnærmingen. De gjentok hele programmeringsprosessen. Det ble observert at de satt sammen et program som bestod av de mange eller alle de samme blokkene som de hadde laget tidligere. Abstraksjon var noe som så ut til å ikke bli like synlig hos elevene. Det å få arbeidet med å koble sammen fler micro:biter var noe som i mindre grad kom til syne. Mine resultater ser ut til at det å gå de neste stegene var noe samtlige elever slet med. Det å representere pc, hjemmeruter med micro:bitene.

4.3 Elevenes arbeid med programmeringsoppgaven og utfordringer

I denne beskrivelsen er det særlig fokus på hvordan elevene arbeidet med undervisningsopplegget. Skaperskolen (Skaperskolen, 2020c) trekker frem at støttestrukturer er noe som er viktig for elevers læring og noe undervisningen bør bestå av. Mangel på støtte kan føre til at elevene stopper opp i arbeidet. Det dreier seg om å gi elevene tilstrekkelig veiledning til at de kan mestre oppgaven. Ulike utfordringer som oppstod beskrives, samt hvordan elevene tilnærmet seg disse. Bruk og betydningen av undervisningsmaterialet, elevenes forkunnskaper, forarbeidet klassene hadde hatt, samarbeid og tidsaspektet blir trukket frem i de neste avsnittene.

Elevene har i varierende grad behov for hjelp til å forstå instruksjonene. Instruksjonen var utformet slik at den ble gradvis mindre detaljert utover i oppgaven. Dette stilte krav til at eleven selv fant ut av hvordan programmet skulle bygges opp. Flere elever hadde utfordringer med å forstå hva instruksjonene «mente». Dette kom til syne ved at de til stadighet stilte spørsmål til læreren, medelever eller observatører for å få oppklart hva de skulle gjøre. Dette var spesielt for de delene av oppgaven hvor ikke hele programmet som elevene skulle lage var inkludert. Dette var det flere som slet med, og Andrine ordla seg slik omkring dette: *«Jo, eh, det var jo vanskelig. Det var jo, jeg skjønnte litt på starten og så ble det litt vanskeligere, synes jeg.»* Etter å ha undersøkt datamaterialet nærmere, ser det ut til at blant de som trengte hjelp til å forstå instruksjonene, vet det hovedsakelig elever som ikke hadde programmert noe særlig tidligere.

Dette leder oss videre til elevenes forkunnskaper, som er en annen form for støttestruktur. Det ser ut til å ha innvirkning på hvordan elevene arbeider. Deres tidligere erfaringer, kunnskaper og ferdigheter, hva de kjenner til vil styre hvordan de griper fatt på oppgavene de får. Alette sa følgende i intervjuet: *«Det [nettsiden] så komplisert ut, synes jeg.... Ja. Jeg synes sånne nye nettsider er så vanskelig å flytte på ting, jeg synes det, da blir jeg veldig forvirra... Det var så mye på skjermen, så at jeg måtte gjøre så mye også»*. Her kommer det frem at også nettsiden var noe elevene brukte tid på å forstå. Under observasjonene la jeg merke til dette. Enkelte elever brukte tid på å orientere seg på nettsiden de skulle programmere på. Fra datamaterialet er det tydelig at den klassen, klasse C, som hadde hatt mer tid i forkant av undervisningen til å bli kjent med utstyret kom raskere i gang med arbeidet og orienterte seg effektivt på nettsiden.

Noe annet som datamaterialet tyder på er at elevene trengte repetisjon av egen kompetanse, og at programmeringsferdighetene er noe som må vedlikeholdes. Blant de elevene som hadde arbeidet med programmering tidligere, fortalte noen av dem at de ikke husket så mye. Christoffer sa følgende: *«Jeg hadde glemt mesteparten av det før da, så var det nesten sånn start på nytt igjen»*. I observasjonen la jeg merke til at elevene brukte tid på å repetere fremgangsmåten, særlig i starten av undervisningen. Dette gjaldt også for klasse C, som hadde hatt fokus på ferdighetsøving i forarbeidet. Catharina uttrykte noe viktig med tanke på elevens utbytte: *«Ehh jeg trur vi hadde lært mer av en lengre oppgave med flere timer. ... Da lærer vi jo mer da. For det som kommer til å skje nå er jo at det vi legger bort det og så tar vi sikkert ikke opp ene til vi slutter på skolen.... Så får vi null mer kunnskap om ehh programmering.»*

Noe som samtlige elever uttrykte i intervjuene, var de ulike feilkildene som oppstod underveis. Det gikk på at utstyret ikke var like lett å bruke, det oppstod **tekniske utfordringer** som skapte forvirring. En stor andel av elevene at de synes det var utfordrende å bruke utstyret, og de brukte en del tid på å få det til å fungere. Et eksempel på dette var det å få overført programmet de hadde laget i nettsiden over til micro:biten. De etterspurte mer informasjon og tydeligere retningslinjer tilknyttet dette i instruksjonene. Slik Christin gjør her: *«Men det var litt vanskelig da, for vi skjønnte ikke hvordan vi skulle laste ned og få den inn på, eh, den micro:biten.»*

Det at micro:bitene skulle få overført ulike programmer, og at det var fort gjort å miste oversikt over hva som var overført til hvilken micro:bit. I tillegg oppstod det noen situasjoner hvor elevene trodde at de hadde programmert feil, men at det viste seg at feilen lå i utstyret, enten PC-en, micro:biten, at den ikke var synkronisert med PC-en. Dette kom også frem i intervjuene. Christian sa *«Ehm først så trodde vi det var noe galt med micro:biten, så vi bytta den ut eh, men så funka det fortsatt ikke på min pc hvertfall... Ehh så vi bytta pc og da funka det. Så det kan være ja eh jeg vet ikke helt hva som var feil.»*

Det var litt ulikt hvordan elevene samarbeidet, og hvor mye de brukte hverandre. Et mønster i datamaterialet var at de parene hvor minst en av elevene hadde programmert tidligere hadde stor nytte av denne elevens forkunnskaper det for å løse oppgaven. De parene som ikke hadde så mye forkunnskaper, fikk istedenfor hjelp fra andre elever eller læreren. Ved å la elevene arbeide i par tilrettelegger man for samarbeid. Det er en form for støttestruktur som er ment til å kunne gjøre det tryggere og bedre for elevene i problemløsningen. Det er også en av

arbeidsmåtene vi finner i *den algoritmiske tenkeren*. Elevene hadde ulik erfaring og jobbet ut fra ulike utgangspunkt. I arbeidet hjalp de hverandre videre. Innad i parene og på tvers av par støttet hverandre. Flere ganger blir det observert at par som stopper opp i arbeidet, ber om hjelp fra elever som har mer erfaring og som har fått til oppgaven. Det er tydelig at de bruker hverandres ulike forkunnskaper, og kommer frem til en løsning i fellesskap. Alfred beskriver fordelene ved å jobbe sammen på følgende måter: «*Kan tenke smartere med to hjerner enn med en*». Det ser altså ut til at de ble motivert av å få hjelp av hverandre og at de så på det å jobbe i par som noe nyttig.

I klasse A og B var det et fåtall av elevene som hadde arbeidet mye med programmering tidligere. Dermed hadde de ikke den samme ressursen i medelever som i klasse C. I tillegg hadde læreren i klasse A og B lite forkunnskaper og erfaring, noe som gjorde at støtten ble noe mindre enn i klasse C. Lærernes ulike forkunnskaper ser ut til å ha innvirkning på hvor mye støtte elevene får. I intervjuene med elevene fra klasse A og B uttrykte elevene at de savnet å få hjelp fra noen som var veldig gode i programmering. Det å få hjelp fra en «*som kan programmet innvendig og ut*» sa Alice at ho kunne trengt for å løse oppgaven.

Språket brukes til å sette ord på egne tanker og kunne dele det med andre. Datamaterialet tyder på at elevene opplevde at det var utfordrende å få andre til å forstå det de mente. Calle sa følgende: «*Så prøvde jeg å forklare det, men jeg følte at jeg kanskje forklarte det litt dårlig, sånn at det ble litt sånn, ordene mine ble bare sånn derre babbel, hvor de ikke helt forstod hva jeg mente da.*» Eleven kom inn på hvordan han opplevde å skulle forklare egen forståelse til medelever.

5. Diskusjon

Undervisningsopplegget som er utgangspunktet for denne studien, legger opp til at elevene skal bruke programmering. De skal bruke det i arbeidet med det teknologiske systemet Internett. Dette er det i tråd med det som inngår i kjerneelementet *Teknologi* (Utdanningsdirektoratet, 2020a). Min studie ble gjort for å få frem hvordan undervisningsopplegget tilrettelegger for læring av teknologi. I resultatene ble funnene fra observasjonene og intervjuene beskrevet. Det ga et innblikk i hvordan de arbeidet med programmeringsoppgaven, samt deres tenkning. Her var elevenes kunnskaper og ferdigheter, samt tilnærming til problemløsning sentralt. For å få frem hovedtrekkene ved resultatene, er funnene samlet i tre temaer: 1) Forkunnskapenes betydning, 2) elevenes problemløsning og 3) støtte i elevers læring. Det er disse temaene som diskusjonen dreier seg om, og strukturen på diskusjonskapittelet følger den rekkefølgen de er nevnt i. Dette gjøres ved å trekke inn relevant teori og funn fra tidligere forskning, for å sette funnene i en større sammenheng. I kjølvannet av diskusjonen, blir det trukket frem noen styrker og svakheter ved studien. Avslutningsvis blir det presentert noen forslag til hvordan temaet kan undersøkes videre.

Blant det som er aktuelt å få frem i diskusjonen, er hvordan undervisningsopplegget fungerte i praksis. Her trekkes det frem hva det ser ut til at elevene fikk anledning til å lære. Strukturen på diskusjonen som følger er etter temaene som ble nevnt i innledningen til dette kapittelet. Derfor går vi nå inn på betydningen av forkunnskaper.

5.1 Forkunnskapenes betydning

Slik det kommer frem i resultatene, uttrykker elevene kunnskaper og ferdigheter relatert til teknologi. Oppgaven elevene skulle arbeide med, ble løst til ulik grad av de ulike elevene. Elevene fikk til å løse deler av oppgaven, men ikke alt. Det oppstod ulike utfordringer, som var relatert både til det teknologiske systemet og programmeringen. Dette er relevant å få frem, siden utfordringer kan stå i veien for intensjonen bak undervisningsopplegget. Noe av årsakt til utfordringer som oppstod, så ut til å komme av at elevene ikke hadde nok forkunnskaper. Dermed kan det se ut til at dette vil ha betydning for hva elevene får arbeidet med og hvordan de arbeider. I det følgende skal jeg først trekke frem betydningen av forkunnskaper tilknyttet modeller, og deretter programmering.

5.1.1 Forkunnskaper modeller

Oppgaven elevene arbeidet med i undervisningsopplegget, dreide seg om å lage en modell av en del av Internett. Modeller kan brukes for å få frem hvordan en teknologisk produkt fungerer, og vise hvordan naturvitenskap og teknologi henger sammen (Pajchel et al., 2019). De skulle altså bruke en sentral naturvitenskapelig praksis, å lage modeller. For å kunne bruke slike praksiser, forutsetter det, ifølge National Research Council (NRC) (2012) at elevene har kunnskaper om hva praksisen innebærer. Ut ifra intervjuene med elevene uttrykte de noe kunnskap som omhandler modeller. De klarte å gjengi definisjoner, gi eksempler, og si hva som er hensikten med dem. De trakk spesifikt frem at modellen ikke fikk frem alt ved hjemmerouteren og Internett som system. Her viste de at de hadde forståelse for en sentral egenskap ved modeller, nemlig at de er forenklinger av virkeligheten (Hannisdal & Ringnes, 2015). I tillegg hadde de noe erfaring med å bruke modeller i naturfagundervisning. Det ser derfor ut til at de hadde forkunnskaper tilknyttet denne naturvitenskapelige praksisen, som kunne bli brukt i arbeidet med programmeringen av micro:bitene. NRC sier altså at forkunnskaper er sentralt i naturfagarbeidet. Her kan det videre diskuteres om forkunnskapene elevene hadde om modeller, faktisk ble brukt for å løse oppgaven.

Det forekom situasjoner hvor elevene klarte å programmere micro:bitene uten å nødvendigvis aktivt bruke kunnskapen de hadde om det teknologiske systemet. Det er ikke nødvendigvis slik at elevene tar i bruk forkunnskapene, selv om det var tanken bak undervisningen. Bungum mfl. (2014) har funnet i sin studie av teknologiprosjekter at det ikke er en selvfølge at elevene inkluderer fagkunnskap fra naturfag i denne typen arbeid. Det at problemet kan løses uten å måtte fokusere på fagkunnskapen, slik det kom frem i min studie, bekreftes av studien til Bungum. Selv om Bungum mfl. (2014) sin studie er ganske annerledes enn den som ble gjort tilknyttet denne masteroppgaven, har de likevel noen likheter. Micro:bitene skulle ha en bestemt funksjon, nemlig å sende og motta informasjon. Modellers funksjon er sentralt i teknologiprosjekter. Bungum trakk derfor frem fra sin studie at det å utforme oppgaver hvor fagkunnskapen faktisk er nødvendig å bruke for å løse oppgaven er viktig. Dette kan vi anta også kan sies om micro:bit-oppgaven. For at elevene skal utvikle sin forståelse og kunne bruke modellkunnskapen i arbeidet med micro:bitene, vil det kunne spille inn på hva som kreves for å løse oppgaven.

Selv om elevene hadde noe forkunnskap om modeller, var det likevel utfordrende for noen elever å bruke denne praksisen. Noe av utfordringen så ut til å være relatert til at elevene ikke hadde tilstrekkelig forkunnskaper om det teknologiske systemet elevene skulle lage en modell av. Det gjorde at de trengte hjelp til å vite hva som var de viktigste delene ved hjemmerouteren, og hvordan dette kunne representeres gjennom en modell. Elevene skulle altså drive såkalt *abstraksjon* (Shute et al., 2017). Modellen skulle bestå av en «sender» og en «mottaker». Dette er vesentlige deler ved funksjonen til en hjemmerouter. De kjente til sentrale begreper som inngår i Internett, og kunne komme med forklaring på hvordan hjemmerouteren fungerte. Likevel så det ut til at sammenhengen mellom modellen og hjemmerouteren så ut til å være uklart for en del av elevene. Det kan komme av at de ikke var trygge nok på faginnholdet. For å kunne skjelle de viktigste delene som skal frem i modellen, vil elevene trenge kunnskaper om fenomenet (Aronson, 1996). Dette inkluderer kunnskap om de ulike delene og om helheten for å forstå hvordan de ulike delene henger sammen og er gjensidig avhengig av hverandre. Dette er forutsetninger for å kunne utvikle sin systemtenkning. For å utnytte potensialet som ligger i arbeid med modeller med tanke på utvikling av elevenes faglige forståelse, ser dette ut til å avhenge av elevenes forkunnskaper om både modeller og det modellen skal representere.

Noe som bekrefter dette, er at det var situasjoner fra undervisningen hvor elevene dro nytte av forkunnskapene de hadde om systemet i utformingen av modellen. En slik situasjon var da elevene ble gjort oppmerksom på kunnskaper de hadde om wi-fi, og hvilken funksjon det har for at systemet skal fungere. Ved å aktivere forkunnskaper om en sentral del av systemet, så ut til å gjøre det lettere for elevene å lage modellen. Det å faktisk dra nytte av forkunnskapene, så ut til å være noe elevene trengte hjelp til. De brukte mye av oppmerksomheten sin på å programmere, som gikk ut over hvor mye de fokuserte på det teknologiske systemet som skulle modelleres. Denne utfordringen er det også andre studier som har trukket frem. I programmeringsprosessen ser fenomenets egenskaper ut til å komme i skyggen av hva elevene gjør for å justere programmet (Sørby & Angell, 2012). I studien til Caballero mfl. (Caballero et al., 2014) fant de at koblingen til det naturfaglige innholdet må poengteres.

Det å ta utgangspunkt i sin naturfaglige kunnskap, var noe elevene gjorde etter å ha fått veiledning. Denne typen veiledning så ut til å ha positiv innvirkning på elevenes arbeid. Det vil også kunne lede til læring. Denne tolkningen støttes av Basu mfl. (Basu et al., 2017). Blant

det de har funnet, er at en veileder kan tilrettelegge for at elevene får forbedret kompetanse i faginnholdet. Veiledning kan bidra til at elever går frem på en mer systematisk måte når de skal modellere. Dette får frem noe som er sentralt for hva undervisningsopplegget kan tilrettelegge for. For at det skal fungere optimalt, vil det kunne være hensiktsmessig å bruke tid på å lære om det teknologiske systemet i forkant av arbeidet med å lage modeller, samt å gi påminnelser underveis i arbeidet. Det vil legge til rette for at elevene får tilegnet seg kunnskaper som modelleringen kan integreres til og anvendt det i arbeidet. Dette vil være i tråd med tilnærmingen til undervisning hvor læring skjer gjennom kunnskapsintegrasjon. Ny kunnskap elevene får, må kobles til det de allerede har av kunnskap. Dette kan bidra til en forståelse hvor elevene ser sammenhenger, og kan gjøre at kunnskapen er lettere å huske og bruke i andre sammenhenger (Linn & Eylon, 2011).

Elevene skulle gjennom programmering utforske hvordan det teknologiske systemet fungerer. Et system består av mange deler som passer sammen, og virker sammen for å få systemet til å fungere (Aronson, 1996). I undervisningsopplegget som ble undersøkt i denne studien er Internett et eksempel på et slikt system. En sentral del av arbeidet i programmering og algoritmisk tenkning, er å bryte ned en sammensatt oppgave. I programmering er en sentral prosess å få oversikt over problemet som skal løses, dele det opp i mindre delproblemer og løse disse. Samtidig vil det å holde oversikt over helheten være avgjørende for å få programmet til å fungere (Shute et al., 2017). Denne måten å tilnærme seg problemer på, har likhetstrekk med systemtenkning (Aronson, 1996). Slik det kommer fram i Eidsvigs artikkel (Eidsvig, 2021), kan programmering tilføre mye til naturfagundervisningen. Dette blir videre tatt opp i diskusjonen om programmering ved å se på modellering.

Selv om det var en del av elevene som trengte hjelp til å modellere fenomenet, var det også noen av elevene som så ut til å ha en progresjon i løpet av arbeidet. Dette kom til uttrykk ved at de trengte en del hjelp i starten av arbeidet til å se sammenhengene mellom micro:bit og hjemmeruteren. I løpet av timen ble de mer og mer selvstendige i arbeidet, og forklarte for medelevene sine slik at også de forstod det bedre. Da disse situasjonene ble snakket om i intervjuene, bekreftet de at modellen bidro til å gjøre det mer forståelig. Elevene mente selv at de forstod det teknologiske systemet bedre etter å ha arbeidet med modellering med fysisk programmering. Det kan derfor se ut til at modellering av det naturfaglige fenomenet med micro:bit kan være til hjelp for elevenes forståelse for det teknologiske systemet. Dette støttes av Berland og Wilensky (Berland & Wilensky, 2015). De undersøkte fysiske roboter, og de

kom frem til at programmering av fysiske roboter så ut til å legge til rette for systemtenkning. Dette er overførbart til funnene fra analysen i denne oppgaven. Micro:bitene inngår i en sammenheng, og fungerer som et system. Deres funn underbygger funnene tilknyttet micro:bit, at det legger til rette for er systemtenkning. Det ser derfor ut til at programmeringen kan tilrettelegge for utvikling av forståelse for hvordan det teknologiske systemet fungerer. Noe annet modeller kan bidra til, er at elevene får ta del i utforskende arbeid. Dette skal diskuteres i de neste avsnittene.

Samtidig som elevene programmerte micro:bitene, snakket de sammen om hva de så for seg at kom til å skje og hvorfor. Dette inngår i sentrale praksiser i utforskende arbeid i naturfag (Sørvik, 2016). Fra observasjonene så det ut til at modellen støttet elevene i diskusjonen, og gjorde at de naturlig ble nysgjerrige på å komme frem til forklaringer. Dette støttes av Haug (Haug et al., 2021), som sier at det å lage og bruke modeller kunne tilrettelegge for utforsking i elevers arbeid med naturfag. Modellen kan få frem hva elevene antar at vil skje. Etter hvert som eleven får ny informasjon, kan det gjøres endringer som får frem hva eleven nå ser for seg at vil skje. I programmering kan slike endringer gjøres ved å justere programkoden, slik at den får ønsket resultat. Dette kan gjøre at arbeidet blir utforskende. Denne tilnærmingen til bruk av modeller, er i samsvar med Weintrop (Weintrop et al., 2016) sin forståelse av algoritmisk tenkning. Som legger velt på den rolle programmering har i naturvitenskapelig utforsking. De understreker betydningen av å inkludere bruk av modeller som er laget gjennom programmering i naturfagundervisningen. Det kan bidra til utvikling av naturfaglig forståelse. Noe annet modeller kan tilrettelegge for, er forståelse av naturvitenskapens kunnskapsbasis og naturvitenskap som prosess (Haug et al., 2021). Bruk av modeller støttes også av teorien som inngår i begrepet kunnskapsintegrasjon. Læringssituasjoner hvor elevene må revurdere forkunnskapen etter hvert som nye ideer kommer til syne, vil kunne bidra til at en dypere forståelse som kan anvendes i ulike situasjoner (Linn & Eylon, 2011).

Blant funnene i min studie, er at noen av ferdighetene som inngår i algoritmisk tenkning ble uttrykt og at de fikk arbeidet med sin forståelse for naturfaglig innhold. Ifølge Hoppe og Werneburg (2019), kan det å la elevene lage egne modeller, være en måte å bidra til utvikling av elevenes algoritmiske tenkning. I likhet med min studie, har Taub mfl. (Taub et al., 2015) funn som indikerer at det å lage modeller gjennom programmering kan gi forbedret forståelse for fenomenet. En annen studie har funnet noe av det samme, nemlig Lai og Lai (Lai & Lai, 2012). Ifølge dem kan blokkprogrammering bidra til at elevene mestrer bedre å modellere

naturfaglige fenomener. For å samle trådene fra dette, ser det altså ut til at programmering tilrettelegger for læring av elementer som inngår i kompetansemålene og kjerneelementet som undervisningen dreide seg om.

I denne delen av diskusjonen er det blitt trukket frem hva det ser ut til at elevene trenger av forkunnskaper for at undervisningsopplegget skal tilrettelegge for læring. De vil ha behov for tilstrekkelig forkunnskaper, både tilknyttet modeller generelt og det modellen skal representere. Det min studie og annen litteratur og forskning får frem, er at forkunnskapen kan legge til rette for at potensialet som ligger i modeller kan bli utnyttet. Utforskende arbeid kan komme frem og bli brukt ved å la elevene arbeide med modeller. I undervisningsopplegget som er utgangspunkt for denne masteroppgaven, er det et teknologisk system det dreier seg om. Momentene som er blitt trukket frem i diskusjonen sier oss likevel noe om hvordan modeller kan brukes i arbeid med andre deler av innholdet som inngår i naturfag. Modeller inngår i kjerneelementet *Teknologi* som det står at skal brukes i arbeid med de andre kjerneelementene. Ett av disse er *Jorda og livet på jorda*. Her vil det være andre systemer enn teknologi elevene skal arbeide med, noe som kan gjøres gjennom bruk av modeller. Derfor kan det også knyttes til kompetansemål til ulike naturfaglige fenomener. Den neste delen av diskusjonen skal dreie seg om elevenes forkunnskaper tilknyttet programmering.

5.1.2 Forkunnskaper programmering

Nå skal vi gå videre til betydningen av forkunnskaper tilknyttet programmering. I denne delen av diskusjonen er hensikten å få frem hvordan elevenes forkunnskaper innen programmering så ut til å ha innvirkning på hva de arbeidet med og hvordan de utførte det. Det trekkes frem hvordan elevene behersket utstyret og programmeringsspråket, og betydningen av forarbeidet blir drøftet.

Ut ifra analysen av observasjon og intervju, så det ut til at samtlige par fikk til å programmere selv om de hadde ulike forkunnskaper. Micro:bit kan ut ifra dette se ut til å være en nyttig måte å introdusere elever for programmering. Dette er i samsvar med det Tyrén mfl (2018) har funnet i sin studie, som også har funnet at undervisning med micro:bit kan løses ganske enkelt av elever uavhengig av erfaring. Det ser altså i utgangspunktet ut til at bruk av micro:bit fungerer slik det var tiltenkt av produsentene, å være en tilnærming til programmering som samtlige kan klare (Ball et al., 2016). Det at programmeringen benyttet

seg av blokkprogrammering, så ut til å være nyttig ved at elevene enkelt valgte de blokkene som var beskrevet i instruksjonene. Som beskrevet i teorikapittelet, er blokk-editoren visuell, og blokkene har en form som gjør at elevene setter sammen programmet ved å sette dem inn i blokkgrupper med passende form (Kluge, 2021).

Et moment tilknyttet brukervennligheten til micro:bit, er tekniske feil som kan oppstå. Dette skjedde i samtlige grupper som deltok i undervisningsopplegget. Det var flere av elevene som ble forvirret da det de hadde gjort ikke ga ønsket resultat. Da dette ble snakket om i intervjuene utdypet elevene det med å si at slike feil gjorde dem ekstra usikre og at de ga litt opp. Dette er det også funnet i andre studier som har undersøkt bruk av micro:bit, blant annet Tyrén (Tyrén et al., 2018). De skriver i sin forskningsartikkel at elever har en tendens til å stille seg likegyldige eller bli motløse når de møter slike feil. Da slike feil oppstod i undervisningen, spurte parene om hjelp fra medelever og læreren for å finne ut hva som var årsaken. Det ble både observert at dette hjalp elevene og bekreftet av dem i intervjuene at dette bidro til å holde dem i gang med å fortsette arbeidet. Dette tyder på at det å få tilstrekkelig støtte og oppmuntring fra læreren og medelever var nødvendig, noe også Tyrén mfl. (2018) påpeker. Denne delen av diskusjonen får frem at micro:bit har noen styrker og svakheter. For at denne programmeringstilnærmingen skal fungere optimalt, vil det være sentralt å unngå at det forekommer for mye unødvendig tidsbruk på tekniske feil.

Noe annet som indikerer nytteverdien av micro:bitene, var at det så ut til å kunne ha positiv innvirkning på elevenes systemforståelse. Det var flere elever som nevnte at det å holde på med micro:bitene gjorde funksjonen til hjemmeruteren mer konkret. Ifølge elevene var det lettere å forstå hvordan den henger sammen med resten av Internett. Dette inngår i kompetansemålene som undervisningsopplegget dreide seg om. En del av kompetansen elevene skal tilegne seg i undervisningsopplegget, går på systemer og systemtenkning. Det er som beskrevet i tidligere i teorigrunnlaget ulike måter å se på systemer. Grunnsteinen i systemtenkning er at det er deler som virker sammen og «skaper» helheten (Aronson, 1996). I tillegg til det elevene fortalte i intervjuene, var det å ha en modell å støtte seg på i samtale omkring systemet til hjelp for elevene. Det ser derfor ut til å være fordelaktig å gi dem oppgave i å lage en modell. Berland og Wilensky (2015) har kommet frem til liknende funn, tilknyttet fysisk programmering. I undervisningsopplegget de undersøkte, skulle elevene drive problemløsning ved å programmere fysiske roboter. Blant de funnene undersøkelsen deres ga, var at denne måten å lage modeller på bidro til at elevene klarte å se ulike deler i en

sammenheng (Berland & Wilensky, 2015). Når dette trekkes videre til hensikten for denne masteroppgaven, er det altså flere ting som tilsier at det å la elevene arbeide med det teknologiske systemet i kombinasjon med fysisk programmering er fordelaktig. Det kan bidra til at elevene får utviklet forståelsen for hvordan det teknologiske systemet Internett fungerer og hvordan de ulike delene henger sammen.

Det var hovedsakelig blant elevene som hadde en del erfaring med programmering som det så ut til at hadde en tydelig progresjon i forståelse for modellen de lagde. Dermed ser det ut til at forkunnskaper i programmering også spiller inn på dette. Blant elevenes forkunnskaper var kjennskap til programmet de skulle bruke. Å ha oversikt over hvilke muligheter som inngår i programmeringen, er noe også Çakıroğlu mfl (2021) kom frem til. Dersom elevene er klar over hvilke blokker som kan brukes og hvordan dette kan gjøres, vil de lettere kunne overføre løsningene til nye situasjoner. Det vil videre ha positiv innvirkning på abstraksjonen, altså å få frem sentrale deler ved det som skal modellere. Elevenes forkunnskaper i programmering ser altså ut til å ha betydning for hvor godt aktiviteten fremmer læring. For at programmeringen skal legge til rette for læring om teknologi, vil det være fordelaktig å få tid i forkant på å lære grunnleggende programmering. I rapporten til Bocconi mfl. (Bocconi et al., 2022) blir denne typen ferdighetstrening trukket frem som noe de anbefaler at skolene prioriterer å bruke tid på. Dette kan bidra til at kjerneelementets innhold, bruk av programmering, kan bli lagt til rette for.

Det kommer frem fra analysen av datamaterialet at elever med programmeringserfaring fikk gjort større deler av oppgaven enn de som hadde arbeidet lite med det tidlige. Erfarne elever så ut til å komme raskt i gang og hadde flere strategier for å komme frem til løsninger av oppgaven. Mindre erfarne elever brukte lang tid og hadde utfordringer med å komme seg videre. Det ble brukt mye tid på å bli kjent med det helt grunnleggende i programmering, de fikk ikke fullført hele oppgaven. Det var mye nytt for mange elevene å sette seg inn i, noe også Tyrén mfl (Tyrén et al., 2018) kom frem til i sin studie. Et forslag disse kommer med, er å introdusere elevene for sentrale konsepter innen programmering, samt feilsøking, i forkant av undervisningen. Dette leder oss videre til diskusjonen omkring hvordan elevene skal forberedes for å tilrettelegge for læring.

Blant det som skilte klassene i min studie fra hverandre, var at de hadde ulike forarbeid før undervisningen. Dermed hadde elevene i utvalget litt ulike utgangspunkt. I klasse C var

innholdet i forarbeidet nettopp det Tyrén (Tyrén et al., 2018) foreslår. De fikk tid til å bli kjent med utstyret og ble introdusert for programmeringsspråket og blokk-editoren de senere skulle bruke. Mange av elevene i denne klassen ble det observert at navigerte seg lett på nettsiden og så ut til å ha en oversikt over hvilke muligheter de hadde i programmeringsspråket. I klasse A og B trengte elevene hjelp til å bruke dette, og store deler av undervisningen gikk til at elevene ble kjent med utstyret og programmeringsspråket. Å øve på å programmere var noe elevene i klasse A og B gjorde samtidig som de skulle løse en oppgave hvor det også var annet naturfaginnhold involvert. Underveis i arbeidet var det noen av disse elevene som ga uttrykk for frustrasjon som så ut til å være relatert til det at de ikke fikk til oppgaven. Noen spurte ofte om hjelp for å finne en løsning, andre elever forlot pulten og gikk til noen andre elevpar. I noen situasjoner stoppet arbeidet opp ved at elevene stilte seg mer passive og så ut til å gi opp. I intervjuene utdypet dette ved å blant annet si at de ble forvirret av at det var mye informasjon å forholde seg til på nettsiden og at det ikke var så lett å vite hvordan den fungerte.

Disse funnene kan tyde på at elevene mer ulik erfaring ble utsatt for ulik kognitiv belastning. Forarbeid med fokus på grunnleggende programmeringsferdigheter var det en av klassene som fikk, og disse elevene var blant dem som fikk til oppgaven uten for mange utfordringer. De hadde fått tid til å øve på ferdigheter de skulle bruke i undervisningen i forkant. Dette la til rette for at elevenes kognitive belastning ble redusert. Slike fordeler av å øve på ferdigheter på denne måten, er i tråd med Sweller mfl. (1998) sin forståelse av kognitiv belastning. Øving i forkant kan bidra til at tankevirksomheten i større grad kan brukes til å løse problemet de arbeider med. Også Taub (Taub et al., 2015) påpeker at dette er en hensiktsmessig måte å tilnærme seg programmering i skolen. Det er altså flere studier og teoretiske perspektiver som peker i samme retning. Dette tyder på at dette er vesentlig innen undervisning og opplæring i programmering i skolen og spesielt naturfag.

For at programmering skal fungere slik det er tiltenkt, trenger både læreren og elevene nok tid til å øve på å programmere. Og som andre ting man lærer, vil kompetansen måtte **holdes ved like**, altså være noe elevene stadig får jobbet med. Forståelsen for de programmeringstekniske konseptene er ifølge Caballero et al. (2014) ferskvare. Det kan gjøre at elevene ikke må bruke unødvendig tid på å repetere før de får kommet i gang med å løse oppgaven. Det kan gjøre at elevene har mer kognitiv kapasitet som kan brukes til å kombinere programmeringen og det naturfaglige innholdet. Dette kan videre knyttes til hvordan programmering blir integrert i

læreplanen. Her er det en tydelig sammenheng mellom kompetansemålene i matematikkfaget og i naturfag. Som nevnt i teorien (se kap 2.1) er det matematikkfaget som har hovedansvaret for å lære elevene å programmere. Dette skal elevene videre bruke blant annet i arbeidet med naturfag. Dette tilsier at disse fagenes innhold er tilpasset hverandre. Resultatet fra undersøkelsen som er presentert i denne masteroppgaven, understreker at for at elevene skal bruke ferdighetene, er det nødvendig å gi dem anledning til å få øvd på dem. Siden matematikk og naturfag har den tydelige sammenhengen med tanke på programmering, tilsier naturfag må samarbeide med matematikk slik det er skrevet inn i læreplanens progresjon i begge fag. Dette poenget understreker viktigheten av tverrfaglig samarbeid.

5.2 Elevenes problemløsning

I denne delen av diskusjonen dreier det seg hovedsakelig om hvordan undervisningsopplegget tilrettelegger for elevenes algoritmiske tenkning. Dette er altså problemløsningsferdigheter. Slik det ble beskrevet i teorikapittelet om *forankring i læreplanen*, er dette begrepet nevnt i Fagfornyelsen. Det er en måte å tilnærme seg problemer hvor programmeringen settes i et bredere perspektiv. Her vil jeg ha fokus på algoritmisk tenkning, spesielt feilsøking og mønstre.

5.2.1 Elevenes tilnærming til feilsøking

Blant hovedfunnene i min studie, er at flere av ferdighetene innen algoritmisk tenkning kom til syne i elevenes arbeid og i intervjuene i etterkant. Dermed ser det ut til at programmeringsaktiviteten legger til rette for at elevene får drevet med problemløsning. Som beskrevet i resultatkapittelet, brukt elevene mye tid på å feilsøke. Dette er en av AT-ferdighetene som både inngår i Udirs modell og Shutes rammeverk (2017; 2019a). Dette er altså en sentral arbeidsmåte i problemløsningen og programmering. Det ble observert at elevene hadde noen feilsøkingstrategier. Måten de gjorde dette på var også noe de satt ord på i intervjuene. Det ble nevnt å starte programmeringsarbeidet helt på nytt, andre går gjennom programmet systematisk, prøver å forutse resultatet, endrer deler av koder hvor de tror feilen ligger. Til tross for at elevene hadde slike strategier, var det tydelig at mange likevel slet med å feilsøke. Dette funnet støttes av Sørby and Angell (2012). Elevene i deres utvalg, hadde utfordringer med feilsøkingen. Både observasjonene og intervjuene tilsier at mange av elevene så ut til å gjøre en form for retningsløs utforskning, altså at problemløsningen hos var usystematisk. Elevenes tilnærming var preget av *prøving-og-feiling*, noe også andre studier

har funnet. I Kohl og Finkelstein (2008) sin studie er dette noe som kjennetegner mange elevers måte å tilnærme seg feilsøkingen i programkoden på. Samtidig var det tydelig at de mer erfarne elevene gikk frem på en mer systematisk måte. Dette er et annet av funnene til Kohl og Finkelstein (2008) tilknyttet elever med programmeringserfaring. I intervjuene beskrev disse elevene fremgangsmåten sin, og kom det blant annet frem at de først brukte tid på å få oversikt over oppgaven, og på å forstå hva oppgaven spurte etter. Deretter la de opp en plan for arbeidet. Det ser altså ut til at elevenes erfaring med programmering vil kunne ha noe å si for hvor systematisk de går frem.

Det så ut som at en utfordring elevene møtte på i arbeidet, var nettopp feilsøking. Modellering av et naturfaglig fenomen med programmering kan gjøre at elevene blir usikre på årsaken til eventuelle feil som oppstår. Noen av elevene sa at de ikke alltid visste om feilen lå i programkoden eller om det kom av at de ikke forstod det naturfaglige innholdet korrekt (Caballero et al., 2014). Det kan underbygges av det Sørby og Angell (2012) har funnet. Ved å tydelig relatere modelleringen til det naturfaglige fenomenet, så kan det bli lettere for elevene å gå systematisk frem i arbeidet.

I likhet med funn fra studien til Sparf et al. (2022) kom det frem at elevene tilnærmet seg programmeringen først og fremst gjennom prøving-og-feiling. I deres studie påpeker de at også denne måten å drive problemløsning på tilrettelegger for utvikling av sentrale ferdigheter i programmering. Samtidig vil det være nødvendig å bistå med støtte. Det var nemlig en god del elever som trengte hjelp med å finne feilen i programkoden. Medelever og lærere sørget for dette i undervisningen, og gjorde at elevene kunne mestre oppgaven. Behovet for støtte har også Sørby og Angell trukket frem som sentralt i elevenes problemløsning. Det kan bidra til at elevene kan tilegne seg en mer systematisk fremgangsmåte (Sørby & Angell, 2012). En måte elever kan oppdage feil og finne ut hvordan de kan løse dem, er ved å gå gjennom steg for steg muntlig hva de ulike blokkene man har satt inn fører til. Det kan gjøre at eleven oppdager hvor feilen ligger (Tyrén et al., 2018).

I undervisningsoppleggets instruksjoner, var det lagt inn egne sider hvor elevene skulle samtale om programkoden. Det var ment til å støtte elevene i feilsøkingen. Ved å gå gjennom disse samtale med medelevene kunne de finne ut hvordan de kan gå frem for å finne og rette opp i feil. Denne form for modellering, kan være en effektiv måte å støtte elevene i feilsøkingen (Rehmat et al., 2020). Det ble observert at elevene ikke brukte så mye tid på

disse samtalene. Det så ut til at de var fokusert på å programmere. Det var altså lagt opp til at elevene kunne få oppøvd sine problemløsningsstrategier. Funnene tilsier at det potensialet som lå i instruksjonene muligens kunne blitt utnyttet enda bedre. Ved å ta disse studiene i betraktning, får det frem at utviklingen av feilsøkingsferdigheter er noe lærere kan legge til rette for ved å understreke og følge opp elevene i arbeidet og samtalene.

De tekniske feilene som oppstod tilknyttet den fysiske programmeringen med micro:bit, skapte en del ekstra forstyrrelser i elevenes arbeid. Dette var det noe samtlige elever trakk frem som utfordring i intervjuene. Et interessant funn tilknyttet dette Caballero (2014) trekker frem. Programmeringstekniske utfordringer som oppstår, har lett for å komme i veien i elevenes systematiske feilsøking. Det kan skape en ekstra kognitiv belastning (Taub et al., 2015). For å demme opp om dette, vil igjen elevenes forkunnskaper være avgjørende. I klasse C brukte elevene tid i forkant på å bli vant til å bruke micro:bitene og øvde på å programmere dem. Også disse elevene slet med tekniske utfordringer. Det var en stund mellom forarbeidet og gjennomføringen av undervisningsopplegget. Det kan tyde på at ferdighetene er noe som må vedlikeholdes og krever tid (Caballero et al., 2014). Dette poenget med tekniske utfordringer er det sannsynlig at ikke vil være noe elevene bestandig vil møte på. Lærere kan gi støtte og oppfølging underveis i arbeidet bidra til å moderere tekniske utfordringer elevene møter i arbeidet (Nordby, 2019).

5.2.2 Koding

Etter nå å ha diskutert elevenes feilsøking, skal jeg trekke inn elevenes *koding*. Det er et sentralt aspekt ved algoritmisk tenkning. Det går på å se mønstre, effektivitet i programmeringen osv (se rammeverk). Å justere, effektivisere koden. Bruke tidligere løsninger i nye situasjoner. Det var ulikt hvor effektive elevene var i programmeringsarbeidet. Problemløsingen påvirkes av hvor godt man klarer å dra nytte av tidligere løsninger i nye situasjoner (Woolfolk, 2016). Det ble observert at noen elever valgte å gjenbruke kode istedenfor å starte på nytt og gjenta hele programmeringen. Dette indikerer generalisering, som vil si å kunne bruke løsninger i nye situasjoner (Shute et al., 2017). Dette leder oss videre i diskusjonen på bruk av utforsking i programmeringsarbeidet.

En av arbeidsmåtene i *den algoritmiske tenkeren* er å *fikle*. Det innebærer å utforske og eksperimentere. Ved å fokusere på denne arbeidsmåten i undervisningsopplegget, vil det kunne tilrettelegge for at elevene kan bruke programmering til å utforske. Slik denne

modellen blir beskrevet, er det naturlig å ta i bruk flere av arbeidsmåtene (Utdanningsdirektoratet, 2019a). *Å fikle* vil være naturlig å relatere til blant annet *å skape*. Elevene skulle lage en modell av hjemmeruteren, og skulle gjøre det ved å programmere.

Med tanke på elevenes utforskende arbeid, vil det å oppmuntre elevene til å være kreative og ikke ha like gitte rammer for oppgaven være hensiktsmessig. Noe av hensikten med undervisningsopplegget til Skaperskolen, var skapende aktivitet. I undervisningsopplegget fulgte elevene en instruksjon for å løse oppgaven, og aktiviteten var lærerstyrt (Skaperskolen, 2020a). Hensikten med dette var å kunne lede elevene i arbeidet til å utvikle en modell for det teknologiske systemet. Det ble observert at elevene derfor hovedsakelig gjorde slik instruksjonen sa. Dette var også noe de bekreftet i intervjuene. Noen av elevene sa også at de gjerne skulle hatt enda tydeligere instruksjoner. De opplevde at det var utfordrende å vite hvordan programkoden skulle settes sammen på egen hånd. Disse elevene hadde lite programmeringserfaring. Det kan tenkes at de opplevde en stor kognitiv belastningen i arbeidet, noe som kan gå ut over deres kapasitet til å drive problemløsning (Taub et al., 2015). Disse elevene ble observert at gikk frem uten en klar plan, noe også andre studier har funnet (Sørby & Angell, 2012). I programmeringsarbeidet ser utforskingen ut til å være tilfeldig. Ifølge Taub (2015) vil en tydelig oppgavetekst kunne bidra til at læringsprosessene og metodene elevene skal ta i bruk faktisk blir benyttet. På denne måten kan det sies at måten undervisningsopplegget var planlagt, samsvarer med annen forskning.

Dette får frem at støtte i instruksjonene kan være en måte å hjelpe elevene i arbeidet og tilrettelegge for læring. Det ble observert at de elevene med mer programmeringserfaring utforsket andre måter å sette sammen programkoden på. Det kan tenkes at ved å gi elevene tilstrekkelig anledning til å lære seg programmering og «lekt» seg med det, vil kunne gjøre dem bedre forberedt til å være nytenkende i arbeidet. Dette kunne lagt ytterligere til rette for å få frem en kreativ problemløsning, og videre algoritmisk tenkning (Utdanningsdirektoratet, 2019a). Det var altså et potensiale i undervisningsopplegget tilknyttet å få elevene til å være skapende i arbeidet. Dette forutsetter at elevene forstår hva de skal gjøre, og har oversikt over programmeringen.

5.3 Støtte i elevers læring

I denne delen av diskusjonen ønsker jeg å trekke frem måter elevene kan bli støttet i arbeidet, og dermed tilrettelegge for læring. Hovedfunnene fra resultatene er blitt undersøkt ved å se dem på tvers av forskningsspørsmålene. I diskusjonen som følger dreier seg om følgende tre hovedtemaer: 1) samarbeid mellom elevene, 2) undervisningsmateriell og 3) støtte fra læreren. Det er denne rekkefølgen diskusjonen struktureres etter.

5.3.1 Samarbeid mellom elever

I undervisningsopplegget programmerte elevene modellen av hjemmeruteren sammen med en medelev. Det er en av støttestrukturene som Skaperskolen har lagt inn i sitt undervisningsopplegg, for å unngå at elevene stopper opp i arbeidet (Skaperskolen, 2020c). Dette virket til å være hensiktsmessig av flere grunner. I de følgende avsnittene trekkes det inn elevenes språklige ferdigheter og begrepsforståelse. Begrepslæring er en stor del av naturfagundervisningen. Likevel er det gjort en avgrensning, slik at det er kun noe av dette som blir tatt med i diskusjonen. Det neste som inkluderes under samarbeid, er betydningen av dette med tanke på å kunne gjøre at elevene får til større deler av oppgaven, altså at det tilrettelegger for å få støtte i arbeidet.

Blant hovedfunnene tilknyttet samarbeid, er at det tilrettelegger for utvikling av elevenes språklige ferdigheter. I samtalene tilknyttet arbeidet med modelleringen, ble sentrale begreper for undervisningen tatt i bruk (Se vedlegg 5, resultat tabeller). Derfor indikerer resultatene at gjennom samarbeid kan elevene utvikle sin begrepsforståelse. Elevene så ut til å være på ulike steder i sin begrepsforståelse. Noen klarte å gjengi definisjoner av begrepene, som er et tegn på en passiv begrepsbruk. Det var andre elever som brukte begrepene i ulike setninger og i utforskningen av hjemmeruteren. Dette indikerer en aktiv begrepsbruk (Haug, 2016). Elevene fikk diskutert blant annet ved å trekke inn naturfaglig innhold inn i det praktiske arbeidet med modellene. Å diskutere inngår som en sentral del av det utforskende arbeidet Dette så ut til å gjøre elevene tryggere både på modellen og det teknologiske systemet, som underbygger det Mork (Mork, 2016) nevner i sin artikkel. Diskusjon kan tilrettelegge for utvikling av elevenes begrepsforståelse.

Når elevene får satt ord på det de tenker og arbeider med, kan de bli mer oppmerksomme på egen forståelse av begrepene. I tillegg vil det å få høre hvordan andre setter ord på

faginnholdet, kunne styrke elevenes språklige ferdigheter og begrepsforståelse (Mork & Erlien, 2017). Elevene brukte kunnskapen de hadde og revurderte egne ideer ut ifra ny informasjon de fikk. Dette er sentralt i kunnskapsintegrasjon (Linn & Eylon, 2011). Det at resultatene av samarbeid kan tilrettelegge for at elevene får koblet ny kunnskap til det de allerede har lært. Både i lys av dette og sosiokulturell læringsteori (Vygotsky, 1987), kan elevenes læring bli styrket ved å samarbeide med andre.

Samtidig som elevenes språklige ferdigheter kan utvikles gjennom samarbeid, ble det tydelig at elevenes kommunikasjon hadde betydning for hvor godt dette faktisk fungerte. I intervjuet var det en elev som sa at han noen ganger hadde utfordringer med å forklare seg på en måte som var forståelig for medeleven. Dermed fikk han erfare noe Webb og Mastergeorge (2003) trekker frem som sentralt for et godt samarbeid. Personer som skal hjelpe andre, vil gjøre dette på en god måte dersom forklaringene er presise. Derfor la undervisningsopplegget til rette for at elevene får utviklet sine språklige ferdigheter. Samarbeidet kan gjøre at elevene selv blir oppmerksomme på viktigheten av språket som videre vil kunne forsterke effekten samarbeid kan ha på elevenes språklige ferdigheter.

Det neste hovedfunnet er at samarbeid tilrettela for at elevene ble støttet i læringsprosessen. Underveis i observasjonene delte elevene ideer og løsningsforslag, noe som gjorde at de kunne spille på hverandre og komme frem til en løsning på problemet. I Udirs modell (2019a) av algoritmisk tenkning, inngår samarbeid blant sentrale arbeidsmåter i problemløsingen. Fra observasjonene var det flere ganger samarbeid mellom elever gjorde at de fikk til oppgaven. I samtlige intervjuer uttrykte elevene at dette var noe de hadde nytte av. Dette har også Cao and Xu (2005) funnet i sin undersøkelse av parprogrammering hos elever. Ut ifra disse funnene og sosiokulturell læringsteori, ser det ut til at samarbeid er fordelaktig med tanke på elevenes læring.

Selv om elevene arbeidet i par og uttrykte at dette var noe de hadde nytte av, ble det observert at parenes samarbeid fungerte ulikt. Dette leder oss inn på diskusjonen omkring parsammensetning. Fra min studie var det tydelig at i de parene hvor begge elevene hadde noe erfaring med programmering hadde større progresjon og var preget av at begge var delaktige i arbeidet. De skilte seg fra parene hvor begge elevene hadde lite programmeringserfaring. Her var det ikke like mye snakk mellom elevene og arbeidet stoppet forttere opp.

Parsammensetning så dermed ut til å være avgjørende for hvordan samarbeidet fungerte, noe

som støttes av funnene fra Cao and Xu (2005) sin undersøkelse. Måten parene hvor de stoppet opp i arbeidet håndterte det med å få hjelp av elever i andre par med mer programmeringserfaring. I lys av forskningen og mer generell læringsteori, begrepet *proksimal utviklingszone* forutsetter det at arbeidet støttes av en person med mer erfaring (Woolfolk, 2016). Derfor vil det å sette elevene sammen på en måte hvor elevene har med seg kompetanse som trengs inn i arbeidet. Derfor vil det at minst en av elevene har arbeidet noe med programmering tidligere kunne bidra til å få utnyttet potensialet ved et samarbeid.

I intervjuene uttrykte både elever med mye og mindre erfaring at de hadde lært mye av å få og gi hjelp. De trakk frem at det var nyttig å repetere fagkunnskapen og ferdighetene, i tillegg til at det å skulle forklare til andre gjorde at de forstod det bedre. Dette støttes av flere studier, både Cao and Xu (2005) og Denner mfl. (2014). Ifølge Denner vil de elevene som ikke har så mye erfaring kan lære mye av andre, og de elevene som har mer erfaring får større selvtillit og mer positive holdninger til det å programmere når de arbeider med en som har mindre erfaring enn seg selv. For å trekke linjer mellom det som kommer frem i denne delen av diskusjonen, ser det altså ut til at samarbeid er en god måte å la elevene arbeide med undervisningsopplegget på. Samtidig forutsetter det at elevene settes sammen på en måte hvor de kan støtte hverandre.

Samarbeidet la også til rette for at elevene kunne bygge videre på hverandres tanker. Elevene snakket sammen om løsningene de hadde programmert. Ifølge Denner (2014) vil elevenes programmeringskunnskaper kunne bli høyere når elevene arbeider på denne måten. I flere tilfeller i observasjonen ble løsningene mer avanserte ved at de arbeidet med dem i fellesskap. Dette funnet underbygges av flere studier. Harlow og Leak sin studie (2014), hvor parprogrammering er blitt undersøkt. Samarbeidet ledet elevene til at de justerte oppbyggingen av programkoden og resultatet (output) ut ifra det de sammen ble enige om. Slike fordeler ved samarbeid har også Lee mfl (2011) presentert i resultatene sine. Det å gi elevene anledning til å dele og sette ord på det de arbeider med, er viktig for deres utvikling av AT. Ut ifra disse studiene, som står i samsvar med min, sier det oss at parprogrammering er en hensiktsmessig tilnærming. I planlegging av undervisning vil det derfor være viktig at læreren er bevisst på måten elevene settes sammen. I klasser hvor det er ulikt hvor mye elevene kan i programmering, så kan samarbeid være hensiktsmessig. De vil da kunne rådføre seg med en annen som har mer erfaring. Dermed blir elevene ressurser for hverandre.

5.3.2 Undervisningsmaterieil

Elevene jobbet med oppgaven ved å følge en PowerPoint. Den bestod av ulike deloppgaver. Disse hadde en progresjon, som vil si at det etter hvert ikke bare var for elevene å kopiere eksemplene (programkodene) som stod. For å få micro:bitene til å fungere, var de nødt til å selv finne ut hvilke blokker som «manglet». Mange av elevene hadde utfordringer med å forstå instruksjonene. De trengte hjelp, spesielt de delene som var mer åpne, at elevene måtte bruke forkunnskapene til å selv vite hvordan programmet skulle settes sammen. Det får frem at det er viktig at elevene jobber med oppgaver som er passe utfordrende for at det skal fremme læring.

Elevene fulgte som sagt instruksjonens deloppgaver. Derfor var det føringer for hva elevene skulle programmere. Derfor var de avhengige av å forstå hva de skulle gjøre. Her kan det trekkes inn oppgaveformen i programmering. At den bør være på en måte hvor elevene kan prøve seg frem. Det skal være et fleksibelt verktøy. Dersom oppgaven er lukket og det er bestemte føringer, kan det gjøre at elevene blir redde for å gjøre feil.

Micro:bit er i utgangspunktet designet for at det skal være lett å ta i bruk, at lærere kan ta det i bruk i klasserommet uten å måtte bruke mye tid på å sette seg inn i det i forkant (Ball et al., 2016). Likevel ble det tydelig i min studie at utstyret ikke alltid fungerte slik det skulle, noe som skapte uromomenter i elevenes arbeid. Oppstod situasjoner hvor det var vanskelig for elevene og læreren å vite hva som var årsaken til at programmet ikke fungerte. De måtte derfor i tillegg til å holde styr på hvordan programkoden skulle settes sammen, også sjekke utstyret underveis. Dette kan distrahere, skape unødvendig frustrasjon, fokus vekk fra selve programmeringen. Tar fokus fra det som er relevant. En ekstra kognitiv belastning (Taub et al., 2015). Ytre.

Både micro:bitene og PC-ene ble byttet ut med andre, og noen ganger gjorde det at feilen ble fikset. En ting er at micro:biter ofte er noe klassene har nok av til å kunne være rustet til slike tekniske feil. En annen ting er PC-er. Det er ikke alle skoler som har én til én data. I slike klasserom vil det gjøre at elevene ikke like lett får kommet videre i arbeidet. En måte å løse det på er ved at elevene jobber sammen med grupper hvor PC-ene fungerer. Det som er en uting ved dette er at da vil ikke elevene få programmert like mye, siden det er flere elever som må dele på det samme utstyret. Tekniske feil kan være et hinder for læringen. Her lurere jeg derfor på om det da vil være hensiktsmessig å bruke en annen programmeringstilnærming enn

micro:bit? Kan det teknologiske systemet arbeides med på andre måter enn dette fysiske programmeringsverktøyet? Den diskusjonen tar jeg ikke videre, men det er noe som kunne blitt gjort.

Bruk av ulike støttestrukturer kan demme opp for at tekniske utfordringer står i veien for at de får løst oppgaven (Nordby, 2019). Modellere utstyret, vise elevene hvordan det skal brukes. For å kunne gi denne støtten, trenger lærere tilstrekkelig forkunnskaper. Kjenne til typiske tekniske feil som kan oppstå. Det kan gjøre det lettere å vite hva den kan gjøre for å hjelpe elevene videre. Læreren som støtte er det neste som skal diskuteres.

5.3.3 Støtte fra læreren

Det kommer frem fra analysen av datamaterialet, at lærerens forkunnskaper hadde betydning for hvordan elevenes læring ble støttet. Lærerne hadde ulike forkunnskaper innen programmering og erfaringer med å bruke det i undervisning. Den ene virket tryggere på temaet enn den andre. Dette gjenspeiler lærerne «der ute». Det kommer frem fra flere studier at mange lærere har lite kompetanse i programmering (Bocconi et al., 2022; Moreau, 2021). Det disse studiene peker på at noen vil føle seg trygge på å gjennomføre slike undervisningsopplegg. Andre vil gjøre det uten den samme selvtilliten. Den av dem som hadde undervist i programmering tidligere og hadde god innsikt i programmering, så ut til å kunne gi elevene nødvendig veiledning og støtte. I flere situasjoner hvor dette ble gitt, så det ut til at elevene klarte å tilnærme seg problemet på en mer systematisk måte. Dette kan relateres til algoritmiske tenkning. Rehmat mfl. (2020) har funnet i sin studie at støtte fra læreren kan gi forbedrede ferdigheter i algoritmisk tenkning. Dermed kan det se ut til at lærere med denne forkunnskapen kan tilrettelegge for at støttestrukturene fremmer læring hos elevene (Woolfolk, 2016).

5.4 Metodediskusjon og videre forskning

Temaet denne studien tar for seg er nokså nytt i forskningssammenheng. De poengene denne studien har kommet frem til, sier noe om hvorfor og hvordan programmering kan tilrettelegge for læring i naturfag. Dette kan få frem hvordan undervisning tilknyttet programmering kan bli gjennomført, og hvilke utfordringer som kan oppstå. Dette kan videre bli undersøkt og belyst gjennom mer forskning.

Datamaterialet er samlet inn fra noen få gjennomføringer av undervisningsopplegget. Derfor er det ikke grunnlag til å generalisere, men resultatene kan være gjenkjennelige og dermed ha overføringsverdi. Datainnsamlingen ble gjort ved å tilnærme meg undervisningsopplegget og elevene fra ulike vinkler, både fokusgruppeintervju og observasjon. Styrken ved denne formen på undersøkelsen, er at datamaterialet kan komplementere hverandre. Data fra observasjonene, kunne utdypes og bli ytterligere belyst ut ifra data som er samlet inn fra fokusgruppeintervjuene. Noe av hensikten med studien var å få frem læring tilknyttet programmering. Dette er komplekst, men triangulering kan frem ulike aspekter som til sammen kan si noe om elevenes læring.

Utvalget i studien er ikke stort nok til at funnene kan si noe generelt, men det kan være overførbart til liknende kontekster utvalget ble hentet fra. Elevene som deltok, var plukket ut fra en helt gjennomsnittlig klasse. Det var ikke noe ved elevene som gjør at de skiller seg ut fra andre elever. I metodekapittelet, er utvalget beskrevet, for at leseren selv kan vurdere om det er gjenkjennelig til den situasjonen leseren sammenlikner med. Utvalget bestod av elever med ulike utgangspunkt med tanke på programmering, noe som er sannsynlig at det også vil være i andre klasser. Dermed kan funnene i studien få frem flere sider ved å undervise i klasser hvor elever har ulike forutsetninger. Et slikt gjennomsnittlig utvalg, bidrar til at resultatene vil kunne være gjenkjennelige også i andre klasser.

Også blant lærerne som gjennomførte undervisningsopplegget var det en bredde. Lærernes kompetanse var ulik, slik at de i ulik grad kunne fungere som støtte for elevene. Utvalget av disse ble tatt ut ifra at lærerne hadde deltatt på Skaperskolens kurs i *Internett fanger*. Bortsett fra dette skiller de seg ikke ut på noe måte. Den ene læreren hadde lite kjennskap til grunnleggende programmering, samt lite erfaring med å undervise i dette. Den andre læreren hadde høyere kompetanse i programmering og var vant til å bruke det i egen undervisning. Det er sannsynlig at situasjonen er lik på mange skoler og at lærere vil kunne kjenne seg

igjen. Måten undervisningsopplegget ble gjennomført, vil på kunne relateres til liknende situasjoner. Det skal i de neste avsnittene, diskuteres noen av studiens begrensninger.

Hensikten med studien var å undersøke bruk av programmering i naturfag. Derfor ble Skaperskolens undervisningsopplegg aktuelt å ha som utgangspunkt, siden programmering inngår her. En svakhet med studien er at konteksten programmeringsaktiviteten var ment å settes inn i ikke ble tatt med i undersøkelsen. Det er sannsynlig at resultatene vil kunne blitt annerledes dersom elevene hadde deltatt i hele undervisningsopplegget til Skaperskolen. Funnene vil likevel kunne gi en indikasjon på hvordan opplegget fungerer.

I kvalitative studier, vil forskeren kunne spille inn på hva som kommer frem under og i etterkant av datainnsamlingen. Det ble fulgt råd fra litteraturen som omhandler forskningsmetode og fra veileder. Likevel vil måten datamaterialet bli samlet inn, vil være avhengig av min kompetanse. Både observasjon og fokusgruppeintervjuer, er metoder som er komplekse. Det ble gjort flere forberedelser i forkant, slik som pilotering, som var for å styrke kvaliteten på datamaterialet. Likevel kan det tenkes at min erfaring og fagkompetanse vil kunne spille inn på hvordan observasjonene og fokusgruppeintervjuene blir gjennomført og tolket. Dersom det var noen andre forskere som gjennomførte studien, kan det ha gitt andre resultater.

Neste del av dette delkapittelet dreier seg om forslag til videre forskning. Skaperskolens undervisningsopplegg består av totalt åtte deler, men det er kun én av dem som er inkludert i denne studien. Dermed kan det i etterkant av denne studien være aktuelt å undersøke større deler av dette undervisningsopplegget. En tilnærming kan være å følge elever over lengre tid, ved at de deltar på hele eller større deler av opplegget. Dette kan få frem hvordan konteksten spiller inn på elevenes arbeide med programmering, og gi ytterlige innsikt om forarbeidets betydning for hva elevene får arbeidet med.

I denne studien har det vært fokus på bruk av programmering tilknyttet læring av et teknologisk system. Et aktuelt forskningstema vil være å se på bruk av programmering i opplæringen av andre naturfaglige temaer. Det kan gi kunnskap som kan sammenliknes med funn fra denne studien. Dermed kan det komme frem om det samsvarer med undervisningsopplegg hvor det er andre systemer som blir arbeidet med.

Noe av det sentrale som kommer frem fra denne undersøkelsen, er samarbeidet som er naturlig å ha mellom naturfag og matematikk. Slik det kommer frem fra LK20, er kompetansemålene i disse fagene relatert til hverandre. Grunnet den nylige innføringen av programmering i skolen, vil elever og lærere ha mer erfaring og kompetanse i programmering i fremtiden enn det utvalget som ble undersøkt i denne studien. Derfor hadde det vært interessant å undersøke hvordan programmering tilrettelegger for læring av naturfaglige temaer i fremtidens skole. Det kan gi andre resultater siden da vil både lærere og elever ha kommet seg gjennom innkjøringsfasen med å implementere programmering i fagene.

6. Konklusjon

Hensikten med denne studien, var å undersøke hvordan programmering tilrettelegger for læring om teknologi i naturfag. Dette er aktuelt å undersøke siden programmering nylig ble innført i grunnskoleopplæringen. Programmering inngår sammen med modellering i kjerneelementet *Teknologi* i den nye læreplanen for naturfag. Her står det blant annet at elevene skal bruke programmering i arbeid med naturfag. Noe av grunnen til det, er at programmering kan bidra til at elevene utvikler problemløsningskompetanse. Dette defineres som algoritmisk tenkning, og er noe elevene vil kunne trenge i fremtidig arbeids- og hverdagsliv. Ifølge litteratur og tidligere forskning, er det et stort potensial for læring ved bruk av programmering. Det kan bidra til at elever får utviklet sin systemtenkning og kompetanse om modellering, som er sentralt innen naturvitenskapen.

For å belyse hensikten, er det blitt gjennomført fokusgruppeintervjuer og observasjoner av elever som deltok i et undervisningsopplegg hvor programmering inngår.

Undervisningsopplegget er utarbeidet av Skaperskolen, og her skal elevene programmere micro:bit for å modellere funksjonen til en hjemmeruter. Blant hovedfunnene fra denne studien, er at elevene fikk anledning til å utvikle kunnskap omkring det teknologiske systemet i arbeidet med oppgaven. Flere av ferdighetene innen algoritmisk tenkning ble også synlig, blant annet å feilsøke og å se mønstre. Noe annet som kommer frem, er at forkunnskapene elevene har, vil ha innvirkning på hva elevene får arbeidet med og hvordan de jobber. For å få unyttet potensialet som ligger i bruk av programmering i naturfagundervisningen, forutsetter det at elevene har forkunnskapene de trenger. Dette er relatert både til det teknologiske systemet og programmering. Elevenes forkunnskaper omkring det teknologiske systemet og at de får støtte til å dra nytte av disse forkunnskapene, vil spille inn på arbeidet med modelleringen. Elever som hadde programmeringserfaring, så ut til å i større grad kunne fokusere på det naturfaglige innholdet. Det ser derfor ut til å være fordelaktig at elevene får tilstrekkelig tid til å øve på å programmere, og på å sette seg inn i det naturfaglige innholdet.

Elevenes behov for støtte, ser også ut til å være avgjørende for hvordan arbeidet til elevene fungerer. Det var tydelig at mange elever tilnærmet seg problemløsingen gjennom prøving- og-feiling, og at de trenger støtte for at problemløsingen skal bli mer systematisk. Lærerens måte å veilede på kan bidra til dette. Tydelige instruksjoner som får frem nødvendig informasjon, kan gjøre at elevene raskere kommer i gang og vet hvordan de skal gå frem for å

løse oppgaven. Dette vil kunne bidra til økt progresjon i arbeidet. For at læreren skal kunne vite hva slags støtte og til hvilket tidspunkt, vil også lærerens kompetanse spille en rolle på hvordan programmeringen tilrettelegger for læring.

En implikasjon fra denne studien er viktigheten av å tilpasse undervisningen etter elevenes forkunnskaper. Slik det kommer frem fra studien, vil elever med ulike forutsetninger kunne dra nytte av programmering på ulike måter. Det vil være forskjell på hvordan oppleggene utformes ut ifra forutsetningene til elevene. Dersom klassen er ny for programmering, vil det være nødvendig å legge inn tilstrekkelig med tid til å øve og holde vedlike ferdighetene i programmering. Det kan bidra til at elevene kan bruke programmering slik det er tenkt i LK20. Som i andre temaer i naturfag, vil klassen bestå av en elevgruppe med ulike forkunnskaper. Dette kan bli en ressurs i arbeidet, ved at de elevene som allerede har en del erfaring kan lære dette videre til sine medelever. Fra denne studiens resultater, ble det synlig at samarbeid på denne måten kan tilrettelegge for læring. I tillegg til at det kan være nyttig å dra nytte av erfarne elever for lærere som ikke har så mye kompetanse i programmering.

Programmering er et tema som nylig ble innført i skolens innhold, og resultatene fra studien gjenspeiler den spesielle situasjonen vi er. Tekniske utfordringer og tid til å lære seg grunnleggende programmering, preget gjennomføringene, siden det er nytt både for elever og lærere. På sikt vil dette sannsynligvis forandre seg. Da kan potensialet ved bruk av programmering i naturfagundervisningen komme bedre til sin rett, ved at de da har fått utviklet de forkunnskapene de trenger. Det vil derfor være spennende å følge utviklingen i bruk av programmering i undervisning i tiden som kommer.

7. Litteraturliste

- Andersen, S. S. (2013). *Casestudier - Forskningsstrategi, generalisering og forklaring*. Bergen: Fagbokforlaget.
- Angell, C., Bungum, B., Henriksen, E. K., Kolstø, S. D., Persson, J., & Renstrøm, R. (2011). *Fysikkdidaktikk*. Kristiansand: Høyskoleforl.
- Aronson, D. (1996). Overview of systems thinking. Retrieved from <http://www.atmosedu.com/WSU/esrp310-550/articlesSysDyn/SystemsThinkingOverview.pdf>
- Balanskat, A., Engelhardt, K., & Ferrari, A. (2017). *The integration of Computational Thinking (CT) across school curricula in Europe*. Retrieved from Brussels: http://www.eun.org/documents/411753/665824/Perspective2_april2017_onepage_def.pdf/70b9a30e-73aa-4573-bb38-6dd0c2d15995
- Ball, T., Protzenko, J., Bishop, J., Moskal, M., Halleux, J. d., Braun, M., . . . Riley, C. (2016). Microsoft Touch Develop and the BBC micro:bit. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7883359>
- Basu, S., Biswas, G., & Kinnebrew, J. (2017). Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, 26. doi:10.1007/s11257-017-9187-0
- Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, 11(1), 13. doi:10.1186/s41039-016-0036-2
- Berland, M., & Wilensky, U. (2015). Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking. *Journal of Science Education and Technology*, 24(5), 628-647. doi:10.1007/s10956-015-9552-x
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagiené, V., Wastiau, P., Engelhardt, K., . . . Stupuriené, G. (2022). *Reviewing Computational Thinking in Compulsory Education*. Retrieved from Luxembourg:
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3, 77-101. doi:10.1191/1478088706qp063oa
- Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research*, 87(4), 834-860. doi:10.3102/0034654317710096
- Bungum, B., Esjeholm, B.-T., & Lysne, D. A. (2014). Science and Mathematics as part of practical projects in technology and design: An analysis of challenges in realising the curriculum in Norwegian schools. *Nordic Studies in Science Education*, 10(1), 3-15.
- Caballero, M. D., Burk, J. B., Aiken, J. M., Thoms, B. D., Douglas, S. S., Scanlon, E. M., & Schatz, M. F. (2014). Integrating Numerical Computation into the Modeling Instruction Curriculum. *The Physics Teacher*, 52(1), 38-42. doi:10.1119/1.4849153
- Çakıroğlu, Ü., Çevik, İ., Köşeli, E., & Aydın, M. (2021). Understanding students' abstractions in block-based programming environments: A performance based evaluation. *Thinking Skills and Creativity*, 41. doi:10.1016/j.tsc.2021.100888
- Cao, L., & Xu, P. (2005). *Activity patterns of pair programming*. Paper presented at the Proceedings of the 38th Annual Hawaii International Conference on System Sciences.
- Christoffersen, L., & Johannessen, A. (2012). *Forskningsmetode for lærerutdanningene*. Oslo: Abstrakt forlag.
- Cohen, L., Manion, L., & Morrison, K. (2018). *Research methods in education* (8 ed.). New York: Routledge.

- Dalland, O. (2017). *Metode og oppgaveskriving*. Oslo: Gyldendal Akademisk.
- Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair Programming: Under What Conditions Is It Advantageous for Middle School Students? *Journal of Research on Technology in Education*, 46(3), 277-296. doi:10.1080/15391523.2014.888272
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39. doi:10.1145/2998438
- Drageset, S., & Ellingsen, S. (2010). Å skape data fra kvalitativt forskningsintervju. *Sykepleien Forskning*, 5(4), 332-335. doi:<https://doi.org/10.4220/sykepleienf.2011.0027>
- Duckworth, D. (2019). Assessing computational thinking. *Australian Council for educational research*. Retrieved from https://research.acer.edu.au/cgi/viewcontent.cgi?article=1359&context=research_conference
- Eide, B. A., Lauritzen, D. O., & Bernitz, M. (2022). *Ingenting er umulig - Om utfordringer, mestring, viljestyrke og fellesskap*. Oslo: Kagge forlag.
- Eidsvig, P.-E. (2021). Slik kan du kombinere programmering og tverrfaglighet. *Utdanningsnytt.no*. Retrieved from <https://www.utdanningsnytt.no/fagartikkel-pedagogikk-programmering/slik-kan-du-kombinere-programmering-og-tverrfaglighet/282835>
- Fox-Turnbull, W. (2009). Stimulated recall using autophotography-A method for investigating technology education. *Strengthening the position of technology education in the curriculum. Delft, The Netherlands: International Technology and Engineering Educators Association*.
- German, S. (2019). Computational thinking. *Science Scope*, 42(9), 36-39. Retrieved from <https://www-jstor-org.ezproxy.oslomet.no/stable/26899029>
- Gjøvik, Ø., & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten - tidsskrift for matematikkundervisning*(3), 31-37. Retrieved from chrome-extension://efaidnbmninnibpcapjpeglclefindmkaj/viewer.html?pdfurl=http%3A%2F%2Fwww.caspar.no%2Ftangente%2F2019%2FTangente%25203%25202019%2520Gj%25C3%25B8vik%2520Torkildsen.pdf&clen=171265&chunk=true
- Grover, S., & Pea, R. (2013a). Computational Thinking in K-12. *Educational Researcher*, 42(1), 38-43. doi:10.3102/0013189x12463051
- Grover, S., & Pea, R. (2013b). Computational Thinking in K-12 A Review of the State of the Field. *Educational Researcher*, 42, 38-43. doi:10.3102/0013189X12463051
- Halfacree, G. (2017). *The official BBC Micro: bit user guide*: John Wiley & Sons.
- Hannisdal, M., & Ringnes, V. (2015). *Kjemi for lærere*. Oslo: Gyldendal Akademisk.
- Harlow, D. B., & Leak, A. E. (2014). Mapping students' ideas to understand learning in a collaborative programming environment. *Computer Science Education*, 24(2-3), 229-247. doi:10.1080/08993408.2014.963360
- Haug, B. S. (2016). Begrepsforståelse og vurdering underveis i en utforskning. In M. Ødegaard, B. S. Haug, S. M. Mork, & G. O. Sørvik (Eds.), *På forskerfotter i naturfag*. Oslo: Universitetsforlaget.
- Haug, B. S., Sørborg, Ø., Mork, S. M., & Frøyland, M. (2021). Naturvitenskapelige praksiser og tenkemåter-på vei mot et tolkningsfellesskap. *Nordic Studies in Science Education*, 17(3), 293-310. doi:<https://doi.org/10.5617/NORDINA.8360>
- Holt, A., Øyehaug, A. B., & Voll, L. O. (2019). Undervisning for dybdelæring. In L. O. Voll, A. B. Øyehaug, & A. Holt (Eds.), *Dybdelæring i naturfag*. Oslo: Universitetsforlaget.
- Hoppe, H. U., & Werneburg, S. (2019). Computational thinking - more than a variant of scientific inquiry. In S.-C. Kong & H. Abelson (Eds.), *Computational thinking education*. Singapore: SpringerOpen.

- Hromkovic, J., Kohn, T., Komm, D., & Serafini, G. (2016). Examples of Algorithmic Thinking in Programming Education. *Olympiads in Informatics, 10*, 111-124. doi:10.15388/ioi.2016.08
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*, 263-279. doi:<https://doi.org/10.1016/j.compedu.2014.11.022>
- Johannessen, A., Christoffersen, L., & Tufte, P. A. (2016). *Introduksjon til samfunnsvitenskapelig metode* (5. utg. ed.). Oslo: Abstrakt.
- Kluge, A. (2021). *Læring med digital teknologi*. Oslo: Cappelen damm.
- Kohl, P. B., & Finkelstein, N. D. (2008). Patterns of multiple representation use by experts and novices during physics problem solving. *Physical Review Special Topics - Physics Education Research, 4*(1), 010111. doi:10.1103/PhysRevSTPER.4.010111
- Kunnskapsdepartementet. (2017). *Overordnet del - verdier og prinsipper for opplæringen*. Retrieved from <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/kompetanse-i-fagene/?lang=nob>
- Kvale, S., & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg. ed.). Oslo: Gyldendal akademisk.
- Lai, C., & Lai, M. (2012, 4-6 June 2012). *Using Computer Programming to Enhance Science Learning for 5th Graders in Taipei*. Paper presented at the 2012 International Symposium on Computer, Consumer and Control.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., . . . Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads, 2*(1), 32–37. doi:10.1145/1929887.1929902
- Leseth, A. B., & Tellmann, S. M. (2018). *Hvordan lese kvalitativ forskning?* Oslo: Cappelen Damm.
- Linn, M. C., & Eylon, B.-S. (2011). *Science Learning and Instruction : Taking Advantage of Technology to Promote Knowledge Integration*. Florence, UNITED STATES: Taylor & Francis Group.
- Lodi, M., & Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Science & Education, 30*(4), 883-908. doi:10.1007/s11191-021-00202-5
- Louca, L. T., & Zacharia, Z. C. (2008). The Use of Computer-based Programming Environments as Computer Modelling Tools in Early Science Education: The cases of textual and graphical program languages. *International Journal of Science Education, 30*(3), 287-323. doi:10.1080/09500690601188620
- Micro:bit Educational Foundation. (2022). Makecode - BBC micro:bit. Retrieved from <https://microbit.org/>
- Mishra, P., & Koehler, M. J. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record, 108*(6), 1017-1054.
- Moreau, H. N. (2021, 31.12.21). Skal lære elevene koding, men forstår det ikke selv. Retrieved from <https://www.nrk.no/innlandet/laerere-trenger-hjelp-til-a-knekke-koden-pa-koding-1.15781343>
- Mork, S. M. (2016). Å diskutere som del av utforskende arbeid. In M. Ødegaard, B. S. Haug, S. M. Mork, & G. O. Sørvik (Eds.), *På forskerfotter i naturfag*. Oslo: Universitetsforlaget.
- Mork, S. M., & Erlie, W. (2017). *Språk, tekst og kommunikasjon i naturfag* (Vol. 2). Oslo: Universitetsforlaget.
- National Research Council. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. Washington, D.C.: The National Academies Press Retrieved from <https://doi.org/10.17226/13165>

- Newby, P. T. (2014). *Research methods for education* (2nd ed. ed.). London: Routledge.
- Nordby, S. T. (2019). *Programmering og algoritmisk tenkning i fysikkundervisning*. NTNU, NOU 2020:2. (2020). *Fremtidige kompetansebehov III*. Retrieved from <https://www.regjeringen.no/contentassets/053481d65fb845be9a2b1674c35d6d14/nou/pdfs/nou202020200002000dddpdfs.pdf>
- Nygård, K. (2018). *Programmering i skolen*. Oslo: Pedlex.
- Pajchel, K., Ramton, A. M. T. S., & Sollid, P. Ø. D. (2019). Modeller og modellering i naturfag. In L. O. Voll, A. B. Øyehaug, & A. Holt (Eds.), *Dybdelæring i naturfag*. Oslo: Universitetsforlaget.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books, Inc.
- Postholm, M. B. (2005). Observasjon som redskap i kvalitativ forskning på praksis. *Norsk Pedagogisk Tidsskrift*, 89(2), 146-159. doi:10.18261/ISSN1504-2987-2005-02-07
- Rehmat, A. P., Ehsan, H., & Cardella, M. E. (2020). Instructional strategies to promote computational thinking for young learners. *Journal of Digital Learning in Teacher Education*, 36(1), 46-62. doi:10.1080/21532974.2019.1693942
- Repenning, A., Webb, D., & Ioannidou, A. (2010). *Scalable game design and the development of a checklist for getting computational thinking into public schools*. Paper presented at the Proceedings of the 41st ACM technical symposium on Computer science education, Milwaukee, Wisconsin, USA. <https://doi.org/10.1145/1734263.1734357>
- Rolfe, G. (2006). Validity, trustworthiness and rigour: quality and the idea of qualitative research. *Journal of Advanced Nursing*, 53(3), 304-310. doi:<https://doi.org/10.1111/j.1365-2648.2006.03727.x>
- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristiansen, T. E., . . . Voll, L. O. (2016). *Teknologi og programmering for alle*. Retrieved from <https://www.udir.no/globalassets/filer/tall-og-forskning/forskningsrapporter/teknologi-og-programmering-for-alle.pdf>
- Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). *"Creating Cool Stuff": Pupils' Experience of the BBC micro:bit*. Paper presented at the Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, Washington, USA. <https://doi.org/10.1145/3017680.3017749>
- Sevik, K. (2016). Programmering i skolen. Retrieved from https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. doi:10.1016/j.edurev.2017.09.003
- Skaperskolen. (2020a). Internett fanger, 8.-10. trinn Retrieved from <https://skaperskolen.no/undervisningsopplegg-8-10/internett-fanger/>
- Skaperskolen. (2020b). Planleggingsverktøyet «Boblemodellen». Retrieved from <https://skaperskolen.no/planleggingsverktoyet-boblemodellen/>
- Skaperskolen. (2020c). Støttestrukturer. Retrieved from <https://skaperskolen.no/pedagogisk-verktoykasse-gammel/stottestrukturer-2/>
- Sparf, M., Löfgren, H., & Kreitz-Sandberg, S. (2022). Design for learning programming - approaches taken by novice learners. *Nordic Studies in Science Education*, 18. doi:<https://doi.org/10.5617/nordina.8251>
- Store norske leksikon. (2009, 05.12.2019). programvare. Retrieved from <https://snl.no/programvare>
- Store norske leksikon. (2020). Hjemmenettverk. Retrieved from <https://snl.no/hjemmenettverk>

- Svensson, M. (2011). Tekniske system i grundskolan – kritiska aspekter som didaktisk möjlighet. *Nordic Studies in Science Education*, 7.
doi:<https://doi.org/10.5617/nordina.241>
- Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. C. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10(3), 251-296.
doi:10.1023/A:1022193728205
- Sørby, S. A., & Angell, C. (2012). Undergraduate students' challenges with computational modelling in physics. *Nordic Studies in Science Education*, 8(3), 283-296.
doi:<https://doi.org/10.5617/nordina.534>
- Sørvik, G. O. (2016). Å forberede en utforsking. In M. Ødegaard, B. Haug, S. M. Mork, & G. O. Sørvik (Eds.), *På forskerfötter i naturfag*. Oslo: Universitetsforlaget.
- Taub, R., Armoni, M., Bagno, E., & Ben-Ari, M. M. (2015). The effect of computer science on physics learning in a computational science environment. *Computers & Education*, 87, 10-23.
- Tyrén, M., Carlborg, N., Heath, C., & Eriksson, E. (2018). *Considerations and Technical Pitfalls for Teaching Computational Thinking with BBC micro:bit*. Paper presented at the Proceedings of the Conference on Creativity and Making in Education.
- Utdanningsdirektoratet. (2019a). Algoritmisk tenkning. Retrieved from <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2019b). Hva er kjerneelementer? Retrieved from <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hva-er-kjerneelementer/>
- Utdanningsdirektoratet. (2020a). *Læreplan i naturfag (NAT01-04)*. Oslo Retrieved from <https://www.udir.no/lk20/nat01-04?lang=nob>
- Utdanningsdirektoratet. (2020b). *Læreplanen for matematikk (MAT01-05)*. Retrieved from <https://www.udir.no/lk20/mat01-05?lang=nob>
- Voll, L. O. (2019). Kompetanse i naturfag og teknologi. In L. O. Voll, A. B. Øyehaug, & A. Holt (Eds.), *Dybdeløring i naturfag*. Oslo: Universitetsforlaget.
- Voll, L. O. (2022). Hva er teknologi? *Naturfag*, 4-7. Retrieved from www.naturfagsenteret.no/c1405589/binfil/download2.php?tid=2354834
- Voll, L. O., & Sollid, P. Ø. D. (2022). Teknologi i naturfag. *Naturfag*, 8-11. Retrieved from www.naturfagsenteret.no/c1405589/binfil/download2.php?tid=2354834
- Vygotsky, L. S. (1987). *The collected works of LS Vygotsky: The fundamentals of defectology* (Vol. 2): Springer Science & Business Media.
- Walford, G. (2001). *Doing qualitative educational research* London: Continuum.
- Waters, J. B. (2020). *Programmering og dybdeløring i naturfag*. (Master). Universitetet i Oslo, Retrieved from <https://www.mn.uio.no/fysikk/forskning/grupper/skolelab/masteroppgaver/waters-programmering-og-dybdeløring-i-fysikk-endig.pdf> DUO vitenarkiv database.
- Webb, N. M., & Mastergeorge, A. (2003). Promoting effective helping behavior in peer-directed groups. *International Journal of Educational Research*, 39(1), 73-97.
doi:[https://doi.org/10.1016/S0883-0355\(03\)00074-0](https://doi.org/10.1016/S0883-0355(03)00074-0)
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology volume*, 25, 127-147.
doi:10.1007/s10956-015-9581-5
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
Retrieved from <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Woolfolk, A. (2016). *Pedagogisk psykologi* (Vol. 6). Bergen: Fagbokforlaget.

Yin, R. K. (2018). *Case study research and applications : design and methods* (Sixth edition. ed.). Los Angeles, California: SAGE.

Øverby, H., & Dvergsdal, H. (2021). Internett. Retrieved from <https://snl.no/internett>

8. Vedlegg

Vedlegg 1 Godkjenning NSD

13.05.2022, 21:55 Meldeskjema for behandling av personopplysninger

[Meldeskjema](#) / [Masteroppgave programmering](#) / Vurdering

Vurdering

Referansenummer

503924

Prosjekttittel

Masteroppgave programmering

Behandlingsansvarlig institusjon

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

Prosjektperiode

16.08.2021 - 31.08.2022

[Meldeskjema](#) 

Dato

26.10.2021

Type

Standard

Kommentar

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 26.10.2021, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

DEL PROSJEKTET MED PROSJEKTANSVARLIG

For studenter er det obligatorisk å dele prosjektet med prosjektansvarlig (veileder). Del ved å trykke på knappen «Del prosjekt» i menylinjen øverst i meldeskjemaet. Prosjektansvarlig bes akseptere invitasjonen innen en uke. Om invitasjonen utløper, må han/hun inviteres på nytt.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 31.08.2022.

LOVLIG GRUNNLAG

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

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

LÆRERES TAUSHETSPLIKT

Lærere har taushetsplikt, og det er viktig at intervjuene gjennomføres slik at det ikke samles inn opplysninger som kan identifisere enkeltelever eller avsløre taushetsbelagt informasjon. Vi anbefaler at du er spesielt oppmerksom på at ikke bare navn, men også identifiserende bakgrunnsopplysninger må utelates, som for eksempel alder, kjønn, navn på skole, diagnoser og eventuelle spesielle hendelser. Vi forutsetter også at dere er forsiktig ved å bruke eksempler under intervjuene.

Studenten og læreren har et felles ansvar for det ikke kommer frem taushetsbelagte opplysninger under intervjuet. Vi anbefaler derfor at studenten minner læreren om taushetsplikten før intervjuet startet.

PERSONVERNPRINSIPPER

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

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

DE REGISTRERTES RETTIGHETER

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art.

17), begrensning (art. 18), og dataportabilitet (art. 20).

<https://meldeskjema.nsd.no/vurdering/61320371-8c86-4398-83b5-e7c1ab0f96c1> ½

13.05.2022, 21:55

17), begrensning (art. 18), og dataportabilitet (art. 20). Meldeskjema for behandling av personopplysninger

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

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

FØLG DIN INSTITUSJONS RETNINGSLINJER

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

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

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

MELD VESENTLIGE ENDRINGER

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

<https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-enderinger-i-meldeskjema>. Du må vente på svar fra NSD før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

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

Kontaktperson hos NSD: Silje Fjelberg Opsvik

Lykke til med prosjektet!

Vil du delta i forskningsprosjektet ”Algoritmisk tenkning og motivasjon i programmeringsaktivitet”?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke hva elever får jobbet med, refleksjoner underveis og i etterkant av undervisning som inkluderer programmering. I tillegg skal vi se på hva som påvirker elevenes motivasjon i undervisningen. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

I forbindelse med våre masteroppgaver ved OsloMet - storbyuniversitetet, ønsker vi din deltagelse i våre forskningsprosjekter. Vi tar vår master i naturfag og ønsker å studere algoritmisk tenkning og motivasjon ved programmering. Formålet med prosjektene er å undersøke hva som påvirker elevens motivasjon og hvordan undervisningen tilrettelegger for utvikling av algoritmisk tankegang.

Vi ønsker å studere hvilke strategier dere bruker, hvordan dere jobber når dere skal finne en løsning på oppgaven. I tillegg hva elever tenker er interessant, engasjerende og motiverende ved undervisningen. For å få innsikt i dette vil vi både observere og intervjuere elever som deltar i leksjonen.

I undervisningen skal dere arbeide både skriftlig og muntlig i grupper, og programmere MicroBit. Det naturfaglige temaet for undervisningen er Internett som system.

Hvem er ansvarlig for forskningsprosjektet?

OsloMet – storbyuniversitetet er ansvarlig for prosjektet. Prosjektansvarlig er Katarina Pajchel, førsteamanuensis ved OsloMet - storbyuniversitetet, og Siv Gundrosen Aalbergsjø, førsteamanuensis ved OsloMet - storbyuniversitetet.

Hvorfor får du spørsmål om å delta?

Vi kom i kontakt med naturfaglæreren deres da vi var på samme kurs. Her fikk vi en innføring i et undervisningsopplegg som vi ønsket å undersøke i forbindelse med våre masteroppgaver. Læreren deres underviser elever i riktig alder, samt var villig til å teste ut det

vi lærte på kurset i sin undervisning. Vi avtalte med læreren deres at vi kunne komme å se på gjennomføringen av dette.

Hva innebærer det for deg å delta?

Elever som velger å delta i prosjektet vil bli observert i undervisningen læreren din gjennomfører. Elever vi kun bli observert når vi er til stede, og det er når elever arbeider på stasjon 2. Som en del av undervisningen vil elever også levere inn noen refleksjonsnotater de skriver underveis i arbeidet. Dette ønsker vi å bruke som datamateriale i våre masteroppgaver. Det vil bli tatt opp lyd underveis i undervisningen. Lydopptakerne som brukes, plasseres slik at det kun er de elevene som har gitt samtykke i å delta, som blir med. Under observasjonene vil det også bli skrevet notater av det vi ser dere gjør, hvordan deres engasjement er, valg dere tar og løsningsmetoder. Vi vil bare gjøre notater av de som deltar i forskningen.

Du vil kunne bli spurt om å være med i et fokusgruppeintervju. Hvis du velger å delta på det, vil du bli med på en samtale sammen med noen andre elever. Her vil vi spørre om hva du/dere syntes om undervisningen, få høre mer om hvordan dere løste oppgaven(e) og tanker rundt deres motivasjon. Det vil bli tatt lydopptak av samtalen, slik at det vil være lettere for oss å huske hva som ble sagt. Alle lydopptakene vil bli slettet etter endte prosjekter, og det er kun anonymiserte utdrag fra opptakene som brukes i masteroppgavene.

Dersom noen foresatte ønsker mer informasjon om gjennomføringen, eventuelt ønsker innsyn i spørsmålene som stilles i intervjuet, kan dere få tilgang på dette ved å ta kontakt med oss på forhånd.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta, eller senere velger å trekke deg. Det som blir gjort i forskningsprosjektet, vil ikke påvirke ditt forhold til skolen/lærer. Det vil heller ikke påvirke karakterer.

Dersom du ikke ønsker å delta i prosjektet kan du likevel delta i undervisningen, og arbeidet ditt blir ikke samlet inn. Det vil bli sikret at de som ikke ønsker å delta ikke kommer med i lydopptak til forskningen.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

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

Våre masteroppgaver er en del av et større forskningsarbeid/-prosjekt, TRELIS, Referanse/prosjektnummer: 165347.

(Navn på meldeskjemaet: TRELIS – lærerstudenters og læreres oppfatninger og erfaringer med programmering og modellering – men TRELIS gar flere meldeskjemaer i NSD, en per arbeidspakke. Denne er for arbeidspakke 5).

Opplysninger om deg vil kun være tilgjengelig for masterstudenter, masterveiledere og TRELIS-prosjektet. Dette forskningsprosjektet deler vi datamaterialet med.

På refleksjonsnotatet skriver dere fornavn. Fornavnet vil vi erstatte med en kode som lagres på egen navneliste adskilt fra øvrig data. Dersom det kommer noen personopplysninger frem i lydopptaket eller i andre deler av studien, vil de bli slettet og inngår ikke i forskningen. Det som inngår i datamaterialet er lydopptak, refleksjonsnotater og observasjonsnotater. Dette krypteres og lagres på en harddisk som kun er tilgjengelig for de som skal ha tilgang.

Innsamlet elevarbeid lagres sikkert.

Deltakerne i prosjektet vil ikke kunne bli gjenkjent i publikasjon.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er i juni 2022. Opplysninger om deg og ditt arbeid vil bli lagret kryptert og sikkert underveis. Ved godkjente masteroppgaver, vil alt datamateriale lagres etter publisert forsknings standard. Det innebærer transkripsjoner av opptak, refleksjonsnotater og observasjonsnotater. TRELIS vil muligens bruke transkribert og anonymisert datamateriale videre i fremtidig arbeid.

Dine rettigheter

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

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

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra OsloMet - storbyuniversitetet har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?

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

- OsloMet – storbyuniversitetet ved Martin Løvdahl - student
 - o Mail: s198097@oslomet.no
- OsloMet - storbyuniversitetet ved Kristin Ausland – student
 - o Mail: s313589@oslomet.no
- OsloMet – storbyuniversitetet ved Katarina Pajchel - veileder
 - o Mail: katarina.pajchel@oslomet.no
- OsloMet - storbyuniversitetet ved Siv Gundrosen Aalbergsjø - veileder
- Vårt personvernombud: Ingrid S. Jacobsen
 - o Mail: personvernombud@oslomet.no

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

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

Med vennlig hilsen

Siv Gundrosen Aalbergsjø
Katarina Pajchel
(Veiledere)

Martin Løvdahl
Kristin Ausland
(Studenter)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «Algoritmisk tankegang og motivasjon tilknyttet programmering», og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i observasjon under undervisning
- å delta i fokusgruppeintervju
- å dele refleksjonsnotater som gjøres i undervisningen
- at masterstudentene kan innad i prosjektene dele relevante opplysninger om meg

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

(Signert av prosjektdeltaker, dato)

Intervjuguide (Samtale med elever)

Velkommen til denne uformelle samtalen om programmering i naturfag.

Som dere kanskje har fått med dere holder vi på med å **undersøke** elevers **algoritmiske tenkning og motivasjon for naturfag** ved arbeid med undervisningsopplegget *Internett fanger*. Målet er å finne ut hvordan programmering kan brukes på en god måte i naturfag, samt forbedre undervisningsopplegget dere har vært med på.

Deres deltakelse vil være til **stor hjelp for oss** i arbeidet med dette.

Det som skal skje nå er at vi kommer til å stille dere noen spørsmål om **hva dere gjorde og hvordan dere tenkte** da dere løste oppgaven med **Micro:Bit** i undervisningen dere deltok i. I tillegg vil jeg stille spørsmål om hva som var **interessant, engasjerende og spennende** ved opplegget. Og så kommer jeg til å spørre dere litt om hva dere tenker om naturfag generelt. Her er det **ingen rette eller gale svar**. Jeg er ute etter deres personlige meninger. Dere må gjerne **kommentere og diskutere hverandres svar**, da målet er å få dere til å prate mest mulig. Innimellom kommer jeg til å bryte inn med et spørsmål for å **passer på at vi dekker de temaene vi undersøker**. Men det finnes helt sikkert interessante tanker som jeg ikke har tenkt på at jeg kunne spørre om. Derfor er det veldig bra hvis dere drar diskusjonen litt utover konkrete svar på akkurat det jeg spør om. Bare prøv å pass på at det **kun er en person som snakker om gangen**. Ellers kan det bli vanskelig å høre gjennom lydopptaket etterpå.

Jeg vil nemlig ta **lydopptak av samtalen** deres. Opptaket skal brukes som et datagrunnlag i undersøkelsen. Det er bare forskergruppen som har tilgang til opptaket. Og dere vil ikke bli identifisert med navn eller kunne gjenkjennes i rapporter fra forskningen. Vi har **taushetsplikt** om hvem som har sagt hva.

Det er selvfølgelig helt **frivillig å delta** i samtalen. Dere kan **trekke dere når som helst** uten grunn. Da er det bare å forlate rommet stille og rolig.

Er det noe som er uklart, eller har dere noen spørsmål før vi begynner?

Er det noen som har lyst til å trekke seg nå før vi starter?

Den er god, da setter vi i gang.

Innledning:

Da tenkte jeg vi skulle ta en rask **navnerunde**. ... “Takk”.

Deltakernes tidligere erfaringer med og kunnskaper om programmering

- Har dere lært om programmering tidligere?

Hvis nei: Gå videre til neste punkt

Hvis ja:

- Har dette vært en del av undervisningen, eller har det vært et frivillig opplegg utenom skoletid?
- Driver dere med noe liknende på fritiden?
 - Gaming, app-utvikling
- Hadde dette noe å si for hvordan dere jobbet med oppgaven i timen?

- Hva synes dere om å jobbe med programmering?

- Liker --> Hvorfor/ hva er det som gjør det?
 - Mestring, lett, utfordrende, kreativitet
- Liker ikke --> Hvorfor/ hva er grunnen til det?
 - Er det *noen* deler som kan være gøy, andre deler kjedelig eller vanskelig?
 - *Jeg la merke til at... Virket som at dere synes dette... var spennende...*

- Hvordan vil dere forklare hva programmering er?

- Hva gjør dere når dere programmerer? Kan dere sette ord på dette?

Hvis eleven kun svarer: «Jeg henter frem blokkene i programmet» osv..

- Hvordan vet du hvilke blokker du skal bruke?
- Forklare begreper innen koding. Variabel, løkke, ...
- ..

- Har dere eksempler på hva dere har brukt programmering til på skolen?

Hvis nei:

- Hva kunne det ha blitt brukt til?
- Har dere hørt om noen andre klasser, skoler, venner har brukt programmering på skolen?

Hvis ja:

- Hadde det vært morsomt å bruke det i andre fag/andre sammenhenger?
 - Hvorfor ville det vært gøy?

- Hvorfor tenker dere det er lurt?
- **Kan dere komme på eksempler på noe dere bruker i hverdagen som er programmert?**

(Kan komme med eksempel)

 - Har noen av dere Vipps?
 - Apper (Vipps), nettsider (VG), spill, relevans for yrkeslivet
 - Hva hjelper disse eksemplene oss til?
 - *For at disse skal fungere, så må noen programmere dem.*
- **Vet dere om noen yrker hvor en bruker programmering?**
 - Hvordan brukes programmeringen, hvordan er det til hjelp for jobben?
 - *Skal gjøre hverdagen/jobben enklere.*
 - Hvor har dere lært om dette? På skolen, foreldre, TV, Internett?
 - Er dette noe dere kunne tenkt dere å jobbe med?

Kort om intervjuobjektets forventinger

- **Hva har dere lært om i timen?**
 - Hvis de svarer hva de har gjort, så kan vi spørre: Hva var hensikten med å gjøre dette (programmere)?
 - Få frem Internett som system, Micro:Bit, hvordan enheter kan kobles til en trådløs ruter. Sender/mottaker.
 - Hvordan hjelper modellen/programmeringsaktiviteten til å vise hvordan hjemmeruteren fungerer?
 - Powerpoint m/instruksjoner
 - *«Da dere jobbet med ..., la jeg merke til at... (observasjonseksempel). Kan dere snakke litt om hva dere tenkte/gjorde her?».*
 - ...
 - ...

Hoveddel:

Deltakernes tanker om selve undervisningen

- (Merge de to neste punktene?)

- **Vil noen av dere gjenfortelle hva dere har gjort?**

Hvordan gikk dere frem da dere løste oppgaven?)

Be elevparene om å forklare/beskrive “stegene” de gjorde da de jobbet med micro-biten. Først be par 1, så få par 1 til å forklare det på sin måte.

(La dem bruke/forklare ut ifra modellen de har laget. Oppskriftshefte/bilde)

- Forklar prosessen, hvordan dere tenkte?
 - Kan dere forklare hvorfor dere har valgt å ta med akkurat disse delene?
Tegne det slik, utforming av modell?
 - På hvilke måter er disse micro:bitene en modell for hjemmerouteren?
 - Brukte dere noen spesielle løsningsmetoder?
 - *Hint: Steg for steg, bryte ned, samarbeid, prøv og feile, generalisering (Den algoritmiske tenkeren)...*
 - *Vise til konkret eksempel fra observasjon*
 - *Hvor, hvordan har dere lært dere disse?*
 - **Kodene i oppskriften**
 - **Hva gjør de ulike “stegene”?**
 - **Hvorfor fungerer de?**
 - **Hvilke andre måter kunne man satt opp “oppskriften”?**
 - Hva gjorde dere da dere stod fast i en oppgave?
 - *Arbeids- og tankemåter*
 - *Se på andre (modellering), starte på nytt, feilsøke...*
 - Hva gjorde dere som hjalp dere til å forstå problemet?
 - *Hvordan hjalp dette dere videre i oppgaven?*
- **Var det noe dere prøvde som ikke fungerte?**

Komme med eksempler fra observasjon

 - *Hvordan fant du ut at det ikke kom til å lede deg til en løsning?*
 - *Hvordan forstod du at du måtte gjøre noe annerledes?*
 - *Hvordan opplevde dere å stå fast/at det ikke ble riktig med en gang?*
 - **Kan noe dere gjorde i denne oppgaven hjelpe dere til å løse andre oppgaver?**
 - *Dere sa jo tidligere at... Bryte ned i mindre biter... Er ikke det overførbart? Brukes i andre oppgaver?*

- Micro:bit til andre ting i naturfag?
- *Jeg la merke til at noen gjorde dette... Er denne måten å tenke på nyttig for andre problemer?*

- Hvordan opplevde dere å gjennomføre denne aktiviteten?

(Mulig dette kommer med under hva gjort i timen!)

- Var det noe ved oppgaven som dere opplevde som spesielt spennende, interessant, motiverende eller engasjerende?
 - Hva var det som gjorde det mer spennende, interessant eller motiverende?
 - Hvilke deler/ har dere noen eksempler på hvorfor?
 - Hva er grunnen til at dere synes dette?
 - *Jobbe på PC/iPaden og få se resultatet på micro:biten.*
(Software/Hardware)
 - *Jeg la merke til at dere*
 - *Koble til hendelser/observasjoner*
 - «Da dere jobbet med..., la jeg merke til at... Kan dere snakke litt om hva dere tenkte/gjorde her?».
- Var det deler ved oppgaven som var mer utfordrende/ ikke like interessant?
 - Hvilke(n) og hvorfor?
- Sammenliknet med annen naturfagundervisning – hvordan var dette?
 - Liker dere/ liker dere ikke naturfag? Hvorfor?
- Hvordan syntes dere det var å jobbe i par/grupper?
 - Skulle dere helst jobbet alene?
 - Hvordan ville det ha vært annerledes i forhold til gruppearbeid?

Deltakernes refleksjon omkring læringsutbytte/ videre arbeid/ problemløsning

- **Tenkte dere at dere kom til å få til oppgaven med en gang dere fikk oppgaven?**
 - Hvorfor/ Hvorfor ikke?
 - Var det deler av oppgaven dere var kjent med fra før?
 - *Dere sa jo tidligere at dette.... var utfordrende. Var dette noe dere hadde sett for dere?*

- Hva fikk dere til å holde på med oppgaven, fortsette...(selv om du ikke nødvendigvis fikk det til med en gang)?
 - Støttestrukturer (Medelever, lærere, nettsider, kravspesifikasjon...)
 - Motivasjon (Indre, ytre)

- **Etter opplegget, har dere lyst til å jobbe mer med liknende oppgaver?**
 - Interesse eller for karakterens skyld?
 - Vil dere lære mer om programmering?
 - Hva slags tanker har dere omkring Internett/Hjemmeruter nå?
 - Hva vil dere da jobbe mer med?
 - *Koble til det de tidligere har snakket om. Snakket i stad om andre måter å bruke programmering på i naturfag...*

- **Kan dere si noe om hva dere har lært?**
 - Kan dere nevne et eksempel på hva dere har lært?
 - Om programmering.
 - Problemløsning.
 - Naturfag/teknologi. Modell – en måte å beskrive et fenomen
 - Hva betyr modell?
 - Hvordan får modellen frem at dette er et system?
 - Hva forstår du bedre nå enn før timen?
 - Programmering.
 - Naturfag.
 - Hvorfor er dette noe du husker/ nevner dette eksemplet?

Avslutning:

- Til slutt: Er det noe dere tenker at jeg burde spørre om, eller noe mer dere vil dele?

Avhenger av elevenes eventuelle innspill. Hvis elevene ikke har noe å tilhøye:

- Tusen takk for at dere ble med.

Vedlegg 4 Shutes fasetter

Tabell 11: Shutes fasetter min oversettelse (Shute et al., 2017, p. 151)

Fasett	Beskrivelse av fasett
Dekomposisjon	Dissekere et komplekst problem/system inn i håndterbare deler. Disse er ikke tilfeldige deler, men funksjonelle elementer som sammen omfatter hele systemet, problemet.
Abstraksjon	<p>Dra ut essensen av et (komplekst) system. Denne komponenten har tre underkategorier:</p> <ul style="list-style-type: none"> a) Datainnsamling og analyse: Samle inn den mest relevante og viktige informasjonen fra flere kilder og forstå forholdene blant datamaterialets ulike «lag». b) Oppdage mønstre: Identifisere mønstre/ regler som inngår i datamaterialet/ informasjonsstrukturen c) Modellere: Bygge modeller eller simuleringer for å representere hvordan systemet fungerer, og eller hvordan et system vil fungere i fremtiden.
Algoritmer	<p>Design logiske instruksjoner, satt i riktig rekkefølge, for å gjengi løsningen til problemet. Instruksjonene kan bli utført av et menneske eller en datamaskin. Denne kategorien har fire underkategorier:</p> <ul style="list-style-type: none"> a) <i>Algoritmisk design:</i> Lage en serie av steg, som har en bestemt rekkefølge, for å løse et problem. b) <i>Parallellisme:</i> Utføre et gitt antall steg på samme tid c) <i>Efficiency:</i> Designe færrest antall steg for å løse et problem, fjerne overflødige og unødvendige steg d) Automasjon: Gjøre «execution»/kjøringen av prosedyren automatisk når det er nødvendig å løse liknende problemer <ul style="list-style-type: none"> a. loop
Feilsøking (debugging)	Oppdage og identifisere feil, og videre fikse feilene, når en løsning ikke fungerer som den skulle.
Iterasjon	Gjenta designprosessen for å finjustere løsningen, helt til det ideelle resultatet er oppnådd.
Generalisering	Overføre algoritmisk tenkning-kompetanser i et bredt spekter av situasjoner/områder for å løse problemer effektivt/optimalt.

Vedlegg 5 Resultat tabeller

Tabell 12 resultater forskningsspørsmål 1:

Temaer	Undertemaer	Funn
Teknologi	Begrepsbruk	Kjenner til begreper tilknyttet internett, slik som ruter, sendere, lagring og base.
		Kan forklare hjemmeruterens vesentligste funksjoner.
		Bruker et hverdagslig språk, og varierer mellom elevene hvor utfyllende de uttrykker seg.
	Modeller	Klarer å forklare sammenhenger mellom micro:bit og hjemmeruterens, men kan se ut til å være litt utfordrende for noen av elevene.
		Noen av elevene ser ut til å bli mer sikre på det hjemmeruterens funksjon ved å arbeide med oppgaven (programmere micro:bit)
		Kjenner til og kan beskrive hva modeller er, og hva modeller kan brukes til. Mange kjenner til atommodell.
Programmring	Forklaring	Elevene klarer å forklare begrepet programmering.
	Relevans	Kjente til eksempler på hva det brukes til. Programmerte «produkter» de kjente til: spill, nettsider, apper, kalkulator, roboter.
		Ser ut til at et fåtall av elevene så «personlig relevansen» av programmering, utenom at det kunne være nyttig i fremtidig jobb.
	Koding	Elevene får arbeidet med sentrale programmeringskonsepter. <i>Input, output, software og hardware.</i> Noen av elevene klarer å gi en forklaring på og bruke løkker. Samtlige elever klarer å bruke betingelser.
		Radio-gruppen får elevene arbeidet mye med. Noen trenger mer veiledning for å forstå hvordan denne fungerer.
		Elevene kjenner til ulike programmeringsspråk, syntaks- regler. Noen elever klarer å sammenlikne tekst- og blokkprogrammering.
Problemløsning	Feilsøking/ debugging	Dette får samtlige elever arbeidet mye med. Mye grunnet diverse tekniske feil som oppstår, men også en del i program-koden. Syntaks-feil oppstod hos flere av elevene.
		Ulikt hvor systematisk elevene feilsøker. Prøve-og-feile-tilnærming. Noen starter helt på nytt, andre går gjennom programmet systematisk, prøver å forutse resultatet, endrer deler av koder hvor de tror feilen ligger.
		Forteller at de vet at man kan kjøre programmet samtidig som man følger koden, men at dette er litt tungvint med micro:bit (må overføre programmet etter hver gang man justerer koden)
	Mønstre	Elevene tilnærmer seg programmeringen ulikt, ved at noen koder mer effektivt (gjenkjenner og gjenbraker hele eller deler av programmet, abstraksjon)

Tabell 13 resultater forskningsspørsmål 2

Temaer	Undertema	Funn fra datamaterialet
Utfordringer	Undervisningsmaterieell	Tekniske feil oppstod underveis, noe som gjorde at elevene brukte tid på å få til å bruke utstyret.
		Mange av elevene hadde utfordringer med å forstå instruksjonene. De trengte hjelp, spesielt de delene som var mer åpne, at elevene måtte bruke forkunnskapene til å selv vite hvordan programmet skulle settes sammen.
		Oppgaven var nokså «lukket», altså det var mange føringer for hva elevene skulle programmere. Derfor var de avhengige av å forstå hva de skulle gjøre.
	Elevenes forkunnskaper	Mange av elevene trengte hjelp og fortalte at det oppstod mye usikkerhet tilknyttet utstyret og programmeringsspråket.
		Forarbeidets innhold og hvordan dette ble brukt i undervisning varierte mellom klassene. Innhold: Teknologisk system, programmerings-ferdigheter
Arbeidsmåter	Samarbeid	Dette var en måte mange hadde nytte av, men hvor godt det fungerte var avhengig av elevenes kompetanse. Dele kunnskap, forklare og hjelpe hverandre. Noen elever fungerte som en ressurs også for andre elevpar.
Rammebetingelser	Tid	Undervisningen varte relativt kort, 60 og 90 minutter.
	Lærerens forkunnskaper	De to lærerne hadde ulik mengde erfaring, og kunne derfor i ulik grad veilede elevene i arbeidet. Måten undervisningen ble lagt opp var da også ulikt. Bruk av støttestrukturer.