



Høgskulen
på Vestlandet

MASTEROPPGAVE

En lærers erfaringer med bruk av ressurser fra det nasjonale Super: bit-prosjektet i matematikkundervisningen

A teacher's experiences with using the resources from the
national Super: bit project in teaching mathematics

Emilie Larsen

Master i matematikdidaktikk (MGUMA550)

Fakultet for lærerutdanning, kultur og idrett (FLKI)

Veiledere: Frode Olav Haara og Lene Hayden Taraldsen

16.mai 2022

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

Forord

Innleveringen av denne masteroppgaven markerer avslutningen på fem år som lærerstudent på Høgskulen på Vestlandet i Sogndal. Studietiden har vært en fin tid, hvor jeg har møtt mange flotte mennesker, fått nye opplevelser og lært mye. Det er ikke til å legge skjul på at arbeidet med denne oppgaven har vært en berg og dalbane av oppturer og nedturer. Heldigvis har jeg hatt medstudentene på biblioteket, som har bidratt til at lunsj klokka 12 har blitt et høydepunkt å se frem til på tunge arbeidsdager. Det er godt å være i mål etter utallige timer, og jeg ser nå frem til å ta fatt på arbeidslivet.

Mine to veiledere, Frode Olav Haara og Lene Hayden Taraldsen, fortjener en stor takk for å ha støttet meg gjennom denne prosessen. Uten deres engasjement og konstruktive tilbakemeldinger ville ikke oppgaven vært det den er i dag.

Emilie Larsen

Sogndal, mai 2022

Sammendrag

Høsten 2020 ble den nye læreplanen, Kunnskapsløftet (LK20), tatt i bruk i norsk skole. For matematikkfaget medførte det blant annet at programmering ble innført som en del av faget. I forbindelse med den økte satsingen på programmering i skolen, gikk Lær Kidsa Koding, Vitensenterforeningen og NRK i 2019 sammen om et felles prosjekt med navnet Super:bit. Formålet med prosjektet var å skape engasjement, og øke elevers og læreres kompetanse innen programmering. I dette masterprosjektet ønsker jeg å undersøke en lærers erfaringer med bruk av ressurser fra dette prosjektet. Problemstillingen er:

Hvilke erfaringer gjør en lærer på en fådelt skole seg med bruk av ressurser fra Super:bit-prosjektet i matematikkundervisningen på mellomtrinnet?

For å besvare problemstillingen har jeg benyttet meg av et forskningsdesign som er inspirert av forskningsmetodikken Design-based reseach (DBR). I studiet har det blitt samlet inn kvalitative data gjennom observasjon og intervju av en lærer. Denne læreren har jeg hatt et tett samarbeid med i hele perioden, og vi har sammen utviklet et undervisningsopplegg med utgangspunkt i ressurser fra Super:bit. Opplegget har blitt gjennomført i klassen til den nevnte læreren, som arbeider på mellomtrinnet på en fådelt skole.

Funn fra studiet viser at læreren er svært positiv til Super:bit-prosjektet. Prosjektet har lagt til rette for at det er enklere å komme i gang med programmeringsundervisningen, og elevene på den fådelte skolen oppleves som motiverte. Læreren beskriver prosjektet som «*en unik mulighet for alle skoler*», og trekker frem den utleverte utstyrs pakken og de andre tilgjengelige ressursene på nett. Samtidig erfarer læreren at et slikt støtteverktøy ikke er nok. Etter hennes oppfatning er det behov for ytterligere kompetanseheving for lærere.

I undervisningsperioden har læreren benyttet seg av en utforskende tilnærming til læring. I denne oppgaven har teori om Inquiry-based learning (IBL) blitt brukt for å identifisere kjennetegn ved dette i undervisningen. Undervisningen har foregått på en fådelt skole, med en aldersblandet elevgruppe, hvor en organisering der elevene arbeider i læringspar ser ut til å fungere godt. I tillegg viser funn fra studiet at elevene tar i bruk flere algoritmiske arbeidsmåter i undervisningen. Arbeidsmåtene feilsøke, fikle, holde ut og samarbeide har vært fremtredende. Erfaringer fra undervisningsperioden kan tyde på at programmering egner seg godt for å arbeide med algoritmisk arbeidsmåter.

Abstract

The new curriculum, «Kunnskapsløftet» (LK20), entered the Norwegian schools from the autumn of 2020. With the new curriculum programming was included in the subject of mathematics. In 2019, with the increased focus on programming in school, Lær Kidsa Koding, Vitensenterforeningen and NRK began to work together on a project named Super:bit. The aim of the project was to create excitement among students and to give students and teachers experience with programming. The purpose of this master thesis is to investigate a teacher's experiences with using the resources from this project. The research question is:

What experiences does a teacher acquire in a mixed-aged school with using the resources from the Super:bit project in teaching mathematics for students in 5-7th grade?

To answer the research question, I have used a research design that is inspired by the research methodology Design-based research (DBR). In this study, qualitative data has been collected through observation and interview by a teacher. I have worked closely with this teacher throughout the period, and we have together developed and implemented a teaching plan based on resources from Super:bit.

Findings from this study shows that the teacher has a positive attitude towards the Super:bit project. The project has made it easier to get started with teaching programming, and the students at the mixed-aged school seems to be motivated. The teacher describes the project as «*a unique opportunity for all schools*», and highlights the handed out equipment package and the other resources available online. At the same time, the teacher experiences that such a support tool is not enough. In her opinion, there is a need for further competence development for teachers.

During the teaching period, the teacher used an exploratory approach to learning. In this thesis, Inquiry-based learning theory (IBL) has been used to identify the characteristics of this in the teaching. The teaching took place with a mixed-age group of students, where organizing the students to work in learning pairs seems to work well. In addition, findings from the study show that the students use several computational thinking approaches. The approaches debugging, persevering and collaborating have been prominent. Experiences from the teaching period may indicate that programming is well suited for working with computational thinking approaches.

Innhold

1	Innledning	1
1.1	Bakgrunn	1
1.2	Valg av tema og problemstilling	2
1.3	Oppgavens oppbygning	3
2	Teoretisk rammeverk	4
2.1	Programmering	4
2.1.1	Begrepet programmering.....	4
2.1.2	Programmering i matematikkfaget.....	5
2.2	Algoritmisk tenkning.....	7
2.2.1	Sammenhengen mellom algoritmisk tenkning og programmering	10
2.3	Super:bit-prosjektet	14
2.3.1	De ulike aktørene i prosjektet.....	15
2.3.2	Utstyrspakken.....	18
2.3.3	Programmeringsspråk.....	20
2.4	Sosiokulturelt perspektiv på læring	22
2.4.1	Læringspartner	23
2.5	Inquiry-based learning	24
2.5.1	Kjennetegn ved IBL	25
2.6	Fådelt skole.....	31
2.7	Oppsummering av teoretisk rammeverk.....	32
3	Metode	34
3.1	Kvalitativ metode	34
3.2	Design-based research	34
3.3	Informant og forskerrolle.....	36
3.3.1	Utvelgelse av informant og lærerens rolle	36
3.3.2	Forskerrollen	38

3.4	Datainnsamlingsverktøy	38
3.4.1	Intervju	39
3.4.2	Observasjon	40
3.4.3	Kombinasjon av observasjon og intervju	42
3.5	Planlegging, gjennomføring, evaluering og justering av undervisningsopplegg	42
3.5.1	Planleggingsmøte	42
3.5.2	Økt 1	44
3.5.3	Økt 2	46
3.5.4	Økt 3	48
3.6	Analyse av data	49
3.6.1	Fase 1	50
3.6.2	Fase 2	50
3.6.3	Fase 3	52
3.6.4	Fase 4	53
3.6.5	Fase 5	55
3.7	Reliabilitet og validitet	56
3.8	Etikk	58
4	Presentasjon av funn	59
4.1	Super:bit og koblingen mot matematikkfaget	59
4.2	Algoritmiske arbeidsmåter	62
4.3	Lærerrollen	63
4.4	Organisering av undervisning	64
4.5	Oppsummering av funn	65
5	Drøfting	66
5.1	Integrering av programmering i matematikkfaget	66
5.2	Algoritmiske arbeidsmåter i undervisningen	68
5.3	Veiledende lærerrolle i undervisningen	70

5.4	Organisering av aldersblandet elevgruppe	71
6	Avslutning.....	74
6.1	Konklusjon.....	74
6.2	Refleksjoner og veien videre	75
	Litteraturliste	76
	Vedlegg	82
	Vedlegg 1: Intervjuguide 1	82
	Vedlegg 2: Intervjuguide 2	84
	Vedlegg 3: Godkjenningbrev fra NSD.....	86
	Vedlegg 4: Informasjonsskriv og samtykkeskjema.....	88

TABELLISTE

Tabell 1: Kjerneferdigheter i algoritmisk tenkning (Bocconi et al., 2016; Torkildsen & Gjøvik, 2019).....	12
Tabell 2: Tilnærminger i algoritmisk tenkning (Bocconi et al., 2016)	13
Tabell 3: Dimensjoner i algoritmisk tenkning (Bocconi et al., 2016).....	14
Tabell 4: Plan for økt 1.....	45
Tabell 5: Plan for økt 2.....	47
Tabell 6: Plan for økt 3.....	49
Tabell 7: Samlet oversikt over kodene som ble brukt i analysen.....	51
Tabell 8: Tabell for å systematisere datamaterialet i analysen.....	54
Tabell 9: Utdrag fra hvordan datamateriale ble systematisert i analysen.....	55

FIGURLISTE

Figur 1: Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019)	9
Figur 2: Utklipp av oppgavesiden (Lær Kidsa Koding, u.å.-a).....	16
Figur 3: Micro:bit med markerte funksjoner (Micro:bit Educational Foundation, u.å.)	19
Figur 4: BitBot XL med markerte funksjoner (Vitensenterforeningen, u.å.-b)	20
Figur 5: Smartby-matta (Vitensenterforeningen, u.å.-a)	20
Figur 6: Eksempel på blokkprogrammering i MakeCode	21
Figur 7: Kjennetegnet ved IBL (Abril et al., 2013, s. 8).....	26
Figur 8: Arbeidsmåter i IBL (Primas, 2011, s. 11)	27
Figur 9: Forskningsforløpet i prosjektet, basert på DBR.	36
Figur 10: Spesifiseringer av koden lærerrollen	52
Figur 11: Oversikt over kategoriene i analysen	53
Figur 12: Eksempel på oppgaver i «Tegne etter instruksjoner fra læringspartner», økt 1	60
Figur 13: Eksempel på oppgave i «Utforske geometriske figurer med programmering», økt 2	61

1 Innledning

1.1 Bakgrunn

Ett av skolens formål er å forberede barn og unge på framtidens samfunns- og arbeidsliv. I opplæringsloven står det at «*Opplæringa i skole og lærebedrift skal, i samarbeid og forståing med heimen, opne dører mot verda og framtida og gi elevane og lærlingane historisk og kulturell innsikt og forankring*» og at «*Elevane og lærlingane skal utvikle kunnskap, dugleik og holdningar for å kunne meistre liva sine og for å kunne delta i arbeid og fellesskap i samfunnet*» (Opplæringslova, 1998, § 1-1). I løpet av de siste tiårene har det skjedd store endringer i måten vi lever på, og hvilke kompetanser som er nødvendige i samfunnet. Med bakgrunn i dette er det nødvendig å vurdere og gjøre endringer i skolens innhold og praksis, slik at elevene får et relevant innhold som gjenspeiler den kompetansen som er sentral for framtiden (Kunnskapsdepartementet, 2017). I lys av dette trekkes teknologi og digitale ferdigheter fram med behov for kompetanseheving (NOU 2014: 7). Som en konsekvens av det økende behovet for teknologi og digitale ferdigheter presenterte regjeringen i 2017 en digitaliseringsstrategi, med det overordnede målet om å gi norske skoleelever den kompetansen innen teknologi som er nødvendig i dagens og framtidens samfunn. Et av tiltakene som ble lagt fram var å innføre teknologi og programmering i skolens læreplaner (Kunnskapsdepartementet, 2017).

Programmering er ingen ny ide i skolen. Allerede på slutten av 1960-tallet ble programmeringsspråket LOGO utviklet med tanke på bruk i utdanning. Situasjonen er imidlertid annerledes i dag, ved at programmering har kommet inn i læreplanen med innføringen av Kunnskapsløftet 2020 (LK20), programmeringsverktøy er lett tilgjengelig og vi står midt i en samfunnsutvikling hvor vi hele tiden må forholde oss til digitale produkter (Nygård, 2018). Fagene kunst og håndverk, musikk, naturfag og matematikk har alle fått kompetansemål som omhandler utvikling av ferdigheter innenfor programmering i LK20 (Utdanningsdirektoratet, 2020).

Utdanningsdirektoratet har lagt til rette for oppstarten med programmering for skoler i hele landet gjennom satsingen «Den teknologiske skolesekken» (Utdanningsdirektoratet, 2019b). Som en del av denne satsingen gikk Lær Kidsa Koding, Vitensenterforeningen og NRK i 2019 sammen om et felles prosjekt for å lære barn og unge programmering. Satsingsprosjektet fikk navnet «Super:bit». Planen for prosjektet var at disse tre aktørene de neste tre årene skulle gi barn og unge programmeringsopplæring, utfordringer, teknologiforståelse og glede

ved å arbeide med programmeringsverktøyet micro:bit. Målet var å skape engasjement landet rundt, fremme programmeringskompetanse for elever og lærere på mellomtrinnet, og ikke minst gi like muligheter for elever på alle landets 2400 grunnskoler (Moldøyen, 2019).

1.2 Valg av tema og problemstilling

I en praksisperiode i løpet av min grunnskolelærerutdanning deltok jeg i matematikkundervisning i programmering hvor ressurser fra Super:bit-prosjektet ble brukt aktivt. Dette vekket en interesse hos meg, både for selve programmeringsaktiviteten, og for hvordan det kan brukes i matematikkfaget. Jeg ønsket å tilegne meg mer kunnskap om dette, ved å undersøke hvordan en lærer bruker ressurser fra prosjektet i matematikkundervisningen. Begrepet ressurs forstås som undervisningsopplegg, oppgaver og annet tilgjengelig materiale som ble utviklet i forbindelse med Super:bit-prosjektet.

I oppstarten av dette masterprosjektet undersøkte jeg om andre hadde gjort noe lignende tidligere. Jeg fant to masteroppgaver, en som omhandler Super:bits overføringsverdi til matematikk (Parken & Jakobsen, 2021), og en som omhandler læreres vurdering av hvordan materiellet og innsikten fra Super:bit skole kan være nyttig i egen undervisning (Aas, 2020). Denne masteroppgaven skiller seg fra de to nevnte masteroppgavene ved at jeg intervjuer og observerer en lærer over lengre tid. Ved å være deltakende i undervisningen ønsker jeg å få dypere innsikt i lærerens erfaringer med Super:bit. Gjennom å se lærerens bruk av ressursene i praksis kan en tilegne seg kunnskap om hvordan læreren bruker sin kunnskap og materiellet i en konkret undervisningssituasjon.

I samarbeid med læreren, som arbeider på en fådelt skole, har jeg vært med å planlegge, gjennomføre, evaluere og justere et undervisningsopplegg med utgangspunkt i det nevnte materialet. Opplegget har blitt gjennomført i en klasse på mellomtrinnet. Grunnet at prosjektet foregikk på en fådelt skole, og dermed med de mulighetene og utfordringene som en aldersblandet elevgruppe gir, vil fordeler og ulemper med dette også bli belyst.

Med bakgrunn i dette lyder problemstillingen for denne masteroppgaven slik:

Hvilke erfaringer gjør en lærer på en fådelt skole seg med bruk av ressurser fra Super:bit-prosjektet i matematikkundervisningen på mellomtrinnet?

Begrepet programmering vil forstås ut fra definisjonene til Nygård (2018) og Sevik (2016). I deres definisjoner er det ikke bare instruksjonene som blir gitt til en datamaskin eller lignende som legges i begrepet, læringen som skjer i prosessen frem til resultatet er like sentralt. Denne

avgjørelsen har blitt tatt på bakgrunn av at det er denne forståelsen som er mest brukt i skolen (Sevik, 2016). Jeg vil bruke begrepet programmering, og ikke koding, da det er dette begrepet som står nevnt i læreplanen (Utdanningsdirektoratet, 2020).

1.3 Oppgavens oppbygning

I kapittel 2 vil det teoretiske rammeverket bli presentert. Videre, i kapittel 3, vil de metodiske valgene som har blitt tatt i dette prosjektet gjøres rede for. Metodedelen inneholder også en beskrivelse av utviklingen av et analytisk rammeverk. Dette rammeverket har blitt brukt for å komme frem til funnene som blir presentert i kapittel 4. Funnene blir drøftet opp mot det teoretiske rammeverket i kapittel 5, før det hele oppsummeres i oppgavens avslutning. Dette gir grunnlag for å besvare oppgavens problemstilling.

2 Teoretisk rammeverk

Dette kapittelet starter med en redegjørelse for hva som ligger i begrepet programmering og hvordan programmering synliggjøres i læreplanen. Videre vil algoritmisk tenkning, Super:bit-prosjektet, sosiokulturelt perspektiv på læring, Inquiry-based learning og fådelt skole bli presentert. Dette vil sammen med tidligere forskning om bruk av programmering i matematikkundervisningen utgjøre det teoretiske rammeverket for denne masteroppgaven. Avslutningsvis (avsnitt 2.7) vil jeg oppsummere kapittelet, og peke fremover mot utviklingen av et analytisk rammeverk for prosjektet.

2.1 Programmering

2.1.1 Begrepet programmering

Begrepet programmering har tradisjonelt blitt brukt for aktiviteter der man skriver programkoder, altså instruksjoner som fører til at en datamaskin eller andre digitale enheter utfører en oppgave. Senere har begrepet blitt utvidet, og selve prosessen med å komme frem til programkoden blir inkludert i begrepet (Sevik, 2016). Denne forståelsen blir støttet i Nygård (2018) sin definisjon:

Programmering handler om å lage et sett med regler og uttrykk for å styre digitale enheter. I dette inngår prosessen fra å identifisere problemer og utforme mulige løsninger, til å lage kode som kan forstås av en datamaskin, systematisk feilsøke og forbedre denne koden, og dokumentere løsningen på en forståelig måte (Nygård, 2018, s. 7).

Nygård (2018) problematiserer at begrepet programmering har skapt mye forvirring og uenighet i skolen, da spesielt i skillet mellom programmering, koding og algoritmisk tenkning. Denne problematikken trekker også Kaufmann et al. (2018) fram. De beskriver at det i enkelte tilfeller kan være et skille i måten begrepene brukes. Begrepet koding forbindes gjerne med å lage mindre programmer med begrenset kompleksitet og konsekvens, mens programmering på sin side er mer omfattende og et eget fag. Likevel argumenteres det for at det ikke er noe klart definert skille mellom begrepene «programmering» og «koding», og at de i dagligtale brukes om hverandre (Kaufmann et al., 2018). I læreplanen er det, som nevnt i innledningen, begrepet programmering som brukes. På bakgrunn av dette vil jeg benytte begrepet programmering videre i denne oppgaven.

2.1.2 Programmering i matematikkfaget

Ved innføringen av LK20 ble programmering en del av matematikkfaget. Argumenter for dette er blant annet at programmering kan utvikle elevers logiske tankegang og evne til problemløsning (Kaufmann et al., 2018). Det er likevel stor uenighet om programmerings plass i matematikkfaget, både i Norge og internasjonalt, men på en annen side stor enighet om at programmering bør inn i skolen (Kaufmann et al., 2018). Ser en på praksisen i Europa, er det svært ulikt hvilke løsninger land har valgt (Balanskat & Engelhardt, 2015). Enkelte land argumenterer for at programmering henger tett sammen med det matematikkfaglige, og som følge av det har en naturlig plass innenfor dette fagområdet (Bueie, 2019). Andre land har valgt å innføre programmering som eget fag eller et valgfag (Balanskat & Engelhardt, 2015).

Sevik (2016) presenterte på vegne av Senter for IKT i utdanningen tre ulike måter programmering kunne innlemmes på i skolen. Disse tre alternativene var at programmering skulle bli en del av et eget IKT-fag, integreres i eksisterende fag eller som fagovergripende kompetanse i flere fag. Sevik (2016) kom med anbefaling om at Norge burde integrere programmering slik det var gjort i blant annet England, hvor programmering hadde blitt et eget IKT-fag. Dette skjedde imidlertid ikke, og beslutningen ble at programmering ble en del av eksisterende fag, deriblant matematikkfaget, slik som i eksempelvis Sverige og Finland (Kaufmann et al., 2018).

Et av hovedargumentene mot innføringen av programmering i matematikkfaget var at programmering ikke er en undergren av matematikk, men en egen fagdisiplin som kalles for informatikk. Argumentene for innføringen ble begrunnet i at programmering brukes mye i matematiske fag i dag. Utdanningsdirektoratet hevder at et viktig fellestrekk mellom programmering og matematikk er algoritmisk tenkning. Ved å se matematikk og programmering i forenlighet argumenter de for at elever vil kunne utvikle evne til algoritmisk tenkning (Utdanningsdirektoratet, 2017).

Nytt i LK20 var innføringen av kjerneelementer i de ulike fagene. Dette er elementer som blir sett på som sentrale i faget, og skal utgjøre en retning eller ramme for hva læreplanene skal inneholde. I matematikkfaget er kjerneområdene «utforsking og problemløsning», «modellering og anvending», «resonnering og argumentasjon», «representasjon og kommunikasjon», «abstraksjon og generalisering» og «matematiske kunnskapsområder» (Utdanningsdirektoratet, 2020). Med tanke på programmering, står kjerneelementet om utforsking og problemløsning sentralt. Hva dette innebærer står beskrevet på følgende måte i læreplanen:

Utforsking i matematikk handler om at elevene leter etter mønstre, finner sammenhenger og diskuterer seg fram til en felles forståelse. Elevene skal legge mer vekt på strategiene og framgangsmåtene enn på løsningene. Problemløsning i matematikk handler om at elevene utvikler en metode for å løse et problem de ikke kjenner fra før. Algoritmisk tenkning er viktig i prosessen med å utvikle strategier og framgangsmåter for å løse problemer og innebærer å bryte ned et problem i delproblemer som kan løses systematisk. Videre innebærer det å vurdere om delproblemene best kan løses med eller uten digitale verktøy. Problemløsning handler også om å analysere og omforme kjente og ukjente problemer, løse dem og vurdere om løsningene er gyldige (Utdanningsdirektoratet, 2020, s. 2).

Sett i sammenheng med Nygård (2018) og Sevik (2016) sine definisjoner av programmering, ser en at dette sammenfatter godt. Programmeringsprosessen handler om å identifisere et problem, finne mulige løsninger, skrive koder, feilsøke og forbedre, noe som samsvarer godt med forståelsen av hva utforsking i matematikk innebærer (Opheim & Simensen, 2017). Videre vil det også stilles krav til problemløsning, da elevene skal løse et problem ved hjelp av programmering. Disse ferdighetene er også sentrale i algoritmisk tenkning, og en kan argumentere for at det er sammenheng mellom algoritmisk tenkning og programmering. Denne sammenheng vil jeg komme tilbake til i avsnitt 2.2.

Også i de digitale ferdighetene, som er en av de fem grunnleggende ferdighetene som er gjeldende i alle fag i skolen, nevnes programmering. I læreplanen står det at digitale ferdigheter i matematikkfaget blant annet innebærer å kunne «*bruke programmering til å utforske og løse matematiske problemer*» (Utdanningsdirektoratet, 2020, s. 5). At programmering nevnes konkret i digitale ferdigheter, synliggjør at programmering ses på som en viktig ferdighet i skolen i dag.

I tillegg til kjerneelementene og de digitale ferdighetene blir programmering nevnt som en ferdighet elevene skal tilegne seg i kompetansemålene i matematikk fra 5.årstinn. På de yngre trinnene fokuseres det på ferdigheter knyttet til algoritmisk tenkning (Utdanningsdirektoratet, 2020). Siden denne masteroppgaven er knyttet opp mot mellomtrinnet, er det kompetansemålene etter 5., 6., og 7.trinn som er mest relevant for denne oppgaven. Følgende kompetansemål i læreplanen er knyttet til utvikling av programmeringsferdigheter (Utdanningsdirektoratet, 2020):

- Etter 5. trinn: «lage og programmere algoritmer med bruk av variabler, vilkår og løkker»
- Etter 6.trinn: «bruke variabler, løkker, vilkår og funksjoner i programmering til å utforske geometriske figurer og mønstre»
- Etter 7.trinn: «bruke programmering til å utforske data i tabeller og datasett».

Programmering som en del av matematikkfaget i norsk skole er relativt nytt, og det er behov for mer utvikling og forskning på feltet. Det er likevel blitt gjennomført en del relevant forskning, som omhandler programmering i faget. Forsström og Kaufmann (2018) gjennomførte et litteraturstudie hvor de gjennomgikk 15 artikler som omhandlet bruk av programmering i matematikkundervisning. Funn i deres litteraturgjennomgang viser at flere av studiene fant en økning i motivasjon, og en positiv effekt på elevenes læringsutbytte i faget. Når det er sagt, poengterer forskerne at disse funnene ikke kan generaliseres. I studiet trekkes også programmerings plass i matematikkfaget fram, gjennom sammenhengen mellom programmering og algoritmisk tenkning. I følge studiet til Forsström og Kaufmann (2018) er programmering gunstig for utvikling av ferdigheter som problemløsning, logisk tenking og kreativitet. I studiet til Calder og Taylor (2010) viser funn, i likhet til funn fra forskningen til Forsström og Kaufmann (2018), at programmering kan være en motiverende arbeidsform i matematikkfaget. Samtidig trekker de frem utfordringen ved å lage oppgaver i programmering som fører til matematisk læring. Funn fra studiet til Calder og Taylor (2010) viser til at elevene tok i bruk matematisk tenkning i arbeidet med programmering, men i hvilken grad det oppsto ny matematisk læring er usikkert.

2.2 Algoritmisk tenkning

Utdanningsdirektoratet definerer algoritmisk tenkning slik:

Algoritmisk tenkning innebærer å bryte ned komplekse problem til mindre, mer håndterlige delproblemer som lar seg løse. Det inkluderer å organisere og analysere informasjon på en logisk måte og å lage fremgangsmåter (algoritmer) for å komme fram til ønsket løsning. Det handler også om å lage abstraksjoner og modeller av den virkelige verden ved å fjerne unødvendige detaljer og fokusere på det som er relevant for den aktuelle problemstilling og løsning. En løsning på et spesifikt problem kan ofte generaliseres, slik at den kan brukes til å løse lignende problemer, og løsninger på flere delproblemer kan kombineres for å løse mer komplekse problem (Utdanningsdirektoratet, 2019a).

Utdanningsdirektoratet definerer altså begrepet algoritmisk tenkning eksplisitt som en problemløsningsmetode (Mæland & Myklebust, 2022). Videre poengterer de at algoritmisk tenkning er den norske oversettelsen av det engelske begrepet «computational thinking» (Utdanningsdirektoratet, 2019a). Mæland og Myklebust (2022) skriver at Computational thinking er en overordna tilnærming, som inneholder flere kognitive komponenter, eksempelvis abstrahering, algoritmekunnskap, logikk og generalisering.

Wing (2010) definerer computational thinking på følgende måte: «*Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent*» (s.1). Wing inkluderer også selve prosessen hvor en arbeider med et problem og har likhetstrekk med Utdanningsdirektoratet sin definisjon. Videre skriver hun at: «*Computational thinking is not just or all about computing. The educational benefits of being able to think computationally transfer to any domain by enhancing and reinforcing intellectual skills*» (Wing, 2010, s. 5). Wing poengterer altså i sin definisjon at algoritmisk tenkning handler om mer enn programmering.

Torkildsen og Gjøvik (2019) problematiserer oversettelsen og forståelsen av begrepet algoritmisk tenkning. Ifølge dem er det språklig forvirrende, ved at algoritmisk tenkning og algoritmisk tankegang kan få et likt innhold på norsk. Utdanningsdirektoratet (2019a) skriver på sine nettsider at det finnes en rekke ulike definisjoner av algoritmisk tenkning, og tydeliggjør dermed at de er bevisst på denne pågående debatten om hva som skal legges i begrepet. Samtidig argumenterer de for at hovedtrekkene ved de ulike definisjonene er sammenfattende, og har med utgangspunkt i det og modellen til Barefoot Computing (UK) utarbeidet figuren Den algoritmiske tenkeren (figur 1). Figuren viser nøkkelbegrepene som inngår i algoritmisk tenkning og typiske arbeidsmåter for å løse et problem.



Figur 1: Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019)

Den algoritmiske tenkeren kjennetegnes ved at elevene må være analytiske og systematiske, men også skapende, eksperimenterende og åpne for alternative løsninger (Utdanningsdirektoratet, 2019a). En slik tilnærming krever at elevene er nysgjerrige og utforskende i sitt arbeid. Gjennom å prøve og feile må det være rom for å gjøre feil, og en må ha en strategi for å oppdage og rette opp feilene. Dette krever at elevene har evnen til å holde ut, og ikke gir opp dersom de ikke kommer fram til riktig løsning umiddelbart. Algoritmisk tenkning handler om å bryte ned komplekse problem til mindre deler som er enklere å løse. Det er nødvendig å kunne organisere og analysere informasjonen en får i et problem, og ut fra dette lage en framgangsmåte for å løse det aktuelle problemet. Ved å komme fram til gode framgangsmåter, også kalt algoritmer, får en mulighet til å generalisere løsninger slik at de også kan brukes for å løse lignende problemer senere. Algoritmisk tenkning handler ofte om å finne mønstre, som igjen kan brukes for å løse problemer effektivt. Det er viktig at elevene klarer å lage abstraksjoner og modeller ved å fjerne informasjon som ikke er nødvendig for problemstillingen, og på den måten fokusere på den informasjonen som er viktig for å komme fram til en løsning. Et siste sentralt kjennetegn ved Den algoritmiske tenkeren er tanken om at

gode ideer ikke skapes individuelt. På bakgrunn av det trekkes samarbeid frem som en egnet arbeidsmetode (Nygård, 2018; Utdanningsdirektoratet, 2019a).

2.2.1 Sammenhengen mellom algoritmisk tenkning og programmering

Det finnes en rekke sammenhenger mellom algoritmisk tenkning og programmering.

Balanskat og Engelhardt (2015) skriver følgende: «*Computational thinking is typically associated with coding and computer programming, but is more than that, involving solving problems, designing systems, and understanding human behaviour*» (s.7). Programmering og algoritmisk tenkning er ikke synonymer, men algoritmisk tenkning kan være en god tilnærming når en arbeider med programmeringsaktiviteter (Sevik, 2016). Nygård (2018) argumenterer også for denne sammenhengen. Hun hevder at arbeidet med programmering krever alle nivåer fra å forutse og analysere hva et program skal gjøre, til å kjenne igjen mønster, eksperimentere og evaluere mulige løsninger. I tillegg er det nødvendig at elevene må kunne samarbeide med andre. Summen av alle disse ferdighetene kan en ifølge Nygårds argumentasjon kalle for algoritmisk tenkning. Undervisning i programmering handler som tidligere nevnt ikke bare om at eleven skal lære å forstå datamaskiner eller utvikle kompetanse i bruk av digitale verktøy. Like sentralt står tanken om at programmering skal lære elever å løse oppgaver og problemer, ved hjelp av gode strategier (Sevik, 2016).

Brennan og Resnick (2012) gjennomførte en studie hvor de så på hvordan programmeringsaktiviteter i Scratch støttet utviklingen av algoritmisk tenkning hos 7-14 år gamle elever. Ut fra analysen utviklet de et rammeverk hvor de identifiserte tre elementer som sentrale i algoritmisk tenkning: computational concepts, computational practices og computational perspectives.

Computational concepts handler om hva man bruker når man programmerer. I studien utpekte sju begreper seg som sentrale: sekvenser, løkker, parallellitet, hendelser, betingelser, operatorer og data (Brennan & Resnick, 2012). Videre inngår prosessen hvor elevene tenker og lærer i det Brennan og Resnick kaller for computational practices. Deres analyse viste at elevene tok i bruk ulike strategier og fremgangsmåter, der de fire mest sentrale var: justere og gjenta, teste og feilsøke, gjenbruke og remikse og abstrahere og modulisere (Brennan & Resnick, 2012). Disse strategiene samsvarer godt med arbeidsmåtene vi finner i Utdanningsdirektoratet (2019a) sin modell Den algoritmiske tenkeren (se figur 1). Det siste elementet, Computational perspectives, handler om fordeler ved å arbeide med algoritmisk tenkning. I forskning trekkes mulighet til å uttrykke seg, kommunikasjon og samhandling

med andre, deling av ideer og mulighet for å stille spørsmål som positivt (Brennan & Resnick, 2012).

Bocconi et al. (2016) har også utviklet et rammeverk som identifiserer sentrale kjennetegn ved utviklingen av algoritmisk tenkning. Deres rammeverk er basert på forskning som viser sammenfall mellom ferdigheter og kjennetegn ved algoritmisk tenkning. Rammeverket har dermed et annet grunnlag enn Brennan og Resnick (2012) sitt, som bygger på egen empiri. Rammeverket til Bocconi et al. (2016) er også delt inn i tre: kjerneferdigheter, tilnærminger og dimensjoner.

Kjerneferdighetene inneholder fem kjennetegn på algoritmisk tenkning. Disse kjennetegnene har Gjøvik og Torkildsen (2019) oversatt til de norske begrepene abstrahering, generalisering, dekomponering, automatisering og algoritmebehandling. Tabell 1 viser forklaring på begrepene.

Tabell 1: Kjerneferdigheter i algoritmisk tenkning (Bocconi et al., 2016; Torkildsen & Gjøvik, 2019)

Kjerneferdighet i algoritmisk tenkning	Forklaring
Abstrahering	Å kunne trekke ut essensen av flere eksempler eller tilfeller. Fjerne unødvendige detaljer, og på den måten gjøre problemet enklere å løse uten at man mister viktige detaljer.
Generalisering	Å gjenkjenne mønstre og sammenhenger, og på den måten lage allmenne regler og metoder. På den måten kan en løse nye, lignende problemer ved hjelp av tidligere erfaringer.
Dekomponering	Å kunne bryte ned et problem i mindre deler. Delene kan forstås og løses hver for seg, som igjen vil gjøre det enklere å løse hele det komplekse problemet.
Automatisering	Å finne en mest mulig effektiv arbeidsprosess for å løse et problem. For eksempel instruere en datamaskin til å løse problemer, på den måten går prosessen mer effektivt enn hvis mennesket skulle løst problemet.
Algoritmebehandling	Å følge og forklare trinnvise instruksjoner.

Kjerneferdighetene Bocconi et al. (2016) trekker frem finner vi igjen i nøkkelbegrepene i Den algoritmiske tenkeren (Utdanningsdirektoratet, 2019a). Begrepene abstraksjon, dekomposisjon og algoritmer samsvarer helt ordrett. Nøkkelbegrepet «mønster» med underoverskriften «finne og bruke likheter» kan gå under generaliseringsbegrepet til Bocconi et al. (2016). De to siste nøkkelbegrepene i modellen, «logikk» og «evaluerings», som kort forklart handler om å analysere, forutse og gjøre vurderinger, passer inn under flere av de nevnte kjerneferdighetene.

Videre trekker Bocconi et al. (2016) frem fire ulike tilnæringer når en arbeider med algoritmisk tenkning. Dette er feilsøking, kreativitet, samarbeid og utholdenhet. Tabell 2 viser forklaringer på hva som legges i de ulike tilnærmingene.

Tabell 2: Tilnæringer i algoritmisk tenkning (Bocconi et al., 2016)

Tilnæringer i algoritmisk tenkning	Forklaring
Feilsøking	Å vurdere eget arbeid, undersøke feil og løse dem.
Kreativitet	Å skape nye løsninger ved å se problemer fra ulike perspektiver.
Samarbeid	Å arbeide sammen med andre for å løse et problem.
Utholdenhet	Å holde ut i arbeidet med utfordrende problemer.

Disse tilnærmingene samsvarer godt med arbeidsmåtene som er presentert i Utdanningsdirektoratet (2019a) sin modell. Samarbeid og feilsøking er nevnt ordrett, mens utholdenhet samsvarer godt med å «holde ut» og «fikle», og «skape» kan plasseres under kreativitet.

Den siste kategorien i rammeverket er dimensjoner. Ett av hovedfunnene Bocconi et al. (2016) gjorde var at algoritmisk tenkning handler om mer enn programmering. Tabell 3 viser forklaringer på hva som legges i de ulike dimensjonene.

Tabell 3: Dimensjoner i algoritmisk tenkning (Bocconi et al., 2016)

Dimensjoner i algoritmisk tenkning	Forklaring
Med teknologi	Algoritmisk tenkning kan fremmes gjennom bruk av teknologi, eksempelvis gjennom programmering på en datamaskin.
Uten teknologi	Algoritmisk tenkning kan være separert fra teknologi. Denne dimensjonen legger vekt på at man kan fremme algoritmisk tenkning i aktiviteter som ikke innebærer bruk av teknologi.

I følge Bocconi et al. (2016) har altså algoritmisk tenkning to ulike dimensjoner: en med teknologi, og en uten teknologi. Dette samsvarer med Wing (2010) sin definisjon, som understreker at algoritmisk tenkning handler om mer enn programmering.

Utdanningsdirektoratet (2019a) knytter heller ikke algoritmisk tenkning direkte til programmering i sin definisjon.

2.3 Super:bit-prosjektet

I 2017 la regjeringen fram digitaliseringsstrategien «Framtid, fornyelse og digitalisering 2017-2021» for grunnsopplæringen. Bakgrunnen for satsingen er at samfunnet trenger mer spesialisert og generell IKT-kompetanse for å utnytte mulighetene som digitalisering gir. Eksempelvis er arbeidslivet avhengig av at utdanningssystemet utdanner arbeidstakere med den kompetansen som er nødvendig i yrkesutøvelsen. Regjeringen foreslo en rekke tiltak, deriblant å innføre teknologi og koding i skolens læreplaner (Kunnskapsdepartementet, 2017). Som en del av denne satsingen ble «Den teknologiske skolesekken» utviklet. Gjennom «Den teknologiske skolesekken» får skoleeier, ledere og lærere mulighet til å motta tilskuddsordninger og andre tiltak for å sikre en digital utvikling i skolen. Formålet er å gi elever (Utdanningsdirektoratet, 2019b):

- Kunnskap om og forståelse for teknologi, algoritmisk tenkning og programmering.
- Tilgang til digitale læremidler.
- Gode, trygge og moderne innloggingsmuligheter som gjør det enkelt å bruke digitale læremidler.

Satsingen på digitalisering førte til at Lær Kidsa Koding, Vitensenterforeningen og NRK gikk sammen om et felles prosjekt, med navnet «Super:bit». Prosjektet består av aktiviteter på vitensentrene, kodeklubber og innhold på NRKs flater. Gjennom «Den teknologiske skolesekken» ble det bevilget 50 millioner kroner til prosjektet, som skulle brukes til opplæring i skolen (Moldøyen, 2019). I tillegg støttet Sparebankstiftelsen DNB prosjektet med cirka 20 millioner kroner for å kjøpe klassesett med programmeringsutstyr, og utvikle oppgaver og materiell. Alle landets 2400 grunnskoler fikk tilbud om å bli med på prosjektet, og etter at lærerne deltok på lærerkurs i regi av sitt lokale vitensenter i skoleårene 2019/20 og 2020/21 mottok de klassesett med programmeringsutstyr (Vitensenterforeningen, u.å.-c).

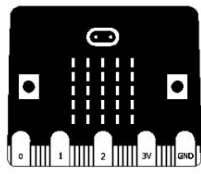
Oppsummert handler Super:bit-prosjektet om at barn og unge i Norge skal lære seg programmering ved bruk av programmeringsverktøyet micro:bit. Gjennom prosjektet er målet at barn skal få interesse av å forstå og skape det digitale (Super:bit, u.å.).

2.3.1 De ulike aktørene i prosjektet

Lær Kidsa Koding

Lær Kidsa Koding er en frivillig bevegelse som arbeider for at barn og unge skal lære å forstå og beherske sin egen rolle i det digitale samfunnet. Gjennom ulike aktiviteter er formålet å hjelpe unge til å ikke bare bli brukere, men også skapere med teknologi som verktøy. På den måten ønsker Lær Kidsa Koding å bidra til rekruttering til IT-yrker og realfag (Lær Kidsa Koding, u.å.-b). Lær Kidsa Koding har over 200 registrerte kodeklubber rundt om i landet. Totalt har kodeklubbene 1400 arrangement der 25 000 barn og unge utforsker programmering. I tillegg arrangerer Lær Kidsa Koding Kodetimen hvert år. Kodetimen er en skoletime med programmering, som når over 60 000 elever fra over 1000 skoler (Super:bit, u.å.). Kodetimen arrangeres i desember, men opplegget ligger tilgjengelig på Lær Kidsa koding sine nettsider.

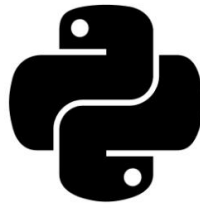
På nettsidene til Lær Kidsa Koding finnes en rekke ressurser, deriblant oppgaver og informasjon for kodeklubber og skoler. Skoler kan finne forslag til undervisningsopplegg, studiesteder og andre ressurser som er relevant for de som arbeider i skolen. Det ligger også svært mange oppgaver tilgjengelig, som kan løses ved hjelp av ulike programmeringsverktøy (se figur 2).



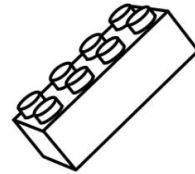
Micro:bit ⓘ
Oppgaver: 55



Scratch ⓘ
Oppgaver: 47



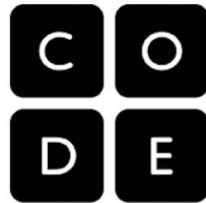
Python ⓘ
Oppgaver: 36



LEGO Mindstorms ⓘ
Oppgaver: 24



Web ⓘ
Oppgaver: 20



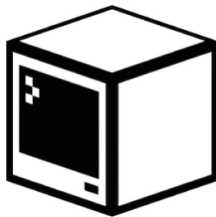
CodeStudio ⓘ
Oppgaver: 18



Processing ⓘ
Oppgaver: 13



Elm ⓘ
Oppgaver: 11



ComputerCraft ⓘ
Oppgaver: 8



App Inventor ⓘ
Oppgaver: 5



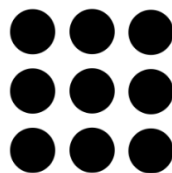
Arduino ⓘ
Oppgaver: 5



Learn To Mod ⓘ
Oppgaver: 5



Uten datamaskin ⓘ
Oppgaver: 5



Diverse ⓘ
Oppgaver: 4

Figur 2: Utklipp av oppgavesiden (Lær Kidsa Koding, u.å.-a)

Innenfor hvert programmeringsverktøy finner man et kurs eller en «samleside» med ulike oppgaver som kan løses med det valgte verktøyet. Alle kursene er delt inn slik at oppgavene er kategorisert i fire nivåer: introduksjon, nybegynner, erfaren og ekspert. Det er også mulig å søke etter oppgaver ved å legge til ulike filter i søkemotoren, slik at en finner oppgaver som er tilpasset et bestemt klassetrinn, tema eller fag.

Vitensentrene

Super:bit skole er undervisningsdelen av Super:bit-prosjektet, og er utviklet av de regionale vitensentrene i Norge. Som del av «Den teknologiske skolesekken», fikk

Vitensenterforeningen, som består av tolv regionale vitensentre rundt om i landet, et 5-årig oppdrag fra Kunnskapsdepartementet via Utdanningsdirektoratet. Oppdraget foregår i perioden 2018-2022, og går ut på å fremme programmeringskompetanse og forståelse hos elever og lærere på mellomtrinnet (Super:bit, u.å.).

Super:bit skole består av fem deler: et lærerkurs, forarbeid, Super:bit-oppdraget, etterarbeid og bonusoppdrag. Lærerkurset og Super:bit-oppdraget er det de regionale vitensentrene som har ansvar for å gjennomføre, mens de resterende delene av undervisningsopplegget er det skolene og lærerne selv som må vurdere om de ønsker å gjennomføre (Vitensenterforeningen, u.å.-a). Målet er at alle landets sjetteklassinger skal få delta på programmeringsaktiviteter med sitt regionale vitensenter i skoleåret 2019/20 eller 2020/21. I tillegg ønsker de å gi lærere på mellomtrinnet et solid støtteverktøy, slik at de senere kan bruke ressursene fra Super:bit med sine elever.

Super:bit skole starter med et lærerkurs, som arrangeres av de regionale vitensentrene. Kurset består av informasjon om Super:bit, undervisningsopplegget og presentasjon av utstyrspakken som skolene får utlevert etter endt kurs. Utstyret er tilstrekkelig for å gjennomføre alle oppgavene i Super:bit skole. Videre har lærerne som har deltatt på kurset ansvar for å gjennomføre forarbeidet med sine elever. Målet med forarbeidet er å forberede elevene til gjennomføringen av Super:bit-oppdraget. Forarbeidet består av tre enkle og praktiske oppgaver, som kan gjennomføres uten internett og teknisk utstyr.

Selve Super:bit-oppdraget er det de regionale vitensentrene som har ansvar for å gjennomføre, enten ved at skolen kommer på besøk eller ved at de besøker den aktuelle skolen. Oppgavene i oppdraget skal gi elevene er introduksjon til praktisk bruk av programmering og ta i bruk utstyret i den utdelte pakken. Oppdraget består av fire deler: «programmering uten skjerm», «kom i gang med programmering», «kjør en meter med BitBot» og «kjør en løype med BitBot».

Etterarbeidet og bonusoppdraget består av oppgaver læreren kan gjennomføre med sin klasse. Alt utstyret som trengs for å gjennomføre oppgavene følger med i den utdelte utstyrspakken. Oppgavene er varierte, og handler om å programmere en micro:bit til å gjøre ulike ting som å lage et trafikklys, lage en veibom og lage en danserobot. Alt av undervisningsopplegg og

lærerveiledning ligger tilgjengelig på Vitensenterforeningen sine nettsider, <https://www.vitensenter.no/superbit/>.

NRK

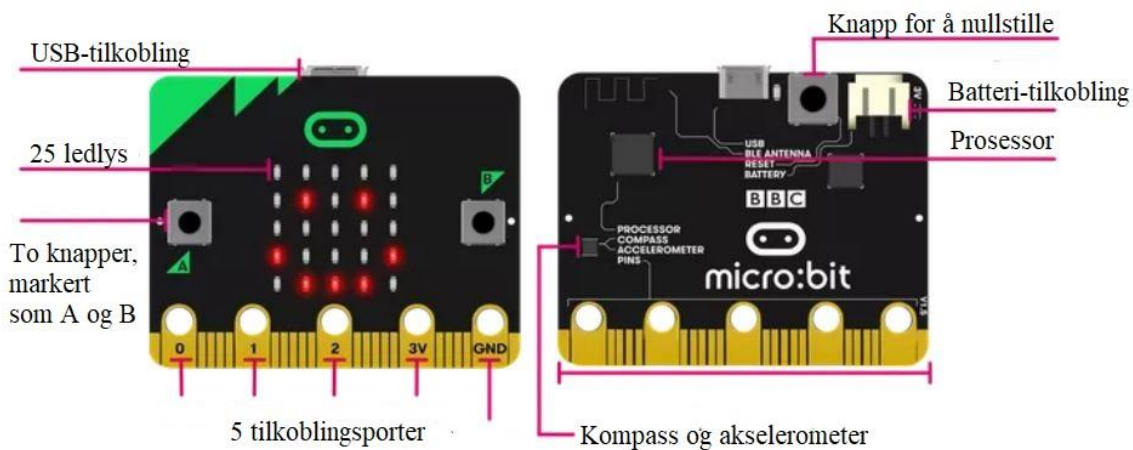
NRK sin rolle i samarbeidet er å eksponere prosjektet for landets barn og unge gjennom NRK Super og NRK Skole. På nettsidene finnes det en rekke videoer om prosjektet, fra hva utstyrspakken inneholder og hvordan en skal komme i gang med utstyret til hvordan en kan løse vanlige «problemer» knyttet til programmeringen. I tillegg er det laget videoer hvor det vises og forklares hvordan bonusoppgavene i Super:bit skole kan løses.

2.3.2 Utstyrspakken

Utstyrspakken, som deles ut gratis til alle grunnskoler der lærere har deltatt på lærerkurset i Super:bit skole, inneholder et klassesett med programmeringsutstyr. Pakken inneholder utstyr til cirka 20-25 brukere og består av (Vitensenterforeningen, u.å.-c):

- 20 stk. micro:bit
- 10 stk. BitBot XL
- Ultrasensor til BitBot XL
- 10 mm hvite, gule, grønne, blå og røde lysdioder
- 5 mm hvite lysdioder
- Servoer, 5 stk. 180 graders og 5 stk. 360 graders
- 5 høyttalere
- 1 RBG lysdiodstripe
- Krokodilleklemmer i ulike farger
- 5 ruller svart elektrisk tape
- 1 oppbevaringskasse
- 1 Smart-by Super:bit-matte.

I undervisningsopplegget i forbindelse med dette masterprosjektet var det i hovedsak micro:bitene, BitBotene og Super:bit-matta som ble benyttet. **Micro:bit** er en liten programmerbar datamaskin, som er designet for å gjøre programmering enkelt og gøy. Enheten ble opprinnelig utviklet av BBC, med formålet at den skulle brukes i opplæringen i den engelske skolen (Nätt, 2020).

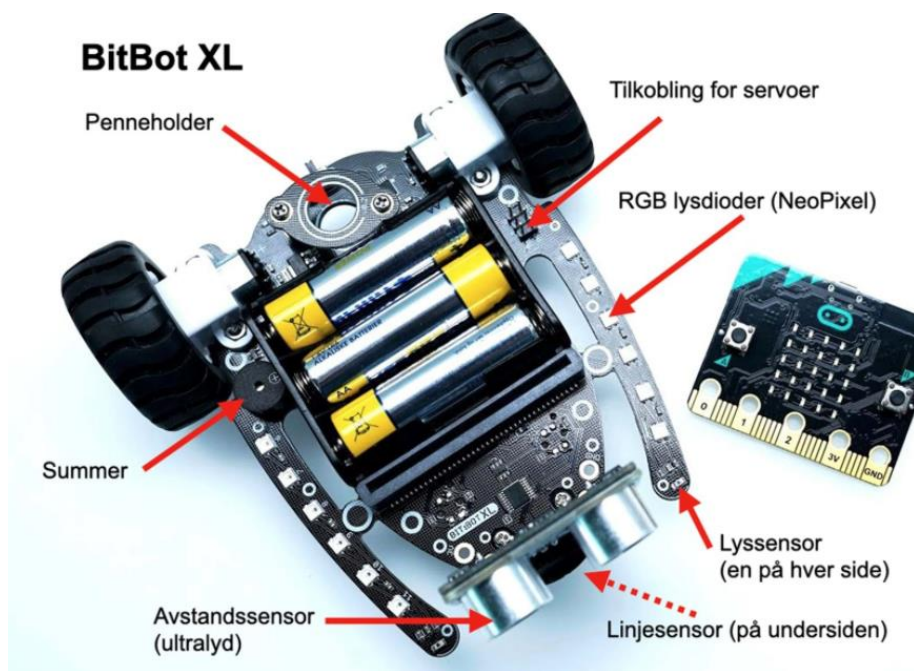


Figur 3: Micro:bit med markerte funksjoner (Micro:bit Educational Foundation, u.å.)

Selve micro:biten er på størrelse med en fyrstikkeske, noe som gjør den lett å bruke og ta med seg. Den drives av en liten batteripakke med to AAA-batterier eller tilkoblet til strøm via en USB-ledning. Enheten har en rekke funksjoner og muligheter, som gjør det mulig å programmere, kontrollere og tilpasse ulike funksjoner. Den har 25 ledlys som kan vise ulike meldinger, to knapper som kan programmeres til å styre programmer eller spill, og sensorer som kan styre bevegelser og retninger (Kalelioglu & Sentance, 2020).

Kalelioglu og Sentance (2020) gjennomførte et studie hvor de undersøkte læreres erfaringer med bruk av micro:bit i undervisning. I studiet trakk lærerne frem at fordelene med micro:bit er at den er enkel å lære og bruke for elevene, det engasjerer og at verktøyet gir mange muligheter. Ved bruk av et fysisk redskap i programmeringen får elevene umiddelbar tilbakemelding på arbeidet de gjør, noe som gir gode muligheter for utforskning. På den måten kan micro:biten skape stort engasjement og være et motiverende arbeidsverktøy (Kalelioglu & Sentance, 2020).

Micro:biten kan programmeres og kobles til **BitBoten**, som kan beskrives som en slags bil. Elevene kan lage programmeringskoder som får BitBoten til eksempelvis å kjøre, svinge, tute eller lyse i ulike farger.



Figur 4: BitBot XL med markerte funksjoner (Vitensenterforeningen, u.å.-b)

I tillegg til micro:biten og BitBoten benyttet vi oss av **Smartby-matta**, som også blir utlevert i utstyrspakken. Matten kan brukes ved at elevene programmerer BitBoten til å kjøre på veiene, eller at elevene programmerer eksempelvis lyktestolper og vegbommer til byen.



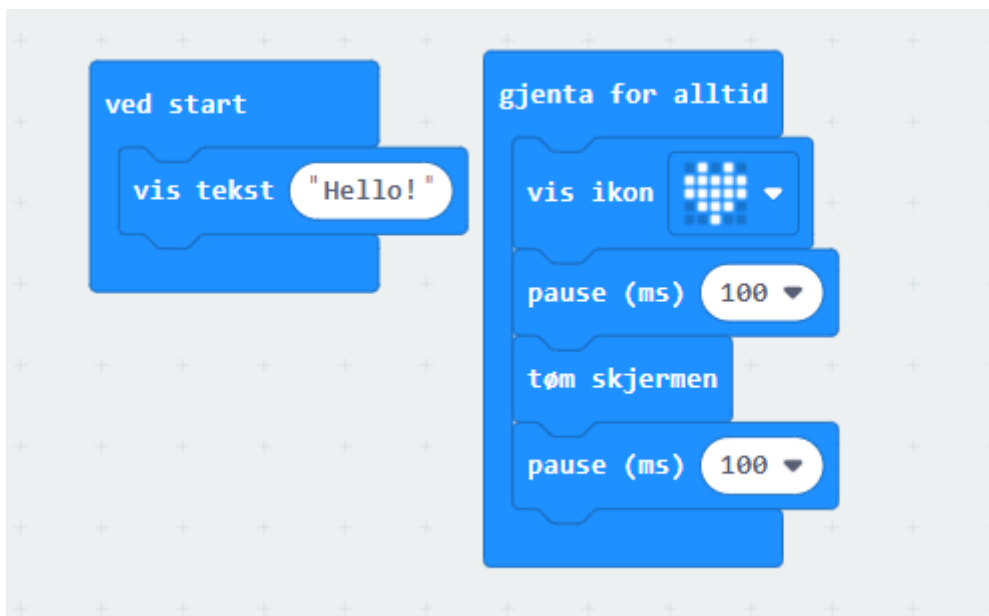
Figur 5: Smartby-matta (Vitensenterforeningen, u.å.-a)

2.3.3 Programmeringsspråk

I Super:bit-opplegget trenger ikke elevene å skrive egne koder ved bruk av tekstprogrammering. Elevene benytter seg av blokkprogrammering, som er ferdige blokker som settes sammen. En av fordelene ved dette er at elevene ikke skriver kommandoene selv, og en kan dermed unngå en del potensielle skrivefeil. Blokkprogrammering er av den grunn

lurt å starte med for yngre elever og elever som ikke har så mye programmeringserfaring (Haraldsrud et al., 2020; Nygård, 2018).

I Super:bit-opplegget programmeres micro:biten med blokkspråket MakeCode, som er utviklet av Microsoft. Dette er tilsvarende likt programmeringsspråket Scratch, som også blir mye brukt i skolen (Haraldsrud et al., 2020). Programmet fungerer ved at man drar inn de blokkene en ønsker å bruke fra menyen, og setter dem sammen i ønsket rekkefølge. Dette kan være blokker som «ved start», «når knapp A trykkes», «gjenta for alltid» eller «kjør fremover med fart 60%». Enheten vil videre utføre de handlingene vi ønsker.



Figur 6: Eksempel på blokkprogrammering i MakeCode

Figur 6 viser et eksempel på blokkprogrammering, programmert i MakeCode. Ved hjelp av de ulike blokkene har jeg i dette eksempelet programmert en micro:bit til å vise teksten «Hello!» ved oppstart, for så å vise et blinkende hjerte i ubegrenset tid. Dette var en av oppgavene elevene som deltok i undervisningsopplegget i forbindelse med denne masteroppgaven gjennomførte, for å bli kjent med programmet og micro:biten.

I tillegg til blokkprogrammering legger noen av oppgavene i Super:bit opp til analog eller unplugged programmering. Dette er programmering som ikke krever datamaskin eller andre digitale verktøy. Nygård (2018) trekker frem at analog programmering fungerer godt som introduksjon. Gjennom analog programmering arbeider man med forståelse av begreper og kommandoer (Statped, 2021). I forarbeidet, som skal gjennomføres før Super:bit-oppgaven slik det ble beskrevet tidligere, ligger det forslag til tre oppgaver som skal løses ved hjelp av

analog programmering. Disse oppgavene er gode for å visualisere ulike begreper, og gjennom fysiske øvelser får elevene opplevd hvordan ulike kommandoer fungerer. Dette kan være en god øvelse for å synliggjøre behovet for å være nøyaktig og forstå hvor lett det er å tolke beskjeder ulikt. Analog programmering er dermed en god tilnærming for å arbeide med forståelse av årsak, virkning, begreper og kommandoer før elevene starter på den digitale programmeringen (Statped, 2021). Funn fra forskning understreker at lærere og elever ser fordeler ved å bruke analog programmering i undervisningen. Tilnærmingen kan være motiverende, og bidra til at elevene får bedre forståelse når de programmerer med digitale verktøy (Bell & Vahrenhold, 2018).

Programmering kan som nevnt også skje gjennom bruk av tekstprogrammering, der elevene selv må skrive kodene. Tekstprogrammering er mer utfordrende enn blokkprogrammering, og passer best for eldre elever (Statped, 2021). Tekstprogrammering har ikke blitt behandlet i denne masteroppgaven.

2.4 Sosiokulturelt perspektiv på læring

I studiet til Kalelioglu og Sentance (2020) understrekte lærerne som brukte en tilnærming med utforskende læring i programmeringsundervisning elevenes behov for støtte, spesielt i oppstarten av undervisningen. Funn fra forskningen viser at det er viktig å finne en balanse mellom å gi elevene nok frihet til å utforske, men samtidig ha nok informasjon og evne til å kunne utforske. Dette er i tråd med det sosiokulturelle perspektivet på læring, som vektlegger at elevene trenger støtte for å kunne lære og utvikle seg (Säljö, 2016). Det sosiokulturelle perspektivet på læring kan være et sentralt perspektiv i en utforskende tilnærming til læring, slik som Inquiry-based learning, som jeg vil gjøre rede for i avsnitt 2.5.

Røttene til det sosiokulturelle perspektivet finner vi i arbeidet til den russiske psykologen Lev Vygotskij. Et viktig utgangspunkt er synet på læring, som i det sosiokulturelle perspektivet forstås som en aktiv prosess der læring er et resultat av interaksjon med andre i læringsmiljøet (Skaalvik & Skaalvik, 2021). Læring skjer altså i samspill med andre, ved at man approprierer kunnskap. Appropriering vil si å ta til seg, å låne eller å ta over og gjøre til sitt (Säljö, 2016). Eksempelvis lærer barn språklige uttrykk og måter å oppføre seg på, ved at de kjenner det igjen og gradvis lærer å ta det i bruk selv. Ut fra denne forståelsen er det det sosiale felleskapet, kulturen og språket som danner grunnlaget for utvikling og læring (Imsen, 2020).

Sentralt i det sosiokulturelle perspektivet står Vygotskij sine tanker om den proksimale utviklingssone. I følge Vygotskij (1978) bygger læring på en grunnleggende ubalanse, hvor

det er en voksen, eller mer kompetent, som må støtte eleven inn i ny kunnskap. Dette er basert på at barnet har to mentale utviklingsnivåer. Det første utviklingsnivået representerer den oppnådde kunnskapen, altså det barnet klarer å utføre alene, uten hjelp eller støtte. Det andre utviklingsnivået kaller Vygotskij for den proksimale utviklingssone. Her ligger de utfordringene barnet greier å utføre med hjelp og støtte fra en kompetent annen. Målet er at barnet skal utvikle seg, slik at en greier å håndtere disse utfordringene på egenhånd. På den måten skjer det en utvikling, og undervisningen i skolen må ut fra denne forståelsen skje innenfor den proksimale utviklingssonen. Ifølge det sosiokulturelle perspektivet er det på bakgrunn av dette nødvendig med tilpasset opplæring, slik at oppgaver, innhold og vanskelighetsgrad er tilpasset den enkelte elevs proksimale utviklingssone (Säljö, 2016).

Ettersom at undervisningen bør knyttes til den proksimale utviklingssonen, betyr det at elevene trenger veiledning og støtte i arbeidet sitt. Begrepet «scaffolding», eller støttende undervisning, brukes gjerne om denne veiledningen og støtten (Imsen, 2020). Et viktig poeng i denne veiledningen er at elevene ikke bare skal være passive mottakere av informasjon ved at de får svaret servert. Elevene skal være bidragsytere og medspillere i utviklingen av kompetanse, og må få veiledning og støtte slik at de selv finner løsninger ved å få tilstrekkelige hint, forklaringer, korrigeringer og oppmuntringer. Av den grunn er språket eller dialogen helt sentralt i dette perspektivet (Skaalvik & Skaalvik, 2021). Vygotskij understrekte at det viktigste redskapet vi har er språket, og kalte det for «*the tool of the tools*» (Vygotskij, 1978, s. 53).

Tanken om redskaper er sentralt i det sosiokulturelle perspektivet. Redskapene medierer handlingene vi gjør, de utgjør instrumenter vi bruker og er avhengige av. Menneskers evner avgjøres ikke av biologiske forutsetninger, men av at individet utvikler og bruker ulike redskaper. Ifølge Vygotskij (1978) bruker vi intellektuelle redskaper også når vi tenker og kommuniserer. Vi er altså avhengige av både fysiske og mentale redskaper i en læringsprosess (Säljö, 2016). Et enkelt eksempel som kan forklare dette er bruken av en micro:bit. En micro:bit er et fysisk redskap, som kan hjelpe oss når vi skal løse ulike oppgaver i matematikk. Vi er likevel helt avhengige av det mentale redskapet som gir oss kunnskap om hvordan vi kan bruke micro:biten til å utføre disse oppgavene.

2.4.1 Læringspartner

I et sosiokulturelt perspektiv på læring er det som nevnt sentralt at læring skjer i et sosialt samspill. Metoden med å bruke læringspartnere i undervisningen har bakgrunn i denne tankegangen. Gjennom arbeid med en partner skjer læringen ved bruk av språket,

interaktivitet, dialog og deltakelse i en sosial praksis. Tanken er at når elevene samarbeider, kommuniserer og diskuterer, vil de kunne lære og utvikle seg (Olsen & Aasland, 2013).

Kommunikasjon, samhandling og deltakelse er sentrale kompetanseområder i skolen. Dette er ferdigheter elevene må mestre for å være rustet for arbeidslivet og hverdagen generelt. I forhold til programmering skjer programmeringsutvikling sjeldent alene, og samarbeid med andre er dermed sentralt. Programmering er en fin arena for å lære elevene samarbeid, og arbeid i par trekkes frem som en gunstig organisering. På den måten kan de dra nytte av hverandre, ha dialog og diskusjon rundt utfordringene de møter, reflektere og sammen vurdere arbeidet de har utført (Sevik, 2016).

I boka til Olsen og Aasland (2013) siteres en lærer på følgende måte: *«Læringspar er en metode der elevene arbeider sammen i par med det formålet å øke eget og medelevers læringsutbytte»* (s.7). Gjennom dialog mellom elevene, legges det opp til at de sammen skal skape forståelse og tilegne seg stoffet. Tanken er at responsen eller tilbakemeldingen fra læringspartneren skal være med på å skape læringsutbyttet hos begge elevene. Videre listes det opp en rekke fordeler ved denne arbeidsmetoden. Disse fordelene er samarbeid, kommunikasjon, forklaring, diskusjon og refleksjon. I tillegg trekkes det frem at elevene blir aktive, de blir tryggere når de ikke er alene om svaret, flere svar kommer frem og elevene blir mer bevisst sin egen kompetanse. Arbeid i læringspar kan også være med på å skape nye, gode relasjoner mellom elevene, og er av den grunn også en god metode for å arbeide med læringsmiljøet i klassen (Olsen & Aasland, 2013).

I studiet til Kalelioglu og Sentance (2020) trekkes det også frem flere fordeler ved å arbeide i par. For det første gir det mulighet til å diskutere og rette opp eventuelle feil. Organiseringen ses også på som positiv fordi elevene får mulighet til å støtte og hjelpe hverandre.

Eksempelvis forteller en av lærerne i studiet til Kalelioglu og Sentance (2020) at hun utnytter denne muligheten ved å sette en dyktig elev med en mindre dyktig elev, slik at de kan hjelpe hverandre. På den måten vil den dyktige eleven kunne være en kompetent annen, som Vygotskij trakk frem som nødvendig i den proksimale utviklingssonen.

2.5 Inquiry-based learning

Som nevnt i avsnitt 2.1.2 stiller det nye kjerneområdet utforskning og problemløsning krav om at undervisningen skal legge til rette for at elevene kan arbeide utforskende i matematikkfaget. Opheim og Simensen (2017) argumenterer for at en utforskende tilnærming til matematikkundervisning er sentral for å forstå matematiske begreper og for mestringen i

faget. Kjennetegn med en utforskende tilnærming i undervisningen er å utfordre elevene til å selv finne sammenhenger og løsningsstrategier, fremfor at læreren skal trene dem opp til å følge en gitt prosedyre. På denne måten får elevene selv en aktiv deltakende rolle i undervisningen (Stedøy, 2018). Denne tilnærmingen samsvarer godt med Inquiry-based learning (IBL).

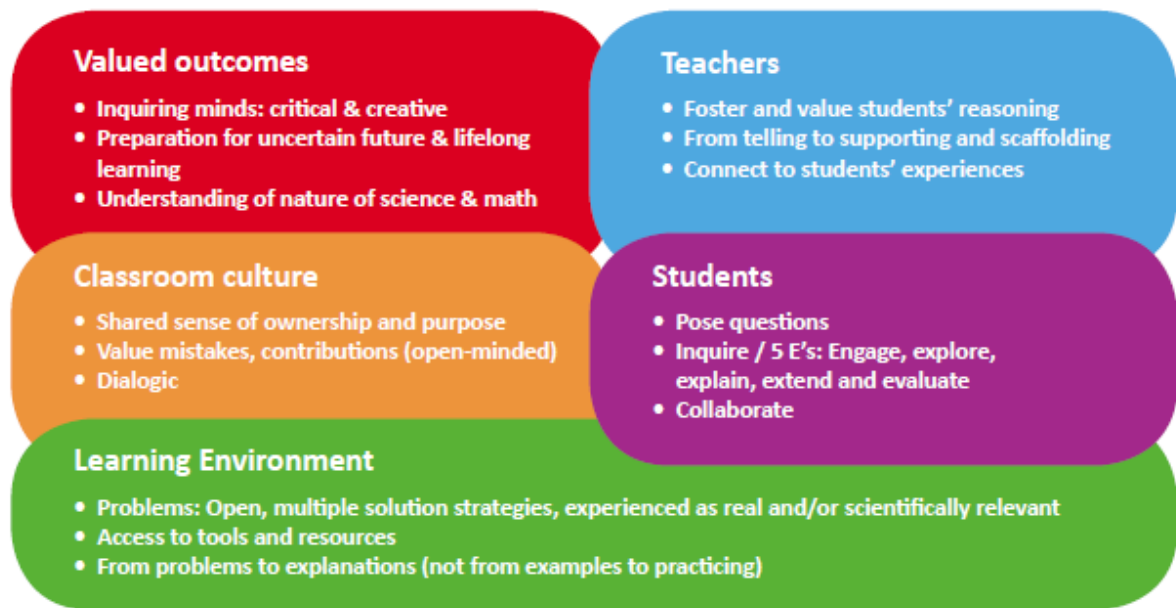
IBL bygger på en pedagogisk filosofi om at utdanning skal være for alle, stimulere den enkeltes interesse for å lære, styrke autonomi og danne mennesker som skal være en aktiv del av dagens samfunn. Utviklingen av IBL som pedagogisk tilnærming skjedde i et skille hvor kunnskap tidligere var blitt sett på som noe som var gitt, til at kunnskap er basert på tenkning, refleksjon, eksperimentering og vitenskap. Mennesket ble på denne måten satt i sentrum for utviklingen av kunnskap (Artigue & Blomhøj, 2013). Den amerikanske filosofen og pedagogen John Dewey var sentral i fremveksten av IBL. Dewey var blant de første teoretikerne som la vekt på individets aktive medvirkning i en læringsprosess. Dewey hevdet at læring skjer gjennom at en høster erfaringer fra en handling (Imsen, 2020). En forutsetning for læring er altså at individet forstår sammenhengen mellom handlingen og resultatet.

Maaß og Artigue (2013) skriver at «*The term «inquiry-based learning» generally refers to student-centered ways of teaching in which students raise questions, explore situations, and develop their own ways towards solutions*» (s.780). Begrepet inquiry-based learning omfatter altså en undervisningspraksis der elevene er i sentrum. Elevene skal stille spørsmål, eksperimentere og utforske for å utvikle ny kunnskap. Artigue & Blomhøj (2013) skriver at slike begreper har blitt brukt i en rekke politiske dokumenter og i læreplaner knyttet til naturfag- og matematikkundervisning det siste tiåret, deriblant i LK20. Dette indikerer at utviklingen av en spørrende og vitenskapelig holdning står sentralt i den nye læreplanen (Sikko & Grimeland, 2020). Eksempelvis finner vi begreper som «utforske» «forklare», «formulere og løse problem», «diskutere» og «beskrive» i kompetansemålene i matematikkfaget (Utdanningsdirektoratet, 2020). Matematikk handler om mer enn å kunne regler og prosedyrer, en må eksempelvis kunne oppdage mønster og sammenhenger, systematisere og finne måter å representere funnene sine (Fuglestad, 2010; Sikko & Grimeland, 2020).

2.5.1 Kjennetegn ved IBL

Gjennom EU-prosjektet Primas, Promoting Inquiry in Mathematics and Science Education Across Europe, ble det utviklet en modell som identifiserer en rekke kjennetegn ved IBL. Modellen deler inn IBL i fem aspekter: valued outcomes (ønsket utbytte), classroom culture

(klasseromskultur), teachers (lærere), students (elever) og learning environment (læringsmiljø) (Abril et al., 2013). Jeg vil gjøre rede for de tre aspektene i modellen som er mest sentrale for å svare på problemstillingen i denne masteroppgaven: ønsket utbytte, klasseromskultur og lærerrollen. Elevenes rolle er ikke tatt med, da det er lærerens rolle og erfaringer jeg har undersøkt. Utviklingen av et godt læringsmiljø vil heller ikke belyses, da dette omhandler faktorer som ligger utenfor tematikken i dette prosjektet.



Figur 7: Kjennetegn ved IBL (Abril et al., 2013, s. 8)

Ønsket utbytte

Ett av formålene ved IBL er at undervisningen skal ligne på måten matematikere og forskere arbeider. I forbindelse med det nevnte PRIMAS-prosjektet ble det utarbeidet en modell (figur 8), som viser hvilke arbeidsmåter som er gunstige i undervisning som bygger på IBL.



Figur 8: Arbeidsmåter i IBL (Primas, 2011, s. 11)

I figur 8 ser vi at arbeidsmåter som eksempelvis kommunikasjon, eksperimentering, utforskning, og eksemplifisere og strukturere trekkes frem som gunstige. Flere av arbeidsmåtene i figuren finnes også i det tidligere nevnte kjerneelementet «utforskning og problemløsning», som jeg i avsnitt 2.1.2 har argumentert for at er spesielt sentralt i programmeringsundervisning:

Utforskning i matematikk handler om at elevene **leter etter mønstre, finner sammenhenger** og **diskuterer** seg fram til en felles forståelse. Elevene skal legge mer vekt på **strategiene og framgangsmåtene** enn på løsningene. Problemløsning i matematikk handler om at elevene **utvikler en metode** for å løse et problem de ikke kjenner fra før. Algoritmisk tenkning er viktig i prosessen med å **utvikle strategier og**

framgangsmåter for å løse problemer og innebærer å **bryte ned et problem i delproblemer** som kan løses systematisk. Videre innebærer det å vurdere om delproblemene best kan løses med eller uten digitale verktøy. Problemløsning handler også om å **analysere** og omforme kjente og ukjente problemer, løse dem og **vurdere** om løsningene er gyldige (Utdanningsdirektoratet, 2020, s. 2).

Ved bruk av de nevnte arbeidsmåtene er det ønskede utbyttet at elevene skal utvikle seg til å bli tenkende, kreative og kritiske. Dette er egenskaper som tradisjonelt har blitt mindre vektlagt i matematikkundervisningen (Sikko & Grimeland, 2020). Likevel ser det som nevnt ut til at dette er ferdigheter som gjør seg mer gjeldende i LK20. Dette ser vi også i «fagets relevans og sentrale verdier» hvor det blant annet heter at «*Matematikk skal forberede elevene på et samfunn og arbeidsliv i utvikling ved å gi dem kompetanse i utforskning og problemløsning*» (Utdanningsdirektoratet, 2020, s. 2). Gjennom IBL er ikke målet for matematikkundervisningen at elevene bare skal tilegne seg en bestemt matematisk ferdighet, men de skal være med i konstruksjonen, evalueringen og refleksjonen rundt den kunnskapen de tilegner seg. På den måten er målet at elevene skal sitte igjen med kompetanse som er mer nyttig i dagliglivet (Engeln et al., 2013). Maaß og Artigue (2013) argumenter for at matematikkundervisningen får et utvidet sett med mål, som å styrke elevenes kompetanse i matematisk tenkning, bygge elevenes motivasjon for å lære, utstyre elevene med strategier for videre læring i fremtiden, og hjelpe elevene med å tilegne seg kompetanser de trenger.

I undervisningsopplegget som ble utarbeidet i forbindelse med dette prosjektet ønsket vi at elevene skulle tilegne seg matematisk kompetanse gjennom programmeringsaktivitetene. Kaufmann og Stenseth (2020) argumenterer i sin forskning for at integrering av programmering i matematikkfaget er mulig, men trekker frem at det stilles store krav for at matematikken ikke skal «forsvinne» i aktiviteten. Rodríguez-Martínez et al. (2020) evaluerte potensialet i programmeringsaktiviteter med Scratch for å fremme sjetteklasseselevens tilegning av matematiske kunnskap og algoritmisk tenkning. Funn i studiet viste ikke en større effekt i forhold til en mer tradisjonell tilnærming til undervisning, men på en annen siden heller ingen negativ effekt. Det viste seg altså at elevgruppa som arbeidet med matematikk i Scratch hadde like godt læringsutbytte som kontrollgruppa som ikke brukte Scratch. Flere andre studier støtter konklusjonen om at en ikke ser noe signifikant forbedring i matematiske ferdigheter gjennom bruk av programmeringsverktøy i undervisningen, men samtidig heller ingen negativ effekt (Ardito et al., 2014; Hussain et al., 2006; Ríos Félix et al., 2020). Forskning viser imidlertid en forbedring i interesse og engasjement gjennom

programmeringsaktiviteter, som igjen kan gi en positiv virkning på motivasjonen for matematikkfaget (Husain et al., 2017). Denne konklusjonen støttes i forskningen til Ríos Félix et al. (2020), som undersøkte bruk av programmeringsverktøy i undervisningen i algoritmisk tenkning. Flere av elevene i eksperimentgruppa, som fikk bruke programmeringsverktøy, viste interesse for å lære mer etter endt undervisningsperiode enn elevene i kontrollgruppa.

Klasseromskultur

IBL medfører at en går fra en deduktiv lærerstil til en aktiv form for undervisning og læring hvor eleven står i sentrum. Utforskning er en sentral del av læringen, og dette har betydningen for kulturen i klasserommet. Engeln m. fl. (2013) understreker betydningen av at læring skjer i en sosial prosess, og dialog er dermed sentralt. I et utforskende klasserom er det viktig at alle innspill blir verdsatt. Det må være rom for å gjøre feil, og feil og misforståelser skal brukes som en mulighet for læring (Sikko & Grimeland, 2020).

I følge Taylor et al. (2010) er klimaet læreren skaper i klasserommet en sentral faktor for elevenes samarbeid. En klasseromskultur hvor elevene respekterer og lytter til hverandre vil styrke samarbeidet. Taylor et. al (2010) argumenterer for at det vil gi rom for at elevene kan ta roller de ikke vanligvis har i matematikkundervisningen, og på den måten vise frem kunnskapen de har i matematikk på en annen måte. I deres forskning trekker de frem to elever som i utgangspunktet hadde lav kapasitet i matematikk, men som viste seg å bli en ressurs i samarbeid med medelever som i utgangspunktet hadde høyere ferdigheter i matematikk og problemløsning. Videre trekker Taylor et al. (2010) frem at organiseringen med elevsamarbeid gir mulighet til å dele kunnskap og lære av hverandre. Dette er i tråd med synet på læring i det sosiokulturelle perspektivet, som jeg gjorde greie for i avsnitt 2.4. For at det skal skje en utvikling, må eleven få støtte av en kompetent annen, en rolle en medelev med mer kompetanse kan ha.

Lærerrolle

Lærerens holdning til matematikkfaget er sentral for hvordan undervisningen blir lagt opp (Engeln et al., 2013). Eksempelvis trekker Engeln et al. (2013) frem at mange matematikklærere ser på faget som mindre relatert til empiriske funn, og mer aksiomatisk, deduktivt og strukturert enn naturfagslærere gjør. For at IBL skal fungere i matematikkfaget, er det viktig at læreren har tro på denne utforskende tilnærmingen til kunnskap og læring. Nedenfor er det listet opp en rekke kjennetegn ved lærerrollen i IBL (Primas, 2011):

- Velge passende utfordringer for elevene
- Tydeliggjøre hensikten med oppgaven
- hjelpe elevene å se hvordan de bør samarbeide
- Gjenkjenne og bygge på elevenes forkunnskaper og erfaringer
- Oppmuntre elevene til å utforske og utveksle ideer
- Oppmuntre til diskusjon av alternative metoder og forståelser
- Fjerne frykten fra å gjøre feil, og se på det som en mulighet for læring
- Utfordre elevene ved å stille spørsmål
- Organisere og gjennomføre diskusjoner i små grupper og hel klasse
- Dra ut det viktigste fra hver time
- Ikke gi en «bruksanvisning» på hva elevene skal gjøre
- hjelpe elevene å lage koblinger mellom ideene sine og matematikken.
- Støtte og veilede elevene i deres selvstendige arbeid

En av utfordringene knyttet til programmering i skolen er lærernes kompetanse (Stigberg & Stigberg, 2020). Et av funnene Kaufmann og Stenseth (2020) gjorde gjennom sin forskning om programmering i matematikkundervisningen var nettopp knyttet til denne problematikken. Forskerne advarer mot å tro at et økt fokus på programmering i skolen i seg selv er løsningen, da forskningsresultater viser at kompetansen til lærere innenfor programmering er mangelfull. Kaufmann og Stenseth (2020) poengterer at dersom programmeringen skal integreres på en vellykket måte i matematikken, er lærerens kompetanse helt avgjørende. Funn fra forskningen til Nouri et al. (2020) viser at kunnskapsforholdet mellom lærer og elever er ganske symmetrisk i programmering sammenlignet med tradisjonelle fag. Mange av lærerne har altså ikke tilstrekkelig kunnskap for å kunne veilede og undervise elevene. På bakgrunn av dette hevder Nouri et al. (2020) at det er nødvendig med en kompetanseheving blant lærere for å utnytte læringspotensialet i programmeringen. Også Stigberg og Stigberg (2020) støtter en slik konklusjon. I deres forskning på lærernes kompetanse i matematikk og programmering viser funnene at det trengs støtte til å utvikle lærernes programmeringskunnskaper, samt utdyping av forholdet mellom programmering og matematikk for å sette programmeringen inn i en matematisk kontekst.

Den pedagogiske praksisen i programmeringsundervisningen i matematikkfaget var også en del av diskusjonene i litteraturstudiet til Forsström og Kaufmann (2018). I deres gjennomgang ble det synliggjort at samarbeid mellom elever var en hyppig organiseringsform. Læreren har dermed et ansvar for å tilrettelegge for at samarbeidet mellom elevene fungerer, blant annet

ved å løse opp i konflikter og skape en trygg klasseromskultur. Funn i litteraturgjennomgangen viser til at læreren spiller en vesentlig rolle i undervisningen. Forsström og Kaufmann (2018) trekker frem at lærerens rolle bør være å veilede elevene, ikke forelese. Denne konklusjonen støttes av Hussain et. al (2006), som understreker at lærerens rolle i undervisningen er å støtte, hjelpe og forstå elevene. De poengterer også at det er viktig at læreren kan programvaren, og at lærerens rolle er sentral med tanke på kvaliteten på undervisningen.

2.6 Fådelte skole

En fådelte skole blir definert som en grunnskole hvor et lavt elevtall fører til at ulike klassetrinn blir slått sammen i undervisningen. Fådelte skoler er motsetningen til fulldelte skoler, hvor elevene er delt etter årstrinn (Nilssen, 2021). Kvam (2013) skriver at det er enighet blant pedagoger med interesse for fådelte bygdeskoler at aldersblanding i skolen kan være tjenlig for utvikling av sosial kompetanse. Myndighetene har også i nyere tid omtalt læringsmuligheter som følge av aldersblanding, også i skoler som i utgangspunktet er fulldelte, som positive (Kvam, 2013).

Melheim (1998) hevder, med bakgrunn i forskningsresultater, at aldersblanding er en fremtidsrettet modell for organisering av læringsgrupper i skolen. Denne konklusjonen er tatt på bakgrunn av at aldersblandingen og færre elever per lærer gir gode muligheter for utvikling av aktive og selvstendige elever. Melheim (1998) hevder at elever som får undervisning i aldersblandede grupper kommer godt ut både faglig og sosialt. Rønning og Leuven (2016) har også forsket på læringsutbyttet for elever som får aldersblandet undervisning. Deres forskning viser en lav, men positiv, effekt på læringsutbyttet ved å gå i aldersblandede klasser versus vanlige klasser. Den positive effekten ved aldersblandingen skyldes at elevene drar nytte av å gå i klasse med mer modne medelever fra høyere klassetrinn. Forskerne poengterer at den positive effekten forutsetter at klasserommet er balansert i forhold til klassetrinnene, og advarer om at læringsutbyttet kan være negativt dersom en ikke kan dra nytte av de eldre elevene.

En utfordring ved fådelte skoler er tilpasning av fagstoff. Melheim (1998) poengterer at aldersblanding fører til en ekstra utfordring for læreren, ved at en i tillegg til å ta hensyn til spekteret av elever med ulik faglig og sosial kompetanse, må ta hensyn til aldersspredningen i elevgruppa. Dersom læreren velger en middelvei i formidlingen av fagstoff, kan konsekvensen være at det blir for vanskelig for de yngste og for kjedelig for de eldste elevene.

Det er grunn til å tro at den aldersblandede organiseringen kan føre til at læreren må arbeide mer med tilpasset opplæring, slik at alle elevene møter utfordringer som er tilpasset deres nivå (Melheim, 1998).

Samtidig som aldersblandingen skaper utfordringer i forhold til tilpasning av fagstoff, bringer det også med seg noen muligheter. Melheim (1998) peker på at elever som trenger repetisjon, kan få dette sammen med elever på lavere klassetrinn. Elevene i aldersblandede klasser vil ha mulighet for hyppigere repetisjon ved å se hva de yngre elevene arbeider med, og de yngste elevene vil se eksempler på fagstoff de vil møte senere.

2.7 Oppsummering av teoretisk rammeverk

I dette kapittelet har det teoretiske rammeverket blitt presentert. Dette avsnittet vil oppsummere og spesifisere hvordan rammeverket er relevant for mitt prosjekt.

Kjerneelementet «utforskning og problemløsning» i LK20 legger til rette for at elevene skal ha en utforskende tilnærming i undervisningen. Forskning viser at IBL kan være en god tilnærming, også i arbeid knyttet til programmering i matematikkfaget (Maaß & Artigue, 2013). Gjennom tilnærmingen er målet at elevene skal arbeide utforskende for å utvikle ny kunnskap. Kjennetegn ved ønsket utbytte, klasseromskultur og lærerrollen, som er tre av totalt fem aspekter Abril et al. (2013) trekker frem som sentrale i IBL, vil være sentralt i min analyse.

Forskning viser at organisering der elevene arbeider sammen i par kan være effektiv for læringsutbyttet dersom elevene greier å dra nytte av hverandre (Leuven & Rønning, 2016). Grunnet at denne oppgavens problemstilling blant annet skal belyse muligheter og utfordringer en fådelt skole gir i undervisning av programmering, vil organiseringen med aldersblanding og læringspar belyses i min analyse. Organiseringen har rot i det sosiokulturelle perspektivet på læring, som legger vekt på at læring skjer i et sosialt samspill (Imsen, 2020).

Programmering og algoritmisk tenkning henger sammen: Programmering kan være et godt verktøy for å arbeide med algoritmisk tenkning og omvendt (Sevik, 2016). I mitt prosjekt var det nødvendig å finne et hjelpemiddel som kan identifisere kjennetegn på algoritmisk tenkning i undervisningen. Utdanningsdirektoratet (2019a) sin modell Den algoritmiske tenkeren har mange fellestrekk med rammeverkene Bocconi et al. (2016) og Brennan og Resnick (2012) utarbeidet for å identifisere algoritmisk tenkning hos elever. Jeg har valgt å legge hovedvekt på Utdanningsdirektoratet sin modell, grunnet at den er laget med tanke på

den norske læreplanen. Forskningen er likevel viktig for å synliggjøre at dette er kjennetegn som er kjent i et større perspektiv. I analysen har jeg valgt ut noen av begrepene fra modellen til Utdanningsdirektoratet, og sett på hvordan de gjør seg gjeldende i mitt datamateriale. En nærmere beskrivelse av mitt analytiske rammeverk vil komme i avsnitt 3.6.

3 Metode

Dette kapittelet tar for seg masteroppgavens metode. Her vil jeg gjøre rede for den benyttede fremgangsmåten for å belyse min problemstilling. Kapittelet består av 8 delkapitler. Først vil jeg gjøre rede for de metodiske valgene som har blitt tatt i denne oppgaven, ved at jeg presenterer forskningsdesignet Design-based research (DBR), informant- og forskerrollen og datainnsamlingsverktøyene. Videre vil jeg beskrive undervisningsopplegget som ble utviklet i forbindelse med dette prosjektet, før jeg gjør rede for hvordan analysen av data har foregått. Avslutningsvis vil jeg vurdere egen forskning ved å belyse prosjektets reliabilitet og validitet, samt etiske prinsipper.

3.1 Kvalitativ metode

Denne masteroppgaven har til hensikt å undersøke hvilke erfaringer en lærer gjør seg med ressurser fra Super:bit-prosjektet i matematikkundervisningen på mellomtrinnet. I kvalitativ datainnsamling samles det inn data som har som mål å beskrive og forstå menneskers handlinger og meningsskaping i deres naturlige kontekst (Postholm & Jacobsen, 2018). Kjennetegn ved kvalitative metoder er at man søker innsikt og forståelse, og dette gjøres ved at forskeren går mer i dybden. Jeg ønsket å få innsikt i den deltakende lærerens erfaringer rundt tematikken, og kvalitativ datainnsamling passet dermed til dette formålet. Basert på datainnsamlingsverktøyene intervju og observasjon ønsket jeg å gjøre rede for og belyse erfaringene læreren fikk i undervisningen. Disse verktøyene gir mulighet til å få en dypere innsikt i lærerens forståelse, og hvilke konsekvenser dette har for handlingsvalgene i undervisningen.

Oppmerksomheten blir altså rettet mot lærerens perspektiv, og hvordan dette samspiller med det perspektivet jeg som forsker har. Kvalitative metoder gir mulighet for et nært samarbeid mellom forsker og informant, og kunnskapen produseres i dette samspillet (Postholm & Jacobsen, 2018). Jeg vil ha et nært samarbeid med min informant, gjennom planlegging og gjennomføring av undervisningsopplegg, samt ved innsamling av data i form av observasjon og intervju.

3.2 Design-based research

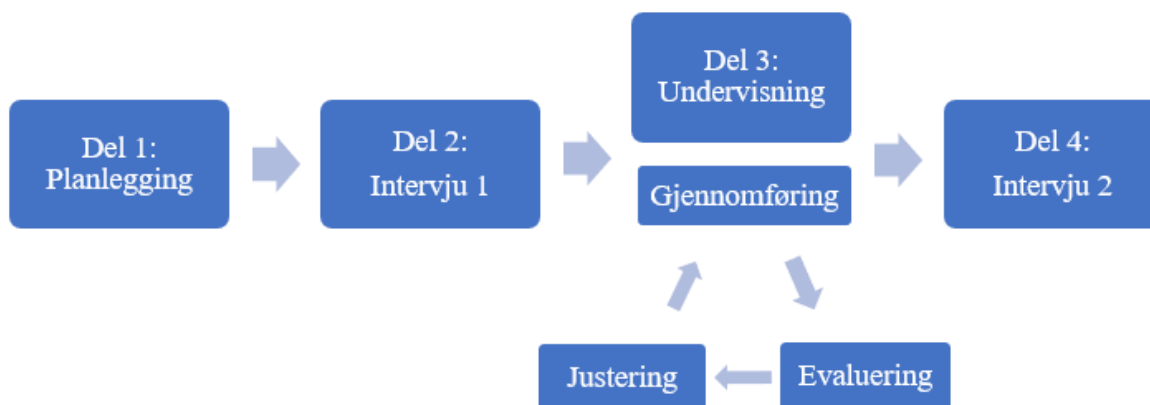
Den metodiske tilnærmingen i denne masteroppgaven er basert på forskningsmodellen Design-based research (DBR). DBR er en forskningsmetodikk som har som mål å bygge bro mellom forskning og praksis. Metodikken hadde sin framvekst tidlig på 2000-tallet, og ble i utgangspunktet designet av og for lærere. Gjennom designet ønsket en å øke effekten og

overføringsverdien mellom utdanningsforskning og praksis (Anderson & Shattuck, 2012). Bakgrunnen for dette er at det er enighet blant mange forskere og praktikere om at utdanningsforskning ofte er distansert fra problemer og utfordringer som finner sted i den daglige skolepraksisen. Det er derfor nødvendig med en metodikk som knytter forskning og praksis nærmere sammen, slik at en utvikler såkalt «brukbar kunnskap» (The Design-Based Research Collective, 2003).

Samarbeidet forskningsmetodikken DBR gir mulighet for har styrke ved at læreren og forskeren kan bidra med kompetanse på hver sin måte. Mens læreren på sin side kan være for travel til å drive forskningsarbeid, kan forskeren bidra med teorikunnskap på feltet. Forskeren på sin side har ikke samme innsikt og forståelse for praksisfeltet (Anderson & Shattuck, 2012). Målet er at partene i prosjektet skal dra nytte av hverandre ved at forskningen setter lys på de ulike forholdene i klasseromssituasjonen. Dette vil bidra til at det bygges bro mellom teoretisk kunnskap og praksis (The Design-Based Research Collective, 2003). I dette prosjektet kan jeg som forsker bidra med min kunnskap om programmering og undervisning som har blitt opparbeidet gjennom utdanning og praksiserfaringer. Læreren på sin side bidrar med kunnskap og erfaringer som har blitt opparbeidet gjennom utdanning, videreutdanninger, kurs og yrkeslivet.

The Design-Based Research Collective (2003) trekker fram DBR som en egnet forskningsmetodikk når en skal designe, bruke og forske på pedagogiske verktøy og materialer i en naturlig kontekst. Gjennom å teste ut og sette inn ulike tiltak er målet å skape forbedring. Den direkte koblingen til praksisfeltet gir på denne måten mulighet for å komme tett på det materialet en ønsker å undersøke. Metodikken var på den måten godt egnet til denne masteroppgaven, da det er erfaringer rundt bruken av ressurser fra Super:bit-prosjektet som skal belyses, og hvordan disse fungerer i en undervisningssituasjon i en klasse på mellomtrinnet.

Forløpet i DBR foregår i en syklus av gjennomføringer, evalueringer og justeringer. En av metodikkens utfordringer er dermed å vurdere om og når forskningen er fullført (Anderson & Shattuck, 2012). I mitt tilfelle var prosjektets omfang bestemt fra prosjektstart. Dette resulterte i at vi på forhånd visste hvor mange undervisningsøkter vi hadde å forholde oss til, og prosjektet hadde en naturlig slutt da disse var gjennomført. Forskningsdesignet gir stor metodefrihet, og en må velge metoder som er best egnet for å besvare problemstillingen. I dette prosjektet har det blitt samlet inn data gjennom observasjoner og intervjuer.



Figur 9: Forskningsforløpet i prosjektet, basert på DBR.

Forløpet i dette prosjektet er illustrert i figur 9. Det består av 4 deler: planlegging, intervju 1, undervisning (gjennomføring, evaluering og justering) og intervju 2. I den første delen, planlegging, utarbeidet læreren og jeg en grovplan for undervisningen som skulle gjennomføres. Denne grovplanen ble videre brukt og utarbeidet til mer detaljerte planer i del 3, hvor syklusen av evalueringer og justeringer skjer. Basert på de erfaringene læreren og jeg gjorde i gjennomføringen av undervisningen planla vi hvordan den neste undervisningsøkta skulle gjennomføres. Under hele prosjektperioden har informanten og jeg hatt en tett og åpen dialog om planlegging og gjennomføring der det har skjedd mange små og store justeringer av undervisningsopplegget underveis. Jeg vil komme nærmere inn på dette i avsnitt 3.5. Forskningsforløpet innebar også to intervjuer. Ett intervju før undervisningsperioden (intervju 1), og ett intervju etter endt periode (intervju 2).

3.3 Informant og forskerrolle

Kvalitative metoder og DBR åpner som nevnt for samarbeid mellom forsker og deltaker, og kunnskapen skapes i dette samspillet. DBR har likevel noen særtrekk, ved at forskeren er svært involvert i forskningsdesignet gjennom tett samarbeid. Det er av den grunn nødvendig å redegjøre for hvilke roller informanten og jeg hadde i prosjektet.

3.3.1 Utvelgelse av informant og lærerens rolle

Ved siden av studiene har jeg jobbet som vikar på en fådelt skole. Gjennom jobben kom jeg i kontakt med lærere som var svært positive til at jeg kunne gjennomføre prosjektet mitt på skolen deres. De så på det som en god mulighet til å komme enda bedre i gang med undervisning i programmering, selv om de allerede hadde arbeidet litt med dette tidligere. Det ble bestemt at jeg kunne gjennomføre prosjektet i 5.-7.trinn. Videre var det litt frem og tilbake

hvem som skulle bli den læreren jeg samarbeidet med i perioden. Hverdagen i skolen er ikke forutsigbar, og det gjenspeilte seg også i planleggingen av dette prosjektet. Det endte opp med at samarbeidet ble med en lærer som ikke underviser i matematikk i klassen per dags dato, men som har ansvar for programmeringen på skolen. Læreren har arbeidet som lærer i 30 år og er interessert i teknologi. I løpet av yrkeskarrieren har læreren vært IKT-veileder eller IKT-pedagog i 25 år. I dag underviser hun mest i spesialpedagogikk, men har også undervisningsfagene naturfag, kroppsøving og IKT. Læreren har tidligere undervist i matematikk, men etter de nye kravene om studiepoeng for å kunne undervise i basisfagene har hun foreløpig ikke utvidet sin formelle kompetanse i faget.

Utvalget i denne studien kan sies å være et bekvemmelighetsutvalg, på bakgrunn av at jeg forsker på en lærer jeg hadde kjennskap til fra før (Høgheim, 2020). Samtidig kan en argumentere for at utvelgelsen også er kriteriebasert. I en kriteriebasert utvelgelse må informanten oppfylle spesielle krav (Johannessen et al., 2021). Denne masteroppgaven har til hensikt å belyse lærerens erfaringer med bruk av ressurser fra Super:bit-prosjektet i undervisningen i matematikk. Et krav i utvelgelsen av min informant var at læreren hadde innsikt i dette. Informanten hadde deltatt på lærerkurset i regi av Vitensenteret og selv tatt i bruk den utdelte utstyrspakken i egen undervisning tidligere. Det ville vært fordelaktig om læreren var matematikklærer, da dette masterprosjektet er knyttet til matematikkdiridaktikk. På en annen side har læreren undervist i matematikk tidligere i yrkeskarrieren, og har dermed opparbeidet seg erfaring og kompetanse. På bakgrunn av det vurderte jeg at læreren, til tross for manglende formell kompetanse i matematikk per dags dato, tilfredsstilte kravene for utvalget i dette masterprosjektet.

Lærerens rolle i prosjektet har vært å planlegge, gjennomføre og evaluere undervisning, samt delta i to intervjuer. Gjennom samarbeidet med meg som forsker, har vi sammen utarbeidet undervisningsopplegg, men det er læreren som har hatt hovedansvaret og gjennomført oppleggene i praksis (med unntak av en økt, grunnet nødvendig fravær). Læreren har på den måten bidratt med sine erfaringer og kompetanse til å belyse den aktuelle problemstillingen, og studiet kan dermed knyttes til fenomenologien. Giorgi og Moustakas hevder at *«fenomenologiske studier beskriver den mening mennesker legger i en opplevelse knyttet til en erfaring av et fenomen»* (sitert i Postholm & Jacobsen, 2018, s. 76). I dette masterprosjektet er formålet å belyse erfaringer med bruk av ressurser fra Super:bit i matematikkundervisningen, og en fenomenologisk tilnærming var av den grunn et naturlig valg. Tilnærmingen baserer seg på antagelsen om at alle oppfatninger av verden er «sanne»

for den enkelte, og det interessante er å få økt forståelse og innsikt i den enkeltes oppfatning (Postholm & Jacobsen, 2018).

3.3.2 Forskerrollen

Høgheim (2020) skriver at forskerens subjektivitet er utfordrende i all forskning. Videre trekker han frem utfordringene ved at forskerens perspektiver kan komme tydelig frem i forskningen, og at forskeren kan la seg påvirke av nærheten til feltet og subjektet som forskes på. Jeg har som nevnt hatt en aktiv rolle gjennom hele perioden, gjennom et tett samarbeid med min informant. Dette har skapt en vanskelig balansegang i forskerrollen. Jeg har forsøkt å være bevisst min rolle ved å påvirke informanten i minst mulig grad. Masterprosjektets formål er å belyse lærerens erfaringer, og det har vært viktig at mine tanker og erfaringer ikke farger informanten for mye. Sett i lys av vårt tette samarbeid gjennom perioden har det vært utfordrende for meg som forsker å være helt objektiv. På en annen side gir den metodiske tilnærmingen DBR mulighet for det tette samarbeidet mellom meg som forsker og læreren. Gjennom samarbeidet skal man i fellesskap skape kunnskap som kan være nyttig i praksisfeltet (Anderson & Shattuck, 2012).

All forskning er fortolkende, fordi forskeren må bearbeide et datamateriale (Høgheim, 2020). I dette prosjektet har jeg som forsker tolket data som har blitt samlet inn gjennom intervjuer og observasjoner. I denne sammenhengen er hermeneutikk et nyttig perspektiv. Essensen i hermeneutikk er at man som forsker har en forforståelse av det man forsker på, og denne vil være med å forme hvordan man skaper kunnskapen i analysen (Høgheim, 2020). De tidligere møtene jeg har hatt med programmering i skolen og på høghskolen vil prege min forforståelse av temaet. I møte med ny kunnskap vil denne forståelsen utvikles, og det utvikles en større helhetsforståelse for temaet (Postholm & Jacobsen, 2018). Min forskerrolle må tas i betraktning med tanke på dette prosjektets resultater. Styrker og svakheter i forhold til min rolle i prosjektet vil jeg komme tilbake til i avsnitt 3.7, der prosjektets reliabilitet og validitet blir diskutert.

3.4 Datainnsamlingsverktøy

Postholm og Jacobsen (2018) trekker fram intervju og observasjon som gode datainnsamlingsverktøy i kvalitativ forskning, fordi de belyser hvordan mennesker konstruerer virkeligheten og samtidig får frem ulike nyanser og variasjoner. Jeg vil komme nærmere inn på intervju og observasjon som metode for datainnsamling, samt belyse fordeler og ulemper ved å kombinere disse metodene.

3.4.1 Intervju

I de kvalitative intervjuene som ble gjennomført i dette prosjektet var målet å få innsikt i informantens erfaringer rundt tematikken. Intervjuene som ble gjennomført var semistrukturerte. Et semistrukturert intervju kjennetegnes ved at det er rom for utforskning av uforutsette og relevante temaet underveis i intervjusituasjonen. Dette fører til at forskeren kan avdekke og undersøke nye ting under hele forskningsprosessen. I et semistrukturert intervju har forskeren et overordnet mål for intervjuet, og gjerne noen ferdig formulerte spørsmål eller temaer (Høgheim, 2020).

For å svare på problemstillingen ble det i forkant av intervjuene utarbeidet to intervjuguider, en til intervjuet før undervisningsperioden (intervju 1, se vedlegg 1) og en til intervjuet etter undervisningsperioden (intervju 2, se vedlegg 2). En intervjuguide er et manuskript, som strukturerer intervjuforløpet (Kvale & Brinkmann, 2015). Intervjuguidene som ble utarbeidet i forbindelse med dette prosjektet er forholdsvis fleksible. De inneholder flere temaer, som gir ordlyden for hva som skal dekkes i spørsmålene. I intervjuguide 1 er dette eksempelvis bakgrunn, programmering i ny læreplan, Super:bit, elevenes erfaringsbakgrunn, aldersblandet gruppe, planlegging og forventninger til undervisningsgjennomføringen. Under hvert tema er det ferdig formulerte spørsmål eller stikkord, som skal belyse den aktuelle tematikken. Disse spørsmålene eller stikkordene fungerer som mulige oppfølgingsspørsmål. De er altså ikke bindende, men tenkt som et støtteverktøy i intervjusituasjonene. Spørsmålene er fleksible, både med tanke på rekkefølge og det kan stilles flere oppfølgingsspørsmål dersom det er nødvendig. Oppfølgingsspørsmål stilles for at informanten skal utdype eller være mer konkret.

En fordel med semistrukturert intervju er at det gir mulighet for at informanten kan ta initiativ til å ta opp emner eller utdype temaer han eller hun tenker er aktuelle (Kvale & Brinkmann, 2015). Videre kan forskeren stille spørsmål med utgangspunkt i informantens beskrivelser og utsagn, som igjen gir informanten mulighet til å utdype og avklare. Kvale og Brinkmann (2015) legger vekt på at kunnskapen skapes sammen under et intervju, om mulighetene et semistrukturert intervju gir vil dermed kunne bidra til at det skapes en større forståelse rundt tematikken. Den fleksible strukturen er også fordelaktig ved at intervjusituasjonen vil oppleves som en samtale, og situasjonen vil dermed kunne oppleves tryggere for informanten (Kvale & Brinkmann, 2015).

På en annen side problematisere Kvaale og Brinkmann (2015) informantens påvirkning på intervjuet som en ulempe ved det semistrukturerte intervjuet. Den fleksible strukturen gjør at

intervjueren kan bli påvirket av informanten. Det er derfor viktig at intervjueren har en nøytral tilnærming i intervjusituasjonen, noe som kan være utfordrende når temaene er interessante og engasjerende. Den nøytrale tilnærmingen kan også være utfordrende i utformingen av oppfølgingsspørsmål under intervjuene. Spørsmålene skal være saklige og ikke lede eller påvirke svarene informanten gir (Tjora, 2021). For å sikre kvaliteten på spørsmålene var arbeidet med å utarbeide gode intervjuguider viktig, slik at jeg hadde et nyttig støtteverktøy under intervjuene.

Under intervjuene benyttet jeg meg av en lydopptaker, som ble lånt fra høgskolen. Lydopptakeren ble benyttet for å kunne fokusere på samtalen, uten å måtte notere det som ble sagt underveis. Dermed sikret jeg at informasjon ikke gikk tapt. I etterkant av intervjuene ble lydopptakene transkribert. Transkriberingen var et nødvendig arbeid for det analytiske arbeidet med datamaterialet. Årsaken til det er at det er lettere å bearbeide og analysere et skriftlig intervju, enn et intervju på en lydfil (Høgheim, 2020).

For å teste om intervjuguiden til intervjuet før undervisningsperioden (intervju 1) fungerte, og for at jeg skulle få prøve meg i en intervjusituasjon, ble det i forkant av datainnsamlingen gjennomført et pilotintervju. Fordelen med å gjennomføre et pilotintervju er at eventuelle misforståelser eller dårlig formulerte spørsmål kan justeres før datainnsamlingen.

Pilotintervjuet ble gjennomført med en lærer på den aktuelle skolen, som også hadde god kjennskap til elevgruppa hvor undervisningsopplegget skulle gjennomføres. Læreren hadde dermed gode forutsetninger for å kunne svare på spørsmålene i guiden. Under intervjuet ble det brukt lydopptaker for å teste og bli kjent med utstyret før intervjuet til datainnsamlingen. I etterkant av pilotintervjuet ble det gjort noen små justeringer på intervjuguiden. Disse justeringene ble gjort fordi det var behov for å være litt mer konkret på enkelte av spørsmålene, slik at intervjuobjektet og jeg forsto spørsmålene likt. Det ble ikke gjennomført pilotintervju før det avsluttende intervjuet. Denne avgjørelsen ble tatt på bakgrunn av at en lærer, som ikke deltok i undervisningsperioden, ikke ville ha forutsetninger til å kunne svare godt på spørsmålene i intervjuet. Med bakgrunn i det var vurderingen at et pilotintervjuet ikke ville vært en nyttig gjennomgang av spørsmålene.

3.4.2 Observasjon

Observasjonsdata i forbindelse med masterprosjektet har blitt samlet inn gjennom deltakelse i undervisningsopplegget. I rollen som deltakende observatør samlet jeg inn observasjonsdata i gjennomføringen av undervisningen. Ved å ta del i den naturlige settingen der handlingene utspilte seg, var målet å få innsikt i hvordan programmeringsaktivitetene fungerte i praksis.

Observasjonene bidrar med utfyllende informasjon til intervjuene (Postholm & Jacobsen, 2018). Observasjonen har funnet sted på den aktuelle skolen, hvor i alt fire undervisningsøkter har blitt observert.

Observasjoner kan gjennomføres på ulike måter, og to sentrale elementer en må tenke gjennom i den metodiske tilnærmingen er konteksten og forskerens rolle. Observasjonene i dette masterprosjektet har blitt gjennomført i en naturlig kontekst. Det vil si at observasjonssituasjonene ikke er stimulerte eller situerte, men foregår slik de vanligvis skjer (Høgheim, 2020). I mitt tilfelle var konteksten undervisning i klasserom. Observasjonsrollen var åpen og aktiv. At observasjonen var åpen, vil si at informanten kjente til at hun ble observert og hva som ble vektlagt i observasjonen. Som aktiv observatør, befant jeg meg i klasseromssituasjonen som ble observert, og deltok aktivt i aktivitetene. Fordelen med en aktiv observasjonsrolle er at en får et godt innblikk i situasjonene som observeres, og har mulighet til å stille spørsmål underveis (Høgheim, 2020). Basert på problemstillingen og forskningsdesignet i denne masteroppgaven var en aktiv rolle gunstig for å få innblikk i hvordan undervisningen fungerte, selv om en kan argumentere for at en mister noe av overblikket en ville hatt med en mer passiv rolle.

I undervisningssituasjonene observert jeg altså hvordan undervisningsopplegget vi hadde utarbeidet fungerte. Utfordringer elevene møtte på underveis ble registrert, både med tanke på oppgavene, utstyret og organiseringen, men også hva som fungerte godt. I tillegg ble hvilken rolle læreren hadde i undervisningen observert, med spesielt fokus på hvordan hun veiledet elevene. Dette fokuset kom på bakgrunn av at hun i intervju 1, som inngikk i del 2 av forskningsforløpet, hadde lagt vekt på at hun ønsket å ha en utforskende tilnærming i undervisningen. I dette vektla hun å ha en veiledende lærerrolle.

Observasjonene ble registrert ved at det ble skrevet feltnotater underveis og umiddelbart etter undervisningsøktene. Siden jeg hadde en aktiv rolle i undervisningen, var det ikke alltid like lett å få notert alle observasjonene underveis, og noe ble dermed skrevet ned i etterkant. Det er fordeler og ulemper med begge tilnærmingene. Tar en feltnotater underveis, kan en risikere å gå glipp av andre observasjoner da det er vanskelig å fokusere på to ting samtidig. På den andre siden kan en risikere å glemme viktige detaljer dersom en skriver alle feltnotatene i etterkant (Høgheim, 2020).

3.4.3 Kombinasjon av observasjon og intervju

Forskningsdesignet som har blitt benyttet stiller krav til at det blir samlet inn data i de ulike delene i forskningsforløpet. For å samle inn data i alle delene ble datainnsamlingsverktøyene intervju og observasjon kombinert, såkalt metodetriangulering. Postholm og Jacobsen (2018) trekker frem at metodetriangulering er spesielt gunstig når datainnsamlinger strekker seg over tid. Forskningsforløpet i dette prosjektet strekker seg over tid, og består av flere økter. Som forsker har jeg dermed mulighet til å dra nytte av metodetrianguleringen i datainnsamlingen. Gjennom det innledende intervjuet forskningsprosessen startet med, ble det innhentet en del bakgrunnsinformasjon. Denne informasjonen var nyttig i forberedelsene til de påfølgende observasjonene og planleggingen av undervisningen. Eksempelvis slik jeg nevnte i avsnitt 3.4.2, ved at lærerens fokus på en veiledende lærerrolle i det første intervjuet førte til et fokus på dette i observasjonene. Videre ga observasjonene en føring på hva som ble aktuelle tematikker å ta opp i det avsluttende intervjuet. Observasjonene ble på den måten gjennomført for å utforske, bekrefte eller eksemplifisere noe som var blitt sagt i intervjuene. I det avsluttende intervjuet fikk jeg kjennskap til informantens erfaringer knyttet til det som var blitt observert i undervisningen. På denne måten, vet at forskerens antakelser møter datamaterialet, kan metodetrianguleringen bidra til at helhetsforståelse for det som studeres utvikles (Postholm & Jacobsen, 2018).

3.5 Planlegging, gjennomføring, evaluering og justering av undervisningsopplegg

3.5.1 Planleggingsmøte

I forkant av undervisningsperioden avtalte læreren og jeg å ha et planleggingsmøte. Formålet med planleggingsmøtet var å lage en grovplan over undervisningsopplegget som skulle gjennomføres, og inngikk i del 1 i forskningsforløpet. Møtet ble gjennomført uken før den første undervisningsøkta. I forkant av møtet hadde både informanten og jeg satt oss inn i ressurser fra Super:bit-prosjektet, og gjort oss opp noen tanker om hvordan vi ønsket å gjennomføre perioden. På planleggingsmøtet satte vi to læringsmål for perioden. Læringsmålene er lagd med utgangspunkt i kompetansemålene som omhandler programmering i LK20 (tidligere nevnt i avsnitt 2.1.2):

- Elevene skal gjennom arbeidet med programmering lære begrepene variabler, løkker, vilkår og funksjoner, og forstå hvordan de bruker dette i praksis når de programmerer.
- Elevene skal bruke programmering til å utforske geometriske figurer. De skal reflektere over egenskapene til figurene, og regne ut omkrets og areal.

I planleggingen tok vi hensyn til de erfaringene elevene hadde med programmering fra før. Undervisningen ble gjennomført i en aldersblandet 5.-7.klasse, noe som medførte at det var variasjon i hvor mye de ulike klassene hadde arbeidet med programmering på tidligere årstrinn. Elevene som går i 6. og 7.klasse hadde arbeidet med deler av Super:bit-prosjektet tidligere, mens 5.klassingene hadde ikke like mye erfaring. Gjennom Super:bit-prosjektet hadde elevene som går i 6. og 7.klasse hatt besøk av Vitensenteret, og fått en god innføring i mange av oppgavene. De hadde også arbeidet med noen andre oppgaver i klassen.

5.klassingene hadde ikke arbeidet med Super:bit i like stor grad, men også disse elevene hadde programmert micro:bit og prøvd å kjøre BitBot tidligere. Dette måtte vi ta hensyn til i planleggingen, blant annet ved å variere de eksisterende oppgavene og velge ut oppgaver 6. og 7.trinn ikke hadde arbeidet med tidligere. I tillegg fikk denne ulike erfaringsbakgrunnen konsekvenser for selve organiseringen. Vi planla at elevene skulle arbeide sammen med læringspartnere i perioden. Denne beslutningen ble tatt på bakgrunn av at vi ønsket at elevene skulle dra nytte av hverandre i undervisningen, slik som forskning viser at arbeid i læringspar kan bidra positivt til (Leuven & Rønning, 2016). Elevene på den fådelte skolen er svært vant til denne organiseringsformen. I perioden ville vi legge opp til undervisning der elevene arbeidet sammen med en eller to læringspartnere, hvor minst en var fra 6. eller 7.klasse, og dermed kunne være en ressurs for 5.klassingene. På den måten kunne de bruke sin kompetanse til å hjelpe elevene med mindre kompetanse på fagfeltet. Dette førte til at det ble totalt 6 grupper (2 elever per gruppe+ en gruppe med 3 elever).

Informanten og jeg hadde begge gjort oss opp noen tanker, prøvd oppgaver og satt opp et forslag til gjennomføring. På planleggingsmøtet planla vi den første undervisningsøkta, og lagde en grovplan for økt 2. Vi bestemte oss for å vente med planleggingen av den tredje og siste økta, da vi ønsket å vurdere nivået på elevene i de to første øktene, og dermed vurdere hvilke utfordringer elevene kunne mestre. Vi ønsket å vurdere om elevene eksempelvis var klare for å programmere med en sender og mottaker, eller om det ble for sammensatt. Det var uproblematisk å gjøre denne vurderingen, og deretter planlegge hva økta skulle inneholde senere. I forskningsforløpet til dette prosjektet er det å gjøre justeringer ut fra evalueringene i de gjennomførte øktene essensen i del 3. Hovedmålet med planleggingsmøtet i del 1 var å lage en grovplan over undervisningsopplegget, mens detaljplanleggingen skjer i del 3 av forskningsforløpet (med unntak av økt 1, som ble planlagt på dette møtet).

Jeg vil videre beskrive hvordan gjennomføringen, evalueringen og justeringen av øktene foregikk, altså del 3 i forskningsforløpet. Øktplanene informanten og jeg lagde for hver økt vil

bli presentert, med tilhørende beskrivelser av hvordan vi arbeidet for å utarbeide disse. Det var satt av tre skoletimer per økt, hvor en skoletime varer i 45 min. Klassen begynner hver dag med faste oppstarts-rutiner, så tatt det i betraktning beregnet vi omtrent 2 klokketimer per økt, som kunne brukes til programmeringsopplegget.

3.5.2 Økt 1

Målet med den første undervisningsøkta var at elevene skulle få en introduksjon til, eller repetisjon av programmering. Gjennom ulike oppgaver ønsket vi at elevene skulle lære hvor viktig det er å være nøyaktig og gi gode, presise kommandoer. Bakgrunnen for dette fokuset var at informantene fortalte at hun opplevde elevene som utålmodige, og at dette skaper utfordringer i forhold til nøyaktighet. I arbeidet med programmering må elevene være presise for å få datamaskinen til å gjøre det de ønsker. Aktivitetene i den første økta foregikk analogt, med unntak av spillet elevene skulle arbeide med avslutningsvis.

Tabell 4: Plan for økt 1

Tid	Hva	Hvordan	Hvorfor
30 min	Tegne etter instruksjon fra lærer.	Klassen er «datamaskiner» og tegner etter de instruksjonene læreren gir. Læreren gir ikke flere forklaringer, men kan gjenta dersom elevene trenger å høre instruksjonen flere ganger. Individuelt arbeid.	-Lære at det er viktig å gi og ta i mot korte, presise forklaringer -Forstå at instruksjoner kan være tvetydige
40 min	Tegne etter instruksjoner fra læringspartner.	Elevene går sammen i par med læringspartner. Parene setter seg rygg mot rygg. Den ene eleven får i oppgave å gi instruksjoner, mens den andre skal tegne etter instruksjonene. Tegningene får de utlevert fra lærer. Elevene bytter oppgaver underveis, og tegningene blir mer og mer utfordrende.	-Erfare hvor nøye man må være når man instruerer/programmerer -Øve på å forklare ting på ulike måter -Bruke matematiske begreper når de forklarer (geometriske figurer, lengder, forhold...)
20 min	Kompis-programmering	Elevene blir delt inn i 3 grupper, og hver gruppe får utlevert en «Programmeringbane» og 8 programmeringskort. Målet er at elevene sammen skal programmere hverandre, ved hjelp av programmeringskortene, til å gå fra 0 til 10 på banen. Etter at gruppene har planlagt, laget og testet sin løsning skal de presentere for klassen. Utfordring: ta vekk «gå ett skritt frem»	-Lage en algoritme ved hjelp av pilene og kommandoene på programmeringskortene. -Forstå hva de ulike kommandoene gjør, og klare å sette dem sammen slik at «roboten» utfører oppgaven en ønsker
30 min	Spill på data (https://www.vitensenter.no/introlop/index.html)	Elevene arbeider sammen med læringspartner på en datamaskin. Gjennomfører de fire delene i spillet: basis, inndata, løkke og logikk.	-Lære sentrale begreper (inndata, løkker, logikk) -Bli introdusert for Makecode (programmet som brukes for å programmere micro:bit).

I etterkant av økt 1 evaluerte læreren og jeg den gjennomførte undervisningsøkta, og planla neste økt ved å gjøre justeringer ut fra de erfaringene vi hadde gjort oss (del 3 i forskningsforløpet). Gjennomføringen av de planlagte aktivitetene i økt 1 tok lengre tid enn vi hadde antatt, og elevene rakk ikke å bli ferdige med «spill på data». På bakgrunn av at elevene ikke fikk gjennomført alle oppgavene vi hadde planlagt i økt 1, ble planen for økt 2 justert. Vi ønsket at elevene skulle arbeide videre med «spill på data», da vi opplevde dette spillet som en god innføring i grunnleggende programmering. Gjennom spillet ble elevene introdusert for sentrale begreper, og lærte hvordan de skulle programmere micro:biten i Makecode. Vi valgte å justere grovplanen vi hadde utarbeidet for økt 2 ved å la elevene arbeide videre med dette. Etter økt 1 evaluerte vi at organiseringen med læringspar fungerte godt, og vi gjorde av den grunn ingen justeringer med tanke på organiseringen.

3.5.3 Økt 2

I økt 2 begynte vi med en liten repetisjon fra forrige økt, før elevene arbeidet med å programmere gjennom spillet. Målet for økta var å bli kjent med og repetere hvordan de programmerte micro:biten til å gjøre ulike oppgaver. Gjennom programmeringen ville de arbeide med begrepene variabler, løkker, vilkår og funksjoner, ved at de brukte dem i praksis når de programmerte. I denne økta var også geometri et fokus, ved at elevene brukte programmering til å arbeide med ulike geometriske figurer. Ved at elevene skulle programmere ulike figurer var de nødt til å reflektere rundt egenskapene til figurene. Disse refleksjonene var nødvendig for at elevene skulle mestre å programmere de geometriske figurene korrekt. Eksempelvis måtte elevene vite at et kvadrat er en firkant med like lange sider og fire rette vinkler (Vatne, 2021). I tillegg skulle elevene jobbe med å regne ut omkrets og areal av figurene.

Tabell 5: Plan for økt 2

Tid	Hva	Hvordan	Hvorfor
10 min	Powerpoint med analog programmering	Læreren programmerer elevene. Bruker powerpoint fra Super:bit-oppgavet.	-Repetisjon fra forrige økt -Se hvordan blokkprogrammering kan se ut
35 min	Spill på data (https://www.vitensenter.no/introlop/index.html)	Elevene arbeider sammen med læringspartner på en datamaskin. Gjennomfører de fire delene i spillet: basis, inndata, løkke og logikk.	-Få en innføring i helt grunnleggende programmering -Lære sentrale begreper (inndata, løkker, logikk) -Bli introdusert for Makecode (programmet som brukes for å programmere micro:bit)
15 min	Kom i gang med microbit og BitBot	Utforskende læring. Elevene får prøve å programmere micro:biten til å gjøre ulike oppgaver. <ul style="list-style-type: none"> - Lage program som får BitBoten til å kjøre 1 meter Utfordring: snu og kjøre tilbake igjen - Programmerer BitBoten til å kjøre gjennom en løype 	- Arbeide med begrepene og funksjonene til løkker, uendelige løkker, vilkår og variabler -Lære hvordan de programmerer microbiten i Makecode og hvordan de overfører kodene til micro:biten og BitBot
60 min	Utforske geometriske figurer med programmering	Elevene skal programmere BitBoten til å lage ulike geometriske figurer mens de kjører. Legger papir på gulvet og fester en penn i pennholder på BitBoten, slik at den kan tegne mens den kjører. Elevene kan dermed måle sider, vinkler og regne ut omkrets og areal av figurene de programmerer.	-Bruke kunnskapen de har og reflektere over egenskapene til geometriske figurer når de programmerer -Jobbe med å regne ut omkrets og areal

Etter økt 2 satt vi igjen med samme erfaring som etter forrige økt: programmering tar tid. Vi fikk ikke gjennomført alle de tenkte oppgavene denne gangen heller. Elevene arbeidet i ulikt tempo med spillet og oppgavene det innebar, noe som førte til at det var variasjon i hvor mye de ulike læringsparene rakk å gjennomføre. Noen av parene var svært utforskende og ønsket å prøve alt som ble lagt fram for dem i spillet, mens andre gikk litt raskere til verks. Vi lot elevene arbeide og teste ut i det tempoet de ønsket, da vi opplevde at parene arbeidet godt med det de holdt på med. De læringsparene som ble ferdige, fikk begynne å prøve seg med BitBoten. Ett av læringsparene mestret dette godt, og løste raskt oppgavene der de skulle få BitBoten til å kjøre en meter og gjennom en løype. Paret fikk derfor prøve seg på en mer avansert oppgave hvor de skulle programmere en radiostyrt bil. På den måten var det lett å differensiere, slik at alle elevene fikk arbeide med oppgaver som passet deres ferdighetsnivå. Vi besluttet at ingen av læringsparene fikk begynne på oppgaven «Utforske geometriske figurer med programmering», da vi ønsket at alle elevene skulle delta på dette.

I etterkant av økt 2 jobbet jeg som vikar i den aktuelle klassen. Læreren og jeg bestemte at jeg kunne gjennomføre den delen av økt 2 vi ikke rakk med elevene. Oppgaven «Utforske geometriske figurer med programmering» ble derfor gjennomført av meg, uten at læreren var til stede. Dette valget ble tatt på grunnlag av at vi ønsket at elevene skulle få gjennomføre dette opplegget med tanke på deres læringsutbytte. Skolehverdagen er hektisk, og en del av hverdagen er at en må gjøre endringer og vurderinger underveis. Vi vurderte at det var viktig at elevene fikk gjennomføre oppgavene, selv om læreren ikke hadde mulighet til å være med. I etterkant av økta fikk læreren oppdatering på hvordan undervisningsøkta hadde fungert, og på bakgrunn av det gjorde vi noen justeringer før den siste økta. Disse justeringene handlet om organisering av læringspar, da det ikke var alle læringsparene som fungerte like godt. Noen av elevene uttrykte ønske om å få være med og bestemme hvem de skulle samarbeide med. Elevene ble også spurt om hva de ønsket å arbeide mer med, og mange svarte «bilene». I samarbeid med læreren planla vi at elevene skulle få mer medbestemmelse i den siste undervisningsøkta, ved at de selv kunne være med å påvirke hva de ville arbeide videre med. En fordel ved dette er at det vil være enklere å differensiere, slik at alle elevene får arbeide med oppgaver som passer deres ferdighetsnivå.

3.5.4 Økt 3

I økt 3 ble elevene satt sammen i grupper på 3-4 elever. Grunnet litt fravær var det totalt tre grupper. På bakgrunn av evalueringene fra økt 2 var gruppene satt sammen slik at elever på omtrent samme ferdighetsnivå fikk arbeide sammen. Gruppene fikk selv være med å velge

hvilke programmeringsoppgaver de ønsket å løse, ut fra oppgavene som ligger under «etter Super:bit-oppdraget» og «bonusoppdrag» i Super:bit skole.

Tabell 6: Plan for økt 3

Tid	Hva	Hvordan	Hvorfor
120 min	Arbeide med oppgavene som ligger under «etter Super:bit-oppdraget» og «bonusoppdrag».	Organiserte elevene i grupper på 3-4 elever. Gruppene fikk selv velge hvilke oppgaver de ønsket å løse.	-Repetisjon fra de tidligere øktene -Gi elevene medbestemmelse og mulighet til å velge utfordringer som passer deres ferdighetsnivå

I selve økta valgte alle gruppene oppgaven hvor de skulle programmere en radiostyrt bil. Ved hjelp av en video produsert av NRK fikk de stegvis veiledning på hvordan oppgaven kunne løses. To av gruppene mestret dette godt, mens det ble utfordrende for den siste gruppa. Denne gruppa fikk arbeide med en oppgave som passet deres ferdighetsnivå bedre. Læreren og jeg lot dem bruke Smart-by matta, og ga dem i oppgave å programmere BitBoten til å kjøre gjennom byen. I slutten av økta bestemte vi oss for å gjennomføre en konkurranse med de radiostyrte bilene elevene hadde programmert. Dette ble gjort ved at vi festet en ballong og en blyant med en nål på dem. Målet med konkurransen var at elevene skulle styre bilene, slik at de sprakk hverandres ballonger. Også elevene som ikke hadde mestret å programmere den radiostyrte bilen fikk være med på denne aktiviteten, ved at de fikk låne programmeringskodene til medelever.

3.6 Analyse av data

I etterkant av datainnsamlingen må dataene analyseres. I dette prosjektet har det blitt samlet inn data i form av observasjoner i undervisningssituasjon, samt to intervjuer. Det har altså blitt samlet inn en del tekst, som gjør at det er nødvendig med et analysearbeid for å redusere datamengden. Dette gjøres ved å fjerne de delene av datamaterialet som ikke er relevant for å besvare den aktuelle problemstillingen. Videre må dataene som er verdt en tolkning systematiseres og analyseres (Larsen, 2017).

Jeg har valgt å bruke Høgheim (2020) sin framgangsmåte for analyse av kvalitativ data. Modellen han presenterer er basert på en innholdsanalyse, og består av fem faser. Jeg vil videre beskrive hvordan jeg har arbeidet med analyse av dataene i de ulike fasene i modellen.

3.6.1 Fase 1

Målet med fase 1 er å bli kjent med, lese og sammenfatte data (Høgheim, 2020). I mitt prosjekt handlet denne fasen om å transkribere intervjuene og strukturere observasjonsnotatene på en god måte. I etterkant av hver undervisningsøkt ble observasjonsnotatene renskrevet i Word. Dette ble gjort kort tid etter observasjonene, for å sikre at jeg husket meningen bak notatene. På den måten kan man sikre at informasjon ikke går tapt eller at en legger til en ny tolkning (Tjora, 2021).

Ved at intervjuene transkriberes til skriftlig form blir de bedre egnet for analyse. Jeg transkriberte intervjuene i Word. Data i tekstform er mer oversiktlig, og på den måten enklere å strukturere og analysere (Kvale & Brinkmann, 2015). Fase 1 er viktig for å bygge et godt grunnlag for den videre analysen. En vesentlig del av fasen var dermed å bruke tid på å bli kjent med dataene, ved å lese gjennom dem flere ganger, notere antakelser og få ideer for videre arbeid.

3.6.2 Fase 2

I fase 2 er målet å lage koder for å organisere dataene. På den måten skal en redusere rådataene til meningsfulle deler (Høgheim, 2020). Jeg valgte å bruke en abduktiv form for koding. Det vil si en kombinasjon av en deduktiv og en induktiv tilnærming. Den deduktive tilnærmingen er teoristyrte, ved at en baserer kodingen på temaer fra litteraturen (Høgheim, 2020). Dette ble gjort ved at det ble hentet ut begreper fra det teoretiske rammeverket. Eksempelvis ble nøkkelbegrepene i Utdanningsdirektoratet sin modell av Den algoritmiske tenkeren: fikle, skape, feilsøke, holde ut og samarbeide brukt, for å identifisere kjennetegn ved algoritmiske arbeidsmåter i undervisningen. Den induktive kodingen kjennetegnes ved at forskeren tar utgangspunkt i empirien, og baserer kodingen på det som legges merke til i datamaterialet (Høgheim, 2020). Et eksempel på dette er at læreren hadde stort fokus på lærerkompetanse i forhold til programmeringsundervisningen, noe som førte til at det ble laget en kodespesifisering som plukket opp dette aspektet i datamaterialet.

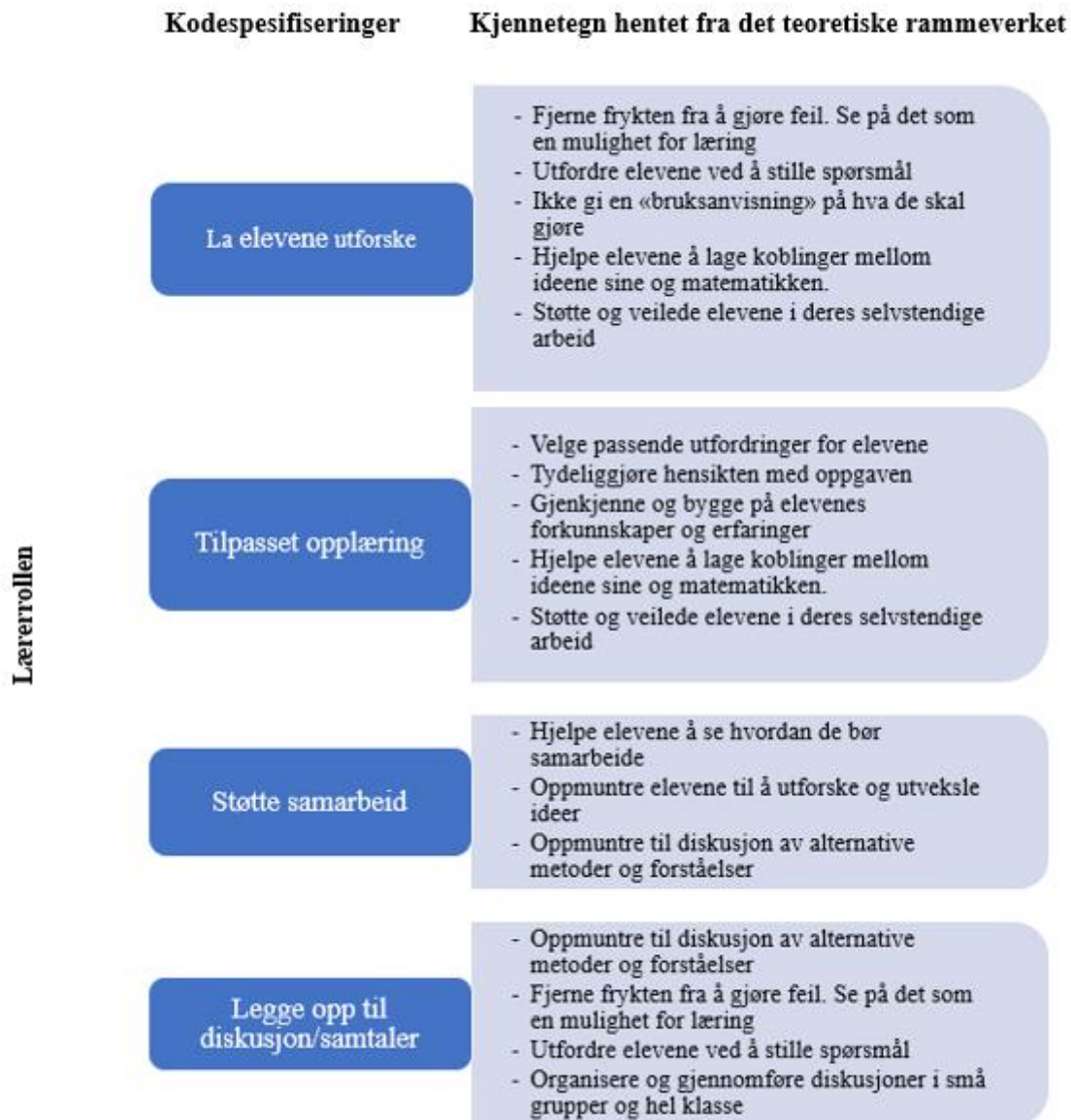
Tabell 7 viser kodene som ble benyttet i kodeprosessen, og tilhørende spesifikasjoner for å belyse hva som ble vektlagt i de ulike kodene. Disse spesifikasjonene bygger på det teoretiske rammeverket som ble presentert i kapittel 2, i tillegg til at det som nevnt ble lagt noen induktive koder basert på aspekter som ble lagt merke til i datamaterialet.

Tabell 7: Samlet oversikt over kodene som ble brukt i analysen

Koder	Spesifisering av kodene
Ønsket utbytte	<ul style="list-style-type: none"> - Lete etter mønster, finne sammenhenger og diskutere - Strategier og fremgangsmåter - Vurdere og analysere
Klasseromskultur	<ul style="list-style-type: none"> - Læring i en sosial prosess - Dialog - Rom for å gjøre feil
Lærerrolle	<ul style="list-style-type: none"> - Tilpasset opplæring - Støtte samarbeid - Legge opp til diskusjon/samtale - La elevene utforske - Kompetansen til lærere
Fikle	<ul style="list-style-type: none"> - Utforske og eksperimentere
Feilsøke	<ul style="list-style-type: none"> - Oppdage og rette feil
Holde ut	<ul style="list-style-type: none"> - Fortsette og prøve igjen
Samarbeid	<ul style="list-style-type: none"> - Dele og arbeide sammen
Aldersblanding	<ul style="list-style-type: none"> - Samarbeid mellom elever på ulikt faglig nivå og med ulik alder - utfordringer og muligheter
Læringspar	<ul style="list-style-type: none"> - Dra nytte av hverandre, ha dialog og diskusjon rundt utfordringene de møter, reflektere og sammen vurdere arbeidet de har utført - Samarbeid, kommunikasjon, forklaring, diskusjon og refleksjon
Utstyr	<ul style="list-style-type: none"> - Micro:bit - BitBot - Tekniske utfordringer
Oppgaver	<ul style="list-style-type: none"> - Blokkprogrammering - Analog programmering
Matematikk	<ul style="list-style-type: none"> - Elevene skal gjennom arbeidet med programmering lære begrepene variabler, løkker, vilkår og funksjoner, og forstå hvordan de bruker dette i praksis når de programmerer. - Elevene skal bruke programmering til å utforske geometriske figurer. De skal reflektere over egenskapene til figurene, og regne ut omkrets og areal.

Figur 10 illustrerer hvordan kodespesifiseringene for koden lærerrollen ble utarbeidet. Med utgangspunkt i teorien, som ble beskrevet i avsnitt 2.5.1, ble det utarbeidet fire kodespesifiseringer (tilpasset opplæring, støtte samarbeid, legge opp til diskusjon/samtale og

la elevene utforske), som belyser kjennetegn ved lærerrollen i IBL. Formålene med disse spesifiseringene var at de skulle hjelpe meg i analysen, ved å synliggjøre hva jeg skulle se etter i datamaterialet.

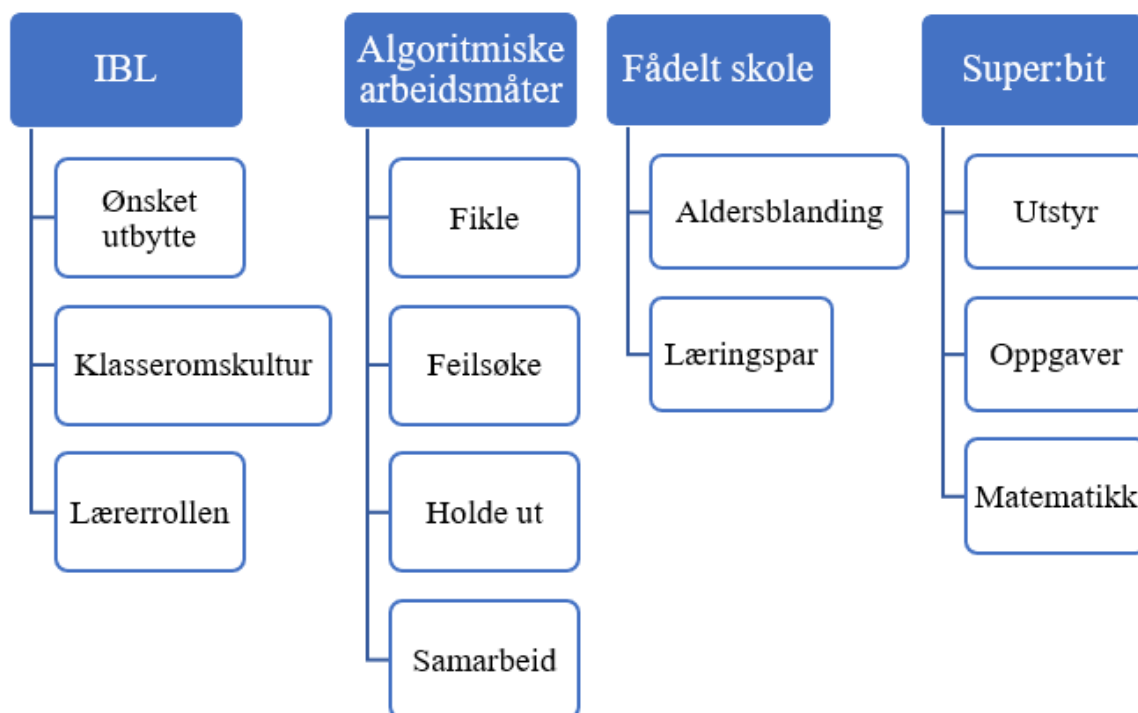


Figur 10: Spesifiseringer av koden lærerrollen

Kodeprosessessen gikk ut på at dataene ble organisert ved å markere deler av dem, og gi dem en kode som representerte den spesifikke delen. Jeg brukte kommentar- og markeringsfunksjonen i Word.

3.6.3 Fase 3

Kodene fra fase 2 ble videre brukt i neste fase, hvor målet var å lage ulike kategorier. En kategori vil si en samling av lignende eller overlappende koder (Høgheim, 2020).



Figur 11: Oversikt over kategoriene i analysen

Figur 11 illustrerer de fire kategoriene som ble utarbeidet i fase 3. Kategoriene inneholder, slik figuren illustrerer, kodene fra fase 2. De fleste kodene var enkle å kategorisere, eksempelvis hadde kodene fikle, skape, feilsøke, holde ut og samarbeide en naturlig tilhørighet i kategorien algoritmske arbeidsmåter. I arbeidet med denne fasen var den største overveielser om matematikk skulle få en egen kategori eller ikke. Valget falt tilslutt på å plassere koden matematikk i kategorien Super:bit. Denne avgjørelsen ble tatt på bakgrunn av at jeg i arbeidet med dette prosjektet har sett spesielt på matematikken i sammenheng med arbeidet med Super:bit. Oppgavene elevene arbeidet med i undervisningsperioden er hentet fra ressurser til Super:bit, og det var derfor hensiktsmessig å se på hvordan matematikken ble synliggjort i dette arbeidet.

3.6.4 Fase 4

Høgheim (2020) skriver at målet med den fjerde fasen er å belyse det dataene inneholder av meninger, betydninger og oppfatninger. I dette prosjektet har det blitt gjennomført en tematisk innholdsanalyse. Dette foregikk ved at det ble lagd fire tabeller, en til hver av de nevnte fire kategoriene som ble utarbeidet i fase 3. Tabell 8 viser en av disse fire tabellene. Tabellen inneholder kodene i kategorien algoritmske arbeidsmåter, og ble brukt ved at det kodede datamaterialet ble systematisert i tabellen. Jeg valgte å skille mellom data fra intervju og

observasjon, og mellom de ulike øktene. I tillegg ble en kolonne med navn «uspesifisert» lagt til. Dette ble gjort fordi det ikke var mulig å tidsbestemme alt av datamaterialet fra intervjuene til en av øktene.

Tabell 8: Tabell for å systematisere datamaterialet i analysen

Algoritmiske arbeidsmåter		Økt 1	Økt 2	Økt 3	Uspesifisert
Fikle	Intervju				
	Observasjon				
Feilsøke	Intervju				
	Observasjon				
Holde ut	Intervju				
	Observasjon				
Samarbeide	Intervju				
	Observasjon				

Høgheim (2020) skriver at denne formen for analyse er gunstig dersom man ønsker å avdekke forståelser ved et fenomen. I denne oppgaven er det nettopp forståelsen, eller de erfaringene læreren gjorde seg i programmeringsundervisningen som skal belyses. Som forsker har jeg systematisert sitater fra intervju og observasjoner, som er verdt en tolkning, i de ulike tabellene.

Tabell 9 viser et utdrag fra en av tabellene jeg arbeidet med i analysen. Utdraget viser noe av dataene som synliggjør den algoritmiske arbeidsmåten holde ut.

Tabell 9: Utdrag fra hvordan datamateriale ble systematisert i analysen

Algoritmiske arbeidsmåter		Økt 1	Økt 2	Økt 3	Uspesifisert
Holde ut	Intervju	Hvis man ikke får til ting med en gang, så gir de litt opp.		«Si svaret også kan jeg bare gjøre». «Kom og vis meg koden også kan jeg bare». «Nei men du skal prøve, sant».	De som er vant til å få svaret, de skal ha et svar og det er to streker under det svaret. Det er ikke noen andre muligheter.
	Observasjon			Elevene fulgte instruksjer på video. Noen av gruppene gjorde ikke de innledende oppgavene nøye nok, det førte til at det ble vanskelig å få til resten. Elevene ble utålmodige og ga litt opp.	

3.6.5 Fase 5

I den femte og siste fasen er målet å finne ut hvilke slutninger en kan trekke på bakgrunn av datamaterialet (Høgheim, 2020). Denne avgjørelsen må tas på bakgrunn av det teoretiske rammeverket, metoden, dataene og analysen. Gjennom tolkning av datamaterialet har jeg i denne fasen kommet frem til funnene i denne masteroppgaven. De fire nevnte tabellene fra fase 4 ble brukt aktivt i denne prosessen. Funnene blir presentert i fire delkapitler i kapittel 4, før de oppsummeres i avsnitt 4.5. Avsnitt 4.1 beskriver generelt hvilke holdninger læreren har til prosjektet Super:bit og koblingen mot matematikkfaget. Det legges vekt på hvordan

læreren opplever at programmeringsaktivitetene kan kobles opp mot faget og hvilke muligheter og utfordringer hun opplever ved dette. I avsnitt 4.2, algoritmiske arbeidsmåter, blir kjennetegnene ved algoritmiske arbeidsmåter som har blitt identifisert i datamaterialet synliggjort. I de to siste avsnittene, lærerrollen og organisering av undervisning, belyses lærerrollen med utgangspunkt i kjennetegn fra IBL og aspektet ved at dette er en fådelt skole.

3.7 Reliabilitet og validitet

I arbeidet med kvalitativ forskning er det viktig å være bevisst at alle metoder har styrker og svakheter, og at alle forskere vil vektlegge ulike sider ved de samme fenomenene (Postholm & Jacobsen, 2018). På bakgrunn av dette er det nødvendig å redegjøre for forskningens reliabilitet og validitet. Reliabilitet handler om hvor pålitelig dataene vi samler inn er, mens validiteten til en studie handler om å vurdere sannheten (gyldigheten) til en slutning (Shadish et al., 2002).

For å styrke reliabiliteten i dette masterprosjektet har det blitt vektlagt å gi detaljerte beskrivelser av forskningsforløpet og det analytiske rammeverket. Dette er et nødvendig grep fordi min forforståelse vil ha innvirkning på hvordan datamaterialet tolkes. I kvalitativ forskning er det nødvendig at leseren får et godt innblikk i datamaterialet og metodene som har blitt benyttet, for å styrke påliteligheten til resultatene (Johannessen et al., 2021).

Den indre validiteten til et studie handler om hvorvidt funnene gir svar på forskningsspørsmålet vi stiller (Postholm & Jacobsen, 2018; Tjora, 2021). Et grep man kan gjøre for å styrke den indre validiteten er å bruke metodetriangulering (Johannessen et al., 2021). Metodetriangulering bidrar til å styrke den indre validiteten ved at de ulike metodene beskriver virkeligheten fra flere ulike vinkler, og dermed skaper et mer helhetlig bilde av en kompleks og sammensatt virkelighet. Samtidig er det også svakheter ved å kombinere ulike metoder når en arbeider innenfor en begrenset tidsperiode slik som i dette masterprosjektet. Postholm og Jacobsen (2018) skriver at masterstudenter kan miste fokuset i sitt prosjekt, da innhenting av dataene krever mye tid og ressurser. I mitt prosjekt, hvor DBR ble brukt som design, opplever jeg imidlertid at data fra intervju og observasjoner var nødvendig for å innhente data fra alle delene i forskningsforløpet. Jeg har tidligere, i avsnitt 3.4.3, beskrevet hvordan jeg utnyttet kombinasjonen av observasjon og intervju for å skape en helhetsforståelse. I prosjektet er observasjonene brukt som et supplement, for å støtte opp eller være en motvekt, til de erfaringene læreren la frem i intervjuene.

Ytre validitet handler om i hvilken grad funnene i studien kan overføres eller generaliseres. Denne masteroppgaven inneholder kun en lærers erfaringer rundt bruk av ressurser fra Super:bit i matematikkundervisningen. Grunnet studiets begrensede utvalg kan jeg ikke argumentere for at funnene vil være representative for alle lærere i Norge. På en annen side skriver Postholm og Jacobsen (2018) at generalisering i kvalitative studier er knyttet til gjenkjennbarheten i beskrivelsene, såkalt naturalistisk generalisering. Dette omhandler hvorvidt leseren kan overføre det til sin situasjon. Det er grunn til å tro at andre lærere gjennom å lese om erfaringene i denne oppgaven vil kunne overføre dette til sin situasjon, og på den måten vil det kunne være gjeldende for flere lærere. I denne oppgaven har detaljerte beskrivelser av forskningsprosessen vært et viktig grep for å styrke validiteten, slik at leseren selv kan vurdere overførbarheten til andre kontekster (Johannessen et al., 2021).

Et særpreg ved forskningsdesignet DBR er relasjonen mellom forsker og deltaker. I løpet av arbeidet med dette masterprosjektet har min informant og jeg hatt ulike roller og relasjoner. Dette fører med seg noen utfordringer og er dermed også nødvendig å diskutere. For det første har vi gjennom planleggingen i prosjektet hatt en samarbeidsrelasjon. Denne relasjonen har sin fordel i at valideringen til prosjektet har blitt styrket ved at jeg kontinuerlig har fått tilbakemeldinger på hvordan ulike situasjoner forstås ut fra hennes perspektiv. På den måten har samarbeidet bidratt til at jeg i større grad kunne kontrollsjekke mine tolkninger (Johannessen et al., 2021). På den andre siden skaper også dette samarbeidet utfordringer ved at min teoribakgrunn, erfaringer, holdninger og meninger kan påvirke læreren. Selv om DBR åpner for samarbeid, har jeg i rollen som forsker vært bevisst på å ikke påvirke læreren for mye.

For det andre har jeg hatt rolle som observatør, mens læreren har blitt observert. I rollen som observatør observerer jeg et undervisningsopplegg jeg selv har vært med å planlegge, og det kan diskuteres om jeg er objektiv nok. Samtidig gjør deltakelsen i planleggingen og samarbeidet at jeg har en annen innsikt i det som skal skje enn det en utenforstående ville hatt. Observasjonene i denne studien er ment som et supplement til intervjuene, i og med at det er lærerens erfaringer som skal belyses. Det er ikke mulig å observere erfaringene til læreren, men jeg kan observere situasjoner som underbygger de erfaringene læreren trekker frem.

Sist, men ikke minst, har vi en forsker-informantrelasjon. En av utfordringene ved kvalitativ forskning er at forskeren samler, analyserer og tolker datagrunnlaget. Resultatene kan dermed være preget av mine verdier og holdninger. Gjennom en bevisst holdning til dette har jeg forsøkt å legge vekt på det teoretiske rammeverket, og holdt mine forståelser i bakgrunnen.

Samtidig åpner forskningsforløpet opp for at stemmen min ikke skal gjemmes helt bort, noe som igjen skaper en utfordrende balansegang i forskerrollen.

3.8 Etikk

Allerede i planleggingen startet de etiske betraktningene rundt dette masterprosjektet. Tidlig i prosessen ble det sendt søknad om godkjenning av prosjektet til NSD (se vedlegg 3). I denne prosessen utarbeidet jeg et informasjonsskriv og en samtykkeerklæring (se vedlegg 4) ut fra malen på NSD sine hjemmesider, som senere ble sendt til og underskrevet av min informant. Rektoren på den aktuelle skolen ga muntlig samtykke om at prosjektet kunne gjennomføres på skolen, og læreren informerte elever og foresatte i den aktuelle klassen om prosjektet. Det var ikke nødvendig å innhente samtykke fra elevene eller foresatte siden prosjektet for elevenes del i sin helhet inngikk i ordinær undervisning, men etter lærerens ønske ble det sendt ut litt informasjon om hva prosjektet innebar.

To sentrale, etiske prinsipper man må ta hensyn til er frivillighet og anonymitet. Læreren i prosjektet samtykket til frivillig deltakelse, og ble informert om retten til å trekke seg. Alle personopplysninger har blitt anonymisert, og det vil dermed ikke være mulig å spore seg tilbake til min informant. Datamateriale som inneholder personopplysninger, som i dette tilfelle består av lydopptak av intervjuene, vil bli slettet ved prosjektslutt. Under arbeidet med dette masterprosjektet har lydopptak og transkribering blitt oppbevart passordbeskyttet i OneDrive.

I arbeidet med dette masterprosjektet har jeg hatt et tett samarbeid med min informant. Som forsker har det vært viktig å ivareta henne gjennom hele forskningsforløpet. Jeg har vært opptatt av at prosjektet ikke skal medføre en for stor ekstra arbeidsbelastning for henne. I gjennomføringen av prosjektet har jeg derfor vært fleksibel med tanke på tidspunkt for intervjuene og planleggingen, slik at det ble tilpasset best mulig i hennes arbeidshverdag. I forhold til datainnsamlingen har det vært viktig å behandle dataene vitenskapelig redelig. Det vil si at forskningsprosessen og de utkomne resultatene har blitt formidlet uten å villede eller fordreie (Høgheim, 2020). For å ivareta informanten har det i arbeidet med datamaterialet i dette prosjektet vært viktig å ikke utlevere informanten for mye, eller fremstille funnene på en måte som kan sette henne i et dårlig lys.

4 Presentasjon av funn

I dette kapittelet vil jeg presentere funnene i denne masteroppgaven. Funnene vil bli presentert i følgende fire avsnitt: Super:bit og koblingen mot matematikkfaget, algoritmiske arbeidsmåter, lærerrollen og organisering av undervisning. Avslutningsvis, i avsnitt 4.5, vil jeg gi en kort oppsummering av hovedfunnene.

4.1 Super:bit og koblingen mot matematikkfaget

Generelt kan en si at læreren er svært positiv til Super:bit-prosjektet. Hun trekker fram at tilgjengeligheten gjennom oppgavene på nett, lærerveiledning og den utleverte utstyrspakken gjør det enklere for læreren å komme i gang med undervisningen, og beskriver det som «*en unik mulighet for alle skoler*». Prosjektet er på den måten en god inngangsport til programmeringsundervisningen, og en nødvendig start for å sikre kompetanseheving blant lærere.

En av fordelene ved Super:bit er etter hennes erfaring at det er forholdsvis selvinstruerende, som igjen fører til at det ikke kreves så mye bakgrunnskunnskap, og det er dermed enklere å komme i gang. I undervisningsperioden brukte vi eksempelvis NRK sine videoer med forklaringer ved flere anledninger, og læreren kunne dermed utforske oppgavene i samarbeid med elevene. På en annen side er læreren åpen om at ikke alle lærere deler dette synspunktet. Hun hevder at for lærere uten interesse for programmering kan Super:bit, i likhet med andre programmeringsaktiviteter, være for tidkrevende og utfordrende å sette seg inn i på egenhånd. Hun var selv med på lærerkurset i regi av Vitensenteret, et kurs læreren uttrykker ga godt utbytte og motivasjon til å ta i bruk opplegget med egne elever.

Læreren uttrykker at hun generelt er svært fornøyd med både oppgavene og utstyret i prosjektet. Samtidig er det noen utfordringer, spesielt knyttet til det tekniske. De tekniske utfordringene går på at utstyret ikke fungerer som det skal, noe som skaper stor frustrasjon hos elevene. Dette opplevde vi i undervisningsperioden, blant annet da elevene arbeidet med oppgaven der de skulle få BitBoten til å tegne ulike geometriske figurer (økt 2).

Læreren forteller at Super:bit har skapt stort engasjement blant elevene. Hun opplever at mange av elevene synes det er lystbetont å arbeide med programmering, og at de gleder seg til undervisningen. Denne antakelsen ser ut til å stemme, da flere av elevene i klassen ønsket seg programmeringsutstyr til jul slik at de også kunne ha mulighet til å drive med det på fritiden.

Læreren er opptatt av hvilket læringsutbytte elevene kan få av programmering, og trekker frem nettopp dette som en utfordring i undervisningen. Hun opplever at programmeringsundervisning fort kan bli en «*happening*», og at meningen bak i noen tilfeller kan forsvinne. Av årsaker til dette nevner hun tre ting: integreringen av programmering i fag har ikke lyktes godt nok enda, elevene ser ikke nytteverdien, og manglende kompetanse hos lærerne. Likevel poengterer hun at hun ser på programmering som en viktig kompetanse elevene må tilegne seg, da det vil være en sentral kompetanse i fremtiden.

Et grep vi gjorde i undervisningen for å koble programmeringen nærmere det matematikkfaglige var å rette fokuset mot geometri. Som en konsekvens av dette endret vi noen av oppgavene fra Super:bit litt, slik at de krevde mer kompetanse innenfor geometri. I økt 1, del 2, hvor elevene hadde i oppgave å gi hverandre instruksjoner, lagde vi eksempelvis bilder som inneholdt en rekke ulike geometriske figurer de måtte beskrive. Figur 12 viser eksempel på to av bildene elevene møtte.



Figur 12: Eksempel på oppgaver i «Tegne etter instruksjoner fra læringspartner», økt 1

I undervisningssituasjonen kom det tydelig frem at elevenes matematiske kompetanse innenfor geometri var varierende. Spesielt i oppgaven hvor elevene skulle beskrive de geometriske figurene for hverandre skapte dette utfordringer, da de ikke hadde like stor forståelse for de samme matematiske begrepene. Eksempelvis var det en elev som ikke visste hva et kvadrat var. Elevene ble imidlertid mer drevne etterhvert og fant alternative måter å beskrive figurene på. Dette ble eksempelvis gjort ved at en elev beskrev et kvadrat på følgende måte: «*en firkant hvor alle sidene er like lange*».

Et annet eksempel på grep vi gjorde, var å stille krav til hva elevene skulle tegne når de programmerte tegnerobotene. For å hindre at elevene kun skulle lage vilkårlige «kruseduller» ga vi dem i oppgave å programmere ulike geometriske figurer. Figur 13 viser eksempel på en av oppgavene elevene møtte. De geometriske figurene de fikk i oppgave å programmere måtte de i etterkant måle og regne ut areal og omkrets av. Dette grepet ble gjort for å stille krav til matematisk kompetanse.

Kvadrat

a) Korleis ser eit kvadrat ut? Kva for kjenneteikn har det?

.....
.....
.....

b) Prøv å programmere BitBoten slik at han køyrer i eit kvadrat.

c) Fest ein penn til BitBoten når de er fornøgd med koden, og la han teikne kvadratet.

d) Rekneoppgåve:

a. Kor lange er sidene i kvadratet?

b. Kor stor er omkrinsen?.....

c. Kva er arealet?.....

Figur 13: Eksempel på oppgave i «Utforske geometriske figurer med programmering», økt 2

I flere av oppgavene vi brukte ble det stilt store krav til nøyaktighet, da ulike instruksjoner kan tolkes på ulike måter. Dette var noe vi ønsket at elevene skulle bli bevisst på, da nøyaktighet er svært viktig i programmering. Læreren hevder at det er mye matematikk i programmering, «selv om man ikke alltid tenker over det». Et eksempel fra undervisningssituasjonen hvor matematikken ble synliggjort var fra da elevene arbeidet med blokkprogrammeringen av BitBoten, og elevene måtte reflektere over hvor stor hastigheten skulle være i forhold til tiden for at den skulle kjøre ønsket lengde. I denne øvelsen måtte elevene ha forståelse for både fart og tid.

En annen fordel med programmering i matematikk slik læreren opplever det, er at for elever som sliter teoretisk kan programmering være en god tilnærming for å konkretisere og arbeide

praktisk med matematikken. Dette oppdaget vi også i undervisningen, da enkelte elever overrasket oss med sitt faglige nivå. Læreren legger vekt på at hun har sett nytteverdien av konkretiseringen for de faglige svake elevene, men også for elevgruppa generelt. Hun opplever at elevene gjennom konkretiseringen i programmeringen kan få en annen forståelse enn hvis de sitter og arbeider i en bok eller på datamaskinen. Gjennom programmering får elevene en annen måte å lære på, som igjen ser ut til å virke positivt på motivasjonen.

4.2 Algoritmiske arbeidsmåter

Læreren er svært opptatt av at elevene må lære seg å prøve og feile, *«(...) at alt er ikke bare et svar med to streker under, men at man må prøve også ble ikke det sånn man hadde tenkt. Også prøver man på nytt igjen.»*. Læreren forteller at hun opplever at elevene ikke har tilegnet seg gode løsningsstrategier og metoder, som fører til at det blir utfordrende å løse problemer som dukker opp. Hun uttrykker at elever i dag skal ha svaret på alt, og ser nytten av at elevene tilegner seg løsningsstrategier og metoder slik at de kan komme frem til svaret på egenhånd. Strategien «prøve og feile» ble benyttet av elevene ved flere anledninger i undervisningssituasjonen. Dette så vi eksempelvis når elevene skulle programmere BitBoten til å tegne en trekant. Oppgaven krevde at elevene brukte flere forsøk for å komme frem til programmeringskoden som ga de korrekte vinklene og lengdene på sidene i den geometriske figuren. Gjennom prøving og feiling arbeidet elevene for å komme frem til en løsning, og også i denne oppgaven var det flere mulige løsninger på oppgaven.

Noen av elevene har lett for å gi opp dersom de ikke får til ting med en gang. I undervisningen måtte disse elevene veiledes slik at de forsto hvordan de kunne komme frem til en løsning. I undervisningen opplevde vi at det var vanskelig for en del elever å ikke få en konkret fremgangsmåte. Elevene synes det kan være utfordrende når det finnes flere løsninger, da mange er mer vant til å bli fortalt hva de skal gjøre, *«også er det bare sånn»*. Enkelte elever ble utålmodige og ga opp når de ikke kom frem til løsningen ved første forsøk, og slet med å forstå hvilke endringer de kunne gjøre for å komme frem til riktig programmeringskode. Denne erfaringen gjorde at vi i de neste undervisningsøktene hadde klare noen enklere alternativer til oppgavene, slik at alle fikk mulighet til å oppleve mestring. I den siste økta hvor elevene programmerte radiostyrte biler fikk eksempelvis noen av elevene arbeide med å programmere BitBoten til å kjøre gjennom Smartby-matta, en mindre sammensatt oppgave.

Et annet svært sentralt aspekt ved programmeringen er samarbeid. *«De skal se at det er lurt å samarbeide. For kanskje jeg vet noe, og du vet noe annet. Og til sammen kan vi kanskje komme fram til gode løsninger for å få til dette her på en måte»*. Elevene er vant til å arbeide i læringspar og samarbeide med hverandre, men for enkelte var det likevel utfordrende å involvere partneren sin i arbeidet. Læreren ønsket at elevene gjennom programmeringsoppgavene skulle se nytten av å arbeide sammen med en partner, og at de skulle bruke hverandres sterke sider for å komme frem til en felles løsning. Organiseringen med læringspar vil jeg komme tilbake til i avsnitt 4.4.

4.3 Lærerrollen

Med tanke på lærerrollen i undervisningssituasjonen er læreren svært opptatt av at hun ikke skal ha svarene, men bakgrunnskunnskapen. Hun poengterer, at etter hennes oppfatning, har mange elever mye mer kunnskap i programmering enn lærere generelt. Kunnskapen til elevene må med fordel utnyttes og brukes som ressurser i undervisningen. Det er likevel viktig for læreren at hun setter seg inn i faget slik at hun har mulighet til å veilede elevene i arbeidet. I forbindelse med undervisningsopplegget vi gjennomførte utførte både læreren og jeg oppgavene elevene møtte i forkant av undervisningen. De tilgjengelige lærerveiledningene på Super:bit sine nettsider ga god veiledning, og gjorde det forholdsvis enkelt å forstå hvordan oppgavene skulle løses. Gjennomgangen av oppgavene var god å ha med seg i møte med elevene i undervisningen, da de møtte på noen av de samme problemene vi hadde gjort. Det ble på den måten enklere for oss å veilede elevene når det oppsto et problem.

Læreren sin rolle er altså å veilede, mens elevene skal reflektere og finne frem til en løsning. Disse holdningene til lærerrollen gjenspeilte seg også i observasjonene av undervisningsøktene. Læreren var ærlig med elevene om at hun ikke kunne alt, og at de måtte utforske oppgavene i fellesskap. Dette ble synlig i undervisningssituasjonen ved flere anledninger, da læreren arbeidet sammen med elevene for å komme frem til en løsning.

Det var tydelig at hun ønsket at elevene skulle komme frem til løsninger selv. Læreren veiledet, men ga ikke svarene til elevene. Et eksempel på dette er fra den første aktiviteten i økt 1, «tegne etter instruksjoner». I denne øvelsen veiledet læreren elevene ved at de hadde en dialog i klassen om enkelte begreper i oppgaven som kunne være ukjente for elevene. Dette var begreper som «horisontal» og «vertikal». Utover dette holdt læreren fast ved de små instruksjonene elevene fikk i oppgaven, og ga ikke mer hjelp selv om enkelte spurte. Hun poengterte at det var flere løsninger og muligheter til å tolke instruksjonene, og at denne

tolkningen var en del av oppgaven for elevene. Læreren hadde stort fokus på elevenes egne refleksjoner når hun veiledet elevene. På den måten la hun opp til at elevene selv måtte finne løsninger på problemene som dukket opp. Gjennom diskusjon i helklasse ble ulike løsningsmetoder og tankeganger lagt frem, og det var fokus på at ingenting er «feil».

Læreren er tydelig på at hun selv ønsker seg mer kompetanse innenfor programmering. Under intervjuene stiller læreren spørsmål ved hvem som skal ta ansvaret for opplæringen av lærere i forbindelse med innføringen av programmering i LK20. Hun erfarer at det er nødvendig med en kompetanseheving for at lærere skal føle seg trygge på undervisningen, og trekker frem at slik det er i dag er det etter hennes opplevelse nødvendig med interesse for å sette i gang med programmering i undervisningen. Samtidig poengterer hun at hun er svært positiv til Super:bit-prosjektet, og uttrykker at det er et viktig bidrag for å få programmering inn i skolen og øke kompetansen blant lærerne.

4.4 Organisering av undervisning

Et særpreg ved fådelt skole er den aldersblandede organiseringen. Læreren gir uttrykk for at hun opplever aldersblanding som noe positivt, og hevder at det legger til rette for at elevene kan støtte, motivere og hjelpe hverandre ved at de har med seg ulik kunnskap og erfaringsbakgrunn. På bakgrunn av dette ble undervisningen organisert slik at elevene skulle arbeide i læringspar.

I planleggingen av undervisningen satte vi sammen læringspar hvor de antatt faglig sterkeste elevene kunne hjelpe elever med mindre kompetanse. Det viste seg i første time at noen av de elevene læreren hadde tenkt kom til å mestre programmeringen, synes det var vanskelig. På den andre siden var det elever som viste at de mestrer mer enn de viser til vanlig i faget. En utfordring ved organiseringen er at enkelte av elevene kan oppleve en følelse av å bare «henge på», mens andre må «dra lasset». Læreren poengterer likevel at det er en fin arena for å trene på samarbeid og respekt for at vi er forskjellige. Læreren forteller at elevene arbeider mye på denne måten, men hun opplevde at det kanskje var ekstra utfordrende for enkelte i programmeringsaktivitetene. «Ja, for det at det ikke er et konkret mål på en måte. Det at du skal prøve og feile, prøve og feile, sant. Og da er det noen som lett gir opp».

Selv om det oppsto noen utfordringer i læringsparene, opplevde vi også at det fungerte godt. Elevene var stort sett flinke til å samarbeide, lytte til hverandre og komme med innspill. Elevene som viste god faglig kompetanse hjalp medelevene sine ved flere anledninger i undervisningen, som igjen førte til at læreren kunne bruke mer tid på de læringsparene som

trengte ekstra oppfølging. Ut fra observasjonene så det altså ut til at læreren slapp en del spørsmål ved at elevene hjalp og veiledet hverandre. Dette førte til at læreren i større grad kunne sikre at elevene som trengte ekstra oppfølging også forsto oppgavene og dermed fikk oppleve mestring.

4.5 Oppsummering av funn

I dette kapitlet har funn fra analysen blitt presentert. For å oppsummere vil jeg trekke frem fire sentrale erfaringer læreren gjorde seg i undervisningsperioden. Disse erfaringene vil være grunnlaget for drøftingen i neste kapittel.

1. Læreren opplever at det er utfordrende å integrere programmering i fag, og hevder at en konsekvens av dette kan være at læringsutbyttet kan komme i skyggen av selve aktivitetene.
2. Læreren nevner ikke begrepene algoritmisk tenkning og algoritmiske arbeidsmåter konkret. Gjennom et mer muntlig vokabular og i utøvelsen av selve undervisningsopplegget kommer likevel flere kjennetegn ved algoritmiske arbeidsmåter til uttrykk. Dette er kjennetegn som å prøve og feile, holde ut, samarbeide, utforske og eksperimentere.
3. Læreren tar i bruk en veiledende lærerrolle i programmeringsundervisningen. Gjennom å vise sine egne svakheter og dra nytte av elevenes erfaringer og kunnskap skapes det en dynamikk i undervisningen som fungerer godt. Hun er likevel tydelig på at det er behov for kompetanseheving for lærere for å nå de nye målene i læreplanen.
4. Læreren opplever at organiseringen med læringspar og aldersblanding gir mulighet for at elevene kan støtte og veilede hverandre i læringsarbeidet.

5 Drøfting

I dette kapittelet vil jeg drøfte funnene fra observasjonene og intervjuene opp mot det teoretiske rammeverket og tidligere forskning. Dette skaper grunnlag for å besvare problemstillingen, som jeg vil komme nærmere tilbake til i kapittel 6.

5.1 Integrering av programmering i matematikkfaget

En av utfordringene knyttet til innføringen av programmeringen i LK20 er etter lærerens erfaring å integrere det mot spesifikke fag. Ut fra tidligere erfaringer læreren har gjort seg, forteller hun at det kan være utfordrende å se sammenhengen mellom programmeringen og matematikken. Diskusjonene blant fagfolk om programmerings plass i skolen synliggjør at dette er en aktuell problematikk. Til tross for at programmeringsundervisningen oppleves som utfordrende, er ikke læreren i tvil om at det er viktig at programmeringen har kommet inn i læreplanen. I likhet med mange fagfolk deler hun synspunktet om at programmering er en viktig kompetanse for fremtiden, som elevene må tilegne seg (Kaufmann et al., 2018).

Læreren erfarer at en utfordring ved programmeringen er at det blir en «*happening*». En konsekvens av dette kan være at læringsutbyttet kommer i skyggen av selve aktiviteten. Denne problematikken støttes av Kaufmann og Stenseth (2020), som hevder at det er fare for at matematikken kan «forsvinne» i aktiviteten. At programmeringen ikke er godt nok integrert i faget enda, elevene ser ikke nytteverdien, og manglende lærerkompetanse, trekkes frem som mulige årsaker til denne problematikken. Disse årsakene kan se ut til å ha en viss sammenheng. Dersom lærerkompetansen i programmering ikke er god nok, vil det være nærliggende å tro at det kan være utfordrende å knytte det til fag. Dersom lærerne opplever det som utfordrende å se koblingen mellom programmeringen og matematikkfaget, kan man heller ikke forvente at elevene gjør det. Manglende kompetanse hos lærerne kan av den grunn være en årsak til at aktiviteten oppleves som en såkalt «*happening*». Dette støttes av Kaufmann og Stenseth (2020), som gjennom sin forskning konkluderer med at dersom programmeringen skal integreres på en vellykket måte i matematikken, er lærerens kompetanse helt avgjørende.

Samtidig som læreren problematiserer koblingen mellom matematikk og programmering, poengterer hun at programmering stiller krav til matematisk kompetanse som ikke alltid er like synlig. Dette begrunner hun i at hun opplever at den nye læreplanen stiller krav til matematikk på en «ny» måte. I den nye læreplanen blir egenskaper som å finne sammenhenger, diskutere, vurdere, analysere og utvikle fremgangsmåter og strategier

vektlagt, egenskaper som tradisjonelt ikke har blitt vektlagt i matematikkundervisningen (Sikko & Grimeland, 2020). Dette kan indikere at matematikkfaget er i endring, og at det stilles krav til nye kompetanser innenfor faget. Gjennom den nye læreplanen ser det ut til at fokuset i matematikk ikke lenger kun handler om å finne rett svar, men selve prosessen for å komme frem til svaret er like viktig. Denne tankegangen er ikke motsigende i forhold til programmeringsaktiviteter. I Sevik (2016) og Nygård (2018) sine definisjoner på programmering er også selve prosessen med å arbeide seg frem til en programmeringskode, altså læringsprosessen vektlagt. Programmeringsprosessen handler om å identifisere et problem, finne mulige løsninger, skrive koder, feilsøke og forbedre, noe som samsvarer godt med egenskaper som vektlegges i LK20.

Vår erfaring med bruk av programmering i matematikk var ikke utelukkende positiv. Da elevene skulle tegne geometriske figurer ved hjelp av BitBoten, førte tekniske problemer til at figurene ikke ble geometrisk riktige. Dette til tross for at programmeringskodene var korrekte. En kan dermed diskutere hvor godt dette verktøyet var for å synliggjøre egenskapene til figurene. Poengene med tegningene var at elevene skulle få en visualisering, og at de gjennom programmeringen måtte reflektere rundt hvordan figurene skulle se ut. Utfordringene førte til usikkerhet for elevene, både med tanke på om de hadde forstått programmeringen og om de forsto egenskapene til de geometriske figurene rett. Gjennom dialog med elevene i etterkant opplevde vi at de hadde forståelse for de tekniske utfordringene, men ved en annen anledning ville jeg som lærer brukt en annen tilnærming som er mer nøyaktig. Det finnes andre programmeringsverktøy, eksempelvis Scratch, hvor en har mulighet til å programmere geometriske figurer. Vår erfaring var at BitBot ikke egnet seg så godt til dette, da slike oppgaver krever høy grad av nøyaktighet for å få ønsket læringsutbytte.

I etterkant av undervisningsopplegget trakk læreren frem tre ting hun opplevde som positivt ved å bruke programmering i matematikkfaget: konkretisering, en praktisk tilnærming og motivasjon. Konkretiseringen og den praktiske tilnærmingen i undervisning trekkes også frem i Kalelioglu og Sentance (2020) sin forskning på bruk av micro:bit i undervisningen. Den raske tilbakemeldingen gjør at det er lett for elevene å oppdage feil og dermed arbeide seg frem til nye løsninger. Micro:bit og de mulighetene verktøyet skaper i undervisningen kan på den måten virke motiverende for elevene (Kalelioglu & Sentance, 2020). Dette ble synliggjort i undervisningen ved at elever som vanligvis ikke er så fremtredende i matematikkundervisningen viste høy kompetanse og motivasjon.

Vi opplevde elevene som motiverte i undervisningsperioden, de gledet seg til undervisningen og viste engasjement i timene. Motivasjon og interesse for aktivitetene kan føre til økt motivasjon for faget (Husain et al., 2017; Ríos Félix et al., 2020). På en annen side er det vanskelig å vite om elevene forstår denne koblingen mot matematikken, og ser nytteverdien av god matematisk kompetanse i programmeringen. Vi kan altså ikke vite om programmeringen førte til en økt motivasjon for matematikkfaget, eller om det var selve programmeringsaktiviteten som skapte motivasjon og engasjement. Dette fører oss tilbake til lærerens erfaring med at det kan være vanskelig å sørge for at programmeringen ikke bare blir en «*happening*», men at elevene også ser det faglige utbyttet. Programmeringen har på en annen side uansett skapt en interesse hos enkelte elever, som i etterkant av introduksjonen på skolen anskaffet utstyr de kunne benytte på fritiden. Super:bit har på den måten nådd et av sine mål ved at elever får opp øynene for programmering, og oppdager at dette kan være et interessant fagfelt.

5.2 Algoritmiske arbeidsmåter i undervisningen

Under intervjuene og i gjennomføringen av undervisningsopplegget blir ikke begrepene algoritmisk tenkning og algoritmiske arbeidsmåter nevnt konkret. Kjentegn ved algoritmiske arbeidsmåter kommer likevel til uttrykk gjennom et mer muntlig vokabular og ikke minst i utøvelsen av undervisningen. Selv om læreren ikke bruker fagbegrepene, kan det altså se ut til at læreren likevel har et bevisst forhold til innholdet i det.

Et eksempel på slik muntlig språk er følgende sitat: «(...) *at alt er ikke bare et svar med to streker under, men at man må prøve også ble ikke det sånn man hadde tenkt. Også prøver man på nytt igjen*». I sitatet beskriver læreren fremgangsmåten i den algoritmiske arbeidsmåten feilsøke, som er beskrevet som å «oppdage og rette feil» i Utdanningsdirektoratet (2019a) sin modell. Læreren bruker heller betegnelsen «prøve og feile», men innholdet i strategien kan se ut til å være tilsvarende lik.

Læreren uttrykker at elevene ikke er trente til å finne gode løsningsstrategier, og hevder at de må øve på det. I læreplanen står det at «*Elevene skal legge mer vekt på strategiene og framgangsmåtene enn på løsningene*» (Utdanningsdirektoratet, 2020, s. 2). For å lykkes med dette, slik vi erfarte det i undervisningsperioden, må det skje en endring i fokus hos elevene. Vi opplevde at elevene var svært opptatt av resultat, og ønsket å komme dit så raskt som mulig. Samtidig kan det se ut til at enkelte av elevene fikk bedre utholdenhet ved å bruke micro:biten og BitBoten. Gjennom å få raske, konkrete tilbakemeldinger fra verktøyene var

det enklere å se hvor de eventuelle feilene oppsto og dermed gjøre de nødvendige endringene. Dette trekkes også frem som en fordel i forskningen til Kalelioglu og Sentance (2020). Micro:bit kan på den måten være et motiverende arbeidsverktøy, som hjelper elevene til å holde ut i arbeidet.

Den umiddelbare tilbakemeldingen fra programmeringsverktøyene gir gode muligheter for utforsking (Kalelioglu & Sentance, 2020). Læreren var svært opptatt av at elevene skulle utforske oppgavene selv, hun skulle ikke «*gi svarene*». Dette samsvarer godt med den algoritmiske arbeidsmåten fikle, som innebærer å utforske og eksperimentere (Utdanningsdirektoratet, 2019a). Den algoritmiske tenkeren kjennetegnes ved at elevene må være analytiske og systematiske, men også skapende, eksperimenterende og åpne for alternative løsninger. En slik tilnærming krever at elevene er nysgjerrige og utforskende i sitt arbeid. Ved bruk av en utforskende tilnærming i undervisningen, slik som IBL, legges det til rette for at elevene får god trening i dette. Elevene fikk ikke nøyaktige instruksjoner i hvordan oppgavene skulle løses av læreren, og måtte fikle for å finne ut av hvordan programmeringsverktøyene fungerte og hvordan oppgavene skulle løses.

På en annen side kan en stille spørsmål ved hvor utforskende enkelte av oppgavene i Super:bit er. Trekker jeg fram radiobilen elevene skulle programmere som eksempel, ble denne oppgaven løst ved at elevene fulgte stegvise instruksjoner på en video. Samtidig ga ikke videoen svarene på alt, og elevene måtte selv eksperimentere med eksempelvis fart og hva slags styringsfunksjoner som skulle legges inn. Balansegangen mellom hvor mye instruksjoner som bør gis og hvor mye elevene skal utforske selv kan i denne sammenheng være noe utfordrende. På den ene siden kreves det en del instruksjoner, spesielt for elever som ikke har arbeidet så mye med programmering tidligere. På en annen side vil det å følge stegvise instruksjoner kunne være noe ødeleggende for forståelsen. I det gjennomførte undervisningsopplegget var det av den grunn viktig å legge oppgavene på riktig ferdighetsnivå. Dette er i tråd med Vygotskij sin proksimale utviklingssone (Imsen, 2020).

Et siste sentralt kjennetegn ved algoritmiske arbeidsmåter er samarbeid. Samarbeid gjennom å dele tanker og ideer trekkes frem som en egnet arbeidsmetode for Den algoritmiske tenkeren. Læreren er svært opptatt av verdien av samarbeid, noe som synliggjøres i intervjuene og i organiseringen av undervisningen. Med rot i det sosiokulturelle perspektivet på læring er tanken at elevene lærer best i en sosial prosess (Imsen, 2020). Gjennom samarbeidet ønsket læreren at elevene skulle dra nytte av hverandre, «*De skal se at det er lurt å samarbeide. For*

kanskje jeg vet noe, og du vet noe annet». Jeg kommer nærmere tilbake til organisering og samarbeid i avsnitt 5.4.

5.3 Veiledende lærerrolle i undervisningen

Engeln et. al (2013) skriver at lærerens holdning til faget har betydning for hvordan undervisningen blir lagt opp. Læreren jeg har samarbeidet med gjennom denne undervisningsperioden har tro på en utforskende tilnærming til kunnskap og læring. Hennes holdninger og utøvelse av lærerrollen samsvarer godt med kjennetegn ved lærerrollen i IBL. Dette synliggjøres gjennom at hun hadde en veiledende lærerrolle, eksempelvis ved at hun ikke ga elevene svarene, men lot de utforske og prøve selv, utfordret dem ved å stille spørsmål og oppmuntret til samarbeid og diskusjon.

I en utforskende tilnærming til læring er det sentralt å finne en balanse mellom å la elevene utforske fritt og gi tilstrekkelig veiledning. I undervisningsperioden erfarte vi at elevene som ikke mestret oppgavene ble frustrerte og lei, noe som medførte at de ga opp. Tilstrekkelig kompetanse for å se sammenhenger og finne løsningsstrategier er en forutsetning for at elevene skal oppleve mestring (Kalelioglu & Sentance, 2020). I en utforskende tilnærming til læring er det nødvendig at elevene får veiledning, som gjør dem i stand til å møte utfordringene i undervisningen. Samtidig skal ikke veiledningen være så detaljert at elevene får en «bruksanvisning» de kan følge. Balansegangen kan dermed være utfordrende, og erfaringsmessig var det svært ulikt hvor god utholdenhet elevene hadde i det utforskende arbeidet før de ville ha et svar. Det var dermed nødvendig å tilpasse veiledningen etter nivået elevene var på. Dette samsvarer med det sosiokulturelle perspektivet på læring, som vektlegger behovet for «scaffolding» i læringsarbeidet (Imsen, 2020)

Forskning viser at kunnskapsforholdet mellom lærer og elev er mer symmetrisk i programmering, sammenlignet med andre fag (Nouri et al., 2020). Læreren nølte ikke med å vise svakheter i undervisningen, og var tydelig på at hun ikke var noen ekspert på programmering. Ved å bruke elevene som ressurs ble det skapt en fin dynamikk, der de dro nytte av hverandre. I undervisningen arbeidet læreren og elevene ved flere anledninger sammen for å løse oppgavene. Gjennom dialog ble elevene oppmuntret til å utforske og utveksle ideer, og diskutere disse. Samtidig kan dette symmetriske kunnskapsforholdet føre til at læreren ikke har den nødvendige kompetansen for å veilede elevene på det nivået de er på. En konsekvens av dette er at elevene ikke får tilstrekkelig støtte for å utvikle sin kompetanse videre.

Læreren er tydelig på at det er behov for kompetanseheving for lærere for å nå målene i LK20. Super:bit er et godt bidrag i kompetansehevingen for lærerne, men hun savner oppfølging for å utvikle kompetansen videre. Denne oppfatningen støttes av annen forskning. Kaufmann og Stenseth (2020) advarer mot å tro at økt fokus på programmering i skolen i seg selv er løsningen. Uten tilstrekkelig lærerkompetanse vil en ikke kunne utnytte det læringspotensialet som finnes i programmeringen. Denne advarselen samsvarer godt med læreren sitt spørsmål om hvem som skal ta ansvar for opplæringen. Slik situasjonen er nå ser det ut til at lærerne har fått starthjelp med programmeringen gjennom prosjektet Super:bit, men at en i ettertid står på egne bein. Det er heller ikke alle lærere som har fått mulighet til å delta på lærerкурset i regi av Vitensentrene. Dette kan være en av årsakene til at enkelte lærere opplever det som utfordrende å ta i bruk den utleverte utstyrspakken. Læreren i mitt prosjekt opplevde kurset som en nyttig input, og beskrev at hun var svært motivert for undervisningen i etterkant. Dette synliggjør at kurs hvor en får konkrete tips om oppgaver og hvordan en kan legge opp undervisningen kan gi lærere god hjelp til å komme i gang. Ut fra denne lærerens erfaring med Super:bit ser det altså ut til at læreren opplever at det er et solid støtteverktøy i undervisningen, som også var et av de overordnede målene til prosjektet.

5.4 Organisering av aldersblandet elevgruppe

Organiseringen på en fådelte skole skaper muligheter, men samtidig utfordringer i undervisningssituasjon. I planleggingen av undervisningsopplegget i forbindelse med dette prosjektet var det eksempelvis mer utfordrende å kartlegge elevenes erfaringsbakgrunn, da elevene hadde gått i forskjellige klasser og dermed fått ulik undervisning tidligere. På bakgrunn av dette var det variasjoner i hva slags kompetanser og erfaringer elevene hadde som utgangspunkt før undervisningsperioden. På en annen side er ikke dette ukjent for elevene og lærerne på den fådelte skolen. Gjennom den aldersblandede organiseringen, som er slik i alle fag, er de vant til at elevene er på ulikt faglig nivå. Aldersblandingen er ikke noe læreren velger, men en nødvendighet i skolehverdagen. Læreren trekker frem en rekke fordeler ved den aldersblandede organiseringen, slik som at elevene gjennom ulik kunnskap og erfaringsbakgrunn kan støtte, motivere og hjelpe hverandre. Forskning viser at aldersblanding kan gi god effekt på elevenes læring, dersom elevene klarer å dra nytte av hverandre (Leuven & Rønning, 2016).

For å utnytte ressursene hos elevgruppa organiserte vi elevene slik at de arbeidet sammen i læringspar. Mange av oppgavene i Super:bit legger til rette for at elevene skal arbeide i par eller grupper, og organiseringen med læringspar var dermed hensiktsmessig. Tanken bak dette

var at elevene på denne måten kunne hjelpe og støtte hverandre i læringsprosessen. Elevene med god forståelse vil få utfordringer ved å veilede og forklare medelever på et lavere kompetansenivå. Elevene på et lavere kompetansenivå vil på sin side få forklaringer som muligens kan være enklere å forstå enn dersom det var en lærer som skulle forklare det samme. Dette kan støttes opp ved Vygotskij (1978) sin teori om den proksimale utviklingszone. En mer kompetent elev fungerer som støtte for medeleven på et lavere kompetansenivå. Etter hvert er målet at eleven vil kunne løse oppgavene uten denne støtten, og dermed utvide den proksimale utviklingssonen.

Samtidig kan en stille spørsmål ved om alle elevene fikk utfordringer som var tilpasset deres nivå. I følge Vygotskij er elevene avhengig av støtte og veiledning for å utvikle seg (Imsen, 2020). Elever som arbeider på et nivå de mestrer godt, vil altså ikke få tilstrekkelige utfordringer for å utvikle kompetansen videre. På bakgrunn av dette er det viktig at alle elevene får utfordringer som er tilpasset deres ferdighetsnivå. I læringsparene møtte de faglig sterke elevene færre utfordringer enn de svake. Samtidig fikk de en utfordring ved at de måtte forklare og lære vekk til medelever, noe som igjen krever god forståelse og kompetanse. Arbeid i mindre grupper kan også medføre at det er enklere å ta ordet og uttrykke sine meninger. Det kan tenkes at terskelen for å stille spørsmål og dele ideer med elevene var lavere i læringsparene enn det hadde vært i en situasjon hvor hele klassen var samlet. I undervisningsopplegget var det eksempelvis elever som ikke pleier å være fremtredende som tok sentrale roller i samarbeidet.

Dersom aldersblanding skal fungere viser funn i Rønning og Leuven (2016) sin forskning at å kunne dra nytte av eldre elever er en forutsetning. Denne erfaringen gjorde vi oss i undervisningsperioden, da vi hadde læringspar som ikke fungerte så godt. Elevene læreren på forhånd hadde tenkt skulle være en støtte for andre elever viste seg å ikke være på det kompetansenivået hun hadde forestilt seg. Det ble dermed vanskelig for noen av parene å løse oppgavene, da ingen av dem fungerte som en kompetent annen. I andre læringspar ble det gjort en annen erfaring, ved at den antatt mindre kompetente eleven var den som fikk den veiledende rollen. Gjennom en annen tilnærming til matematikk fikk elever som ikke vanligvis viser seg frem vist frem sine sterke sider. Dette samsvarer med funn i Taylor et al. (2010) sin forskning, som konkluderte med at samarbeid mellom elever skaper muligheter for å vise og dele kunnskap. Ved å skape en annen læringssituasjon i matematikkfaget kan det se ut til at elever tar nye roller. Årsaken til dette kan blant annet være interesse for

programmeringsaktivitet, at de har løst lignende oppgaver tidligere, eller at den praktiske tilnærmingen til matematikk appellerer bedre.

6 Avslutning

Formålet med dette masterprosjektet har vært å undersøke en lærers erfaringer med bruk av ressurser fra Super:bit. Ved å benytte DBR som metodisk tilnærming, som ga mulighet for å være deltakende i planlegging, gjennomføring, evaluering og justering av undervisningsopplegget som ble gjennomført i dette prosjektet, ønsket jeg å få en dypere innsikt i dette. Deltakelsen la til rette for at jeg som forsker fikk se hvordan læreren arbeidet i praksis, og på den måten innhentet jeg informasjon om hvordan læreren brukte sin kunnskap og materiellet i en konkret undervisningssituasjon. Dette ga grunnlaget for å svare på følgende problemstilling:

Hvilke erfaringer gjør en lærer på en fådelt skole seg med bruk av ressurser fra Super:bit-prosjektet i matematikkundervisningen på mellomtrinnet?

I dette kapittelet er hensikten å besvare problemstillingen. Refleksjoner og tanker om videre forskning vil også bli presentert.

6.1 Konklusjon

Ut fra funnene i dette masterprosjektet ser det ut til at Super:bit-prosjektet har nådd målene sine. Elevene har fått lære programmering ved hjelp av micro:bit, og det ser det ut til at prosjektet har greid å skape interesse og engasjement blant elevene. Det er heller ikke tvil om at ressurser fra Super:bit gjør det enklere å komme i gang med undervisningen, ved at det fungerer som et solid støtteverktøy for læreren. Utfordringen ligger i hvordan en skal utvikle seg videre, og i lys av dette etterspør læreren tilrettelegging for ytterligere kompetanseheving for lærere. Dette er etter lærerens erfaring en nødvendighet for at programmeringen skal tas i bruk i skolen på en hensiktsmessig måte, og ikke minst for å klare å integrere det i fagene på en vellykket måte. Funn fra denne studien viser altså, i likhet med forskning på feltet, at det er behov for kompetanseheving for lærere. Læreren som har deltatt i prosjektet er svært positiv til Super:bit, som har vært en nyttig ressurs som del av kompetansehevingen, men er samtidig tydelig på at hun har behov for å ytterligere utvikling av egen kompetanse.

Læreren i denne studien løser kompetanseutfordring ved å dra nytte av elevene. Gjennom en utforskende tilnærming i undervisningen og organisering i læringspar, legges det til rette for at elevene kan bruke hverandre som ressurser. Organiseringsformen fungerte stort sett godt i elevgruppa, ved at elevene støttet og veiledet hverandre i læringsarbeidet.

I undervisningsperioden har de algoritmiske arbeidsmåtene feilsøke, fikle, holde ut og samarbeid vært fremtredende. Dette synliggjør at det stilles krav til algoritmiske arbeidsmåter når elevene arbeider med oppgavene i Super:bit. Erfaringer fra undervisningsperioden kan tyde på at programmering egner seg godt for å arbeide med algoritmisk arbeidsmåter. Læreren vektla flere av de algoritmiske arbeidsmåtene i undervisningen, selv om hun tilsynelatende ikke hadde et bevisst forhold til fagbegrepene.

6.2 Refleksjoner og veien videre

Det er viktig å poengtere at denne masteroppgaven kun representerer én lærers erfaringer rundt tematikken, og funnene er dermed ikke representative for alle lærere. På en annen side kan naturalistisk generalisering føre til at funn fra denne masteroppgaven kan overføres til andre situasjoner. Studiet er presentert så transparent som mulig, slik at leserne av denne oppgaven selv kan vurdere om funnene er gyldige for dem og deres situasjon. Et annet særpreg ved denne oppgaven er at undervisningen foregikk på en fådelte skole. I og med at man også kan praktisere aldersblandet undervisning på fulldelte skoler, gjør det at denne masteroppgaven og de resultatene den viser også kan være aktuell for den fulldelte skolen. En kan også diskutere om organiseringen på den fådelte skolen fører til andre resultater, da det også vil være elever med ulik kompetanse og erfaringsbakgrunn på fulldelte skoler.

Programmering er et tidsrelevant tema, og det er grunn til å tro at det vil få større plass i skolen de neste årene. Det vil av den grunn være nyttig med flere studier som undersøker hvordan lærere arbeider med kompetansemålene i programmering i matematikkfaget. I dette prosjektet har programmeringsverktøyet micro:bit blitt benyttet i undervisningen. I videre forskning kan det vært interessant å se på hvilke erfaringer lærere gjør seg med bruk av andre programmeringsverktøy. I tillegg kan det være interessant å ta med elevenes perspektiv, for å synliggjøre hvordan de opplever undervisningen i programmering.

Litteraturliste

- Abril, A. M., Aguirre, D., Aldorf, A.-M., Andras, S., Antal, E., Ariza, M. R., Blomhøj, M., Boer, C. den, Bronner, P., Ceretkova, S., Doorman, M., Dorier, J.-L., Escobero, J. M., Farrugia, J., Febri, M. I. M., Garcia, Fco. J., Kooij, H. van der, Kontai, T., Lyngved, R., ... Len, R. (2013). *Inquiry-based learning in maths and science classes: What it is and how it works - examples - experiences* (K. Maaß & K. Reitz-Koncebovski, Red.). University of Education Freiburg. https://primas-project.eu/wp-content/uploads/sites/323/2017/11/primas_final_publication.pdf
- Anderson, T. & Shattuck, J. (2012). Design-Based Research: A Decade of Progress in Education Research? *Educational Researcher*, 41(1), 16–25. <https://doi.org/10.3102/0013189X11428813>
- Ardito, G., Mosley, P. & Scollins, L. (2014). We, robot: Using Robotics to Promote Collaborative and Mathematics Learning in a Middle School Classroom. *Middle Grades Research Journal*, 9(3), 73–88.
- Artigue, M. & Blomhøj, M. (2013). Conceptualizing inquiry-based education in mathematics. *ZDM – Mathematics Education*, 45(6), 797–810. <https://doi.org/10.1007/s11858-013-0506-6>
- Balanskat, A. & Engelhardt, K. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. European Schoolnet.
- Bell, T. & Vahrenhold, J. (2018). CS Unplugged-How Is It Used, and Does It Work? I H.-J. Böckenhauer, D. Komm, & W. Unger (Red.), *Adventures Between Lower Bounds and Higher Altitudes* (s. 497–521). Springer International Publishing.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A. & Engelhardt, K. (2016). *Developing computational thinking in compulsory education: Implications for policy and practice*. Publications Office of the European Union. <https://data.europa.eu/doi/10.2791/792158>
- Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association*, 1, 1–25.
- Bueie, H. (2019). *Programmering for matematikklærere*. Universitetsforlaget.
- Calder, N. & Taylor, M. (2010). Scratching below the surface: Mathematics through an alternative digital lens? I L. Sparrow, B. Kissane, & R. Hurst (Red.), *Shaping the Future of Mathematics Education: Proceedings of the 33rd annual conference of the*

- Mathematics Education Research Group of Australasia* (s. 117–124). Freemantle: MERGA.
- Engeln, K., Euler, M. & Maaß, K. (2013). Inquiry-based learning in mathematics and science: A comparative baseline study of teachers' beliefs and practices across 12 European countries. *ZDM – Mathematics Education*, 45(6), 823–836.
<https://doi.org/10.1007/s11858-013-0507-5>
- Forsström, S. E. & Kaufmann, O. T. (2018). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18–32. <https://doi.org/10.26803/ijlter.17.12.2>
- Fuglestad, A. B. (2010). Læringsfellesskap og inquiry. *Tangenten – tidsskrift for matematikkundervisning*, 21(4), 2–6.
- Haraldsrud, A. D., Sveinsson, H. A. & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Husain, H., Kamal, N., Ibrahim, M. F., Huddin, A. B. & Alim, A. A. (2017). Engendering problem solving skills and mathematical knowledge via programming. *Journal of Engineering Science and Technology*, 12(12), 1–11.
- Hussain, S., Lindh, J. & Shukur, G. (2006). The effect of LEGO Training on Pupils' School Performance in Mathematics, Problem Solving Ability and Attitude: Swedish Data. *Educational Technology & Society*, 9(3), 182–194.
- Høgheim, S. (2020). *Masteroppgaven i GLU*. Fagbokforlaget.
- Imsen, G. (2020). *Elevens verden. Innføring i pedagogisk psykiatri* (6. utg.). Universitetsforlaget.
- Johannessen, A., Christoffersen, L. & Tufte, P. A. (2021). *Introduksjon til samfunnsvitenskapelig metode* (6. utg.). Abstrakt forlag.
- Kalelioglu, F. & Sentance, S. (2020). Teaching with physical computing in school: The case of the micro:bit. *Education and Information Technologies*, 25(4), 2577–2603.
<https://doi.org/10.1007/s10639-019-10080-8>
- Kaufmann, O. T. & Stenseth, B. (2020). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 52(7), 1029–1048. <https://doi.org/10.1080/0020739X.2020.1736349>
- Kaufmann, O. T., Stenseth, B. & Holone, H. (2018). Programmering i matematikkundervisningen. I A. Norstein & F. O. Haara (Red.), *Matematikkundervisning i en digital verden* (s. 73–95). Cappelen Damm Akademisk.

- Kunnskapsdepartementet. (2017, 25. august). *Digitaliseringstrategi for grunnsopplæringen 2017–2021*. Regjeringen. <https://www.regjeringen.no/no/dokumenter/framtid-fornyelse-og-digitalisering/id2568347/>
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju*. Gyldendal akademisk.
- Kvam, V. (2013). Bygdeskoler med fådelt ordning – en aktuell politisk og pedagogisk utfordring. *Bedre skole*, 25(2), 19–23.
- Larsen, A. K. (2017). *En enklere metode veiledning i samfunnsvitenskapelig forskningsmetode* (2. utg.). Fagbokforlaget.
- Leuven, E. & Rønning, M. (2016). Classroom Grade Composition and Pupil Achievement. *The Economic Journal*, 126(593), 1164–1192. <https://doi.org/10.1111/eoj.12177>
- Lær Kidsa Koding. (u.å.-a). *Kurs*. Hentet 6. april 2022 fra <https://oppgaver.kidsakoder.no/>
- Lær Kidsa Koding. (u.å.-b). *Om Lær Kidsa Koding*. Hentet 8. september 2021 fra <https://www.kidsakoder.no/om-lkk/>
- Melheim, K. (1998). *Arbeid i fådelt skule*. Samlaget.
- Micro:bit Educational Foundation. (u.å.). *BBC micro:bit overview*. Hentet 6. april 2022 fra <https://microbit.org/get-started/user-guide/overview/>
- Moldøen, H. (2019, 10. mai). *Super:satsing på programmering med micro:bit for barn i hele Norge*. Lær Kidsa Koding! <https://www.kidsakoder.no/2019/05/10/supersatsing-pa-programmering-med-microbit-for-barn-i-hele-norge/>
- Mæland, M. & Myklebust, M. (2022). Programmering i matematikkfaget: Forventning, forvirring og forundring. *Tangenten – tidsskrift for matematikkundervisning*, 33(1), 20–26.
- Maaß, K. & Artigue, M. (2013). Implementation of inquiry-based learning in day-to-day teaching: A synthesis. *ZDM – Mathematics Education*, 45(6), 779–795. <https://doi.org/10.1007/s11858-013-0528-0>
- Nilssen, F. H. (2021, 7. mai). Fådelt skole. I *Store norske leksikon*. http://snl.no/f%C3%A5delt_skole
- NOU 2014: 7. (2014). *Elevenes læring i fremtidens skole- Et kunnskapsgrunnlag*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/NOU-2014-7/id766593/?ch=9>
- Nouri, J., Zhang, L., Mannila, L. & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1–17. <https://doi.org/10.1080/20004508.2019.1627844>
- Nygård, K. (2018). *Programmering i skolen*. Pedlex.

- Nätt, T. H. (2020, 16. april). Micro:bit. I *Store norske leksikon*. <http://snl.no/microbit>
- Olsen, H. Ø. & Aasland, M. (2013). *Læringspartner undervisningsvurdering i praksis*. Pedlex.
- Opheim, L. G. & Simensen, A. M. (2017). Matematikk- utforskning av mønstre og de store sammenhengene. I S. Bjørshol & R. Nolet (Red.), *Utforskning i alle fag* (s. 101–132). Cappelen Damm akademisk.
- Opplæringslova. (1998). *Lov om grunnskolen og den videregående opplæringa* (LOV-1998-07-17-61). Lovdata. <https://lovdata.no/dokument/NL/lov/1998-07-17-61>
- Parken, M. H. & Jakobsen, B. N. B. (2021). *Super:bits overføringsverdi til matematikk* [Masteroppgave, UiT Norges arktiske universitet]. UiT Munin. <https://munin.uit.no/handle/10037/22305>
- Postholm, M. B. & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm akademisk.
- Primas. (2011). *Guide for professional development providers*. University of Education Freiburg. https://primas-project.eu/wp-content/uploads/sites/323/2017/11/FINAL_WP4_Guide_PD_providers_licence_150708.pdf
- Ríos Félix, J. M., Zatarain Cabada, R. & Barrón Estrada, M. L. (2020). Teaching computational thinking in Mexico: A case study in a public elementary school. *Education and Information Technologies*, 25(6), 5087–5101. <https://doi.org/10.1007/s10639-020-10213-4>
- Rodríguez-Martínez, J. A., González-Calero, J. A. & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327. <https://doi.org/10.1080/10494820.2019.1612448>
- Sevik, K. (2016). *Programmering i skolen*. Senter for IKT i utdanningen. https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Shadish, W. R., Cook, T. D. & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin company.
- Sikko, S. A. & Grimeland, B. (2020). Kritisk matematisk literacy i ein inquiry-basert kontekst på småskulesteget. *Nordisk tidsskrift for utdanning og praksis*, 14(1), 104–117. <https://doi.org/10.23865/up.v14.2065>
- Skaalvik, E. M. & Skaalvik, S. (2021). *Skolen som læringsarena* (4. utg.). Universitetsforlaget.

- Statped. (2021, 1. mars). *Programmering*.
<https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/analog-programmering/>
- Stedøy, I. M. (2018). *Utforskende matematikkundervisning*. Realfagsløyper.
<https://realfagsloyper.no/sites/default/files/2018-08/T2.P1.M3A%20Artikkel%20Utforskende%20undervisning.pdf>
- Stigberg, H. & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures in Education*, 18(4), 483–496. <https://doi.org/10.1177/1478210319894785>
- Super:bit. (u.å.). *Hva er super:bit?* Hentet 16. april 2021 fra <https://www.superbit.no/hva-er-superbit/>
- Säljö, R. (2016). *Læring: En introduksjon til perspektiver og metaforer*. Cappelen Damm akademisk.
- Taylor, M., Harlow, A. & Forret, M. (2010). Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking. *Procedia - Social and Behavioral Sciences*, 8, 561–570.
<https://doi.org/10.1016/j.sbspro.2010.12.078>
- The Design-Based Research Collective. (2003). Design-Based Research: An Emerging Paradigm for Educational Inquiry. *Educational Researcher*, 32(1), 5–8.
<https://doi.org/10.3102/0013189X032001005>
- Tjora, A. (2021). *Kvalitative forskningsmetoder i praksis* (4. utg.). Gyldendal.
- Torkildsen, H. A. & Gjøvik, Ø. (2019). Algoritmisk tenkning. *Tangenten – tidsskrift for matematikkundervisning*, 30(3), 31–37.
- Utdanningsdirektoratet. (2017, 21. januar). *Kjerneelementer i matematikk, men hvorfor programmering?* Udirbloggen.no.
- Utdanningsdirektoratet. (2019a, 27. mars). *Algoritmisk tenkning*. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2019b, 3. juli). *Den teknologiske skolesekken*.
<https://www.udir.no/kvalitet-og-kompetanse/nasjonale-satsinger/den-teknologiske-skolesekken#a147262>
- Utdanningsdirektoratet. (2020). *Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020.
<https://www.udir.no/lk20/mat01-05>
- Vatne, J. E. (2021, 27. januar). Kvadrat. I *Store norske leksikon*. <http://snl.no/kvadrat>

- Vitensenterforeningen. (u.å.-a). *Super:bit skole*. Hentet 16. april 2021 fra
<https://www.vitensenter.no/superbit/>
- Vitensenterforeningen. (u.å.-b). *Tips, triks og feilsøking*. Hentet 14. oktober 2021 fra
<https://www.vitensenter.no/superbit/tips-triks-og-feilsoeking/>
- Vitensenterforeningen. (u.å.-c). *Utstyr til super:bit*. Hentet 16. april 2021 fra
<https://www.vitensenter.no/superbit/utstyr-til-superbit/>
- Vygotskij, L. S. (1978). *Mind in society: The development of higher psychological processes*.
Harvard University Press.
- Wing, J. M. (2010). Computational thinking: What and why? *Carnegie Melon School of
Computer Science Discussion Papers*.
<https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Aas, C. (2020). *Matematikklærere om programmering* [Masteroppgave, Universitetet i
Agder]. Aura. <https://uia.brage.unit.no/uia-xmlui/handle/11250/2680637>

Vedlegg

Vedlegg 1: Intervjuguide 1

Bakgrunn

1. Hvilke fag underviser du i?
2. Hvor lenge har du jobbet som lærer?
3. Hva legger du i begrepet programmering?
4. Hvilke erfaringer har du med bruk av programmering i skolen før denne undervisningsperioden?
 - a. Micro:bit
 - b. Super:bit-prosjektet
 - c. Bruk av andre digitale verktøy

Programmering inn i ny læreplan

1. Hva er dine tanker om at programmering har kommet inn i den nye læreplanen?
 - a. Ulike fag
2. Hva er dine tanker om programmering i matematikkfaget?
 - a. Hvordan vil du innføre det i din matematikkundervisning?
 - b. Hvilke matematiske-emner tenker du programmering passer inn under?
 - c. Hvilke utfordringer og muligheter ser du?

Super:bit

1. Fortell om hvordan din skole og du ble introdusert for Super:bit-prosjektet og de muligheter det har ført med seg.
 - a. Lærerkurs
 - b. Utstyrspakken
2. Hvilke tanker har du om prosjektet?

Elevenes erfaringsbakgrunn

1. Har elevene erfaring med programmering fra før?
 - a. Deltatt i super:bit-prosjektet?
 - b. Micro: bit
 - c. Andre digitale verktøy?

2. Motivasjon og interesse (Hvordan opplever du elevens motivasjon og interesse for å arbeide med programmering?)
3. Det matematikkfaglige? (Ferdighetsnivå i matematikk, hvilke holdninger har elevene til matematikkfaget?)

Aldersblandet gruppe

1. Hvilke tanker har du om ferdighetsnivå?
 - a. Store forskjeller?
2. Hvilke muligheter ser du for differensiering og tilpasninger?
3. Aldersblandet gruppe i programmering
 - a. Muligheter
 - b. utfordringer

Planlegging

1. Hvilke erfaringer har du gjort deg om planlegging av undervisningen?
2. Er det noe du opplever som spesielt utfordrende i planleggingen?

Forventninger til undervisningsgjennomføring

1. Forventninger
 - a. Hvilke forventninger har du til den kommende undervisningsperioden?
2. Mål
 - a. Hva er målet for perioden?
 - b. Hva ønsker du at elevene skal lære?

Oppsummering

1. Er det noe mer du ønsker å legge til eller utdype?

Avslutning

Takk for intervju.

Vedlegg 2: Intervjuguide 2

Gjennomføring av undervisningsprogram

1. Planlegging
 - a. Fortell om hvordan du planla undervisningen.
 - b. Var det noe du tok spesielt hensyn til/la vekt på i planleggingen?
2. Gjennomføring
 - a. Oppgaver
 - i. Noe som fungerte bra?
 - ii. Oppgaver som ikke fungerte?
 - b. Tidsbruk
 - c. Elevene
 - i. Hvordan opplevde du at elevene synes det var å jobbe med programmering?
 - ii. Motivasjon og interesse
 1. Hvordan opplevde du elevenes motivasjon for å arbeide med oppgavene?
 2. På hvilken måte viste eller viste ikke elevene interesse for programmering?
 3. Opplevde du at det var stor forskjell mellom elevene, og hva tenker du i så fall er årsaken til dette?
 - d. Lærerrolle
 - i. Beskriv hvordan du opplevde lærerrollen.
 - ii. Var det noe spesielt du la vekt på i lærerrollen?
 - iii. Møtte du på noen utfordringer i undervisningen?
 - e. Utstyrspakken
 - i. Noe du savner?
 - ii. Fungerer det slik det skal? Tekniske utfordringer?
 - iii. Tror du at du kommer til å bruke utstyret videre?
3. Etterarbeid/ evaluering
 - f. Gjorde du noen endringer mellom timene?
 - i. Hvis ja, hvilke endringer?
 - g. Ble undervisningen slik du hadde sett for deg?

- i. Hva tenker du er grunnen til dette?
 - ii. Noe du ville gjort annerledes dersom du skulle gjennomført undervisningen igjen?
- h. Samlet sett, hvordan evaluerer/vurderer du undervisningen?

Læringsutbytte

1. Hvordan opplever du elevenes læringsutbytte?
2. I hvor stor grad opplever du at dere oppnådde læringsmålene for undervisningen?
3. Hvordan opplevde du at det matematikkfaglige kom til uttrykk i undervisningen?
4. Hvordan vil du bruke programmering videre i din undervisning?
 - a. Tenker du at du kommet til å bruke ressursene fra super:bit videre?

Aldersblandet gruppe

1. Hvordan opplevde du elevenes ferdighetsnivå?
2. Hvordan opplevde du mulighetene for differensiering og tilpasninger?
3. Aldersblandet gruppe i programmering
 - a. Muligheter
 - b. utfordringer
4. Hvordan opplevde du at organiseringen med læringspartnere fungerte?
 - a. På hvilken måte opplevde du at elevene dro nytte av hverandre?

Programmering inn i ny læreplan

1. Hva legger du i begrepet programmering?
2. Hva er dine tanker om at programmering har kommet inn i den nye læreplanen etter denne perioden med programmering?
3. Tanker om programmering i matematikkfaget
 - a. Hvordan vil du arbeide videre med dette i din matematikkundervisning?
 - b. Hvilke emner?
 - c. Muligheter og utfordringer?

Oppsummering

1. Er det noe mer du ønsker å legge til eller utdype?

Avslutning

Takk for intervju.

11.05.2022, 18:05 Meldeskjema for behandling av personopplysninger

NSD NORSK SENTER FOR FORSKNINGSDATA

Vurdering

Referansenummer

613215

Prosjektittel

Programmering i matematikkfaget

Behandlingsansvarlig institusjon

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

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Lene Hayden Taraldsen, lene.hayden.taraldsen@hvl.no, tlf: 98821853

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Emilie Larsen, [REDACTED]

Prosjektperiode

01.09.2021 - 20.06.2022

Vurdering (1)

07.10.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg 07.10.2021. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 20.06.2022.

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.

PERSONVERNPRINSIPPER NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i

<https://meldeskjema.nsd.no/vurdering/615449ad-f257-4f26-a5e4-dbb02fcd17fd> 1/2

11.05.2022, 18:05 Meldeskjema for behandling av personopplysninger personvernforordningen om:

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om ogsamtykker 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: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), dataportabilitet (art. 20).

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

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

FØLG DIN INSTITUSJONS RETNINGSLINJER

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

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

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://nsd.no/personvernombud/meld_prosjekt/meld_endringer.html

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.

Lykke til med prosjektet!

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

Vil du delta i masterprosjektet

«*Programmering i matematikkfaget*»?

Dette er en forespørsel til deg om å delta i et masterprosjekt om programmering i matematikkfaget. I dette skrivet får du informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Gjennom grunnskolelærerutdanningen 5-10 på Høgskulen på Vestlandet skal jeg arbeide med et masterprosjekt i matematikdidaktikk. Formålet med prosjektet er å undersøke hvordan en lærer bruker sin kunnskap og ressursene fra super:bit-prosjektet i matematikkundervisningen, og få innsyn i hvordan dette fungerer i praksis. Prosjektet har som mål å belyse hvordan en kan arbeide med de nye kompetansemålene som omhandler programmering i matematikkfaget. Undervisningen vil foregå på en fådelt skole, med en aldersblandet elevgruppe på mellomtrinnet. Muligheter og utfordringer med undervisning i programmering i en aldersblandet klasse vil derfor også bli belyst.

Problemstillingen for oppgaven er:

Hvilke erfaringer gjør en lærer på en fådelt skole seg med bruk av ressursene fra super:bit-prosjektet i matematikkundervisningen i programmering på mellomtrinnet?

Hvem er ansvarlig for forskningsprosjektet?

Den ansvarlige for dette prosjektet er Emilie Larsen, som er masterstudent i matematikk på grunnskolelærerutdanningen 5-10 ved Høgskulen på Vestlandet. Prosjektet skjer under veiledning av Frode Olav Haara og Lene Hayden Taraldsen.

Hvorfor får du spørsmål om å delta?

Utvalget vil bestå av en lærer som arbeider på en fådelt skole som har tatt del i det nasjonale super:bit-prosjektet.

Hva innebærer det for deg å delta?

Deltakelsen i prosjektet innebærer å utarbeide et undervisningsopplegg i samarbeid med forskeren (Emilie Larsen). I forkant og etterkant av gjennomføringen av undervisningsopplegget vil det bli gjennomført et intervju. De to intervjuene vil begge ha et omfang på cirka en time. Det vil bli tatt lydopptak av intervjuene. Lydopptakene vil bli transkribert. I tillegg vil undervisningen bli observert. Emilie Larsen vil delta i undervisningen som en åpen, deltagende observatør.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. Det er kun jeg, Emilie Larsen, og mine veiledere, Frode Olav Haara og Lene Hayden Taraldsen, som vil ha tilgang til datamaterialet. Datamaterialet vil bli lagret passordbeskyttet i OneDrive. I oppgaven vil informanten bli anonymisert ved bruk av et fiktivt navn, og det er dermed ikke mulig å spore opplysninger tilbake til deg eller skolen der du arbeider.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Prosjektet skal etter planen avsluttes i mai 2022. Personopplysninger og datainformasjon vil bli slettet ved prosjektslutt, og senest desember 2022.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

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

Dine rettigheter

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

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

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

- Emilie Larsen [REDACTED]
- Høgskulen på Vestlandet ved Frode Olav Haara (Frode.Olav.Haara@hvl.no) eller Lene Hayden Taraldsen (Lene.Hayden.Taraldsen@hvl.no).
- Vårt personvernombud: Trine Anikken Larsen, tlf.: 55587682

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

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

Med vennlig hilsen

Emilie Larsen
Masterstudent

Frode Olav Haara og Lene Hayden Taraldsen
Veiledere

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «Programmering i matematikkfaget» og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i to intervjuer, hvor det vil bli tatt lydopptak
- bli observert i undervisning

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

(Signert av prosjektdeltaker, dato)