



Høgskulen
på Vestlandet

MASTEROPPGÅVE

Utprøving av analog programmering for å hjelpe lærar med å fremje skaparglede, engasjement og utforskartrong hjå elevar

Testing unplugged programming to assist teacher to promote joy of creation, engagement and urge to explore among students

Øystein Riksheim Østvik

Master i matematikdidaktikk (MGUMA 550)

Fakultet for lærarutdanning, kultur og idrett (FLKI)

Rettleiarar: Odd-Eivind Holo og Karin Elisabeth Sørli Street

15. Mai. 2023

Eg stadfestar at arbeidet er sjølvstendig utarbeida, og at referansar/kjeldetilvisingar til alle

kjelder som er brukt i arbeidet er oppgitt, jf. *Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.*

Forord

Då var eg endeleg ferdig med fem fine år på HVL i Sogndal. Det har vore eit turbulent studie prega av både omstrukturering til femåring lærarutdanning og Covid19 midt i studielauget. Sjølv om det har vore utfordrande til tider har det har vore ei flott tid der eg har utvikla meg og tatt eit stort steg som person. Eg har dei siste året på studiet flytta heim og masteroppgåva mi har på den måte blitt skreve på deltid. Her må eg derfor utlyse ein takk til Halbrend Skule som har vore fleksible i samband med arbeidsoppgåvene mine og alltid sørge for at studiet fekk førsteprioritet.

Så vil eg rette ein stor takk til mine to rettleiarar Karin Elisabeth Sørli Street og Odd-Eivind Holo. Dei har vore ein fantastisk støtte gjennom heile prosessen, og eg er sikker på at oppgåva mi hadde vore langt mindre vellykka utan deira profesjonelle rettleiing og konstruktive tilbakemeldingar. Dei har alltid vore oppmuntrande, støttande, og hjelpt meg med å halde fokus og motivasjon når det har vore vanskeleg.

Øystein Riksheim Østvik

Sogndal, mai 2022

Samandrag

Forskning seier at skular, lærarar og studentar har behov for at forskingsmiljøet held fram med å arbeide med undervisning i programmering i grunnskulematematikk. Eit sentralt mål for forskinga bør vere å hjelpe lærarar å fylle forventingane til elevane sin digitale kompetanse. Eit anna mål vil vere å undersøke moglegheiter og utfordringar knytt til det å lære matematikk gjennom utforsking ved hjelp av algoritmisk tenking. Hensikta med denne masteroppgåva vil vere å finne ut om analog programmering kan vere ein god inngang til bruk av programmering for læraren i matematikkundervisninga, samt leggje til rette for at eleven skal kunne fremje skaparglede, engasjement og utforskartrong.

Med dette som bakgrunn og interesse for temaet har eg formulert følgjande problemstilling:

I kva grad og korleis kan analog programmering vere ein attraktiv tilnærming for å nå måla om skaparglede, engasjement og utforskartrong i matematikkundervisninga?

For å svare på problemstillinga har eg nytta meg av eit forskingsdesign inspirert av forskingsmetodikken Design-based research (DBR). I oppgåva har det blitt samla inn kvalitative data gjennom observasjon av ei klasse på 6 trinn i den norske skulen, samt intervju frå læraren i denne klassa. Denne læraren har eg hatt eit tett samarbeid med gjennom heile perioden, og vi har saman utvikla eit undervisningsopplegg med utgangspunkt i problemstillinga der vi aktivt søkte å gjere opplegget attraktivt for lærar og samtidig nå målet om å fremje skaparglede, engasjement og utforskartrong.

Resultat frå studien viser at læraren har vore svært positiv til undervisningsopplegget med analog programmering. Studien har lagt til rette for ein lettare inngang til programmeringsundervisninga, og vist seg å vere attraktivt for lærar. Ut i frå resultata i denne studien ser det ut til at undervisningsopplegget med analog programmering fremja utforskartrong og engasjement hjå elevane, mens det er meir usikkert i kva for ein grad opplegget fremja deira skaparglede.

Abstract

Research suggests that schools, teachers, and students need the research community to continue working on integrating programming into elementary school mathematics education. A key goal of the research should be to help teachers meet the expectations for students' digital competence. Another goal would be to explore the possibilities and challenges of learning mathematics through exploration using algorithmic thinking. The purpose of this master's thesis is to determine whether unplugged programming can be an effective approach for teachers to introduce programming in mathematics education, while also promoting students' joy of creation, engagement, and urge to explore.

With this background and interest in the topic, I have formulated the following research question:

To what extent and how can unplugged programming be an attractive approach to achieve the goals of fostering joy of creation, engagement, and urge to explore in mathematics education?

To address the research question, I have employed a research design inspired by the Design-Based Research (DBR) methodology. In this thesis, qualitative data has been collected through classroom observations of a sixth-grade class in a Norwegian school and interviews with the teacher of this class. I have maintained a close collaboration with this teacher throughout the entire period, and together we have developed a teaching program based on the research question, actively striving to make it appealing to the teacher while achieving the goal of promoting joy of creation, engagement, and curiosity.

Findings from the study indicate that the teacher has been highly positive about the teaching program involving unplugged programming. The study has provided an easier entry point to programming education and has proven to be attractive to the teacher. Based on the findings of this study, it appears that the teaching program using unplugged programming has fostered curiosity and engagement among the students, although the extent to which it has promoted their joy of creation is somewhat uncertain.

Innhold

1	Innleiing	8
1.1	Bakgrunn	9
1.2	Struktur på oppgåva.....	12
2	Teoretisk rammeverk.....	13
2.1	Programmering.....	13
2.1.1	Historisk blick på programmering i skulen	14
2.1.2	Skulen i dag	16
2.1.3	Programmeringsspråk.....	17
2.1.4	Analog programmering	19
2.1.5	Algoritmisk tenking.....	20
2.2	Programmeringspedagogikk og kompetanse i programmering	22
2.3	Didaktiske rammeverk innanfor programmering og matematikk.	24
2.3.1	Benton 5 e-r.....	25
2.3.2	Thinking Classrooms.....	27
2.3.3	Primm-modellen.....	30
2.4	Oppsummering	32
3	Metode.....	34
3.1	Design.....	34
3.2	Deltakarar, Informant og forskarrolle.....	38
3.3	Datainnsamlingsverktøy	39
3.3.1	Intervju	40
3.3.2	Observasjon	41
3.3.3	Pilotering	45
3.4	Utvikling av intervensjon over fleire forskingssyklusar.....	46
3.4.1	Første økt.....	49
3.4.2	Andre økt.....	51
3.5	Analyse av data	54
3.6	Reliabilitet og validitet	56
3.7	Etikk	58
4	Presentasjon av resultat	60
4.1	Oversikt	60
4.2	Gjennomføring av intervensjonen	63
4.3	Svar på forskings spørsmål 1.....	63
4.3.1	Oppsummering frå intervju med lærar	64

4.3.2	Oppsummering av observasjonar gjort under intervensjonen	65
4.3.3	Anna observasjon	67
4.4	Svar på forskingsspørsmål 2.....	68
4.4.1	Oppsummering av lærar sine tankar og erfaringar etter intervensjon	68
4.5	Oppsummering av resultat.....	70
5	Drøfting	72
5.1	Analog programmering for å fremje skaparglede.....	72
5.2	Analog programmering for å fremje engasjement.....	72
5.3	Analog programmering for å fremje utforskartrøng?	74
5.4	Analog programmering som noko attraktivt for lærarar	75
6	Slutt	76
6.1	Konklusjon	76
6.2	Refleksjon og vegen vidare	77
	Litteraturliste	78
	Vedlegg	85
	Vedlegg 1: Intervjuguide 1	85
	Vedlegg 2: Intervjuguide 2.....	86
	Vedlegg 3: Intervjuguide 3.....	87
	Vedlegg 4: Godkjenningsskriv frå NSD	88
	Vedlegg 5: Informasjonsskriv og samtykkjeskjema.....	89
	Vedlegg 6: Symbol til undervisningsopplegg	92
	Vedlegg 7: Kode til undervisningsopplegg	93
	Vedlegg 8: Oppgåver til undervisningsopplegg	95

Tabelliste

TABELL 1: 5 STEG I PRIMM. (PRIMMPORTAL, 2022)	31
TABELL 2: PRINSIPP FOR Å NÅ MÅL I UNDERVISNING	47
TABELL 3: UNDERVISNINGSOPPLEGG 1 I KVA, KORLEIS, KVIFOR SKJEMA	51
TABELL 4: UNDERVISNINGSOPPLEGG 2 I KVA, KORLEIS, KVIFOR SKJEMA	53
TABELL 5: OVERSIKT OVER DATAMATERIALE MED UTGANGSPUNKT I UNDERVISNINGSOPPLEGG	62

Figurliste

FIGUR 1: ALGORITMISK TENKING, PROGRAMMERING OG KODING I KONTEKST (ETTER EIN IDE AV LIV ODDRUN VOLL, NATURFAGSENTERET UIO)	14
FIGUR 2: SKJERMDUMP FRÅ SCRATCH.MIT.EDU	18
FIGUR 3: DEN ALGORITMISKE TENKEREN (UTDANNINGSDIREKTORATET, 2019)	21
FIGUR 4: VURDERING AV ALGORITMISK TENKING (HTTPS://SCRATCH.MIT.EDU)	22
FIGUR 5: FORDELING AV ELEVAKTIVITET I KLASSEROMMET (LILJEDAHL, 2021, S. 10)	27
FIGUR 6: DESIGN AV STUDIEN	34
FIGUR 7: FIRE SENTRALE FASAR I DBR OVERSETT TIL NORSK (SCOTT ET AL 2019)	37
FIGUR 8: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 12	43
FIGUR 9: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 13	43
FIGUR 10: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 22	44
FIGUR 11: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 23	45
FIGUR 12: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 5	46
FIGUR 13: DEN DIDAKTISKE RELASJONSMODELLEN (BJØRNDAL OG LIEBERG 1978 I HIIM OG HIPPE 2022)	48

1 Innleiing

Holo et al. (2022) skriv at skular, lærarar og studentar treng at forskingsmiljøet held fram arbeidet med undervisning i programmering i grunnskulematematikk. Eit sentralt mål for forskning bør vere å hjelpe lærarar å fylle forventingane til elevar sin digitale kompetanse. Eit anna mål vil vere å undersøke moglegheiter og utfordringar knytt til det å lære matematikk gjennom utforsking ved hjelp av algoritmisk tenking. Eg vil sjå om analog programmering kan vere ein god inngang til bruk av programmering for læraren i matematikkundervisninga, samt leggje til rette for at eleven skal kunne utfalde skaparglede, engasjement og utforskartrøng. Det vil bli interessant å sjå kva erfaringar elevar og lærar gjer ved bruk av analog programmering, og om det kan vere ein god inngang til å lære om programmering i den norske grunnskulen. Hensikta med denne masteroppgåva vil vere å finne ut om analog programmering kan vere ein god inngang til bruk av programmering for læraren i matematikkundervisninga, samt leggje til rette for at eleven skal kunne fremje skaparglede, engasjement og utforskartrøng.

Med dette som bakgrunn og interesse for temaet har eg formulert følgjande problemstilling:

I kva grad og korleis kan analog programmering vere ein attraktiv tilnærming for å nå måla om skaparglede, engasjement og utforskartrøng i matematikkundervisninga?

Og for å svare på dette spørsmålet, tar ein utgangspunkt i to forskingsspørsmål:

1) *«I kva for ein grad og korleis kan analog programmering leggje til rette for 6. klasse elevars skaparglede, engasjement og utforskartrøng?»*

2) *«I kva grad og korleis kan eit undervisningsopplegg med analog programmering vere attraktivt for lærarar?»*

For å danne ei forståing av kva ein legg i dei forskjellige omgrepa har eg valt å definere skaparglede, engasjement og utforskartrøng følgjande:

Skaparglede kan ein omsetja til ein positiv og motiverande haldning til å utforske, skape og vere kreativ i læringsprosessen. Det inneber å ha glede av å generere og uttrykke eigne idear, finne løysingar på problem og vere nysgjerrig på å utforske nye konsept og moglegheiter.

Engasjement handla om å være djupt involvert, interessert og fokusert i læringsaktiviteter. Det inneber å vise ein form for intensitet og vedvarande innsats i oppgåver eller emnar, samt oppleve ein følelse av mening og relevans i det som blir lært. Engasjement kan inkludere

motivasjon, oppslutning og dedikasjon til å oppnå mål og utføre oppgaver på en grundig og reflekterende måte.

Utforskertrang kan ein sjå på det å vere nysgjerrig og ha ein aktiv haldning til å utforske og oppdage nye idear, konseptet og ferdigheiter. Det inneberer en vilje til å stille spørsmål, undersøke, eksperimentere og søke kunnskap på egen hand. Utforskertrang inneber og evna til å tenke kritisk, være open for ulike perspektiv og vere villig til å ta risiko og lære av feil.

Vidare vil eg i denne oppgåva definere «attraktivt» som noko som både er lett å gjennomføre og givande for lærar å bruke i undervisning.

1.1 Bakgrunn

Bakgrunnen til at eg har valt å sjå nærmare på programmering og spesielt analog programmering er at verda i dag blir stadig meir digitalisert og utfordrar måten vi lærer på og kva kompetanse som blir viktig i tida framover. Senter for IKT i utdanningen skriv at ein «Dypere forståelse av underliggende prosesser og systemer, evne til logisk tenkning og ferdigheter i å være skapende og produsere har alltid vært viktige i læring og utdanning» (Sevik, 2016 s. 6). Samtidig det å skape meningsfulle læringsprosesser, utvikle skapende evner og djup forståing krevje andre tilnærmingar enn gårdsdagens skulesystem. Digit-utvalet påpeikte i sin hovudutgreiing (NOU 2013:2) at eit av hindera for digital verdiskaping er manglande kompetanse i programmering blant befolkninga, og eit behov for å legge til rette for at barn og unge ikkje berre er i stand til å bruke, men og skape digitalt innhald og digitale tenester. I 2015 kom Ludvigsen-utvalet med si hovudutgreiing «Fremtidens skole – Fornyelse av fag og kompetanser», med understreking av korleis skulen og skulefaga bør endrast for å møte framtida sitt kompetansebehov (NOU 2015:8).

Desse utgreiingane i lag med andre sentrale aktørar, som til dømes Europakommisjonen og regjeringa sine 8 kompetansar for framtida (NOU 2014:7), samt utvikling av eit meir digitalisert samfunn, gjorde at programmering dei siste åra fekk eit langt større politisk fokus i forhold til korleis skule og utdanningssektoren forvaltar krava til bruk og utvikling av ny teknologi. I fagfornyelsen 2020 kom det tydelege kompetansemål som fekk eit klart feste i matematikkundervisninga. Det er likevel stor usemje om programmeringa si plass i matematikkfaget og skulen generelt, både i Norge og internasjonalt (Kaufmann et al, 2018). Nokre hevdar til dømes at det å ha kunnskap om programmering er så grunnleggjande i dagens samfunn at det bør sjåast på som eit eige fag og sidestillast med basisfaga norsk,

engelsk og matematikk i staden for ein del av matematikken (Nardelli, 2019). Sjølv om det vert diskutert om programmeringa si plass i undervisinga, er det stor semje om at programmering bør inn i skulen (Kaufmann et al., 2018). Argument som støttar dette er blant anna at programmering kan utvikle elevar sin logiske tankegang og evne til problemløysing (Kaufmann et al., 2018). Andre argument for å innføre programmering i skulen er primært retta mot dei digitale kunnskapane som «twenty-first-century-skills» (NOU 2015:8), framtidige behov for kompetanse i næringslivet og evna til å forstå korleis eit stadig meir digitalisert samfunn fungerer.

Utdanningsdirektoratet gjennom senter for IKT skriv at tradisjonelt sett har programmering vore eit omgrep for å skrive programkode (Selvik, 2016). Altså det å kunne lage instruksjonar til digitale einingar for at dei skal kunne utføre ei eller fleire oppgåver. Programmering inkluderer det å skrive kode, men og forklare kva programmet skal gjere ved hjelp av algoritmar. I matematikk er ein algoritme ein fullstendig og nøyaktig skildring av framgangsmåten for løysing av ei oppgåve (Hovde & Grønmo, 2020). Eit døme på kva ein algoritme er, kan vere ei oppskrift der ein følgje ein bestemt framgangsmåte. For å skape algoritmar må vi tenke algoritmisk, og her kjem vi innom eit sentralt omgrep som blir kalla algoritmisk tenking. Algoritmisk tankegang kan ein då sjå på som ein problemløysingsprosess som handlar om det å tilnærma seg eit problem på ein systematisk måte, og ta i bruk digitale hjelpemidler for å løyse problemet. Utdanningsdirektoratet (2019) definerer det på følgjande måte:

«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»

I kompetansemål for matematikk er algoritmisk tenking plassert under kjerneelementa «utforsking og problemløysing». Her blir altså algoritmisk tenking løfta fram i fagfornyinga 2020 som ein av problemløysingsmetodane som er gunstige å bruke (utdanningsdirektoratet 2020). Grunna dette har Utdanningsdirektoratet laga ein modell (Den algoritmiske tenkaren, figur 3) som støttar lærar i arbeid med algoritmisk tenking. Denne kjem vi tilbake til seinare i oppgåva.

Skaparglede, engasjement og utforskartrøng er noko alle faga i skulen skal bidra til (Opplæringslova, 1998). Engasjement og interesse er føresetnader for å nå heilt sentrale mål for opplæringa, også i matematikk. I den nye overordna delen står det vidare om opplæringas verdigrunnlag at elevane skal kunne utvikle skaparglede, engasjement og utforskartrøng (Kunnskapsdepartementet, 2017). Det kjem òg fram i overordna del av læreplanen at elevane skal delta i ulike aktivitetar av stadig aukande kompleksitet for å få god kompetanse i faga. Dette er noko eg vil bruke vidare å sjå om det er mogleg å fremje desse sentrale måla i arbeidet med programmering i matematikkundervisninga.

Eit sentralt problem for mange er at fleire norske lærarar kjenner på ei uvisse knytt til programmering og algoritmisk tenking (Moreau, 2021), og i høyringsinnspel til fagfornyninga kjem det fram at både fagmiljø og lærarar meina at skular ikkje er godt nok budde til å laga god undervisning med programmering. Dette vert støtta opp av Kjällander et al. (2018) som skriv at det er stor bekymring om mangelfull kompetanse på programmeringsfeltet, og også av Nordby et al (2022), som i sin studie kom fram til at fleire lærarar uttrykte at dei var usikre og mangla kompetanse rundt korleis algoritmisk tenking kom fram i læreplanen og på kva algoritmisk tenking faktisk er. Vidare skriv Nordby et al (2022) at mange av lærarane ønska kompetanse på korleis dette kunne bli inkludert i matematikkundervisninga.

Ein god inngang til å lære om programmering og algoritmisk tenking kan vere analog programmering. For å kunne hjelpe lærarar med utfordringar rundt programmering har Whol, Porter og Clinch (2015) gjennomført ein større internasjonal samanlikningsstudie. I studien blei det rapportert at læring ved bruk analog programmering gav ein høgare grad av forståing om konseptet algoritmisk tenking samanlikna med meir tradisjonelle programmeringsverktøy. Analog programmering kan omsetjast til «unplugged programming» (Humble et al., 2020), som i korte trekk går ut på å lage ei oppskrift eller løyse eit problem utan å bruke digitale einingar. Det å programmere utan ein digital eining kan vere med på å lære elevane samanheng innan programmering, slik dei får jobba med forståing av korleis og kvifor ting heng saman innan programmering (Statped, 2021a). Dømer på dette kan vere korleis ein lagar en kopp te, pussar tenner eller lagar seg ei brødiskive med Nugatti. Ifølgje Bell og Vahrenhold (2018) gjer analog programmering det altså mogleg å jobbe med programmering utan at elevane har tekniske kunnskapar om programmering. Internasjonal forskning som undersøker korleis lærarar og elevar i grunnskulen oppfattar analog programmering (Bell & Vahrenhold, 2018; Cortina, 2015; Humble et al., 2020), indikerer at både elevar og lærarar ser fordelar ved analog programmering.

I Norge er det gjort studiar knytt til læraren og elevar si oppfatning av programmering knytt til digitale plattformer, som til dømes Scratch (Giannakos et al., 2013). Det same er i midlertidig ikkje gjort for analog programmering, og det kan derfor vere interessant å undersøke kva for ein rolle analog programmering har eller kan ha i norsk skule. Tonje Berg frå Universitetet i Sørøst-Norge har publisert i Tangenten: tidskrift for matematikkundervisning at analog programmering kan vere ein god inngang til nettopp å legge til rette for bruk av programmering (Berg, 2021). Berg meiner midlertidig at det er viktig å understreke at studien tar utgangspunkt i eit lite utval og at det er behov for vidare forskning.

1.2 Struktur på oppgåva.

I kapittel to presenterer eg det teoretiske rammeverket som bli nytta i studia. Her vil eg legge vekt på forhistorie, programmeringsteori, analog programmering, algoritmisk tenking og korleis ein kan undervise programmering i skulen. Her går eg djupare inn på kva ein kan gjere for at både elev og lærar skal dra nytte av bruk av programmering i skulekvardagen på ein positiv og gjennomførbar måte, og som vil bidra til skaparglede, engasjement og utforskartrøng. I kapittel tre blir metodedelen presentert. Her gjer eg greie for val av metode og instrument, kvalitativ metode, og intervju og observasjon som instrument. Eg vil deretter forklare framgangsmåten som blei brukt for å innhente empirien, og sjå på ulike etiske utfordringar som er knytt til ein slik studie. I kapittel fire presentera eg resultat frå analysen av observasjon- og intervju-data. Her var det ein kontaktlærar og elevane hans på 6. trinnet som blei observert og lærar som vart intervjuet. I kapittel fem drøftar eg resultatata opp mot det teoretiske rammeverket som blei presentert i kapittel to samt kva implikasjonar desse resultatata kan ha for praksisfeltet, til dømes lærarars undervisning i programmering. I kapittel seks vil eg oppsumere resultatata mine, kome med ein konklusjon og evt. forslag til vidare forskning.

2 Teoretisk rammeverk

Dette kapitlet startar eg med å sjå på forhistorie til programmering i skulen samt omgrepsavklaringar rundt programmering, programmeringa si plass i skulen og programmeringsspråk. Dette har eg valt å ta med sidan programmering er eit vanskeleg emne som mange i skulesystemet ikkje har god kontroll på. Med å sjå litt på forhistoria vil vi kunne lettare danne oss eit bilete kva som tidlegare har blitt lagt i programmering og korleis det har påverka dagens undervisning. Vidare vil eg presentere algoritmisk tenking og kvifor det er sentralt innanfor programmeringstemaet, samt sjå på korleis ein kan undervise programmering i skulen. Eg vil vidare gå djupare inn på læringsperspektiva til Lisa Benson med sine fem E-ar og Peter Liljedahl sitt tenkande klasserom (Thinking Classroom), og Susan Sentance sin PRIMM-modell der eg vil sjå nærmare på korleis ein kan bruke programmering i matematikkundervisninga best mogleg med utgangspunkt i relevant læringsteori. Dette vil, i tillegg til tidlegare forskning, utgjere det teoretiske rammeverket for denne masteroppgåva. Til slutt vil eg summere kapitlet og peike framover mot utviklinga av eit analytisk rammeverk for prosjektet.

2.1 Programmering

Tradisjonelt har omgrepet programmering blitt brukt for aktiviteten der ein skriv programkode, altså lage instruksjonar til datamaskin og andre digitale einingar for å utføre ei oppgåve. I nyare tid handlar programmering om meir enn det å skrive programkode, det inkluderer og prosessen med å kome fram til koden. Det vil seie prosessen frå å identifisere eit problem, tenkje ut moglege løysingar for å så skrive kode som kan bli forstått av ei digital eining, og feilsøke og kontinuerleg forbetre koden. Denne forståinga blir støtta 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).

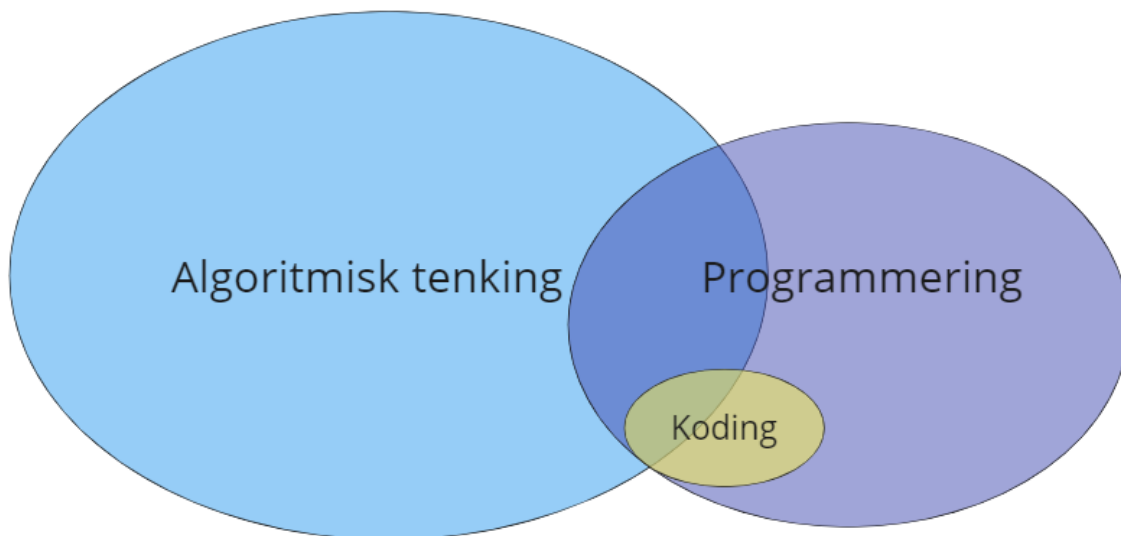
Nygård problematiserer at omgrepet programmering har skapt mykje forvirring og usemje i skulen, då spesielt i skilje mellom programmering, koding og algoritmisk tenking. Denne problemstillinga blir og trekt fram av Kaufmann et al. (2018). Dei skildra at det i enkelte

tilfelle kan vere eit skilje i korleis omgrepa blir brukt. Omgrepet koding forbind ein gjerne med å lage mindre program med avgrensa kompleksitet og konsekvens, mens programmering på si side er meir omfattande og eit eige fag. Likevel blir det argumentert for at det ikkje er noko klart definert skilje mellom omgrepa programmering og koding, og at desse omgrepa i det kvardagslege blir brukt om kvarandre (Kaufmann et al., 2018). I læreplanen er det, som nemnt i innleiinga, omgrepet programmering som blir brukt. På bakgrunn av dette vil eg nytte omgrepet programmering vidare i oppgåva.

I dei nye læreplanane blir algoritmisk tenking i tillegg til programmering vektlagt.

Programmering kan sjåast på som ein del av algoritmisk tenking, og er eit verktøy som kan bli brukt til å løyse problem. I konteksten av den norske læreplanen blir det difor viktig å differensiere mellom koding og programmering. Det å skrive programkode og få ei datamaskin til å utføre koden, er kun ein liten del av programmeringa (Sjå figur 1).

Programmering er alle prosessar knytt til korleis ein kjem fram til ei løysing.



FIGUR 1: ALGORITMISK TENKING, PROGRAMMERING OG KODING I KONTEKST (ETTER EIN IDE AV LIV ODDRUN VOLL, NATURFAGSENTERET UIO)

2.1.1 Historisk blick på programmering i skulen

Introduksjonen til bruk av programmering i undervisinga blir ofte kalla «den første bølga». Formålet var å lage programmeringsspråk der yngre elevar kunne lære matematikk. Denne utviklinga starta på slutten av 1960-talet. I 1967 blei programmeringsspråket Logo laga

(gjennom «The Logo Project») av Seymour Papert og hans kollegaer på MIT (Massachusetts Institute of Technology) (Feurzeig et al., 1969). Logo blei laga som eit spesialisert verktøy for programmering der Papert introduserte «objects-to-think-with», nærmare bestemt ein virtuell skjelpadde som kunne bli programmert til å utføre kommandoar (Papert, 1980).

Seymour Papert var ein pioner for sitt arbeid med å introdusere programmering i undervisinga, og han tok til ordre for ei storstilt skulereform på 1980-talet. Papert hadde eit grunnleggjande syn på at læringsmiljø blir skapt ved å nytte datamaskiner, og at dette kunne forandre måten elevane lærte (Papert, 1980). Elevane kunne programmere relativt avanserte program med Logo, og Papert argumenterte for at læringsprosessane elevane tileigna seg ved å gjere denne typen oppgåver burde vere ideal for elevane si kognitive utvikling: « ... *I believe that certain uses of very powerful computational technology and computational ideas can provide children with new possibilities for learning, thinking, and growing emotionally as well as cognitively*» (Papert, 1980, s. 17).

Pea & Kurland (1984) var ein av Papert sine største kritikarar til påstanden om at programmering med Logo ville gje auka kognitiv effekt. Dei argumenterte med at det var studiar som viste at programmering ikkje førte til betre problemløysing eller matematisk forståing i bruk av variablar i matematikk (Pea & Kurland, 1984). Samstundes var dei einig i at programmering og problemløysing delte nokre eigenskapar som kunne fremje tenkingsdugleik. Pea & Kurland (1984) påpeika at det var naudsynt med meir empirisk forskning på om programmering ga fagleg utbytte og overføringsverdi.

Mayer et al (1986) var og kritiske til Papert sin påstand om at å lære å programmere ga forbetra tenking i andre fag. Mayer et al (1986) påpeika at deira forskning oppmuntra ideen om at programmering kunne gje ein positiv effekt på tankeferdigheit direkte relatert til språket som vart lært. Dei argumenterte for at det på dette tidspunktet ikkje låg føre bevis for at programmering betra elevane sine intellektuelle eigenskapar meir enn andre fag (Mayer et al, 1986)

Forskinga som blei utført på 1980-talet, oppsumerte i si heilheit at programmering ikkje førte til generell problemløysing med overføringsverdi til andre fag. Den pedagogiske revolusjonen som oppfordra til auka bruk av programmering i undervisinga fekk redusert ambisjon, og utviklinga dabba av.

2.1.2 Skulen i dag

Sidan den gong har det kome mykje forskning som støtter opp under synet at programmering er ein viktig ferdigheit som høyrer heime i skulen, og sjølv om programmering ikkje i seg sjølv blir nemnt i den norske læreplanen kan ein rekke kompetanssmål i fleire fag blir nådd ved hjelp av programmering (Sevik, 2016 s. 19). I 2006 kom difor ein ny skulereform kalla LK06 (Kunnskapsløftet 2006). Den la til grunn fem nye grunnleggjande ferdigheiter som elevane skulle tileigne seg gjennom skulelaupet: å kunne lese, skrive og rekne, og ha munnleg og digital kompetanse. På den tida var digitale ferdigheiter rekna som det å kunne bruke digitale verktøy. Skulane forplikta seg til å bruke ressursar som prioriterte at elevane fekk oppfølging til å lære seg dei grunnleggjande ferdigheitene. Den nye læreplanen som blei utarbeida i LK06 fekk og klare og tydelege kompetanssmål, og det blei lagt til rette for at skulane sjølv kunne ha friheit til å lage eigne, lokale læreplanar basert på dei nasjonale. Rammeverk for grunnleggjande ferdigheiter blei revidert av utdanningsdirektoratet i 2012, og digital kompetanse blei erstatta med digitale ferdigheiter. Meir presist tydde dette at lærarar skulle bruke digitale verktøy i undervisninga, og hjelpe elevane i utviklinga av deira digitale ferdigheiter.

I 2020 kom Kunnskapsløftet (LK20), som og var den største innhaldsreforma i norsk skule sidan LK06. Ulike utval gjorde greie for og la grunnlaget for fagfornyninga som reform. Med introduksjonen av nye kjerneelement i 2018 (utdanningsdirektoratet, 2018) har antall tema i faga blitt redusert. For matematikkfaget inkluderer dei algoritmisk tenking med utforsking og problemløysing, abstraksjon og generalisering. Dei nye kompetansmåla i læreplanen for matematikk i grunnskulen fekk fornya innhald som i stor grad viser at algoritmisk tenking og programmering blir vektlagt (utdanningsdirektoratet, 2020). Under har eg lista opp kompetanssmål i programmering/algoritmisk tenking for ulike trinn for matematikkfaget i grunnskulen.

2. trinn: Lage og følgje reglar og trinnvise instruksjonar i leik og spel

4. trinn: Lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer

5. trinn: Lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

8. trinn: Utforske korleis algoritmar kan skapast, testast og forbeholdt ve hjelp av programmering

9. trinn: Simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering

10. trinn: Utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Som vi ser så vil elevar allereie frå 2. klasse bli introdusert for oppbygginga av algoritmar. Frå 5. klasse står det spesifisert (i den nye læreplanen for matematikk) at programmering skal nyttast for å løyse algoritmar ved bruk av variablar, vilkår og løkker. Det første elevane møter av programmering er ofte blokkbasert programmering, i form av Scratch eller Microbit. Det skal vi gå nærmare inn på i neste delkapittel.

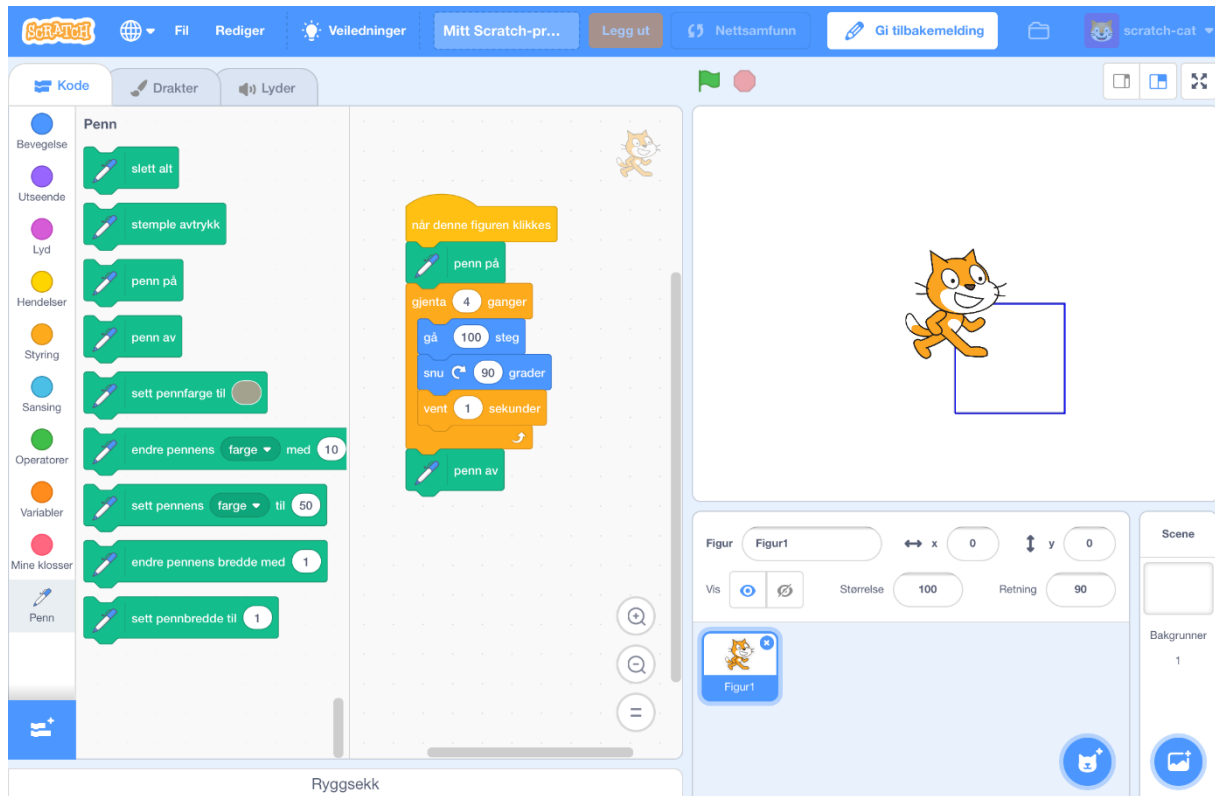
2.1.3 Programmeringsspråk

Ein grunnleggande føresetnad for programmering er at ein prosessor (ein komponent i ei digital eining som tolkar og utfører instruksjonar) kan lese og utføre instruksjonar frå maskinkode. Ein forenkla måte å skrive maskinkode på er med assemblerspråk. Dette er et meir uleseleg tekstbasert språk, som til dømes den klassiske binære maskinkoden med einarar og nullar. Assemblerspråk går føre seg på eit lågt nivå av kompleksitet kjent som låg-nivå programmeringsspråk (Vihovde, 2019).

Høg-nivå programmeringsspråk er eit meir leseleg tekstbasert språk enn assemblerspråk (lav-nivå). Høg-nivåspråka er laga slik at eit definert kodespråk kan nyttast for å få den digitale eininga til å løyse problema. Desse språka blir brukt til å utføre utrekningar og prosessere data utan å vite noko om maskinkode eller assemblerspråk. For å skildre korleis høg-nivå programmeringsspråk fjernar kompleksitet kan ein dra parallell til det å køyre ein bil. Kunnskap om ratt, pedalar og gir trengs for å operere bilen, medan det som skjer inni motoren ikkje er ein like viktig kunnskap for at ein person skal kunne køyre frå A til B.

Python er eit eksempel på høgnivå-språk som blir mykje brukt i dag. Det er fleire årsaker til at dette har blitt eit naturleg val av programmeringsspråk nytta i programmeringsundervisning i skulen. Samanlikna med andre programmeringsspråk som Java og C, har Python vist seg å vere lettare å lære for nybyrjarar i programmering. Python blir sett på som lett å lese, og har ein enkel syntaks som er nært knytt til korleis ein uttrykkjer seg matematisk (Haraldsrud, Sveinsson, & Løvold, 2020). For mange elevar let det seg likevel ikkje gjere å starte med Python som programmeringsspråk på eit så høgt nivå. For å kunne ta det ned til eit enda enklare nivå kan vi bruke blokkbasert programmering.

Blokkprogrammering baserer seg på bruk av objekt i steden for tekstbasert kodespråk (Plaza et al., 2020). Eit døme på blokkbasert programmering kan vere Scratch som er eit populært visuelt programmeringsspråk for nybegynnarar.



FIGUR 2: SKJERMDUMP FRÅ SCRATCH.MIT.EDU

Dette blir ofte nytta i grunnskulen som ein introduksjon til programmering, sjølv om det ikkje er eit programmeringsspråk som vert nytta profesjonelt (Kölling, 2015). Elevane går ofte seinare over til tekstbasert programmering, og i følge Dolonen et al (2019) kan elevar med kunnskap om blokkbasert programmering ha lettare for å forstå tekstbasert programmering. Dolonen et al (2019) hevdar dette er fordi dei har blitt introdusert for fleire grunnleggjande prinsipp tidlegare i undervisinga. Papert argumenterte, ifølge Resnick et al (2009), for at programmeringsspråk burde ha ein låg inngangsterskel, høgt tak og breie veggar. Dette er noko som går igjen i såkalla LIST-oppgåver. LIST betyr låg inngangsterskel og stor takhøgde, og skal gi rom for å jobbe på eit høgt matematisk nivå til tross for at den er enkel å kome i gang med for dei fleste. For å konkretisere programmeringsspråket enda meir kan vi bruke eit anna alternativ, nemleg fysisk programmering, også kalla analog programmering.

2.1.4 Analog programmering

Ein finn mange ulike tilnærmingar på korleis programmering kan løysast i skulen, både digitalt og analogt. Ein kan bruke analog programmering på mange ulike måtar i matematikkundervisninga, som til dømes det å styre ei brikke ved hjelp av koder på eit brettspel, programmere medelevar til å utføre spesifikke oppgåver, eller pusle et puslespel utan at ein sjølv ser puslespelet (Aranda & Ferguson, 2018).

Forskarar trekk fram analog programmering i tilknytninga til å lære algoritmisk tenking (Aranda & Ferguson, 2018; Zhang og Nouri, 2019; Berg, 2021). Eit sentralt omgrep frå algoritmisk tenking som fleire forskarar nemner er feilsøking og dette blir sett på som sentralt i analog programmering. Det blir meint at elevane lærer mykje av å gjere feil for å så rette dei opp igjen (Zhang og Nouri, 2019). Vidare peikar Nouri på kor viktig det er å forstå kva programmering er før digital programmering blir innført (2019). Fleire forskarar viser til samanheng mellom analog og digital programmering, der en variasjon mellom dei er å føretrekke (Berg, 2021; Chibas et al., 2018; Heikkilä & Mannila, 2018; Otterborn et al., 2020).

Berg (2021) trekk fram variasjonen mellom analog og digital programmering som noko meningsfullt for elevane sin motivasjon, samt at det er ein fin måte å tilpasse undervisninga til kvar enkelt elev. Chibas et al. (2018) understreker korleis ein kombinasjon gjer at elevane får ulike tilnærmingar til å arbeide med den kunnskapen dei tileignar seg. Likevel blir det påpeika at gjennom analog programmering vil læraren få eit betre overblikk over kva elevane synest er vanskeleg og kva som kan vere ei utfordring når dei går over til digital programmering (Chibas et al., 2018). Ifølge Berg (2021) kan motivasjonen for programmeringa sjåast i samanheng med at dei analoge programmeringsaktivitetane er kvardagslege og gjer overgangen til digital programmering enklare. Nouri et al (2020) skriv at elevane lærer og utvikla god forståing for programmeringsomgrep gjennom analog programmering. Dette gjeld omgrep som sekvens, gjentakingar (loop), trinnvise instruksjonar og vilkår (Nouri et al. 2020). Spesielt trinnvise instruksjonar blir angitt i forskning i forbindelse med ulike aktivitetar innanfor analog programmering (Heikkilä & Mannila, 2018; Berg, 2021; Chibas et al., 2018; Nouri et al., 2020). Heikkilä & Mannila (2018) fortel blant anna om ein meir kvardagsleg programmeringsaktivitet som inneber trinnvise instruksjonar ved påkledning. Dette går igjen i Berg sin artikkel (2021) som viser til ei oppgåve der elevane skulle bygge eit tårn av centikuber ved hjelp av trinnvise instruksjonar. Heikkilä & Mannila (2018) løftar fram korleis programmering blant dei yngste elevane kan sjåast på som

leikbasert læring, der dei kan ta i bruk fantasien samtidig som problemløysingsferdigheiter blir utvikla. I tillegg presiserer Nouri et al. (2020) at programmering gir elevane moglegheit til å engasjere seg i kreativ problemløysing på ein leiken måte. Aranda & Ferguson (2018) anbefalar analog programmering som ein metode i rolleleik. Videre blir boka «Hello Ruby», eller «Hei Ruby» på norsk, framheva som eit godt hjelpemiddel for unge elevar til å lære programmering på en leiken og fantasifull måte (Aranda & Ferguson, 2018). Det blir gjort eit poeng av at analog programmering er hensiktsmessig å byrje med, då denne forma for programmering tar for seg den grunnleggjande forståinga for kvifor vi programmerer og korleis programmering går føre seg. Å inneha kunnskap om analog programmering blir sett på som ein fordel ved oppstart av digital programmering (Nouri et al., 2020). Likevel er ein kombinasjon av analog og digital programmering å føretrekke (Berg, 2021; Chibas et al., 2018; Heikkilä & Mannila, 2018; Otterborn et al., 2020). I tillegg trekk fleire fram bruk av algoritmisk tenking som eit sentralt aspekt innanfor den analoge programmeringa (Aranda & Ferguson, 2018; Nouri et al., 2020; Berg, 2021).

2.1.5 Algoritmisk tenking

Algoritmisk tenking er eit omgrep som har fått stor merksemd dei siste åra. På utdanningsdirektoratet si nettside skriv dei at det finst ein rekkje ulike definisjonar av algoritmisk tenking. Utdanningdirektoratet sjølv vel å definera omgrepet algoritmisk tenking som ein problemløysingsmetode. Vidare er dei tydeleg på at det framleis er ein pågåande diskusjon om kva som skal leggjast i omgrepet. Dei argumenterer for at hovudtrekka ved dei ulike definisjonane er samanfatta, og har med utgangspunkt i det og i modellen til Barefoot Computing (Lucas et al., 2017) utarbeida figuren «Den algoritmiske tenkeren» (figur 3). Figuren viser nøkkelomgrepa som går under algoritmisk tenking og typiske arbeidsmåtar for å løyse eit problem.

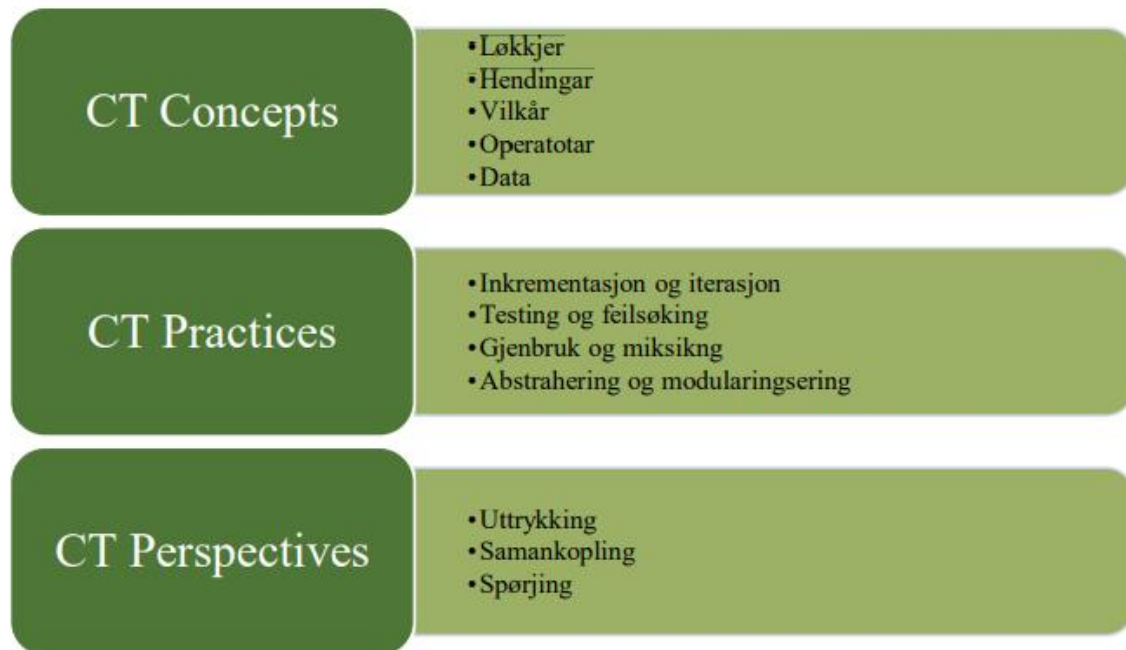


FIGUR 3: DEN ALGORITMISKE TENKEREN (UTDANNINGSDIREKTORATET, 2019)

Vidare seier dei at algoritmisk tenking er den direkte norske omsetjinga av det engelske omgrepet «computational thinking». Computational thinking er ei overordna tilnærming som inneheld fleire kognitive komponentar, som til dømes abstrahering, kunnskap om algoritme, logikk og generalisering. Wing (2010) ein anerkjent forskar innan programmeringsfaget definerer computational thinking på følgande 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» (Wing, 2010, s. 1). Wing inkluderer og sjølve prosessen der ein arbeider med eit problem og har slik likskap med Utdanningsdirektoratet sin definisjon. Videre skriv ho 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 tenking handlar om meir enn berre omgrepet programmering.

For å kunne vurdere elevar si utvikling av algoritmisk tenking har Karen Brennan og Mitchel Resnick (2012) laga eit rammeverk bygd opp rundt tre hovuddimensjonar. Desse er «computational thinking concepts», «computational thinking practices» og «computational thinking perspectives» (Brennan & Resnick, 2012, s. 3). Rammeverket er utvikla med blokkbasert programmering som fokus, men det kan og nyttas innan andre typar programmering. Dette blir støtta av Lye og Koh (2014, s. 52-53) som peikar på at

rammeverket godt kan nyttast til å studere utvikling av algoritmisk tenking gjennom programmering generelt. Figur 4 viser ei oversikt over dei tre hovuddimensjonane og kva som ligg i dei. Eg har brukt desse hovuddimensjonane når eg seinare i oppgåva fann rammeverk som var relevant for min studie.



FIGUR 4: VURDERING AV ALGORITMISK TENKING ([HTTPS://SCRATCH.MIT.EDU](https://scratch.mit.edu))

2.2 Programmeringspedagogikk og kompetanse i programmering

Waite (2018) har i ein omfattande litteraturgjennomgang av programmeringspedagogikk prøvd å finne svar på kva som er effektiv pedagogikk innan programmering. Ho kjem med ein rekke forslag til område ein kan studere for å finne moglege svar. Kva som er god pedagogikk er avhengig av ulike programmeringsspesifikke kontekstar eller utfordringar som ein kanskje ikkje finn igjen i andre fag. Til dømes foreslår Waite at vi må utvikle ein måte å undervise symbol- og eller blokkprogrammering på som gjer overgangen til tekstprogrammering enklare, og vi må finne ut korleis ein kan støtte elevane til å gå frå ein opplæringsituasjon til å klare å bruke det dei har lært til å utvikle eigne prosjekt. Waite reknar programmering for fag som matematikk som ein eigen kontekst, som krev ein eigen pedagogisk framgangsmåte, og som det ikkje er gjort mykje forskning på.

I staden for berre å fokusere på spesifikke programmeringskonsept i undervisninga må vi og lære elevane programmeringsferdigheiter (Lye & Koh, 2014). Forskinga seier lite om kva for ferdigheiter som manglar for dei som aldri meistrar programmering (Waite, 2018), men elevane bør lære om dei ulike perspektiva på kva programmering omfattar og inneber (Lye & Koh, 2014). Opplæringa bør innehalde eit breitt utval av ulike aktivitetar for å dekke desse områda, men det er gjort lite arbeid for å finne ut kva slags aktivitetar og kva slag pensum som eigna seg for ulike alderstrinn (Lye & Koh, 2014).

Kjällander et al, (2018) viser til forskingsprosjekt ved lærarutdanningar i andre nordiske land der ein undersøker korleis lærarar modellerer problemløysing, kreativ tenking, teknologi og IKT. Her meinte Kjällander et al., (2018) at det var stor bekymring for mangelfull kompetanse på programmeringsfeltet. Artikkelen peikar på at det ikkje fanst noko formell kursing av lærarar som dekker didaktiske spørsmål, og at akkurat det kan vere eit problem framover. Forfattarane skriv vidare at det eksisterer fleire tilbod om kurs i programmering på internett, men at dette stort sett er kurs som heva programmeringskompetansen og ikkje fokuserte på dei didaktiske utfordringar som læraren kan møte i kvardagen. Eit behov for kunnskap om tema blir grunngeve med at programmering skal inn i læreplanar og at lærarutdanninga ikkje vil vere i stand til å gje den naudsynte kompetansen på feltet til lærar og lærarstudentar (Koehler & Mishra, 2009). Forfattarane peiker og på behovet for eit didaktisk design innan programmeringsfeltet i framtida.

Dette blir støtta opp i ein rapport frå (NIFU, 2019:3) om lærerans profesjonsfaglege digitale kompetanse i lærarutdanninga der det står blant annet på side 13: «heller ikke lærerutdanningene henger helt med når det gjelder å forberede fremtidens lærere til en skolehverdag der teknologi får stadig større plass».

Videre i rapporten på side 24 står det om bruken av digitale ressursar:

«De tre bruksmåtene som i størst grad anvendes er til kommunikasjon og kontakt med studentene, til å presentere nytt lærestoff og til vurdering av arbeidskrav. Minst bruk er det til å tilrettelegge for omvendt undervisning, til samarbeid mellom studenter og praksisfeltet, til å fremme faglige drøftinger og til å gjøre undervisningen mer praksisrelevant».

For å kunne løyse denne utfordringa såg Nordby et al. (2022) på fire lærarar si forståing av algoritmisk tenking, og korleis dei inkluderte det i matematikkundervisninga i samanheng med innføringa av algoritmisk tenking i dei nye læreplanane. Dei såg vidare på korleis innføringa av algoritmisk tenking kunne påverke den eksisterande og tradisjonelle

klasseromsaktiviteten der lærar står og underviser. Studien viste at lærarane streva med å forstå omgrepet algoritmisk tenking, og korleis dette kunne koplust opp mot matematikkundervisninga. Lærarane si undervisning i programmering og algoritmisk tenking blei ofte løyst på veldig ulike måtar, og datamateriale i studien avslørte to distinkte forståingar av omgrepet algoritmisk tenking. På den eine sida blei algoritmisk tenking sett på som noko som er kopla av matematikken og som ein kan knytte opp mot første møte med programmering, mens det på den andre sida blir forstått som ein del som ikkje er blitt tatt i bruk, men som blir dekkja av allereie eksisterande element i faget, slik som standardalgoritmer (Nordby et al., 2022). Manglande kompetanse hos lærarar blei delt inn i to kategoriar. Lærarar uttrykte at dei var usikre og mangla kompetanse på korleis algoritmisk tenking kom fram i læreplanen og på kva algoritmisk tenking faktisk er. Samstundes uttrykte dei manglande kompetanse på korleis dette kunne bli inkludert i matematikkundervisninga (Nordby et al., 2022). Ei sånn todeling av kompetansen i algoritmisk tenking og programmering finn vi og i rammeverk for lærarar sin profesjonsfaglege digitale kompetanse (PFDK, 2021). Studien til Nordby et al (2022) vart gjennomført tre månader inn i den nye læreplanen. Avslutningsvis peika studien på at lærarar opplevde å ha få utviklingsmoglegheiter, sjølv om det fanst tilgjengelege ressurspakker frå utdanningsdirektoratet. Det var og manglande kunnskap om korleis koplinga mellom algoritmisk tenking og matematikk skulle gå fram, og korleis algoritmisk tenking kunne bli overført til fag og aktivitet i skulen. Nordby (et al., 2020) trekk fram at sjølv om lærarane var i oppstartsfasen, hadde alle lærarane kome i gang med algoritmisk tenking, men på ulike måtar og i ulike retningar. Manglande rettleiing og forbetring til ny læreplan blir peika på som moglege faktorar til korleis algoritmisk tenking i matematikk blir forstått på veldig ulike måtar (Nordby et al., 2022)

2.3 Didaktiske rammeverk innanfor programmering og matematikk.

For å kunne hjelpe lærarar med låg kompetanse har eg valt ut tre didaktiske rammeverk som eg tar utgangspunkt i for å kunne lage eit undervisningsopplegg med analog programmering. Ein finn mange rammeverk å velje mellom, men eg har gjort eit bevist val på tre rammeverk som står fram for meg som mest relevante i arbeidet mitt med lærar.

2.3.1 Benton 5 e-r

Benton et al. (2017) påpeika at programmering er blitt ein stor del av skulen, og det er ein av årsakene til at dei har utvikla eit rammeverk som handlar om undervisning i programmering. Dette rammeverket skulle vere eit «framework for action». diSessa og Cobb (2004) forklarar «framework for action» som ei generalisering på tvers av fleire empirisk støtta teoriar. Ved å sjå på dei generelle samanhengane mellom ulike teoriar og forskning innan programmering har dei bygd opp eit rammeverk basert på desse. Benton et al. (2017) gjekk gjennom relevant litteratur som omhandla undervisning av LOGO (programspråk) og ei rekke andre lærebøker. Lærebøkene var bøker med mål om å støtte lærar i undervisning av programmering og matematikklærebøker for grunnskulen.

I tillegg til å gjennomgå relevant litteratur samla Benton et al. (2017) inn data, som viste til lærar og elev sine behov og utfordringar i programmeringsundervisning i matematikk. Ut i frå data som vart samla inn, utvikla dei eit rammeverk som underbyggjer pedagogiske strategiar for implementering av programmering. Benton et al. (2016) forklarar denne implementeringa av pedagogiske strategiar følgjande: «It has been developed to provide guidance on the pedagogical strategies teachers may adopt to successfully implement different aspects of the SM (ScratchMaths) intervention» (s.28-29). Omsett til norsk er rammeverket til for å underbygge dei pedagogiske strategiane som ein lærar nyttar når ein underviser i programmering. Rammeverket vil kunne nyttast i rettleiing av lærarar som ein del av den profesjonelle utviklinga (Benton et al., 2017). Benton et al. (2016) forklarar at dei gav rammeverket namnet «5Es», då det består av fem undervisningspraksisar. Desse er fleksible og blir nytta ulikt i ulike kontekster, og dei har heller ingen spesifikk rekkefølge. Rammeverket består av undervisningspraksisane; Explore, Envisage, Explain, Exchange og BridgeE (Benton et al., 2016; 2017). Desse kan ein omsette til; Utforske, sjå for seg, forklare, utveksle og å kople. (Vidare vil eg gjere greie for kva desse fem undervisningspraksisane står for.)

Det første omgrepet, *Explore*, eller *utforske*, tydeleggjer kor viktig det er å utvikle og støtte aktivitetar der elevane får utforske idear, prøve og feile, samt rette opp eventuelle feil dersom det er nødvendig. Her handlar det om å la elevane få lov til å ta kontroll over eiga læring og utforske kva som ligg bak det ein gjer. Dette refererer Benton til autonomien og noko som kan styrke elevane sin indre motivasjon. Det kan vi sjå igjen i sjølvrådeteorien til Deci og Ryan.

Det andre omgrepet, *Explain*, eller *forklare*, handlar om det å forstå idear, å kunne forklare kunnskap og artikulere årsakene til den valde tilnærminga på problemet. I eit kognitivt perspektiv kan til dømes det å sette ord på ting legge til rette for forståing, sidan ein må kople det ein skal forstå med det ein kan frå før. Gjennom å svare på spørsmål frå medelevar og uttrykke ideane sine, utvikler elevane evna til å avklare idear. Dette omgrepet uttrykker kor viktig det er å legge opp til reflekterte spørsmål, og opne for klasseroms-diskusjonar både med medelevar og lærar. Her kan det vere alt frå lærarstyrte klassesdiskusjonar, samarbeid med medelevar eller å svare på refleksjonsspørsmål.

Det tredje omgrepet, *Envisage*, eller *å sjå for seg*, som viser til det å ha eit klart mål når ein arbeider med programmering. Her skal ein kunne sjå for seg kva dei moglege utfalla kan bli før ein køyrer koden. Å generere eit tenkt utfall gjennom koding kan vere utfordrande, men blir ikkje resultat som ønskja kan ein sjå over kvar feilen ligg og lære av dette. Omgrepet å sjå for seg tydeleggjer behovet for at læringsaktivitet skal gje elevane moglegheit til å planlegge utfallet og reflektere over dette. Kvifor blei utfallet som det vart? Det er viktig at omgrepet går saman med utforsking, då det mogleggjer å oppdage gjennom utforsking, samtidig som det gjev anledning til å sjå for seg moglege utfall før dei tester programmet/koden

Det fjerde omgrepet, *Exchange*, eller *utveksle*, seier kor viktig det er å dele og bygge på andre sine meiningsfulle idear. Hoyles (1985) foreslår at andre idear potensielt kan resultere i modifiseringar av elevar sin tenkjemåte, og det er særleg hjulpsamt å gjere greie for og forklare idear som ikkje er fullstendig forma enda. Dette omgrepet baserer seg på at elevane skal utveksle sine tankar og idear, og på denne måten blir oppdaginga og utforskinga mindre lærarstyrt.

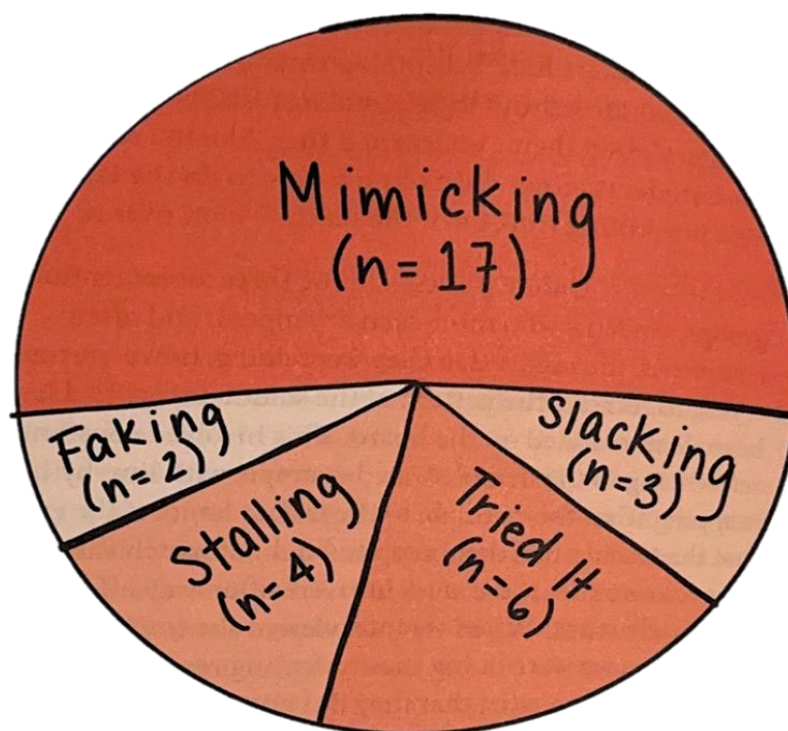
Det femte, og siste omgrepet, *BridgE*, eller det å *kople*. Det inneber det å bygge ei bru eller kople saman noko. Benton et al. (2016) forklarar at «(...) ideas are seen as powerful partly through their connections with other disciplines». Benton et al. (2016; 2017) påpeiker at det er viktig å sjå på programmering i lys av matematikk. Det burde være eksplisitte koplingar mellom programmering og matematikk i aktivitetane og læraren sin praksis. Programmering skapar gode moglegheiter for å lære matematikk, og er eit verktøy ein kan bruke for å lære matematiske konsept. Verktøyet kan til tider overskygge matematikken, og for å sikre at moglegheitene for å lære matematikk ikkje blir oversett må koplinga til matematikk bli tydeleggjort for elevane. For å gjere matematikken tydeleg, kan læraren kople programmeringa til læreplanen i matematikk og bruke spørsmål som skapar diskusjonar

(Benton et al., 2016). Til dømes kan ein bygge bru mellom problemløysing og programmering, der ein må dele problemet inn i mindre deler og finne moglege løysningar.

Disse fem E-ane vil kunne hjelpe lærarar i programmeringsundervisning. Dei kan nyttast til å underbygge læraren sitt val av pedagogiske strategiar, men og vere eit hjelpemiddel i planlegging av undervisning. To av E-ane, Explore og Explain, er verb som blir forklart av Van De Walle, Bay-Williams & Karp (2014) som handlingar der elevane blir engasjert til å gjere matematikk, og som gjev moglegheit til tenking på eit høgare nivå. Dette gjer at bruk av dei 5 E-ane, og dermed utforsking og forklaring, kan vere eit hjelpemiddel for å auke elevaktivitet og konstruksjon av eigen kunnskap.

2.3.2 Thinking Classrooms

Peter Liljedahl starta på slutten av 90-tallet ei stor undersøking (over mange tusen timar) der dei observerte elevar frå ulike skular spreidd over heile Canada for å sjå på elevaktiviteten i klasserommet. Her fann han at det berre var om lag 20 % av elevane som kunne bli kategorisert som tenkande i eit klasserom (Liljedahl, 2021). Liljedahl sjølv definerer tenkande elevar med at dei kan utrykke sjølvstende, sjølvregulering, kritisk tenking, kreativitet og samarbeid.



FIGUR 5: FORDELING AV ELEVAKTIVITET I KLASSEROMMET (LILJEDAHL, 2021, s. 10)

Dei vidare 80 % av elevane (Sjå figur 5) dreiv med ingenting, andre ting, lata som dei gjorde oppgåver eller «mimicking». Mimicking, eller herming, er det elevar gjer når dei skal prøve å atterskape det læraren har gjort på tavla. Dette var noko om lag halvparten av elevane gjorde, og Liljedahl skriv at mange av elevane følte at det var denne måten læraren ønskja at elevane skulle lære seg matematikk på. På bakgrunn av dette valde Liljedahl å utføre eit større forskingsprosjekt, der han saman med fleire hundre lærarar ville finne ut korleis ein kunne endre undervisningspraksisen for at elevane skulle bli meir tenkande i matematikkundervisninga. I løpet av ein periode på fleire år kom Liljedahl fram til fleire konkrete tiltak lærar kan praktisere for å skape gunstige forhold for tenking, og dermed og læring i klasserommet. Denne forskinga resulterte i eit didaktisk rammeverk som blei kalla «Thinking Classroom» eller «Tenkande klasserom».

Liljedahl sjølv definerer eit tankande klasserom som:

« (...) a thinking classroom is a classroom that is not only conducive to thinking but also occasions thinking, a space that is inhabited by thinking individuals as well as individuals thinking collectively, learning together, and constructing knowledge and understanding through activity and discussion. It is a space wherein the teacher not only fosters thinking but also expects it, both implicitly and explicitly». (Liljedahl, 2016, s 364)

Eit Tenkande klasserom skal altså ikkje berre bidra til tenking, men og gjere klasserommet om til eit tenkane kollektiv som kan lære og konstruere kunnskap saman gjennom å diskutere, løyse og forstå matematikk. Eit klasserom der lærar ikkje berre fremjer tenking, men og forventar det både implisitt og eksplisitt. Eit tenkande klasserom inneheld altså elevar som tenker individuelt, men som i tillegg oppfører seg som tenkande individ som skal tenke i fellesskapet av klassa (Liljedahl, 2016, s 364).

Klasserommet skal vere ein stad der elevane får moglegheit til å lære både aleine og saman med andre, og opparbeide seg kunnskap og erfaring gjennom aktivitet og diskusjon. Dette støttar opp om Dewey sin teori om at kunnskap bør ha utgangspunkt i elevane sine egne erfaringar, og at læraren må forvente at elevane tenker og reflekterer. Det som skil eit tenkande klasserom frå eit ikkje-tenkande klasserom, er at det finst ei norm som krev at elevane tenker. Liljedahl (2021) undersøkte klasseromsnormer og identifiserte 14 element som kan seie noko om eit klasserom er tenkande eller ikkje. Liljedahl brukte desse elementa som utgangspunkt for vidare undersøkingar. Dei 14 elementa er variablar som ein lærar kan

sjå nærmare på for å få etablert eit tenkande klasserom (Liljedahl, 2016, 2021). Lista med element er følgande:

1. What types of task we use
2. How we form collaborative groups
3. Where students work
4. How we arrange the furniture
5. How we answer questions
6. When, where, and how tasks are given
7. What homework looks like
8. How we foster student autonomy
9. How we use hints & extensions
10. How we consolidate a lesson
11. How students take notes
12. How we choose to evaluate
13. How we use formative assessment
14. How we grade

Liljedahl skriv at det vil vere vanskeleg å bite over alle elementa og valde difor å dele dei 14 undervisningspraksisane inn i fire verktøykasser. Han skildra dei fire verktøykassane som ei ramme for å hjelpe lærarar å integrere studentaktivitet i klasserommet (Liljedahl, 2021).

Tiltaka i gruppe 1 skal vere dei enklaste å sette i verk for ein lærar, og det er og desse som har størst innverknad på klasseromsnormene (Liljedahl, 2021). I verktøykasse 2 handlar det meir om å finjustere det tenkande klasserommet og legge grunnlag for verktøykasse 3, som skal jobbe med å skape flyt i klasserommet. I verktøykasse 4 krev det meir av læraren, og denne verktøykassa dreier seg om evaluering og reflektering over eigen praksis (Liljedahl, 2021). Han meiner at desse verktøykassane er essensielle for å oppmuntre deltaking og refleksjon, samt å støtte lærar i å strukturere klasserommet for å maksimere elevane si tenkjeevne og engasjement. Liljedahl skriv vidare at det er vanskeleg å gripe over alle dei 14 variablane på ein gong grunna deira store omfang, og har difor råda lesarane til å prøve ut ei-og-ei verktøykasse i rekkefølga gitt over. I min studie har det difor vore naturleg å fokusere på den første verktøykassa, då denne er mest relevant for å få svar på mine forskingsspørsmål. Denne verktøykassa vil og vere den som har størst innverknad på klasserommet. Dei tre undervisningspraksisane i den første verktøykassen er:

1. «Give thinking tasks»: Dersom ein ønskjer at elevane skal tenke, må ein gje dei noko å tenke på. Det må vere noko som ikkje krev tenking åleine, men som og oppfordrar til tenking. Når det kjem til tenkande oppgåver, er problemløysing den beste plassen å starte (Liljedahl 2021). Undervisninga må starte med gode problemløysningsoppgåver. I ein tidleg fase i oppbygginga av tenkande klasserom, må desse vere høgt engasjerande, samarbeidande oppgåver som gjer at elevane ønskjer å snakke med kvarandre medan dei prøver å løyse oppgåva (Liljedahl, 2021). Eit døme på ei slik oppgåve kan vere det som vi i den nye

læreplanen kallar for rike oppgåver (Til dømes LIST oppgåver nemnt tidlegare). Under arbeid med problem vil elevar sitte fast, tenke, og komme seg laus. Når dei gjer akkurat dette vil dei lære om matematikk, om seg sjølve, og dei vil lære korleis ein tenker (Liljedahl, 2021).

2. «Frequently form visibly random groups»: Det er viktig at gruppene blir delt inn foran elevane og at det er synleg for dei at det er tilfeldige grupper. Ideelt sett må dette skje i oppstarten av kvar undervisningsøkt (Liljedahl, 2021). Desse gruppene vil arbeide saman med ei gitt problemløysande oppgåve og sitte eller stå saman under gruppe- eller klasse-diskusjonar (Liljedahl, 2021).

3. «Use vertical non-permanent surfaces»: Elevane skal arbeide på vertikale (ikkje-permanente) tavler, som til dømes «whiteboards» eller vindauge. Dette vil synleggjere arbeidet som blir gjort både for lærar og for dei andre i elevgruppene. Vidare meiner Liljedahl at det berre bør vere ein tusj på kvar gruppe, for å legge til rette for diskusjon. Når elevane arbeider i tilfeldige grupper og på vertikale tavler, tenkte dei lengre, diskuterte meir matematikk og haldt ut lengre når oppgåvene var vanskelege (Liljedahl, 2021).

Peter Liljedahl sitt tenkande klasserom er altså ein pedagogisk modell som vektlegg aktiv læring, ansvarleggjering og demokratiske prosessar. Modellen tar utgangspunkt i at elevane skal ta ansvar for si eiga læring, og at dei må få moglegheit til å reflektere, ta slutningar og løyse oppgåver saman. Modellen har eit ønskje om å gje elevane ei kjensle av meistring og auke deira interesse for læring. Det kan og inspirere dei til å utforske og eksperimentere med nye idear og løysingar.

2.3.3 Primm-modellen

For å svare på nokre av utfordringane innan programmering i skulen har Sentance et al. (2019) (inspirert av Lee et al., (2011) sin Use-Modify-Create-modell) utvikla modellen PRIMM (Predict, Run, Investigate, Modify, Make) for å strukturere klasseromsundervisning i programmering uavhengig av programmeringsspråk eller andre ressursar. Denne modellen er ikkje ein komplett didaktisk pakke, men det er ein start mot ei meir forskingsbasert programmeringsopplæring. Modellen vart i hovudsak utvikla for å hjelpe til med å planlegge undervisningsøktar og aktivitetar med programmering. Stega predict, run, investigate, modify, og make, som er arbeidsprosessane i modellen, kan ein forklare ved hjelp av tabellen under.

Predict	Elevane får utlevert ei ferdig kode, og skal freiste å avgjere kva som skjer når ein køyre koden.
Run	Her skal elevane køyre koden for å evaluere om dei tenkte rett i steget «predict».
Investigate	I tredje steg av modellen skal elevane bli betre kjent med koden. Denne fasen kan og bør helst varierast med ulike typar oppgåver, som gjerne kan innebere å lage feil i koden eller gje namn til dei ulike variablane.
Modify	Her får elevane i oppgåve å gjere bestemte endringar slik at noko endrar seg når ein køyrar koden.
Make	Siste og femte steg i modellen. Her skal elevane lage ein heilt ny kode, med utgangspunkt i den koden dei fyrst fekk utlevert. Det kan vere ei oppgåve der dei kan låne noko frå den opphavslege koden, eller ei heilt ny oppgåve. I begge høve skal den nye koden innebere noko nytt som krev ein ny funksjon.

TABELL 1: 5 STEG I PRIMM. (PRIMMPORTAL, 2022)

Denne modellen har tatt utgangspunkt i Lev Bygotsky sine sosiokulturelle læringsteoriar som vektlegg språk, formidling og overføring av kunnskap og dugleik frå det sosiale til det kognitive plan (Sentance et al., 2019). For å minske startvegring og sikre motivasjon hos elevane, gjev lærar som brukar PRIMM- modellen elevane eit ferdig program dei skal analysere (predict), før dei får køyre det (run).

Elevane skal arbeide parvis der dei må samarbeide og kommunisere om oppgåvene. Gjennom opplegget får elevane i oppgåve å gjere bestemte endringar i programmet, og må då gå inn i koden for å undersøkje korleis endringa skal gjerast (Investigate). Ved å utføre endringane (modify), får dei meir eigarskap til koden og heilt til slutt skal elevane lage sin eigen kode frå botn, med utgangspunkt i den dei fekk utlevert i starten. Gjennom alle trinna i undervisningsopplegget er samarbeid og kommunikasjon med ein eller fleire medelevar viktig (Primmportal, 2022)

PRIMM-modellen er testa ut blant elevar i grunnskulealder i Storbritannia, med mål om å adressere dei mange utfordringane ein finn med programmering i skulen (Sentance et al., 2019). I denne studien viste det seg at elevgruppa som gjennomførte PRIMM-modellen presterte jamt over høgare enn kontrollgruppa, noko som gjorde at Sentance et al saman med lærarane i studie kunne konkludere med at modellen er noko ein bør implementere i

programmeringsundervisning. PRIMM skil seg frå både forelesningsformatet og den ustrukturerte tilnærminga til for eksempel Resnick (2017). I PRIMM skal elevane ikkje kopiere læraren si kode, men strukturert gjennomføre ulike læringsaktivitetar som gradvis går mot at dei skriv sin eigen kode. Eit viktig poeng i PRIMM-modellen er at elevane skal lære å uttrykke seg munnleg om programmeringsomgrep og -prinsipp. Lye og Koh (2014) konkuderte i sin «review» av 27 intervensjonar om algoritmisk tenking med at dette er en viktig eigenskap som ofte blir oversett både i programmeringsundervisning og forskning. Hermans, Swidan og Aivaloglou (2018) oppdaga at elevar som er flinke til å lese kode høgt for andre også presterer betre som programmerarar, og dei hevdar at alle elevar burde lære “kodefonologi” - altså å lese kode høgt. Denne tilnærminga finn vi igjen i Kluge et al. (in press), der elevar i norsk ungdoms- og vidaregåande skule produserte “screencasts”, eller skjermopptak, der dei presenterte og forklarte sin eigen kode munnleg.

2.4 Oppsummering

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

Programmeringstema har vore eit omgrep med mykje forvirring i skulen. Nygård (2018) peikar på at omgrep som programmering og algoritmisk tenking ikkje har klare nok definisjonar og mange i skulesystemet slit med å plassere dei. Dette er noko som går igjen i nyare forskning der blant anna Kjällander et al, (2018) viser til at det bør rettast ei bekymring over den mangelfulle kompetansen lærarar har på programmeringsfeltet.

I den nye læreplanen blir programmering og algoritmisk tenking vektlagt og sett på som eit verktøy for å løyse problem. Det vil seie at programmering og algoritmisk tenking heng saman, der programmering kan vere eit godt verktøy for å arbeide med algoritmisk tenking og omvendt (Sevik, 2016). I mitt prosjekt vil det vere av interesse å sjå på korleis programmering kan bli brukt for å hjelpe elevane med å kjenne skaparglede, engasjement og utforskartrang. Det er lite forskning som peikar på kva som skal til for å nå dei overordna prinsippa, men der er ein god del som peiker på kva som er god pedagogikk innan programmering- og matematikkundervisning. For å kunne lage eit undervisningsopplegg om programmering har eg sett på tre didaktiske rammeverk som skulle leggje grunnlaget for opplegget.

Benton et al. (2017) utvikla eit rammeverk basert på samanheng mellom ulike teoriar og forskning innan programmering. Her kom Benton et al. (2017) fram til fem

undervisningspraksisar som skulle hjelpe lærarar i programmeringsundervisinga. Praksisane skulle nyttast til å underbygge læraren sitt val av pedagogiske strategiar, men og vere eit hjelpemiddel i planlegging av undervisning. Desse fem undervisningspraksisane var Explore, Envisage, Explain, Exchange og BridgeE (Benton et al., 2016; 2017).

Peter Liljedahl forska på kva som skulle til for å auke elevaktiviteten i klasserommet. Forskinga hans enda opp i rammeverket «Thinking Classroom» som å skal bidra til at elevane blir meir tenkande, men og gjere klasserommet til eit meir tenkande kollektiv. Liljedahl (2021) undersøkte klasseromsnormer og identifiserte fjorten element som kan seie noko om eit klasserom er tenkande eller ikkje. Med utgangspunkt i desse fjorten elementa utvikla Liljedahl nokre verktøykassar for å hjelpe lærarar å integrere elevaktivitet i klasserommet (Liljedahl, 2021). I den første verktøykassa var det tre sentrale undervisningspraksisar eg har tatt med meg vidare. Desse var «Give thinking tasks», «Frequently form visibly random groups» og «Use vertical non-permanent surfaces»

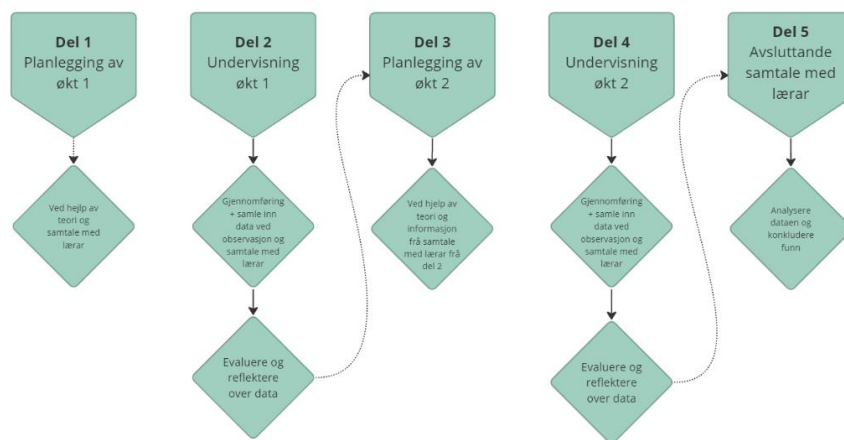
Det siste rammeverket er Primm-modellen til Sentance et al. (2019). Modellen er ikkje ein komplett didaktisk pakke, men vart i hovudsak utvikla for å hjelpe lærarar med å planlegge undervisningsøktar og aktivitetar med programmering. Primm blei testa ut blant elevar i grunnskulen i Storbritannia, der resultatane viste at elevgrupper som bruke Primm presterte jamt over høgare enn kontrollgruppa. Dei fem undervisningspraksisane til Primm-modellen var «Predict», «Run», «Investigate», «Modify», og «Make».

3 Metode

Dette kapittelet tar for seg masteroppgåva sin metode. I arbeidet med mi oppgåve har eg tatt ei rekkje metodiske val som har påverka datainnsamlinga og analysemetoden, og det har fått konsekvensar for studien. Kapittelet består av sju delkapittel. Først vil eg gjere greie for dei metodiske vala som har blitt tatt i oppgåva: Eg vil her presentere forskingsdesignet mitt, som er inspirert av Design-Based Research, informant- og forskarrolla og datainnsamlingsreiskapane. Vidare vil eg skildre undervisningsopplegget som blei utvikla ved forankring i teorikapittelet, før eg vil forklare korleis analysen av datamateriale har gått føre seg. Til slutt vil eg vurdere mi eiga forskning ved å sjå på prosjektets reliabilitet, validitet og nokre etiske prinsipp.

3.1 Design

Denne oppgåva var eit designeksperiment (til dømes Cobb et al., 2003) som eg utførte på ei 6.klasse på mellomtrinnet ved ein norsk grunnskule. Mitt generelle mål i designeksperimentet var å utvikle teoretisk og praktisk kunnskap om måtar å fremje skaparglede, engasjement og utforskartrøng, gjennom design og implementering av instruksjonsintervensjonar, kunnskap og undervisningspraksis om analog programmering, der eg i hovudsak forankra opplegget på Bentons 5 E-ar (kapittel 2.3.1), Liljedahl sitt tenkande klasserom (kapittel 2.3.2) og Primm-modellen (kapittel 2.3.3). I tillegg var det eit mål at undervisningsopplegget vi utvikla skulle vere gjennomførbart og attraktivt for lærar. Designet (sjå figur 6) gjekk altså ut på å planlegge, gjennomfør, evaluere og justere eit opplegg to gongar for å finne best mogleg praksis.



miro

FIGUR 6: DESIGN AV STUDIEN

Med eit forskingsspørsmål om analog programmering kan leggje til rette for skaparglede, engasjement og utforskartrong, og eit ønskje om å undersøke dette i praksis, var det viktig for meg å kome tett inn på dei individa som deltok i prosjektet mitt. Her var det sentralt for meg å forstå menneske sine handlingar og meiningsskaping i naturleg kontekst, og det var difor naturleg for meg å velje ein kvalitativ datainnsamling. Med ein kvalitativ datainnsamling samlar ein inn data som har som mål å skildre, samt forstå, menneske sine handlingar og meiningsskaping i naturleg kontekst (Postholm & Jacobsen, 2018). Eg hadde behov for eit grundig og skildrande datamateriale for å få fram lærar og elev si åtferd rundt arbeidet med analog programmering i matematikkundervisninga (Patton, 1987; Postholm, 2005; Tjora, 2012). Ved hjelp av datainnsamlingverktøya intervju og observasjon ønskte eg å sjå på erfaringane frå lærar og åtferda til elevane i undervisninga. Desse verktøya ga meg høve til å få ei djupare og klarare innsikt i læraren si forståing, og kva konsekvens handlingsval undervegs gjer for undervisninga. Vidare ga det meg eit høve å danne ei meining om elevane si åtferd i undervisninga for å vurdere om undervisningsopplegget la til rette for måla om skaparglede, engasjement og utforskartrong.

For å danne grunnlaget for undervisningsopplegget til økt 1 laga eg eit undervisningsopplegg forankra i teorien frå forrige kapittel. For at opplegget skulle vere tilfredsstillande og gjennomførbart for lærar, så hadde eg hyppige samtalar med lærar undervegs. Det var i hovudsak eg som forskar som la fram forslag, og lærar kunne bruke sin erfaring og lærarstil til å påverke undervisningsopplegget etter sine ønsker og behov. Eit innspel lærar kom med var til dømes det å jobbe med omgrep og omgrepsforståing. Når undervisningsopplegget var klart, fekk lærar gjennomføre opplegget, medan eg observerte og samla inn nødvendige datamateriale. Etter at undervisningøkta var gjennomført, tok vi ein ny samtale der lærar fekk lufta tankar og meiningar om opplegget i samråd med meg som forskar. Her kom lærar med både innspel og idear til kva som kan forbetre opplegget. Eg og lærar tok med oss desse tankane, samt observasjonsmateriale frå økt 1 då vi gjekk vidare til planlegging av økt 2. Undervisningsopplegget i økt 2 vart tilnærma likt, men med nokre justeringar basert på erfaringar og observasjon frå første økt. Undervegs i økt 2 vart det igjen samla inn nytt datamateriale som eg og lærar, gjennom ein lengre samtale, brukte til å evaluere undervisningsopplegget. Vi diskuterte og vurderte om det fungerte betre denne gongen, kva som eventuelt gjorde det betre og korleis vi kunne utvikle undervisningsopplegget enno betre til seinare bruk .

Eit av metodikken sine største utfordringar er å vurdere når forskinga er fullført. Anderson & Shattuck (2012) skriv at det er gjerne ønskjeleg med fleir syklusar, men sidan kvar prosess er tidkrevjande måtte eg ta omsyn til dette, både for læraren sin del og min eigen. Eg valde difor å la designeksperimentet mitt bestå av to forskingsssyklusar, der eg implementerte, analyserte og forbetra undervisningsopplegget etter kvar syklus. Omgrepet forskingsssyklus kan ein sjå på som ein prosess som gjentek seg i same rekkefølge over eit viss tidsrom. Dette resulterte i at både eg og lærar visste på førehand kor mange undervisningsøker vi hadde framføre oss, og at prosjektet hadde ein naturleg slutt då desse var gjennomført. Forskingsdesignet gjev stor metodefriheit, og ein må velje dei metodar som er best eigna til å svare på problemstillinga.

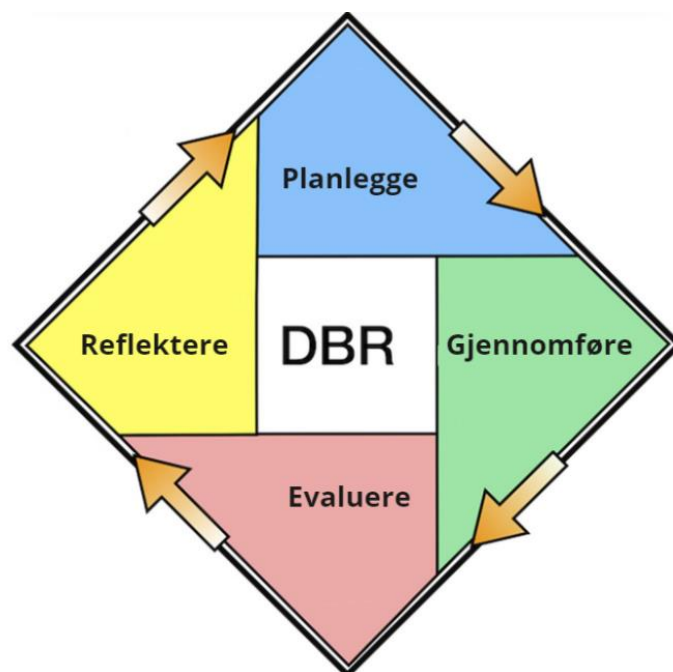
Merksemda i denne oppgåva vil i hovudsak bli retta mot læraren sitt perspektiv, og korleis det samspelet med perspektivet eg har som forskar. Kvalitative metodar gjev moglegheit for eit nært samarbeid mellom forskar og informant, og kunnskapen blir produsert i dette samspelet (Postholm & Jacobsen, 2018). Undervegs i oppgåva mi vil eg ha eit tett samarbeid med informanten min, både gjennom samtaler, planlegging og gjennomføring av undervisningsopplegg og ved innsamling av data i form av observasjon og intervju.

Den metodiske tilnærminga i denne masteroppgåva er basert på forskingsmodellen Design-Based Research (DBR). DBR er ein forskingsmetodikk som har som mål å minske gapet mellom forskning og praksis. Metodikken hadde sin framvekst tidleg på 2000-talet, og vart i utgangspunktet designa av lærarar for lærarar. Det overordna målet i designeksperimentet er å utvikle teoretisk og praktisk kunnskap om måtar ein kan utvikle, gjennom design og implementering av instruksjonsintervensjonar, spesifikke læringsmål knytt til elevane sine kognitive og affektive domen (Anderson & Shattuck, 2012). Metodikken tar utgangspunkt i eit sentralt problem, som i mitt tilfelle vil vere korleis elevane kan fremje skaparglede, engasjement og utforskartrøng og korleis eg kan løyse det i ein reell utdanningskontekst gjennom utforming og utprøving av intervensjonar. I mitt prosjekt ønskte eg gjerne å sjå på to ting. Det første vil vere å auke den teoretiske forståinga for programmering og undervisning rundt emnet, og det andre vil vere å utvikle ein intervensjon som kan nyttast i klasserommet. Dette går igjen i det Anderson & Shattuck, 2012; McKenney & Reeves, 2013 skriv der dei forklarar at DBR som regel har eit dobbelt føremål der teori og praksis står på kvar si side.

Dette prosjektet var prega av eit tett samarbeid mellom forskar og lærar. Eg har då som forskar bidratt med min kunnskap om analog programmering og undervisning som eg har opparbeidd gjennom lesing av forskingslitteratur, medan lærar har bidratt med kunnskap og erfaring som han har opparbeidd gjennom utdanning, vidareutdanning, kurs og

praksiserfaringar. Dette var noko eg såg på som veldig nyttig, då lærarar ofte har ein hektisk kvardag og det er rett og slett for tidkrevjande å drive med forskingsarbeid i tillegg til dei daglege gjeremåla. Eg, som forskar, har på mi side ikkje den same innsikt og forståing for praksisfeltet, og drar difor nytte av læraren sine erfaringar på dette feltet (Anderson & Shattuck, 2012). Dette blir støtta av The Design-Based Research Collective (2003) som skriv at målet i forskingsmetodikken er at partane i prosjektet skal kunne dra nytte av kvarandre ved at forskinga lyser opp dei ulike forholda i klasseromssituasjonen og at dette vil bidra til at det blir bygd bru mellom teoretisk kunnskap og praksis.

Prosjektet mitt hadde ei direkte kopling til praksisfeltet med tanke på gjennomføring av undervisning og samarbeid med lærar. Dette gjer at DBR er ein godt eigna forskingsmetodikk når ein skal designe, bruke og forske på pedagogiske verktøy og materiale i ein naturleg kontekst (The Design-Based Research Collective 2003). Gjennom å teste ut, forandre og utvikle, er målet å skape ei forbetring. Og gjennom den direkte koplinga til praksisfeltet, gjev det meg på denne måten moglegheit for å komme tett på det materialet ein ønskjer å undersøke. Metodikken var på den måten godt eigna for mi masteroppgåve, då eg skulle sjå på bruk av analog programmering for å fremje skaparglede, engasjement og utforskartrøng og om det var lett å gjennomføre for den enkelte lærar. For å få ein oversikt over framgangsmåten for DBR kan ein ta utgangspunkt i Figur 6 med nøkkelorda; Planlegge, Gjennomføre, Evaluere og Reflektere.



FIGUR 7: FIRE SENTRALE FASAR I DBR OVERSETT TIL NORSK (SCOTT ET AL 2019)

Planlegge undervisningsopplegget, spesifisere måla for økta, målgruppa, aktiviteten og kva verktøy som skal brukast

Gjennomføre undervisningsopplegget, overvake og samle inn data

Evaluerer undervisningsopplegget inneber å analysere data for å forstå resultatet av opplegget og kva innverknad dei forskjellige elementa har hatt

Reflektere over resultatata frå undervisningsopplegget, vurdere og reflektere over implikasjonane av resultatata for å hjelpe å identifisere kva som kan betrast.

Metoden er karakterisert ved prosessar der data blir samla inn kontinuerleg for å evaluere ein intervensjon og tilpasse den til lokale behov (Bradley & Reinking, 2011). Slike prosesser resulterer i utvikling av prinsipp for vidare innvending og utvikling av intervensjonen (Anderson & Shattuck, 2012; McKenney & Reeves, 2013). Truverdet til design-basert forskning vert styrka gjennom retrospektiv og systematisk analyse av prosessen, gjennom den teoretiske innramminga av analyse og gjennom tydeleggjering av kriteri, datamateriale og tolkingar, og i tillegg kan lokale praksisnære innovasjonar samanliknast med overordna teoretiske anbefalingar (Cobb, et al. 2003). Design-basert forskning har ofte blitt anvendt i grunnskule og vidaregåande opplæring, samt innanfor utdanningsteknologi (Anderson & Shattuck, 2012).

3.2 Deltakarar, Informant og forskarrolle

I denne masteroppgåva har eg følgd ei klasse på 6. trinn på ein ungdomsskule og læraren deira medan dei gjennomførte ei utprøving av eit nytt undervisningsopplegg i programmering. Elevgruppa i klassa er nokså likt fordelt mellom kjønn, og ligg på eit matematisk nivå som tilsvarer det norske gjennomsnittet samanlikna med nasjonale prøver etter 5 trinn. I klassen er det relativt mykje uro og bråk, noko som gjer at læraren kan ha vanskar med å undervise om ein ikkje har med seg ekstra ressursar. Dette var noko eg og lærar tok omsyn til i planlegginga av undervisningsopplegget.

Kvalitative metodar og DBR har, som nemnt over, eit tett samarbeid mellom forskar og deltakar, og kunnskapen blir skapt rundt dette samspelet. Forskarrolla i DBR er likevel svært involvert i forskingsdesignet, og det er av den grunn naudsynt å gjere greie for kva roller eg og min informant hadde i prosjektet

Utvalet i denne studien kan ein trekke mot eit bekvemmelighetsutval, då eg forskar på ein lærar eg hadde godt kjennskap til frå før (Høgheim, 2020). Dei siste to åra har eg jobba som vikarlærar ved ein grunnskule ved sidan av studiet. Gjennom vikarjobben kom eg i kontakt med ein lærar som var interessert i prosjektet og som ønskja seg ein alternativ inngang til tema programmering. Læraren eg fekk kontakt med såg på dette som eit høve til å få utvikla kompetansen sin i programmering, sjølv om dei hadde jobba litt med tema tidlegare.

Fordelen med eit slik utval er at den er kostnadssparande (Jacobsen 2005). På den andre sida kan ein argumentere for at utveljinga og er kriteriebasert sidan informanten måtte oppfylle spesielle krav som det å undervise i matematikk (Johannessen et al. 2021). Sidan eg i dette prosjektet undersøker læraren sine erfaringar med bruk av analog programmering i undervisninga i matematikk var det naudsynt at informanten var matematikklærar, då dette masterprosjektet er knytt til matematikkdiraktikk.

Læraren si rolle i prosjektet har vore å planlegge, gjennomføre og evaluere undervisning i samråd med meg, samt delta i fleire intervju både før, undervegs, og i etterkant av undervisningsopplegga. Her har vi både hatt meir planlagde/formelle samtalar, men og mange uformelle og korte samtalar som har bidratt til å utvikle undervisningsopplegget og evalueringa av dette. Vi har saman utarbeidd undervisningsopplegg, men det er læreran som har hatt hovudansvaret med gjennomføringa av undervisninga i praksis. Læraren har på den måten bidratt med sine erfaringar og kompetanse til å lyse opp den aktuelle problemstillinga, noko som er ei styrke med denne metoden.

3.3 Datainnsamlingsverktøy

For å kunne sjå om undervisningsopplegget fremja skaparglede, engasjement og utforskartrong hjå elevar var det naturleg at eit av verktøya til å samle inn data falt på observasjon. Dette såg eg på som naturleg, då systematisk observasjon har det føremon at eg kan få eit direkte innblikk i interaksjonane lærar-elev og elev-elev i klasserommet, og desse interaksjonane opp mot undervisningsopplegget. All den tid observasjonen min er ei tolking av kva eg som forskar ser, og ikkje ei direkte måling av omgrepa, vil det vere feilkjelder. For å redusere dette og for å unngå at vurderingane mine vart personlege, tok eg utgangspunkt i eit observasjonsverktøy i tillegg til min eigen observasjon. Eg tok i bruk observasjonsinstrument som tidlegare har blitt brukt og validert, og dette såg eg på ein styrke for å luke bort eventuelle feilkjelder. Det finst mange forskjellige instrument for å observere.

Mange av dei krev trening, kurs eller sertifisering, men det finst og nokon som er fritt tilgjengelege. Med tanke på kva som er gjennomførleg i ei masteroppgåve fall valet mitt på eit observasjonsverktøy som var fritt tilgjengeleg. Det vil vere ein del veikskap ved å nytte desse verktøya, sidan ein ikkje veit i kva for grad forskjellige observatørar vil legge merke til dei same observasjonane. For å styrke min eigen observasjon var det difor naturleg for meg å gjennomføre intervju med lærar etter fullført undervisningsopplegg. Eg ville høyre om lærar hadde oppfatta det same eller hadde andre erfaringar. Læraren kjenner elevane best, og har gjerne innspel som ikkje eg som forskar la merke til. Fordelane med å kombinere intervju og observasjon er fleire. Først av alt kan det gje forskingsresultatet ein betre heilskap, sidan det gjer det mogleg å få informasjon både frå aktiv deltaking og passiv observasjon. I tillegg kan det hjelpe meg som forskar å få bekrefte resultat som vart funnet under observasjon og intervju.

Det andre forskingsspørsmålet mitt var å sjå kva grad og korleis kan eit undervisningsopplegg med analog programmering vere attraktivt for lærarar. Her er det berre lærar sjølv som kan meine noko om opplegget, og det fall naturleg å nytte intervju for å få lærar sine egne erfaringar.

Med bakgrunn i dette bestemte eg meg for at intervju og observasjon vart verktøya eg ville nytte for datainnsamlinga mi. Postholm & Jacobsen (2018) skriv at intervju og observasjon er viktige datainnsamlingsmetodar innanfor kvalitativ forskning. Dette skuldast at dei to metodane bidreg til å gjere greie for korleis menneske konstruerer verkelegheita, samt får fram nyansar og variasjonar som ikkje alltid er like klare frå før. I dette delkapittelet vil eg gå nærmare inn på intervju og observasjon som metode for datainnsamling, skildre kva som har blitt gjort og diskutere fordelar og ulemper med egne val.

3.3.1 Intervju

I intervju eller samtalane som vart gjennomført med lærar i dette prosjektet var målet å få innsikt i denne personen sine erfaringar rundt undervisningsopplegget. Tykte læraren dette var noko som var gjennomførleg? Tykte læraren at elevane viste skaparglede, engasjement og utforskartrøng undervegs i opplegget? Intervju og samtalane eg og lærar gjennomførte vart gjort både på ein semistrukturert måte og ustrukturert. Ved bruk av semistrukturert intervju hadde eg rom for utforsking av uventa, men relevante emne undervegs i intervjusituasjonen. Dette førte til at eg som forskar kunne avdekke og undersøke nye element under heile

forskningsprosessen. I eit semistrukturert intervju har forskaren eit overordna mål for intervjuet, og gjerne nokon ferdig formulerte spørsmål eller tema (Høgheim, 2020). I tillegg til dei semistrukturerte intervjua, hadde eg og lærar fleire uformelle dialogar. Vi jobba tett saman, og det var difor hensiktsmessig for oss å ha fleire små dialogar, dialogar som fall utanfor den strukturerte delen. Her kom det ofte fram små element som vi kunne ta med oss vidare. Noko som kom særst godt fram ved desse samtalane var definisjon av omgrep mellom meg som forskar og lærar. Vi såg på det som viktig og sentralt at vi hadde identisk oppfatning av dei ulike omgrepa, og vi snakka mykje rundt desse for å kome til einigheit.

For å svare på problemstillinga mi vart det i forkant av planlegginga til «økt 1» utarbeidd tre enkle, men presise intervjuguidar. Ein guide vart utarbeidd for samtalen eg og lærar hadde før oppstart av den første undervisningsøkta, ein for samtalen etter gjennomføring av økt 1 og ein for økt 2 (Sjå vedlegg 1-3). Dette vart gjort for min eigen del, slik at eg hadde nokre klare punkt som eg ønska å vite meir om, samt for å halde oversikt og kontroll. Intervjuguidane som vart utarbeidd i høve prosjektet var forholdsvis fleksible. Intervjuguide 2 og 3 tok utgangspunkt i undervisningsopplegget (Sjå tabell 3 og 4). Dei romma nokre element som ga ordlyden for kva som skulle dekkast i spørsmåla. Dette er i tråd med det Kvale & Brinkmann (2015) peikar på ved ein intervjuguide, nemleg at det skal vere eit verktøy for å strukturere intervjuforløpet. Under intervju/samtalane valde eg å notere ned dei elementa frå samtalen som eg såg på som viktige for å kunne analyse datamateriale i etterkant, samt for å nytte det til å forbetre undervisningsopplegget. Her vil eg og påpeike at eg kontinuerleg hadde hyppige samtalar med lærar der vi tok opp spørsmål frå intervjuguidane og andre tankar og idear ofte.

3.3.2 Observasjon

Når eg skulle observere ei klasse for å sjå på skaparglede, engasjement og utforskartrøng hadde eg behov for eit verktøy som kunne hjelpe meg med å måle desse omgrepa. Her valde eg OECD sitt verktøy for observasjon i skulen. OECD sin «Global Teaching Observation tool» (OECD, 2020) er eit verktøy som blir nytta til å støtte lærarar og leiarar i å måle og utvikle klasseromspraksis. Dette verktøyet gir god hjelp til å systematisk analysere og forbetre undervisning, læringsmiljø og læringsresultat. Verktøyet vil då hjelpe meg til å identifisere styrker og utfordringar ved læring og undervisning, samt finne tiltak og iverksette handlingar for å forbetre læringsresultata. Observasjon i klasserommet gjev meg høve til å sjå kva som eigentleg skjer i eit klasserom, i staden for å stole på skildringar og tankar om det som skjer.

Eg har valt å gjere ein ikkje-deltakande observasjon. Det vil gje meg høve til å ta eit steg unna senteret av klasserommet, og eg kan reflektere utan å vere påverka av gjennomføringa av undervisningsopplegget. Dette vil kunne gje meg ein ny type innsikt, og gi meg idear til vidareutvikling av opplegget. Det å la ein anna lærar undervise, og sjå korleis dei handterer ulike situasjonar i klasserommet, vil og vere ei rik kjelde til nye idear.

OECD sitt observasjonsverktøy er delt opp i seks hovuddelar som alle tar for seg eit emne innan klasseromsundervisning. For å svare på problemstillinga mi var det for omfattande å sjå på alle dei ulike delane, så eg valde ut element frå to av desse for å få hjelp til å måle skaparglede, engasjement og utforskartrøng blant elevane. Her har eg valt ut element frå «social-emotional support» og «student cognitive engagement»

- «Social-emotional support».

Her skriv OECD (2020) at elevane må få utføre arbeid, uttrykke idear og tankar som avslører utfordringar og misoppfatningar. For at elevane skal vere villige til å vere sårbare, til at det kan skje, er det viktig at dei kjenner seg godt støtta og er skjerma frå sjenanse. Er dette på plass, så kan det opne for eit trygt klassemiljø der elevane kjenner seg trygge og motiverte. Motivasjonen kan her vere med på å skape engasjement. Vidare vil tryggleik skape eit klassemiljø der det er akseptabelt å kome med nye idear og tankar utan at det skal bli sett ned på. I figur 8 og 9 ser vi korleis det kan brukast og kva element som i følge OECD blir sett på som viktige for å oppnå «social-emotional support»

I. OBSERVATION CHART				
<p>RESPECT</p> <p>The teacher treats all students with respect</p> <p><input type="radio"/> agree</p> <p><input type="radio"/> somewhat agree</p> <p><input type="radio"/> disagree</p> <p><input type="radio"/> unable to judge</p> <hr/> <p>Students demonstrate respect for each other</p> <p><input type="radio"/> agree</p> <p><input type="radio"/> somewhat agree</p> <p><input type="radio"/> disagree</p> <p><input type="radio"/> unable to judge</p>	<p>ENCOURAGEMENT AND WARMTH</p> <p>The atmosphere is</p> <p><input type="radio"/> very warm and friendly</p> <p><input type="radio"/> somewhat warm and friendly</p> <p><input type="radio"/> not at all warm and friendly</p> <hr/> <p>The teacher is encouraging and open</p> <p><input type="radio"/> very</p> <p><input type="radio"/> somewhat</p> <p><input type="radio"/> not at all</p>	<p>RISK-TAKING</p> <p>Students voluntarily share their work and ideas</p> <p><input type="radio"/> agree</p> <p><input type="radio"/> somewhat agree</p> <p><input type="radio"/> disagree</p> <p><input type="radio"/> unable to judge</p> <hr/> <p>Students seem willing to seek guidance or ask questions</p> <p><input type="radio"/> agree</p> <p><input type="radio"/> somewhat agree</p> <p><input type="radio"/> disagree</p> <p><input type="radio"/> unable to judge</p>	<p>PERSISTENCE</p> <p>The teacher encourages students not to give up when they encounter difficulties</p> <p><input type="radio"/> agree</p> <p><input type="radio"/> somewhat agree</p> <p><input type="radio"/> disagree</p> <p><input type="radio"/> unable to judge</p>	<p>PUBLIC SHARING</p> <p>Students are invited to share at the front of the class</p> <p><input type="radio"/> rarely</p> <p><input type="radio"/> sometimes</p> <p><input type="radio"/> frequently</p>

Things that caught your attention relating to social-emotional support:

FIGUR 8: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 12

II. OVERALL SUMMARIES

Most students seemed comfortable participating in the lesson

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

The teacher listened attentively and with interest to students

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

The students were mean or disrespectful to each other

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

The atmosphere in the classroom was cheerful

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

The teacher was encouraging and patient with students who struggled

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

Unable to judge

Students did not seem worried or afraid of making mistakes

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

Would you have done anything differently in this lesson with respect to socio-emotional support?

FIGUR 9: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 13

- «Student cognitive engagement»

OECD skriv og at om elevar skal kunne kome seg vidare med læring og førebu seg på ei verd med problemløysing, så bør dei intellektuelt stimulerast og utfordrast av innhaldet i klasserommet. Det å engasjere elevane i arbeid som krev analyse, skaping, evaluering,

omtanke, samtidig som det promoterer fleire måtar å tenkje på og vektlegging på grunngevinga bak prosesser og metodar, kan vere ein kraftfulle måte å presse eller utfordre elevane til å bygge ein djup konseptuell forståing. Element som skaping, analyse og evaluering kan bidra til skaparglede og utforskartrøng. I figur 10 og 11 ser vi korleis det kan brukast og kva element som i følge OECD blir sett på som viktige for å oppnå «student cognitive engagement»

I. OBSERVATION CHART			
<p>Students were cognitively challenged and stretched intellectually</p> <p><input type="radio"/> yes <input type="radio"/> no</p> <p>The teacher asked students to (check all that apply)</p> <ul style="list-style-type: none"> <input type="radio"/> evaluate strengths and weaknesses of ideas, solutions or approaches. <input type="radio"/> explain or justify ideas, solutions or approaches. <input type="radio"/> Defend an idea, solution or approach. <hr/> <p>Students had opportunities to think critically</p> <ul style="list-style-type: none"> <input type="radio"/> rarely <input type="radio"/> sometimes <input type="radio"/> often 	<p>Teaching asked students to learn a procedure or method</p> <p><input type="radio"/> yes <input type="radio"/> no</p> <p>The teacher asked students to (check all that apply)</p> <ul style="list-style-type: none"> <input type="radio"/> articulate the logic behind their answers <input type="radio"/> state the steps used in their work <input type="radio"/> review and summarize the rationale for each step a procedure or method <input type="radio"/> repeat back or explain steps <hr/>	<p>The focus was as much on learning the steps as it was on understanding them</p> <ul style="list-style-type: none"> <input type="radio"/> agree <input type="radio"/> somewhat agree <input type="radio"/> disagree <hr/> <p>Students had opportunities for deliberate repeated practice of the methods or procedures being learned</p> <ul style="list-style-type: none"> <input type="radio"/> yes <input type="radio"/> somewhat <input type="radio"/> no <hr/>	<p>Students had opportunities to engage in multiple approaches to reasoning</p> <ul style="list-style-type: none"> <input type="radio"/> yes <input type="radio"/> no <hr/> <p>The teacher asked students to:</p> <ul style="list-style-type: none"> <input type="radio"/> explain similarities and differences <input type="radio"/> solve problems using different strategies <input type="radio"/> place themselves in a contrasting situation <input type="radio"/> work backwards or inversely <input type="radio"/> use a counter example <input type="radio"/> other: <hr/>
<p>METACOGNITION</p> <p>Students had opportunities to (check all that apply)</p> <ul style="list-style-type: none"> <input type="radio"/> reflect about their assumptions or beliefs <input type="radio"/> reflect about difficulties they encountered or points of confusion <input type="radio"/> review the steps in their thinking <input type="radio"/> reflect about changes that had occurred in their knowledge or understanding <input type="radio"/> synthesise or summarise their learning 			
<p>Things that caught your attention about cognitive engagement:</p>			

FIGUR 10: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 22

II. OVERALL SUMMARIES

The teaching placed genuine intellectual demands on the students

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

Student understanding was a strong priority in the teaching

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

The lesson exposed students to multiple approaches of thinking or a variety of ideas

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

The amount of practice time was

Too little Just right Too much

The time that students were given to self-reflect/self-assess was

Too little Just right Too much

The teaching made use of technology to support conceptual understanding

Strongly agree Agree Somewhat agree Disagree Strongly Disagree

Would you have done anything differently in this lesson with respect to engaging students intellectually?

FIGUR 11: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 23

Når eg observerte valte eg å sette meg bak i klasserommet. Dette gjorde eg for å få god oversikt utan at eg skulle påverke elevane på nokon måte. Eg tok med meg observasjonsdokumenta som eg hadde skrevet ut fysisk samt eit eige blankt ark for egne observasjonar. Her noterte eg undervegs og etter kvart element frå undervisningsopplegget med utgangspunkt i hovuddelane frå tabell 3 og 4. OECD sitt verktøy åleine var ikkje nok til å kunne svare på alle mine spørsmål. Eg valte difor å gjere nokre egne vurderingar for å kunne kople «eventuell måloppnåing» til dei ulike prinsippa/fasane i gjennomføringa. Dette var meir ein overordna observasjon av totalen frå kvar delane i undervisningsopplegget. Her såg eg på heilheita og gjorde meg nokre tankar som eg ville seinare diskutere/ta opp med lærar.

3.3.3 Pilotering

Før eg starta med å observere valde eg å gjere ei pilotering. Pilotering er nyttig i observasjon, fordi det gir meg høve til å teste og kvalitetssikre datainnsamlingsprosessen før den fullstendige undersøkinga vert gjennomført. Det gir meg og høve til å identifisere og rette

eventuelle feil eller veikskap i designet eller prosessen, noko som igjen kan auke pålitelegheita og validiteten til dei endelege resultatane. Før gjennomføring tok eg utgangspunkt i førebuingsskjemaet til OECD, der dei har skrivne åtte punkt som ein bør ta med seg før ein startar observasjonen. Her valte eg å undervise ein matematikktime i ein anna klasse sidan det var det som var lettast å gjennomføre og passa godt med timeplanen til dei involverte.

BEFORE GETTING STARTED

As you prepare to use these tools, there are several important points for you to keep in mind.

- 1** When you observe others teaching please remember that:
 - The objective is not to evaluate but to reflect, question and consider.
 - You should keep an open mind in order to be as receptive as possible to unfamiliar approaches and new ideas.
 - It will be best to reason about practice based on the concrete evidence that you observe.
- 2** The tools provide you with lenses to help focus your attention on critical aspects of teaching, but these lenses are neither exhaustive nor mutually exclusive. Thus, representing teaching as a pie that can be divided into distinct slices is just a simplification to help organise your thinking. So, please remember that:
 - You should always feel free to exercise your professional judgement as you use our tools.
 - You should not hesitate to extend your observations beyond our suggestions in order to explore areas that are of particular interest to you.

- 3** Teachers are made not born, and professional growth takes time and patience. In fact, researchers have found that observation needs to be carried out in a sustained way to actually have an impact on what teachers notice, believe and, most importantly, do in the classroom. So please remember that:
 - You should not be discouraged, improving practice is a long process.
 - You are not alone. Turn to colleagues to talk to them about your struggles and to share ideas.
 - Take advantage of Global Teaching InSights digital site. It is a rich and unique resource that can empower you to develop a global perspective on practice.

FIGUR 12: OECD GLOBAL TEACHING INSIGHT OBSERVATION TOOLS SIDE 5

For å så kontrollsjekke intervjuguiden før samtalan, samt få prøve meg i ein intervjusituasjon, valde eg og å gjennomføre eit pilotintervju i forkant av datainnsamlinga. Her brukte eg timen som eg observerte for å teste ut observasjonsskjema mine, for å så få stille lærar nokre spørsmål etterpå. For å ikkje bruke for mykje av læraren si tid valde eg berre å bruke nokre minutt på dette.

3.4 Utvikling av intervensjon over fleire forskingsyklusar

Etter å ha lese meg opp på ulike teoriar rundt programmeringsdidaktikk og læringsteori (sjå kap. 2), så har eg kome fram til nokre prinsipp innan tre forskjellige rammeverk som dannar

grunnlaget for utviklinga av undervisningsopplegget. For å skaffe oversikt har eg valt å dele dei ulike måla for økta inn i kategori 1 til 4, og rammeverka frå teori i A, B og C. Prinsippa blir då element frå rammeverka som nyttast for å kunne nå måla (1-14). Tabell 2 viser måla for undervisningsøktene, kva prinsipp frå rammeverka som er brukt og datakjelder for å kunne vurdere om måla blir nådd gjennom bruk av prinsippa

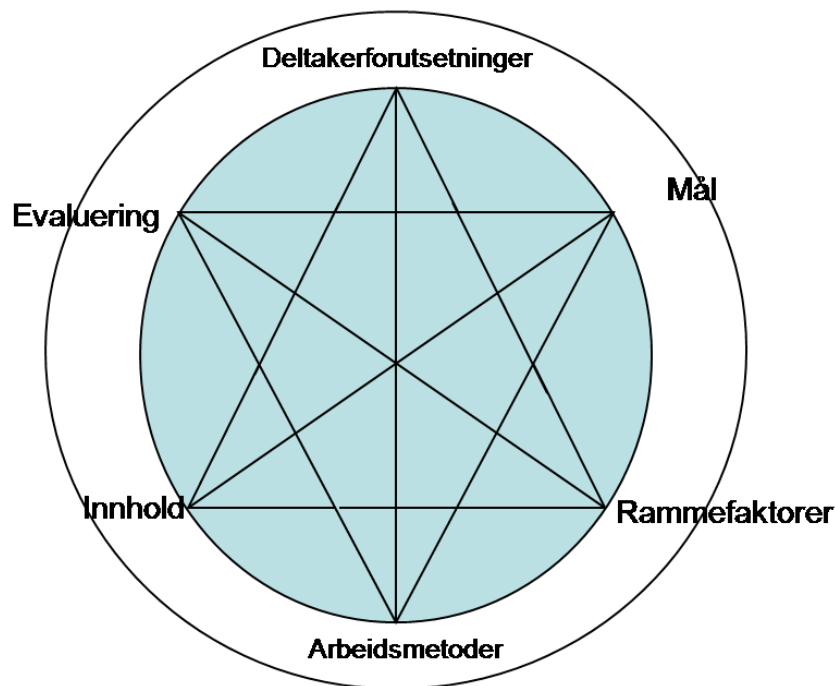
Prinsipp for å nå mål om skaparglede, engasjement og utforskartrøng for elevar samt gjere timen gjennomførbart for lærar				
Mål	Rammeverk	Prinsipp	Korleis samle inn data	Eigen observasjon
1) Fremje skaparglede	A) PRIMM-modellen B) 5Es	1A – 1 – Utforske kode 1A – 2 - Prøve å feile 1B – 3 – Utveksle idear	<ul style="list-style-type: none"> OECD – Student Cognitive Engagement Intervju - Lærar 	Viser elevane skaparglede når dei sjølv får lage kode?
2) Fremje engasjement	C) Tenkande klasserom A) PRIMM-modellen	2C – 4 – Tilfeldige grupper 2C – 5 – Ikkje permanent arbeidstavle 2A – 6 – lage hypotese 2A – 7 – Kode på eigenhand	<ul style="list-style-type: none"> OECD – Social-Emotional Support OECD – Student Cognitive Engagement Intervju - Lærar 	Viser elevane engasjement når dei får vite kva som skal gjerast?
3) Fremje utforskartrøng	C) Tenkande klasserom B) 5Es A) PRIMM-modellen	3C – 8 – Tenkande oppgåver 3B – 9 – Sjå føre seg 3B – 10 – Utforske kode 3A – 11 – Lage eigen kode	<ul style="list-style-type: none"> OECD – Student Cognitive Engagement OECD – Classroom Talk Intervju - Lærar 	Korleis respondera elevane når dei sjølv skal finne ut korleis den gitt koden ser ut?
4) Gjere undervisninga gjennomførbart for lærar	Samtale med lærar om kva som gjer ein time lett gjennomførbart	12 – Få overgangar 13 – Mykje elevaktivitet 14 – Enkle ressursar	<ul style="list-style-type: none"> Intervju - Lærar 	Er det god flyt i undervisningsopplegget?

TABELL 2: PRINSIPP FOR Å NÅ MÅL I UNDERVISNING

For å nå mål 1 til 3, valte eg å trekke inn forskjellige prinsipp frå dei tre rammeverka eg valte meg (Benton et al., 2017, Liljedahl, 2021 og Sentance et al., 2019). For å nå mål 4 der undervisningsopplegget skal vere gjennomførlig og attraktivt valde eg å høyre med læraren kva han meiner er ein god undervisningstime. Her tok eg ein samtale med lærar for å høyre kva element som var ønskeleg å ha med, ved hjelp av intervjuguide (sjå vedlegg 1). Sidan vi hadde eit tett samarbeid i prosjektet, vart desse elementa ofte diskutert i samband med utarbeidinga av undervisningsopplegget og ikkje kun i eit enkelt intervju. Eg ville nytte læraren sine erfaringar i praksis, samt min eigen kunnskap med bakgrunn i teori, til å kunne drøfte og kome fram til korleis ein matematikktime burde og kunne sjå ut. Dette vart gjort for

at eg skulle kunne legge inn nokre ønskelege prinsipp for undervisningsopplegget. Denne samtalen enda opp i tre prinsipp som vi valte å ta med oss i tabell 2. Desse var at undervisningøkta skulle ha få overgangar, mest mogleg elevaktivitet gjennom økta og gjennomføre opplegget med enkle ressursar.

For å planlegge undervisningsopplegget tok eg utgangspunkt i å organisere det etter den didaktiske relasjonsmodellen. Modellen tar utgangspunkt i seks gjensidig avhengige element når ein planlegg undervisning. Desse seks elementa er deltakarforutsetningar (målgrupper), rammefaktorar, mål, innhald, arbeidsmetodar (pedagogisk tilnærming) og evaluering.



FIGUR 13: DEN DIDAKTISKE RELASJONSMODELLEN (BJØRNDAL OG LIEBERG 1978 I HIIM OG HIPPE 2022)

Didaktikk handlar om tankane knytt til verksemda sine Kva (mål og innhald), Korleis (metodikken) og Kvifor (drøfting og grunngevingar). Eg har difor valt å sette undervisningsopplegget opp i eit Kva, korleis og kvifor-skjema for å gjere det oversiktleg både for meg sjølv og for lærar. I planlegginga, utarbeidde eg og lærar saman ei oversikt over korleis undervisninga skulle gjennomførast gjennom kva, korleis og kvifor-skjema og brukte den som utgangspunkt når vi evaluerte og justerte opplegget etter økt 1 og 2.

3.4.1 Første økt

Tanken med den første undervisningsøkta var at elevane skulle få ein introduksjon til analog programmering. Her ville vi i hovudsak at elevane skulle oppleve skaparglede, engasjement og utforskartrøng i arbeidet med analog programmering. Ved sidan av vårt overordna mål ønska vi at elevane skulle få praktisk erfaring i å kode, oppnå ei forståing for kodinga, samt lære og erfare kor viktig nøyaktige og presise instruksar er rundt emnet programmering. Hovudårsaka til dette fokuset var fordi lærar ønska at elevane skulle få ei grunnleggjande forståing rundt kva det vil seie å programmere. Tabell 3 viser undervisningsøktas struktur, innhald, læringsaktivitet og kvifor vi har valt læringsaktiviteten og kva data eg ønskja å samle inn under opplegget.

Undervisningsopplegg 1

Praktisk informasjon		Fysiske rammer	
Forskar: Øystein Østvik	Fag: Matematikk	Aktivitetsleiar: Kontaktlærer	Antall lærarar og fagarbeidarar: 1 kontaktlærer + hjelpearar.
Dato: 15.02.2023	Innhald: Lære å gjere om verkelege handlingar til instruksjonar.	Antall elevar: 19, 6. trinn	Fysisk organisering: 5 grupper i vanlig klasserom fordelt på arbeidsstasjonar.
Klokkeslett: 11.55 – 12.40	Få praktisk erfaring i å kode med symbol. Oppnå ein forståing for presisjon innan koding. Erfare korleis «debugging» brukast til å finne feil i koden.		Whiteboards til å skrive på og koppar til å jobbe med. Ark med oppgåver og forklaringar.

Undervisningas struktur		Undervisningas kva, korleis og kvifor			Datainnsamling
Omtrentleg tidsbruk	Del av undervisningsøkt Namn på delane av undervisningøkta.	Innhald Kort forklaring om fagleg innhald	Læringsaktivitet Her skildre eg undervisningsmetoden og elevaktivitet	Grunngjevnad Her grunngjev eg vala eg har tatt med forankring i teori	Kva ser eg etter i kva del og korleis kan eg måle det?
10 min	Introduksjon	Etablere ein forståing for kva vi skal gjere og jobbe med	Start med å spørje klassa om robotar. Kva er det, kva bruke vi det til?. Kan ei sånn maskin høyre oss snakke? Forstår den «faktisk» det som blir sagt? Svaret på det er «ikkje på same måte som ein person gjer»	Gjennomgang for å i varena prinsipp 12 og 13 (tabell 2) så elevane kan halde på med mykje elevaktivitet og få overgangar. Her vil	Her vil eg ha fokus på om det er lett for elevane å forstå kva som er planen for økta. Og om det er lett for lærar å gjennomgå omgrepa med elevane.

			<p>Robotar treng ein serie med presise «instruksjonar» som ofte blir kalla algoritmar. Spesifikke ting dei har blitt satt til å gjere. For å kunne gjennomføre ei oppgåve, må ein robot altså ha ein rekke algoritmar for å fungere. I dag skal vi lære om kva som trengs for å få det til.</p> <p>Gå igjennom tre omgrep som vi skal vite om:</p> <p>Algoritme – Ein serie av instruksjonar om korleis vi kan nå eit mål. (Døme Korleis rekne vi saman 25+25 (starte med addere bakarste tal osv.. Rett og slett ei <i>oppskrift</i>)</p> <p>Koding – Gjere om handlingar til eit symbolspråk.</p> <p>Debugging – Finne og fikse feil og problem i koden (I symbolspråket)</p> <p>Vis fram Symbol-ark (vedlegg 6). Fortel at vi skal kun bruke dei seks symbola her i oppgåvene og vi skal bruke dei til bygge tårn med koppar. (Dette er då symbolspråket vårt)</p> <p>Forklar første oppgåve før neste bolk: Elevane får ei ferdig utdelt ein kode (vedlegg 7) på tavla si. Her skal elevane diskutere og tippe korleis resultatet blir om ein følgje koden rett. Korleis vil tårnet sjå ut? (minn elevane på at «Eit skritt fram/tilbake» er ein <i>HALV</i> kopplengde)</p>	<p>det og bli ivaretatt</p> <p>14 sida ein treng kun papir/tavle og koppar til oppgåvene.</p> <p>Ordforråd for å hjelpe til å forstå og i vareta prinsipp 1, 2, 3, 6 og 11 seinare i opplegg der elevane bør kunne grunnleggjande omgrep for å utforske og spå kva som skjer i koden.</p>	<p>Dette vil eg finne ut i samtale med lærar etter økt</p> <p>Eg og sjå i eigen observasjon om elevane fremje skaparglede, engasjement og utforskartrong når lærar forklarar kva som skal gjerast</p>
5 min	Inndeling	Inndeling i grupper med ikkje permanente tavler	Dele opp i tilfeldige grupper og fordele dei på arbeidsstasjonar og	For å i vareta prinsipp 4 og 5 (Desse går igjen resten av økta).	Vil tilfeldige valte grupper og ikkje permanente arbeidsflater bidra til meir engasjement i gruppene. Dette vil eg sjå på med hjelp av eigen observasjon og lærarintervju
25 min	Utforsking og utprøving av koding	Utforsking og prøving av ei ferdig gitt kode	Etter nokre minutt når elevane har tankar om korleis tårnet vil sjå ut kan lærar vise på framme på kateter med til dømes ein elev.	For å nå prinsipp 1, 3, 6, 8 og 10 . Der elevane skal få tenke, utforske og utveksle.	Her vil eg fokusere på utforskartrong og skaparglede i hovudsak ved bruk av «Social emotional support» og «Students cognitive engagement».

			Elevane får jobbe med oppgåve A til K.	For å nå prinsipp 7, 8, 9 og 11 der elevane får lage, prøve og samarbeide i arbeid med kode.	
5 min	Oppsummering	Ei gruppe kan vise sin kode	Høyr med elevane om dei fekk det til oppgåvene. Kva var typiske feil? (Prøve å kople det mot at når ein programmere må ein vere presis.) Kunne vi forenkla kodane våre på nokre måtar? (Neste time: Funksjon og parameter)	For å nå prinsipp 3 og 7. Her vil eleva utveksla og køyrt koden sin.	Sjå på mål om skaparglede og engasjement. OECD sitt skjema via «Students cognitive engagement», eigen observasjon om det ser ut som elevane ønskje å jobbe meir med dette og intervju med lærar.

TABELL 3: UNDERVISNINGSSOPPLEGG 1 I KVA, KORLEIS, KVIFOR SKJEMA

I etterkant av første økt evaluerte læraren og eg den gjennomførte undervisningøkta, og la ein grovplan for neste økt ved å gjere justeringar ut i frå dei erfaringane vi hadde gjort oss (Sjå figur 6 design). Gjennomføringa av dei planlagde aktivitetane tok litt lengre tid enn vi først hadde antatt, noko som gjorde at elevane ikkje fekk gjort meir enn eit par oppgåver frå oppgåvearket. Dette tok vi omsyn til i undervisningsopplegg 2, og gjorde nokre justeringar slik at elevane fekk sette i gang med oppgåver tidlegare. Elevane skulle halde fram med arbeidet sitt frå forrige time. Etter undervisningsopplegg 1 valde vi og å gjere nokre justeringar på elevgruppa. Nokre elevar i klassen har læreutfordringar og kontaktlærar ønskte å sende desse ut med hjelpelærar. Dette gjorde at vi fekk ei mindre elevgruppe som var lettare for lærar å følge opp. Eg og lærar såg på dette som noko naturleg å gjere sidan det var normal praksis i andre matematikktimar.

3.4.2 Andre økt

I undervisningsopplegg 2 starta vi med ein liten repetisjon frå forrige økt, før vi introduserte to nye omgrep. Her skulle elevane få lære om funksjonar og parametarar. Tanken med økta var dei same som i undervisningsopplegg 1, men i tillegg skulle dei prøve å få danne seg ei forståing av kor nyttig funksjonar og parametarar kan vere i arbeid med programmering.

Undervisningsopplegg 2

Praktisk informasjon		Fysiske rammer	
Forskar: Øystein Østvik	Fag: Matematikk	Aktivitetsleiar: Kontaktlærer	Antall lærarar og fagarbeidarar: 1 kontaktlærer
Dato: 08.03.2023	Innhald: Lære å gjere om verkelege handlingar til instruksjonar.	Antall elevar: 14, 6. trinn	Fysisk organisering: 3 grupper i vanlig klasserom fordelt på arbeidsstasjonar.
Klokkeslett: 11.55 – 12.40	Få praktisk erfaring i å kode med symbol. Oppnå ein forståing for presisjon innan koding. Erfare korleis «debugging» brukast til å finne feil i koden. <i>Forstå kor nyttig funksjonar og parametar er. (NY)</i>		Whiteboards til å skrive på og koppar til å jobbe med. Ark med oppgåver og forklaringar.

Undervisningas struktur		Undervisningas kva, korleis og kvifor			Datainnsamling
Omtrentleg tidsbruk	Del av undervisningsøkt Namn på delane av undervisningøkta.	Innhald Kort forklaring om fagleg innhald	Læringsaktivitet Her skildre eg undervisningsmetoden og elevaktivitet	Grunngjevnad Her grunngjev eg vala eg har tatt med forankring i teori	Kva ser eg etter i kva del og korleis kan eg måle det?
10 min	Introduksjon	Etablere ein forståing for kva vi skal gjere og jobbe med	Start med å spørje klassa om forgje økt. Kva hugsar vi og kva lærte vi. Noko ein hugsar som var viktig? Reperer dei tre omgrepa som vi lærte forgje time, før gjennomgang av to nye omgrep til dagens økt. Algoritme – Ein serie av instruksjonar om korleis vi kan nå eit mål. (Døme Korleis rekne vi saman 25+25 (starte med addere bakarste tal osv.. Rett og slett ei <i>oppskrift</i>) Koding – Gjere om handlingar til eit symbolspråk. Debugging – Finne og fikse feil og problem i koden (I symbolspråket) Funksjon – Ein del med kode som kan brukast om og om igjen.	Gjennomgang for å i vareta prinsipp 12 og 13 så elevane kan halde på med mykje elevaktivitet og få overgangar. Her vil det og bli ivaretatt 14 sida ein treng kun papir/tavle og koppar til oppgåvene. Ordforråd for å hjelpe til å forstå og i vareta prinsipp 1, 2, 3, 6 og 11 seinare i opplegg der elevane bør kunne grunnleggjande omgrep for å	Her vil eg ha fokus på om det er lett for elevane å forstå kva som er planen for økta. Og om det er lett for lærar å gjennomgå omgrepa med elevane. Dette vil eg finne ut i samtale med lærar etter økt Eg og sjå i eigen observasjon om elevane vert engasjerte/ivrige når lærar forklarar kva som skal gjerast

			<p>Parameter – Ekstra informasjon som kan leggast til <i>Funksjonen</i>, for å tilpasse den.</p> <p>Forklare at no skal vi jobbe vidare med oppgåve arket frå forgje økt.</p>	utforske og spør kva som skjer i koden.	
5 min	Inndeling	Inndeling i grupper med ikkje permanente tavler	Dele opp i tilfeldige grupper og fordele dei på arbeidsstasjonar.	For å i vareta prinsipp 4 og 5 (Desse går igjen resten av økta).	Vil tilfeldige valte grupper og ikkje permanente arbeidsflater bidra til meir engasjement i gruppene. Dette vil eg sjå på med hjelp av eigen observasjon og lærarintervju
25 min	Utforsking og utprøving av koding	Utforsking og prøving av ei ferdig gitt kode	<p>Her får ei utdelt ein ferdig kode og elevane skal prøve å tenke/diskutere seg fram til korleis tårnet vil sjå ut.</p> <p>Etter nokre minutt når elevane har tankar om korleis tårnet vil sjå ut kan lærar vise på framme på kateter med til dømes ein elev.</p> <p>No skal dei prøve sjølv med fleire koppa. Del ut oppgåvearket + symbolark. (vedlegg 8 og 6).</p> <p>Elevane får jobbe med oppgåve A til K.</p>	<p>For å nå prinsipp 1, 3, 6, 8 og 10. Der elevane skal få tenke, utforske og utveksle.</p> <p>For å nå prinsipp 7, 8, 9 og 11 der elevane får lage, prøve og samarbeide i arbeid med kode.</p>	Her vil eg fokusere på utforskartrøng og skaparglede i hovudsak ved bruk av «Social emotional support» og «Students cognitive engagement».
5 min	Oppsummering	Ei gruppe kan vise sin kode	Høyr med elevane om dei fekk det til oppgåvene. Var det lettare å programmere når ein kunne forkorte med funksjonar? Forstår vi nytten til å ha funksjonar når ein skal kode noko kjempestort til dømes eit dataspel?	For å nå prinsipp 3 og 7 . Her vil eleva utveksla og køyrt koden sin.	Sjå på mål om skaparglede og engasjement. OECD sitt skjerma via «Students cognitive engagement», eigen observasjon om det ser ut som elevane ønskje å jobbe meir med dette og intervju med lærar.

TABELL 4: UNDERVISNINGSSOPPLEGG 2 I KVA, KORLEIS, KVIFOR SKJEMA.

Etter andre undervisningsøkt satt vi igjen med mykje av dei same positive inntrykka som frå første undervisningsøkt. Programmering er noko som tar tid, og elevane kjem sjeldan så langt som ein ønsker. Slik er det gjerne i starten når mykje er nytt. Fem elevar i kvar gruppe vart litt i meste laget då vi såg at sterke elevar tok mykje av styringa frå kanskje nokre av dei svakare. Eit anna dilemma som dukka opp var tidsbruken på oppgåvene. Nokon var raskt ferdige,

medan andre trengte lenger tid. Dermed vart det litt ventetid /“dødtid” for enkelte. Her kunne ein kanskje ha anna opplegg/oppgåver medan dei venta på å få programmere.

3.5 Analyse av data

I etterkant av datainnsamlinga måtte all informasjonen analyserast. I dette prosjektet har det blitt samla inn data i form av observasjonar i undervisningssituasjon, samt intervju etter kvar undervisningsøkt. Det har blitt samla inn ein god del tekst, noko som gjorde det naudsynt med eit analysearbeid for å redusere datamengda. Eit slikt analysearbeid gjer ein ved først å fjerne dei delane av datamaterialet som ikkje er relevant for den aktuelle problemstillinga. Vidare må data som har verdi systematiserast og analyserast (Larsen, 2017).

Det var interessant å sjå på i kva grad undervisningsopplegget såg ut til å fungere, men like interessant var det å prøve å finne ut kvifor det fungerte eller kvifor det ikkje fungerte. Her var eg altså ute etter detaljane ved opplegget. Eg ønska ikkje berre å finne ut kva som var bra og ikkje bra, men og peike på deler som eventuelt såg ut til å støtte opp rundt måla eller ikkje. Med dette som fokus hadde eg behov for ei god oversikt over all data eg hadde, og valde å plassere dette i ein tabell (sjå tabell 5) som tok utgangspunkt i undervisningsopplegget 1 og 2 (tabell 3 og 4). Her gjekk eg djupare inn i detaljane på kvar del av undervisningsopplegget. Eg såg etter kva eg og lærar var ute etter i den spesifikke delen av opplegget, kva erfaringar lærar gjorde seg, og kva eg som forskar observerte. Vidare såg eg på kva del av undervisningsopplegget som resulterte i kva, og brukte dette til å analysere vidare. Eg valde å nytte ein tabell fordi det er eit effektivt verktøy for å organisere og analysere informasjon. Det hjalp meg til å identifisere ulike resultat og samanlikne datagrunnlaget mitt. Det gjorde det og enklare for meg å identifisere spesifikke element med undervisningsopplegget som ein kanskje elles lett hadde oversett.

For å systematisere datainnsamlinga mi valde eg å dele intervju og observasjon frå kvarandre, og sjå på måla mine i lys av desse. Eg tok utgangspunkt i forskingsspørsmåla i mi analyse av intervju. Dette var intervju som vart gjennomført i etterkant av kvart undervisningsopplegg. Intervjua tok om lag 30 minutt, der vi gjekk gjennom dei ulike delane i opplegget og kopla det opp mot forskingsspørsmåla. Eg skreiv ned stikkord undervegs i intervjuet, og lagde utfyllande notat i etterkant medan det enno var friskt i minnet. Eg noterte ned kva som var relevant for kvart forskingsspørsmål, før eg vidare systematiserte datamaterialet ved å fylle

det inn i skjema med oversikt over dei ulike delane i økta. (tabell 5). Då eg og lærar hadde eit tett samarbeid, vart og noko av datamaterialet samla inn utanfor dei spesifikke intervju.

For observasjonen sin del valde eg på førehand ut nokre spørsmål til dei forskjellige delane av undervisningsopplegget (Sjå tabell 2). For å måle skaparglede og utforskartrøng tok eg utgangspunkt i «Student cognitive engagement» (Sjå figur 10). Her ville eg sjå om elevane vart stimulert og utfordra av innhaldet i undervisningsopplegget. Opplegget skulle bidra til at eleven fekk byggje ei djup konseptuell forståing for å hjelpe med å fremje skaparglede og utforskartrøng. Det ville vere i overkant å nytte alle delane av observasjonsskjemaet, så eg valde meg ut nokre spørsmål som eg såg på som enklast å måle og som var relevant for å analysere undervisningsopplegget. Følgjande spørsmål (omsett til norsk) var:

1. Elevane vart kognitivt utfordra og intellektuelt stimulert
2. Læraren fekk elevane til å forklare eller rettferdiggjere idear eller tilnærmingar
3. Elevane hadde høve til å reflektere rundt sine meiningar
4. Elevane hadde høve til å oppsumere eller samanfatte si eiga læring

For å måle engasjement tok eg utgangspunkt i observasjonsskjema «Social-Emotional Support». Her ville eg sjå om elevane fekk høve til å utføre arbeid, uttrykke idear og tankar som avslørar utfordringar og misoppfatningar. I undervisningsøktene nytta eg observasjonsskjema (Figur 8) for å sjå om ein nådde måla med å fremje engasjement. Her valde eg meg ut seks spørsmål som kunne hjelpe meg til å svare på dette. Desse spørsmåla (omsett til norsk) var:

1. Behandlar læraren alle elevar med respekt?
2. Viser elevane respekt for kvarandre?
3. Er atmosfæren i klasserommet varm?
4. Er lærar oppmuntrande og open?
5. Er elevane villige til å dele idear og tankar?
6. Er elevane villige til å spørje om hjelp eller rettleiing?

Eg valde å bruke desse skjema om kvarande under undervisningsopplegget, og noterte ned flittig under kvar del av opplegget. Så snart timen var over, satte eg meg ned, tolka og overførte datamaterialet til oversiktsskjema (tabell 5). Her gjentok eg prosessen på nytt etter undervisningsøkt 2.

Ei tredje datakjelde som dukka opp utanfor sjølve undervisningsopplegget var informasjon frå elevar. Som lærarvikar ved skulen fekk eg og tileigna meg informasjon utanfor klasserommet, i form av samtalar med elevar i undervisning og i friminutt (rundt prosjektet analog programmering). Fleire av elevane kom bort til meg og spurte om analog programmering. Nokre av spørsmåla som dukka opp var til dømes: «Når skal vi fortsette med analog programmering?» og «Kan vi ha det oftare?». Dette gjev meg eit klart innblikk i elevane si meining rundt tema og viser heilt klart eit teikn på engasjement og utforskartrøng. Det at elevane sjølv i friminutta bruker tida si på å snakke med meg om analog programmering viser at vi har “trefte” med undervisningsopplegget vårt og at eg og lærar har klart å nå dei måla vi sett oss før utarbeidinga av opplegget. Dette var noko eg ikkje hadde sett føre meg under planlegginga av datainnsamlinga mi og denne informasjonen tok eg med meg inn i samlinga som ei ekstra datakjelde.

I etterkant av kvar undervisningsøkt, når data frå observasjon og intervju var systematisert, satt eg og lærar oss ned og såg over skjema (tabell 5). Dette vart gjort for å nytte datakjeldene, for å få mest mogleg ut av dei. Dei eksisterte ikkje berre side om side, men var og med på å påverke og informere kvarandre. Dette hjalp meg til å gjere observasjonar som eg kanskje ikkje ville gjort utan kjeldene.

3.6 Reliabilitet og validitet

Sidan eg i mitt prosjekt har arbeidd med kvalitativ forskning var det viktig for meg å vere bevist på styrkar og veikskapar i mine val av metodar. Ein må og forstå at menneske er ulike og difor vil forskarar vektleggje ulike sider ved dei same fenomena (Postholm & Jacobsen, 2018). På bakgrunn av dette ser eg det naudsynt å greie ut forskinga mi sin reliabilitet og validitet. Reliabiliteten handlar om kor pålitelege den data eg har samla inn er, samt kor godt eg har gjort greie for prosessen i oppgåva mi. Validiteten tar føre seg kvaliteten på sjølve forskinga.

For å styrke reliabiliteten i dette masterprosjektet har det blitt vektlagt å gje detaljerte skildringar av forskingslaupet og det analytiske rammeverket. Dette er eit naudsynt grep fordi mi forståing vil ha innverknad på korleis datamaterialet vert tolka. I kvalitativ forskning er det naudsynt at lesaren får eit godt innblikk i datamaterialet og metodane som har blitt nytta, nettopp for å styrke pålitelegheita til resultata (Johannessen et al., 2021). Det vil altså vere viktig for meg å gjere greie for prosessen i oppgåva mi av fleire grunnar. For det første gjev

det lesaren ei betre forståing av korleis forskinga vart utført, kva og kven som var involvert i forskingsprosessen. Det andre vil vere at det gjev eit høve for andre å vurdere reliabiliteten og validiteten til studien, samt å kunne reprodusere resultata om ønskeleg. I etterkant har det å gjere greie for prosessen i oppgåva bidratt til å styrke mi eiga forståing av forskingsprosessen og bidratt til å utvikle mi eiga kritiske tenking, problemløysing og refleksjonsevne i arbeidet med denne masteroppgåva.

I mitt prosjektet valde eg å bruke metodetriangulering (Johannessen et al., 2021).

Metodetriangulering bidreg til å styrke den indre validiteten ved at dei ulike metodane skildrar verkelegheita frå fleire ulike vinklar, og dermed skapar eit meir heilskapleg bilete av ei kompleks og samansett verkelegheit. Postholm & Jacobsen (2018) skriv at det kan vere ein veikskap å kombinere ulike metodar når ein arbeider innanfor ein avgrensa tidsperiode, men i mitt prosjekt, der DBR har blitt brukt som design, opplevde eg at intervju og observasjonar var naudsynt for å kunne innhente data frå alle delane i forskingslaupet. Eg har tidlegare, i kapittel 3.3, skildra korleis eg utnytta kombinasjonen av observasjon og intervju for å skape ei heilheitleg forståing. I prosjektet er observasjonane og intervju nytta om kvarandre, både for å støtte opp om eller vere ei motvekt til dei erfaringane læraren la fram, samtidig som intervju vart nytta til å utdjupe det eg som forskar observerte.

Masteroppgåva mi inneheld berre ein lærar og hen si klasse sine erfaringar rundt bruken av analog programmering. Grunna studien sitt lave utval kan eg ikkje argumentere for at resultata vil vere representative for alle lærarar i Norge. På den andre sida skriv Postholm og Jacobsen (2018) at generalisering i kvalitative studiar er knytt til kor lett ein kan kjenne seg att i skildringane, såkalla naturalistisk generalisering. Det handla om kor lett lesaren kan overføre det til sin situasjon. Det er grunn til å tru at andre lærarar gjennom å lese om erfaringane gjort i denne oppgåva vil kunne overføre det til sin eigen situasjon, og på den måten kunne vere gjeldande for fleire lærarar. I denne oppgåva har detaljerte skildringar av forskingsprosessen vore eit viktig grep for å styrke validiteten, slik at lesaren sjølv kan vurdere kor overførbar dette er til andre kontekstar (Johannessen et al., 2021)

Dette masterprosjektet tar for seg forskingsdesignet DBR. Dette designet skil seg ut frå andre forskingsdesign ved å leggje stor vekt på samarbeid mellom forskar og deltakar. Relasjonen mellom forskar og deltakar kan påverke forskingsprosessen og kva resultat som kjem fram, og dermed er det naudsynt å diskutere denne relasjonen og dei utfordringane som kan oppstå. Ein av fordelane med å ha eit tett samarbeidsforhold mellom forskar og deltakar er at valideringa av prosjektet styrkast. Ved å få kontinuerlege tilbakemeldingar frå deltakar på korleis ulike

situasjonar blir forstått frå hans perspektiv, kan gje meg som forskar ei større grad av kontrollsjekk av tolkingar og sikre at forskingsprosess og resultat er gyldige og pålitelege. Dette bidreg til å styrke forskingsdesignet og auke kvaliteten på resultata som kjem fram. På den andre sida kan samarbeidet mellom forskar og deltakar skape utfordringar. Forskaren sin teoribakgrunn, erfaringar, haldningar og meiningar kan påverke deltakaren og dermed og påverke resultata. Difor er det viktig at forskaren er bevisst på å ikkje påverke deltakaren for mykje, sjølv om DBR opnar for samarbeid. Eg som forskar har difor vore open for deltakaren sitt perspektiv og hen sine idear, samtidig som eg har vore bevist på å oppretthalde ei nøytral rolle i samarbeidet.

I sum kan ein seie at relasjonen mellom forskar og deltakar i DBR-forsking er ein viktig faktor som kan påverke forskingsprosessen og resultata. Sjølv om samarbeidet mellom forskar og deltakar kan skape utfordringar, kan det og bidra til å styrke forskingsdesignet og auke kvaliteten på resultata. Difor var det viktig at eg som forskar var bevisst på eiga rolle og min relasjon til deltakar, og jobba aktivt for å oppretthalde ein nøytral og open tilnærming. Avslutningsvis har mitt prosjekt hatt ein forskar-informant-relasjon. Ein av utfordringane ved kvalitativ forskning er at eg som forskar samlar, analyserer og tolkar datagrunnlaget. Resultatet er difor prega av mine verdiar og haldningar. Gjennom å ha ei bevisst haldning til dette, så har eg forsøkt å legge vekt på det teoretiske rammeverket, og haldt mine forståingar i bakgrunnen. Samtidig opner forskingsforløpet opp for at stemma mi ikkje skal bli heilt borte, noko som igjen skapte ein utfordrande balansegang i forskarrolla.

3.7 Etikk

Allereie i planlegginga starta dei etiske betraktningane for masterprosjektet. Tidleg i prosessen vart det sendt søknad om godkjenning av prosjektet til NSD (Norsk senter for forskningsdata) (vedlegg 4). I denne prosessen utarbeidde eg eit informasjonsskriv og ei samtykkeerklæring (vedlegg 5) ut frå malen på NSD si heimeside. Denne vart etterpå sendt til min informant for underskrift. Lærer informerte elevar og føresette i den aktuelle klassen om prosjektet. Det var ikkje naudsynt å innhente samtykke frå elevane eller føresette sidan prosjektet for elevane sin del inngjekk i ordinær undervising.

Frivilligheit og anonymitet er to viktige etiske prinsipp som eg måtte rette meg etter i dette masterprosjektet. Før prosjektet starta fekk læraren som deltok informasjon om sin rett til å trekke seg ut av prosjektet. For å halde informant anonym vart alle personopplysingar

anonymisert, og det er difor ikkje mogleg å spore informant. Alt av data vart oppbevart passordbeskytta i OneDrive, og vert sletta når prosjektet er ferdig.

Gjennom det tette samarbeidet med min informant har eg sørga for å ivareta informanten på best mogleg måte. Eg har vore fleksibel med tanke på tidspunkt for intervju og planlegging, slik at det skulle passe inn i informanten sin arbeidskvardag. For å behandle data vitskapeleg og reieleg har eg sørga for å ikkje fordreie eller villeie resultatata. Samstundes har det vore viktig å ikkje utlevere informanten for mykje, eller stille fram resultatata på ein måte som kan sette informanten i dårleg lys

4 Presentasjon av resultat

Eg vil i dette kapittelet presentere resultatet i for studien min. Eg vil starte med å presentere ei oversikt over resultat i form av ein tabell (tabell 5). Denne tabellen vil eg nytte til å gå meir ned på detaljnivå og freiste svare på forskingsspørsmåla mine seinare i oppgåva. Før eg presenterer resultata vil eg og raskt finne ut i kva grad undervisningsøktene vart gjennomført som planlagt. Her vil eg sjå på kva for nokre moment som ikkje gjekk som planlagt, kvifor, samt kva som vart endra frå økt 1 til 2. Her vil eg sjå på dei fire elementa skaparglede, engasjement, utforskartrong, samt attraktivitet for lærar, og trekkje inn datakjeldene mine for å svare på forskingsspørsmåla. Til slutt vil eg samle alt i ei kort oppsummering av kapittelet.

4.1 Oversikt

For å skaffe meg ei oversikt tok eg utgangspunkt i undervisningsopplegget frå tabell 3 og 4 og forenkla det ned i fleire små, presise delar av undervisningøkta. Her viste eg konkret kva som vart gjort, kva eg såg etter, og kva eg fann ut med forankring i observasjon og samtale med lærar. For å gjere det enklare å halde oversikt over kva eg presenterer i tabellen har eg valt å nummerere det frå 1 til 8. Eg har valt å utelate målet om at det skal vere attraktivt for lærar, då dette er noko som har gått igjen i heile opplegget og vil bli grundigare forklart i delkapittel 4.4.

Resultat							
Kva:	Kva prinsipp vi var ute etter	Kva såg vi etter	Kva fant vi	Kva endra vi til neste økt?	Kva effekt gjorde endringa	Kor sikker er eg i resultata mine	Kva ville vi endra til ei eventuell økt 3?
Del 1 Starta opp timen med å forklare prosjektet		Verka presentasjon av prosjektet noko som kunne fremje engasjement hos elevane?	Observasjon: Det såg ut som elevane tok tema og målet med økta raskt Samtale: Lærar følte elevane viste stor interesse og verka spente på å sette i gang med prosjektet.	Ingen endring	Ingen endring	Her vil eg med stor sikkerheit meine at elevane vart engasjert	Ingen endring
Del 2 Samtale om kva programmering er	1 – Utforske kode 3 – Utveksle idear	Kunne ein samtale om programmering hjelpe elevane til å fremje utforskartrong og engasjement ?	Observasjon: Mange av elevane viste stort engasjement for prosjektet og verka ivrige i å sette i gang	Ingen endring	Ingen endring	Kan med nokon grad av sikkerheit meine at elevane var engasjerte og viste teikn på utforskartrong	Ingen endring

			<p>Samtale: Lærer synes han fekk ein god dialog med elevane når dei snakka om programmering</p>				
<p>Del 3 Forklarte omgrep og viste eit døme</p>	<p>1 – Utforske kode</p> <p>2 – Prøve å feile</p> <p>11 – Lage eigen kode</p>	<p>Kunne ein samtale om omgrep i programmering bidra til skaparglede og utforskartrong?</p>	<p>Observasjon: Nokre av elevane mista litt fokus når ukjente omgrep blei involvert</p> <p>Samtale: Lærer følte omgrepa var naudsynt å lære, men kunne kanskje presenterast på ein «artigare» måte?</p>	<p>Repetisjon + to nye omgrep</p> <p>Bruke omgrepa på ein betre måte som ville gjere det lettare for elevane å forstå og bruke sjølv.</p>	<p>Observasjon: Elevane lærte omgrepa på ein meir konkret måte der alle blei godt brukt og forklart i døme</p> <p>Samtale: Lærer synes elevane forstod omgrepa betre i økt 2, men usikker på om det bidrog til skaparglede eller utforskartrong</p>	<p>Her kan eg ikkje med noko grad av sikkerheit meine at omgrepsforståing bidrog til skaparglede og eller utforskartrong.</p>	<p>Her kunne vi til dømes ha intrigert omgrepa i form av spørsmål/oppgåver for å hjelpe elevane med å fremje utforskartrong.</p>
<p>Del 4 Fordeling av grupper på arbeidsstasjon</p>	<p>4 – Tilfeldige grupper</p> <p>5 – Ikkje permanente arbeidstavle</p>	<p>Kunne tilfeldige valte grupper og ikkje permanente arbeidsflater bidra til meir engasjement i gruppene?</p>	<p>Observasjon: Tilfeldig valte grupper såg ikkje ut til å ha noko effekt på elevgruppa, men å jobbe på felles arbeidstavle såg ut til å bidra positivt på engasjementet i gruppa</p> <p>Samtale: Lærer synes elevane viste mykje engasjement når dei samarbeida på arbeidstavlene</p>	<p>Valte å gjere gruppe-inndeling tilfeldig, men med visse unntak for å unngå nokre elevgrupper</p>	<p>Observasjon: Såg ut til å hjelpe nokre av gruppene til å jobbe meir effektivt, men såg ikkje ut til å fremje engasjement</p> <p>Samtale: Lærer meinte nokre av elevane fekk meir engasjement sida dei slapp å jobbe med spesifikke elevar</p>	<p>Her kan eg ikkje med noko grad av sikkerheit meine at tilfeldige arbeidsgrupper hadde noko effekt på elevgruppa.</p> <p>Eg kan med stor grad av sikkerheit meine at det å jobbe på ikkje permanente arbeidstavle bidrog til å fremje engasjement.</p>	<p>Her kunne vi prøvd å korta ned til grupper på tre og tre og sett kva effekt det har på engasjementet.</p>
<p>Del 5 Utforsking og prøving av ei ferdig gitt kode</p>	<p>1 – Feilsøke kode</p> <p>6 – Lage hypotese</p> <p>8 – Gje elevane tenkande oppgåver</p> <p>9 – Sjå føre seg</p> <p>10 – Utforske kode</p>	<p>Kunne arbeidet med å sjå på ei ferdig laga kode og samarbeide rundt dette vere med på å fremje utforskartrong hos elevane?</p>	<p>Observasjon: Elevane var nysgjerrige på den gitte koden og viste utforskartrong når dei samarbeida for å løyse koden.</p> <p>Samtale: Lærer synes det var ein smart måte å starte med å la elevane få løyse ein gitt kode frå starten av og han synes det verka veldig positivt på elevane i</p>	<p>Ingen endring</p>	<p>Ingen endring</p>	<p>Her kan eg med stor sikkerheit meine at elevane viste stor grad av utforskartrong ved å arbeide med ei ferdig gitt kode.</p>	<p>Ingen endring</p>

			form av utforsking, prøving og feiling.				
Del 6 Presentasjon av den gitte koden	3 – Utveksle idar 6 – Lage hypotese	Kunne det å presentere ideen sin bidra til å fremje engasjement ?	Observasjon: Såg at mange elevar verka lite interessert i å presentere for andre. Samtale: Lærer meinte at fleire i klassen ofte er redd for å gjere feil og at det kunne bli utslagsgjevande for denne delen.	Fjerna denne delen	Auka fokus på samarbeid i gruppa.	Kan med stor sikkerheit meine at det å presentere resultatet av den gitte koda hadde ingen effekt på engasjement. Heller tvert i mot der mange elevar viste lite interesse for å presentere for klassen.	Ingen endring
Del 7 Jobbe med oppgåve	3 – Utveksle idar 7 – Kode på eigenhand 8 – Gje elevane tenkande oppgåver 10 – Utforske kode 11 – Lage eigen kode	Kunne det å jobbe i grupper med å programmere analoge oppgåver bidra til å fremje skaparglede og utforskartrong ?	Observasjon: Stor elevaktivitet og iver i å jobbe med oppgåvene. Bygge tårn saman ved hjelp av programmering såg ut til å ha god effekt Samtale: Lærer var imponert over engasjementet til elevane og synes det var veldig kjekt å sjå eit slik iver etter å jobbe med matematikkoppgåver samanlikna med til dømes oppgåver med multiplikasjon.	Raskare i gang med oppgåve løysing	Tidssparande for lærar og meir effektivt for elevane å kome i gang med sjølv læringa med analog programmering.	Kan med nokso stor sikkerheit meine at det å jobbe i grupper med å programmere analoge oppgåver har effekt på utforskartrong der fleire av elevane ofte vill prøve vanskelegare og vanskelegare oppgåver og nokre ville til og med lage egne oppgåver for andre. Det å lage oppgåver for medelevar kan vise til skaparglede, men sida det kun var eit fåtal kan vi ikkje med sikkerheit meine at det å jobbe med oppgåver bidrog til skaparglede.	Utforde elevane enda meir med tenkande oppgåver. Mange av oppgåvene kunne til slutt bli litt repeterande og om en skulle jobbe vidare med analog programmering bør det absolutt vere eit punkt ein må ta omsyn til.
Del 8 Oppsummering		Viste engasjement for å jobbe meir med analog programmering?	Observasjon: Mange av elevane var delaktig i samtalen når lærar oppsummera timen og viste interesse til å jobbe meir med tema Samtale: Lærer synes det var mange som deltok aktivt i oppsummeringa, men dei «vanlege» meldte seg ut	Ingen endring	Ingen endring	Eg kan med noko grad av sikkerheit meine at elevane viste engasjement for å jobbe meir med analog programmering.	Finne ein måte å involvere enda fleir av elevane i oppsummeringa.

TABELL 5: OVERSIKT OVER DATAMATERIALE MED UTGANGSPUNKT I UNDERVISNINGSSOPPLEGG

4.2 Gjennomføring av intervensjonen

I denne delen vil eg kort gjere greie for den faktiske gjennomføringa av intervensjonen av undervisningsopplegget opp mot den planlagde gjennomføringa, ut i frå strukturen i undervisningsopplegg 1 og 2 (Tabell 3 og 4). Som det vil bli avklart, og som eg og lærar hadde forventa, var gjennomføring av intervensjonen trufast mot planen, sjølv om det vart gjort noko naudsynte justeringar. Den hadde med andre ord ganske høg nøyaktigheit. Gjennomføringa varte i omlag 45 minutt, og fann stad på tilsvarande identisk måte.

Del 1 til 3 (Sjå tabell 5) der lærar gjekk gjennom starten på prosjektet, forklarte programmeringstema, samt omgrep, gjekk det meste som planlagt. Her forventa eg og lærar at det ville kome nokre spørsmål som lærar kanskje kunne bruke til å vinkle seg inn mot det som var ønska i samtalen med elevane. Mot slutten av delen, når lærar viste fram symbol-ark (vedlegg x) for elevane og forklarte korleis ein brukte det, valde lærar å vise eit eksempel for å konkretisere. Dette var ikkje i den opprinnelege planen for undervisningsøkta, men var til god hjelp for elevane då dei skulle gå i gang med del 5, der dei arbeidde med ein ferdig gitt kode. Når vi kom til del 6 der elevane sjølv skulle få presentere svaret på den gitte koda var det ingen av elevane som ønska å presentere svaret sitt for resten av klassa. Lærar valde då å droppe presentasjonen, og gjekk heller raskt rundt til gruppene og let dei vise hen kva dei hadde kome fram til. Etter dette skulle elevane få arbeide med oppgåver knytt til analog programmering. Her gjekk oppgåvene som forventa. Elevane jobba i ulikt tempo, men såg ut til å klare fleire av oppgåvene. Til slutt var det oppsummeringa som stod på planen, og denne delen gjekk og som planlagt. Til undervisningsøkt 2 gjorde vi oss nokon endringar i opplegget, og med erfaringar frå økt 1 så gjekk det meste som planlagt i denne delen. Her brukte elevane litt meir tid på oppgåveløysinga, og det vart kortare tid til oppsummering.

4.3 Svar på forskingsspørsmål 1

For å svare på det første forskingsspørsmålet («*I kva for ein grad og korleis kan analog programmering leggje til rette for 6. klasse elevars skaparglede, engasjement og utforskartrøng?*») nytta eg, som vist i tabell 5, data frå både intervju og observasjon. Eg vil no presentere datamaterialet først frå lærar sin ståstad ut i frå intervju og deretter mine eigne observasjonar.

4.3.1 Oppsummering frå intervju med lærar

I del 1 (frå tabell 5) følte lærar at elevane viste stor interesse for programmeringsprosjektet og verka spente på å sette i gang. Lærar var og tydeleg på at elevane likte godt å jobbe med programmering, men var spent på korleis dei ville takle å jobbe med programmering på ein annleis måte enn dei vanlegvis gjorde.

I del 2 der lærar førte samtale med elevane om kva programmering er, synes lærar at han fekk til ein bra dialog med mange av elevane. Mange delte idear og tankar om kva ein robot er og korleis det kan bli styrt og lært opp. Undervegs i samtalen dukka det opp eit spørsmål rundt kunstig intelligens, noko som elevane synes var spennande. Her meinte lærar at det viste at elevane har interesse for teknologi og forskning.

I del 3 der lærar forklarte og snakka om omgrep synes lærar at elevane fekk litt vanskar. Det var fleire nye omgrep som elevane ikkje har vore borti før. Lærar drog ei samanlikning av omgrepet «debugging» mot bugs (insekt) som noko ein ikkje vil ha i huset sitt, og på den måten fekk hen elevane til å danne ei forståing for omgrepet. Andre omgrep som algoritme var meir utfordrande, for her hadde ikkje elevane noko fast å kople det opp mot. Lærar synes at det å jobbe med omgrep er naudsynt i matematikken, men tenkte kanskje at til neste time kunne ein prøve å forklare på ein alternativ måte for å sjå om elevane tok det betre då. I neste økt gjorde vi difor nokre små endringar, og då synes lærar at elevane forstod omgrepa betre. Om dette var grunna endringane, repetisjon eller ein kombinasjon av desse kunne han ikkje meine noko om.

I del 4 skulle elevane delast i grupper og ikkje-permanente tavler vidare i undervisningøkta. Lærar hadde frå før jobba særst lite i grupper med denne klassa og var noko spent på korleis det kunne fungere. Lærar delte inn i tilfeldige grupper og gjorde seg ikkje nokon direkte tankar om dette i ettertid. Det lærar sjølv synes var interessant, var dei ikkje-permanente tavlene. Her synes lærar at fleire av elevane viste mykje engasjement og samarbeidsvilje. I økt 2 valde vi å endre litt på gruppeinndelinga for å unngå klinsj mellom nokon av elevane som etter lærar sitt syn ikkje burde vere på gruppe saman. Her meinte lærar at det fungerte betre enn i økt 1, men ikkje noko revolusjonerande.

I del 5 fekk elevane utlevert ei ferdig gitt kode på arbeidstavla si som dei skulle utforske. Dette synes lærar var ein artig og interessant måte å jobbe på. Av eiga erfaring i klasserommet synes hen det verka positivt på elevane at dei fekk lov til å samarbeide og tenke seg fram til ei løysing i fellesskap.

I del 6 skulle elevane presentere ideen eller løysinga si frå del 5. Lærar hadde nemnt at det å presentere for resten av klassa er ei utfordring han har jobba ein del med. Fleire i klassa er redd for å gjere feil, så det er veldig vanskeleg å få nokon i til å presentere. Det enda difor opp med at lærar sjølv presenterte den gitte koden. Vi valde å fjerne denne delen til økt 2. Ein tanke eg og lærar gjorde oss då vi snakka saman om del 5 og 6 var om det kanskje kunne vore mogleg å flytte denne delen fram i opplegget. Vi vurderte om det kunne vere lurt å utlevere koden til elevane utan noko informasjon, for å så å vise kvifor ein treng eit programmeringsspråk og ein måte å forstå språket på. Etter det kunne dei på nytt få sjå på koden og prøve å løyse den.

I del 7 der elevane jobba med oppgåver knytt til analog programmering var lærar imponert over engasjementet til elevane og synes det var kjekt å sjå at så mange bidrog og viste interesse for å jobbe med matematiske oppgåver. Lærar samanlikna det med til dømes å jobbe med oppgåver knytt til multiplikasjon, der elevane ikkje viser i nærleiken den same interessa. Lærar var ikkje sikker på om det var måten dei jobba med oppgåver, på eller om det var tema i seg sjølv som bidrog til interessa, men at denne delen var positiv var lærar sikker på.

Avslutningsvis i del 8 der lærar oppsummerte timen synes lærar det var god deltaking blant elevane. Lærar meinte aktiviteten var over gjennomsnittet, men mange av dei elevane som vanleg trekk seg vekk og venter på friminuttet gjorde det i denne timen og. Vi valde å ikkje gjere nokon endring til økt 2, men i ei eventuell økt 3 synes lærar det kunne vere interessant å prøve på ei oppsummering som involverte enda fleire.

4.3.2 Oppsummering av observasjonar gjort under intervensjonen

I del 1 starta lærar opp med å forklare prosjektet. Lærar skapte ei varm atmosfære i klasserommet og behandla alle spørsmål og svar frå elevar på ein fin måte, sjølv om ikkje alle var like relevante. Elevane såg ut til å vere særst interessert i emnet, og ut frå eigen observasjon såg det ut som elevane gledde seg til å sette i gang med opplegget.

I del 2 då lærar starta samtalen med spørsmål om kva vi legg i omgrepet programmering, var fleire av elevane villige til å dele idear og tankar. Lærar var oppmuntrande og open for tankane til elevane og bruker mykje av det elevane seie for å vinkle seg inn mot det han ønskjer å forklare. Undervegs i samtalen kunne eg sjå at mange av elevane verka enda meir interessert i opplegget etter å ha prata litt rundt tema. I denne delen kunne eg og sjå at nokre elevar vart kognitivt utfordra. Lærar stilte ofte spørsmål og let elevane vere med på diskusjon

rundt emnet dei skulle starte opp med. Her viste fleire av elevane engasjement og interesse rundt det nye tema analog programmering.

I del 3 fekk elevane høve til å reflektere rundt egne meiningar då lærar prata om omgrep. Lærar let elevane få tenkje seg fram til kva dei ulike omgrepa tydde og korleis dei kunne nyttast vidare. Her var det likevel ein del av elevane som datt litt ut, særleg då nokre av omgrepa kanskje vart noko vanskelege i høve til å resonnerer seg fram til eit mogleg svar. Eit døme på dette var då elevane skulle forstå «algoritme». Det såg ut til at fleire av elevane ikkje klarte å danne seg ei forståing av kva det betydde. Til neste time endra vi difor litt korleis lærar presenterte omgrepa, og i lag med repetisjonen frå forrige økt, såg det ut til at resultatet betra seg for fleire av elevane. Likevel er vi ikkje sikre på om det å jobbe med omgrep er med på å fremje skaparglede og utforskartrong.

I del 4 såg eg på fordeling av grupper på arbeidsstasjonar med bruk av ikkje-permanente arbeidstavler. Her såg eg at tilfeldige valde elevgrupper ikkje hadde særleg effekt på engasjementet i gruppa. Ein kunne sjå at nokre elevar vart positive då dei fekk vite kva gruppe dei hamna i, medan andre såg litt meir skuffa ut. Det kan ein gjerne trekke mot at alle elevane kanskje ikkje viser like stor respekt for kvarandre, og at nokre av elevane veit kven som «bråkar» eller syner lite arbeidslyst. På den andre sida synes eg mange av elevane fekk seg eit løft då dei kom i gang med å arbeide med ikkje-permanent tavle. Det å kunne teikne på ei større tavle og vise medelevar idear og tankar viste seg å ha stor effekt på engasjementet. I økt 2 prøvde vi å gjere nokre justeringar på gruppeinndelinga. Dette for at fleire av elevane skulle sleppe unna dei elevane som ikkje viste same interesse for opplegget. Sjølv om det betra seg kan eg ikkje med tryggleik seie at tilfeldige valde arbeidsgrupper kontra individuelt arbeid hadde noko effekt på engasjementet i elevgruppa.

I del 5 skulle elevane arbeide med utforsking og utprøving av ei ferdig gitt kode. Her verka mange av elevane nysgjerrige på koden og korleis resultatet ville sjå ut. Elevane verka kognitivt utfordra og fekk høve til å forklare eller rettferdiggjere ideane sine for gruppa si. Her synes eg elevane aktivt prøvde å forklare, lære eller hjelpe kvarandre med å forstå korleis resultatet ville sjå ut. Ut i frå dette kan eg meine at denne delen av opplegget fremja elevane sin utforskartrong på ein god måte.

I del 6 var det elevane sin tur til å presentere løysinga si. Her var det mange elevar som trakk seg vekk og få som var interesserte i å presentere for dei andre gruppene. Elevane viste lita vilje til å dele idear og tankar, og klasseroms-atmosfæren var på dette tidspunktet kaldare enn

tidlegare i timen. Dette stadfestar det lærar forklarte i førre kapittel, at mange av elevane i klassa er redd for å gjere feil når andre ser på og trekte seg difor tilbake då det kom til det å presentere. Her kan eg med stor tryggleik seie at det å presentere resultat eller idear ikkje bidrog til å fremje engasjement i denne klassa. Grunna dette valde vi å fjerne denne delen frå neste økt.

Då vi kom til del 7, der elevane skulle jobbe med spesifikke oppgåver knytt til analog programmering, viste elevaktiviteten seg å vere svært høg og elevane viste stor iver i å jobbe saman for å løyse oppgåvene. Her fekk elevane kognitiv utfordring og intellektuell stimuli med å måtte forklare problemløysingsmetoden deira for resten av gruppa, samt rettferdiggjere det for kvarandre på ein truverdig måte. Særleg etter å ha klart å løyse første oppgåva såg det ut til at gruppa viste enda meir utforskartrøng. Og dei var ivrige etter å forstå og tenkte seg til korleis neste skulle løysast. Samtidig synes eg det var vanskeleg å sjå direkte skaparglede i oppgåveløysinga. Mot slutten av økt 2 var det nokre elevar som ønska å lage eigne oppgåver til kvarandre, noko som kan syne til skaparglede. Men dette var berre eit fåtal av elevane og eg kan ikkje med tryggleik seie at å jobbe med oppgåvene fremja skaparglede. Men eg kan med tryggleik seie at elevane fremja utforskartrøng i arbeidet med oppgåvene.

Mot slutten av timen, del 8, fekk elevane høve til å summere si eiga læring. Her såg eg at elevane viste engasjement for å arbeide vidare med analog programmering. Mange av elevane var delaktige i samtalen og viste både lærd kunnskap og interesse for vidare arbeid. På den andre sida såg eg og at ein del av elevane retta fokus mot friminutt. Så undervisningsopplegget såg absolutt ut til å vere kjekt for mange av elevane, men kanskje ikkje kjekt nok til å kunne måle seg med friminutt.

Totalt såg arbeidet med analog programmering absolutt ut til å vere innom mange av perspektiva vi ønska å skape i undervisninga, og eg vil difor med ein viss grad av tryggleik seie at arbeidet med analog programmering legg til rette for skaparglede, engasjement og utforskartrøng.

4.3.3 Anna observasjon

Ved sidan av observasjon og intervju fekk eg og samla inn data utanfor undervisningsøktene. Nokre dagar etter gjennomføring av økt 1 var eg vakt i friminuttet utanfor mellomsteget der prosjektklassa haldt til. Allereie i starten av friminuttet kom to elevar bort til meg og spurte om prosjektet. Eine eleven spurte «Når skal vi fortsette med analog programmering?». Dette

såg eg på som eit klart teikn på at eleven viste engasjement i arbeidet med analog programmering. Etter litt prat om kva vi skulle gjere vidare kom det eit par andre elevar frå klassa bort og ville prate om prosjektet. Når fire, fem elevar var interesserte i å prate om analog programmering i friminuttet, kan eg med noko grad av tryggleik hevde at elevane viste engasjement i arbeid med analog programmering. Om det gjaldt heile klassen kan eg ikkje svare på, men det er ikkje utenkeleg at det gjekk att hos fleire av elevane.

4.4 Svar på forskingsspørsmål 2

For å svare på det første forskingsspørsmålet («*I kva grad og korleis kan eit undervisningsopplegg med analog programmering vere attraktivt for lærarar?*») nytta eg, som vist i tabell 5, data frå intervju med lærar. Eg vil no presentere lærars tankar og erfaringar ut i frå dette.

4.4.1 Oppsummering av lærar sine tankar og erfaringar etter intervensjon

I starten av prosjektet hadde eg og lærar ein samtale der vi såg på kva som skulle til for å nå målet om at undervisningsopplegget i programmering skulle vere attraktivt for lærar. Her valde vi tre prinsipp som utgangspunkt i undervisningsopplegget; få overgangar, mykje elevaktivitet og enkle ressursar.

Vi starta med å sjå på korleis opplegget klarte å nå prinsippet om få overgangar. Lærar var interessert i dette grunna ei elevgruppe som lett kunne miste fokus når det fleire beskjedar og mykje som skjedde på same tid. Etter at vi hadde gjennomført økt 1 satt vi oss ned for å sjå på prinsippet vi hadde sett oss. I første økt var det totalt tre overgangar der elevane fekk beskjed å gjere noko nytt. Lærar uttrykte sjølv i intervjuet (etter økt 1) at han følte dette var så optimalt ein kunne få det når det kom til overgangar. Når lærar utførte opplegget følte lærar at han lett fekk med seg elevane, og han følte seg trygg på at få overgangar var ein av årsakene. Med bakgrunn i lærar si tilbakemeldingar og synspunkt valde vi difor ikkje å gjere nokon endringar i dette målet (til neste økt).

Det andre prinsippet vi sett oss var mykje elevaktivitet. For å nå dette prinsippet var vår første tanke at det her måtte gå på engasjement og motivasjon for å få elevane til å arbeide med tema. Her synes lærar spesielt elementa frå Liljedahl med tilfeldig valde grupper og ikkje - permanente arbeidstavler var interessant. Dette var noko som lærar aldri hadde vore borti før,

og Liljedahl sitt tenkande klasserom var noko som var heilt nytt. Etter gjennomføring av økt 1 synes lærar elevaktiviteten var særleg høg. Det var langt meir aktivitet blant elevane enn kva lærar opplevde i ein normal matematikktime, og hen trudde det var fleire årsaker til det. Den eine var at programmering alltid har vore noko elevane synes var kjekt, og aktivitetsnivået er ofte høge i desse timane. Ein anna årsak til den høge aktiviteten kunne vere at det å bruke fysisk analog programmering var noko heilt nytt, og mange elevar var nysgjerrige og interessert i å prøve ut opplegget. Det å kunne arbeide praktisk, til dømes å byggje eit tårn av koppar med hjelp av programmeringsspråk verka kjekt for elevane. Til slutt meinte lærar at måten ein samarbeidde og diskuterte felles i grupper var med på å engasjere elevane. Grunna ei elevgruppe med fleire utfordringar har lærar sjølv gått bort frå gruppearbeid, sidan det sjeldan resulterte i noko positivt for timen. Lærar var likevel usikker på om gruppearbeidet ville fungere over lengre tid. Kanskje ville elevane gå lei dersom dei måtte gjere det same fleire gongar eller utfordringane gjorde at framgangen stoppa opp. Eit anna innspel frå lærar som eg som forskar ikkje hadde tenkt ut på førehand var at målet med få overgangar kan ha vore med på å hjelpe elevaktiviteten. Her synes lærar at det å kunne jobbe samanhengane over lengre tid var med på å auke aktiviteten. Desse tankane og erfaringane tok vi med oss når vi starta planlegging og gjennomføring av økt 2. Etter denne økta var tankane frå lærar mykje likt. Elevane viste stor grad av elevaktivitet og oppretthaldt engasjementet frå førre økt. Likevel synes lærar at nokre av elevane var i ferd med å miste interessa. Dette er noko vi tenkte kunne skape utfordringar for ei eventuell økt 3. Likevel synes lærar at opplegget fungerte minst like bra i økt 2, men at meir variasjon hadde vore nødvendig om ein skulle jobbe vidare med analog programmering.

Det siste prinsippet vi hadde sett oss var det å bruke enkle ressursar. Her la lærar ut om alle utfordringane ein møter på skulen når ein skal ta i bruk digitale einingar. Tidsbruk på elevar som ikkje får logga seg inn, oppstart av maskina, internett som er nede og meir. Alt dette krev tid. For å løyse dette såg eg og lærar det som attraktivt å kunne programmere med enkle ressursar. Målet i seg sjølv er kanskje lett å følge opp med at undervisningsopplegget tok i bruk analog programmering, men vi la i tillegg vekt på å gjere den analoge programmeringa enkel. Etter gjennomført økt var lærar veldig positiv til bruken av analog programmering for å nå målet om enkle ressursar. Det å kunne bruke fysiske ressursar som koppar og ikkje permanente arbeidstavler var noko lærar likte veldig godt. Eit sitat lærar sa var at «Ein kunne starte å jobbe med oppgåver frå første sekund». Her slapp lærar at arbeidstavla måtte kople seg på nett og at koppane til dømes måtte bli slått på. Lærar synes akkurat dette var veldig

frigjerande, då ein slepp dei vanlege uromomenta ein gjerne får frå digitale einingar. Her kunne ein fokusere på det å gjennomføre timen best mogleg. Eit anna element lærar tok fram var at det er mykje lettare for han å følge med på elevane og ser om dei gjer som dei har fått beskjed om. Lærar drog og fram poenget at mange av elevane som har vekse opp med datamaskin heile livet er veldig gode på å bruke den. Mange av elevane kan å bruke datamaskina betre enn lærar sjølv, og det ein del elevar som kan halde på med heilt andre ting utan at lærar får det med seg. Ei utfordring lærar kommenterte var at ein måtte ha tilgjengelege ikkje-permanente tavler, og dette er noko ikkje alle skular har. Lærar såg difor tidleg ei løysing der ein heller kunne bruke ark og blyant. Her gjekk ein altså litt vekk frå Liljedahl sitt prinsipp om ikkje-permanente tavler, men lærar såg på det som eit alternativ som likevel kunne gje i noko grad likt utbytte for elevane. Lærar hadde ikkje fleire innspel etter økt 2. Tankane var dei same, og han synes dette var noko som fungerte særst godt.

Etter at økt 1 og 2 var gjennomført spurte eg lærar kva han generelt synes om opplegget. Her sa lærar at han var svært positiv til prosjektet med analog programmering. Lærar var særleg glad for å kunne jobbe med programmering utan å måtte bruke ferdig spikra nettbaserte undervisningsopplegg der ein sjølv følte seg unyttig, og var glad for å kunne ta emnet ned på eit nivå som gjorde det mogleg for lærar å involvere seg og styre undervisningøkta på eigen måte. Lærar skildra prosjektet som ein særst god inngang til å undervise i programmering, der ein dannar seg ei grunnleggande forståing for emnet før ein så utviklar seg mot den meir tradisjonelle programmeringa på ei datamaskin. Dette var noko lærar såg på som nyttig og noko han ville ta med seg vidare i lærarkvardagen.

4.5 Oppsummering av resultat

I dette kapitlet har resultatet frå analysen blitt presentert. For å oppsummere vil eg trekke fram fire sentrale resultat lærar og forskar gjorde seg i undervisningsperioden. Desse resultatane vil vere grunnlaget for drøftinga i neste kapittel.

1. Ein kan ikkje med tryggleik seie at prinsipp i konteksten av analog programmering fremje skaparglede. Eg såg tendensar i datamateriale, men eg synes det var uklart og vil difor ikkje konkludere om undervisningsopplegget har greidd å fremje målet om skaparglede.

2. Med stor grad av tryggleik kan vi seie at analog programmering kan bidra til å fremje engasjement. Lærar og forskar har ein klar einigheit om at det å undervise analog programmering med bakgrunn i fleire av prinsippa. Til dømes såg ikkje-permanente tavler ut til å fungere veldig bra, medan tilfeldige grupper verka meir usikkert.
3. Vi kan med noko grad av tryggleik seie at analog programmering bidrog til å fremje utforskartrøng. Lærar og forskar var einige i at undervisningsopplegget fremja utforskartrøng hos elevane, men er usikker på i kva grad.
4. Lærar synes analog programmering var ein attraktiv måte å undervise programmering i matematikk. Det å kunne undervise i programmering på ein alternativ måte, utan digitale hjelpemidlar, såg lærar på som ein ressurs som både kan og bør nyttast oftare.

5 Drøfting

I dette kapittelet vil eg drøfte resultatane mine frå datamateriale opp mot tidlegare forskning og erfaringar. Eg har allereie danna meg eit godt grunnlag for å kunne svare på problemstillinga basert på resultat frå forrige kapittel. Svaret på mi problemstilling vil då vere at analog programmering i støtte av fleire av prinsippa frå undervisningsopplegget absolutt var er noko ein lærar kan bruke på ein god måte for å fremje engasjement og utforskartrøng. Lærar synes det var lett å ta i bruk og elevane synte godt engasjement og utforskartrøng i heile undervisningsopplegget. På den andre sida kunne vi ikkje vere sikre på at analog programmering fremja skaparglede. Vidare i drøftinga vil eg prøve å løfte dette opp på eit høgare nivå og og forsøke sjå kva betydning det kan ha i ein større samanheng. Diskusjonen vil ta utgangspunkt i resultatane frå forskingsspørsmåla mine.

5.1 Analog programmering for å fremje skaparglede

Ein kan ikkje med tryggleik seie at prinsipp i konteksten av analog programmering fremjar skaparglede. Eg såg tendensar i datamateriale, men eg synes det var uklart og vil difor ikkje konkludere om undervisningsopplegget har greidd fremje målet om skaparglede. Då elevane sjølv arbeidde med oppgåver såg ein tendensar til skaparglede, særleg då elevane sjølv ønskte å lage oppgåver.

5.2 Analog programmering for å fremje engasjement

Eit undervisningsopplegg med analog programmering såg ut til å ha gode resultat for å fremje engasjement hjå elvane. Særleg det å nytte ikkje-permanente tavler såg ut til å ha stor effekt på elevane. Det var med på å synleggjere arbeidet for kvarandre, noko som la til rette for diskusjon og samarbeid om oppgåvene. Dette stemmer godt overeins med det Liljedahl (2021) sjølv skriv, og dette var noko som fungerte frå dag 1. Vidare peika Liljedahl på at desse ikkje-permanente tavlene ville bidra til å hjelpe elevane å halde ut lenger når oppgåver vert vanskelege. Eg og lærar var etter økt 2 litt usikre på om oppgåvene var utfordrande nok, så i kva grad elevane vart motiverte og engasjerte nok til å halde ut lenger ved utfordrande oppgåver kan eg ikkje med tryggleik seie. Dette er noko ein kunne sett meir på om ein skulle gått vidare med ei eventuell økt 3.

Eit anna moment som ikkje såg ut til å stemme godt overeins med forventninga, var bruk av tilfeldige grupper blant elevane. I mitt prosjekt såg tilfeldige valde grupper ikkje ut til å fremje elevengasjement. Dette er i strid med forskinga til Liljedahl (2016), som peika på at dette var noko som bidrog til engasjement. Den første tanken eg gjorde meg då, var eventuelle feilkjelder. Sidan Liljedahl har forska på dette i fleire år, ville eg først trur at det må ha vore feilkjelder i mitt og lærar sitt eige opplegg. Men etter å ha sett gjennom datamateriale og tatt ein diskusjon rundt dette, så meinte vi at vi hadde gjort alt riktig i høve Liljedahl sitt rammeverk. Vi måtte difor tenke på andre årsaker til kvifor det enda slik. Liljedahl sin metodikk blir i media (Jørgensen, 2023; Humstad & Kalinina, 2022) ofte sett på som noko heilt fantastisk og som gjev sær gode resultat. Men for å få eit godt utbytte av metodikken er ein nok avhengig av å jobbe med det over tid. Å implementere ei "Tenkande klasserom"-tilnærming på ein einaste dag eller ei veke kan vere utfordrande. Eit Tenkande klasserom er ikkje eit eingongsarrangement, men heller ein pedagogisk filosofi som krev kontinuerleg engasjement og praksis. Det inneber ei endring i undervisnings- og læringsstrategiar, klasseromsdynamikk og studentengasjement. For å etablere eit Tenkande klasserom, må lærar vanlegvis gjennom ein prosess med kompetanseutvikling, der ein lærer om prinsipp og strategiar knytt til tilnærminga. Deretter kan ein introdusere og integrere desse strategiane gradvis inn i den daglege undervisningspraksisen over ein lengre periode. Dette gjer det mogleg for elev og lærar å tilpasse seg dei nye metodane og utvikle dei naudsynte ferdigheitene og tankesettet for eit tenkesentrert læringsmiljø. Sjølv om det kanskje ikkje er mogleg å fullstendig implementere heile Tenkande klasserom-tilnærminga på ein gong, kan lærar absolutt starte med å gradvis inkorporere element av det (noko vi har prøvd i denne studien). Dei kan introdusere opne spørsmål, fremje samarbeidsretta problemløysingsaktivitetar, oppmuntre til elevdiskusjon og refleksjon, og gi høve for elevutforsking og oppdaging. Desse små stega kan starte prosessen med å dyrke eit meir tenkesentrert klasserom over tid. Det er viktig å vere klar over at eit Tenkande klasserom er ein pågåande prosess, ein prosess som utviklar seg etter kvart som lærar og elev blir meir komfortable og dyktige med strategiane og tankesettet knytt til det. Det handlar om å skape ein kultur for tenking og undersøking, noko som krev kontinuerleg innsats, praksis og støtte både frå lærar og elev.

Det å presentere svar eller idear såg ikkje ut til å fremje engasjement hjå elevane i denne klassen. Basert på erfaring lærar hadde frå før, då det i denne klassa var mange elevar som var ukomfortable med å presentere for resten av klassa, vil eg tru dette har hatt innverknad på

resultatet. Eit utrygt klassemiljø, der elevar ikkje stolar på kvarandre, kan absolutt føre til dårleg samarbeid og ei redsle for å dele idear og tankar med kvarandre. Når Liljedahl (2021) skriv at ein ofte skal lage tilfeldige arbeidsgrupper, var det i hovudsak for å unngå at til dømes ein svakare elev alltid blir para saman med ein sterk ein for at dei skulle klare å løyse oppgåvene. Her har det ikkje blitt tatt omsyn til at eit Tenkande klasserom i utgangspunkt bør vere etablert i ei klasse utan frykt. Altså ei klasse der elevane stolar på kvarandre, ei klasse der ein tør å diskutere og svare feil og som lærer av både egne og andre sine feil. Dette tenker eg er noko som går igjen i heile det Tenkande klasserommet, og som ein matematikklærer bør ta tak i tidleg i grunnskulen. Det å tidleg etablere eit miljø med aksept for å gjere feil, for å så bruke det til å lære, vil vere særst gunstig for einkvar lærar i matematikk og er noko ein aktivt bør jobbe for. For å fremje engasjement blant elevar kan det ofte vere meir effektivt å oppmuntre til aktiv deltaking, samarbeid og dialog. Dette kan inkludere og leggje til rette for diskusjonar i små grupper (Liljedahl nemner grupper på tre som optimalt), oppmuntre til spørsmål og refleksjon, og skape eit trygt og støttande læringsmiljø der elevar kjenner seg oppmuntra til å delta aktivt og dele sine tankar og perspektiv.

5.3 Analog programmering for å fremje utforskartrøng?

Det at elevane fekk arbeide med ei ferdig gitt kode, som dei skulle sjå på og diskutere seg fram til eit mogleg svar, såg ut til å ha stor effekt på utforskartrøng deira. Dette er eit viktig poeng i PRIMM-modellen, der elevane skal lære å uttrykke seg munnleg om programmeringsomgrep og -prinsipp. I PRIMM-modellen er det lagt opp til å arbeide parvis, medan i dette undervisningsopplegget jobba elevane i grupper på 4 og 5.

Eit moment som ikkje såg ut til å fremje utforskartrøng var det å jobbe med omgrep. Det å jobbe med omgrep og omgrepsforståing vert sett på som viktig og kan bli nytta som ein byggjestein i matematikkundervisninga. Lærar nemnte dette som noko han synes var viktig i matematikk og ønskte at elevane skulle arbeide meir med dette. Ei av årsakene til at arbeidet med omgrep ikkje fremja utforskartrøng, tenkjer eg kan vere at dette var heilt nye omgrep for mange og utfordrande å danne seg ei forståing av. Då vi repeterte omgrepa i økt 2 verka det som fleire av elevane hadde ei betre forståing. Her kan ein då tenke seg at det kan vere høveleg å legge gjennomgangen og samtalen om omgrep til slutten av økta når ein allereie har jobba med omgrepa i ein kontekst utan å sette namn på dei.

5.4 Analog programmering som noko attraktivt for lærarar

Ein av dei største fordelane med analog programmering var etter lærar si erfaring at det danna ein grunnleggjande forståing av programmering på eit nivå som er naudsynt for å få alle elevane (og lærar med lite kompetanse) med seg vidare til dei meir utbreidde undervisningsopplegga ein har tilgjengeleg i dag. Dette går igjen med Kjällander et al., (2018) og rapporten til NIFU (2019:13) som påpeikar at det var stor bekymring om mangelfull kompetanse på programmeringsfeltet. Lærarutdanninga gjorde ikkje nok til å kunne utdanne framtidens lærarar i programmering. Særleg handla det om dei didaktiske utfordringane ein lærar møter i arbeidet med programmering. Mange lærarar har dei siste åra gått på kurs og skule for å lære å nytte programmering i undervisninga, men gjengangaren til dei fleste av desse er at det er lite fokus på det didaktiske. Ein kan seie at dei fleste elevar, lærarar og lærarutdanninga er i ei byrjingsfase når det kjem til programmeringsfaget. Erfaringsbanken er låg og forskingslitteraturen er ikkje uendeleg stor, noko eg har erfart i arbeidet med denne studien. Likevel har eg kome fram til at ein analog inngang kan vere føremålstenleg når ein skal byggje opp algoritmisk tankegang (Berg, 2021), og ikkje minst for å få bygd opp engasjement og utforskartrong. Dermed burde det kanskje bli forska meir på analog programmering i skulen, og, ikkje minst, kan dette vere eit viktig tema som lærarutdanninga må ta med seg. Det kan og seiast at kompetansemåla ikkje eine og aleine bør bli forstått som å programmere med digitale verktøy, men at det er rom for å tolke desse i favør av analoge prosessar. Dette er noko eg tenker lærarutdanninga kan jobbe betre med framover. Dette gjeld sjølvstøtt for dei med matematikk som fag, men ein kan finne nytte i andre fag som til dømes naturfag.

Ein anna ting som kan vere formålstenleg å diskutere er i kva grad undervisningsopplegget høver for «alle» elevar. I min studie valde vi i økt 2 å la elevane med størst utfordringar innan matematikk få lov til å arbeide på grupperom med to-lærar. Dette gjorde at elevgruppa vart mindre og dette hjalp lærar som fekk høve til å følgje opp andre elevar tettare. På den eine sida vil den positive effekten av å ta ut dei «svake» elevane peike mot at undervisningsopplegget kanskje ikkje høver for alle, men for dei med gjennomsnittleg nivå i matematikk og oppover fungerte det bra. På den andre sida kunne det vore interessant å finne ut korleis det hadde fungert for lærar om dei «sterke» elevane blei tatt ut. Elevgruppa ville også no vore mindre og lettare å kontrollere. Ein kan då stille spørsmålet om undervisningsopplegget vart betre grunna ei mindre elevgruppe, eller om det var grunna det generelle matematiske nivået til elevane.

6 Slutt

Føremålet med denne studien har vore å undersøke om analog programmering kan bidra til å fremje skaparglede, engasjement og utforskartrøng hjå elevar, og samstundes vere attraktivt for lærar. Ved å nytte DBR som metodisk tilnærming, noko som gav høve til å vere deltakande i planlegging, gjennomføring, evaluering og justering av undervisningsopplegget som vart gjennomført, ønska eg å få ein djupare innsikt i dette. Deltakinga la til rett for at eg som forskar fekk sjå korleis læraren arbeidde i praksis, og på den måten innhenta eg informasjon om korleis læraren brukte sin kunnskap og materiell i ein konkret undervisningssituasjon. Dette ga grunnlaget for å svare på følgjande problemstilling:

I kva grad og korleis kan analog programmering vere ein attraktiv tilnærming for å nå måla om skaparglede, engasjement og utforskartrøng i matematikkundervisninga?

Eg kan ut i frå min studie seie at eit undervisningsopplegg med bruk av analog programmering kan vere ein attraktiv tilnærming for å nå måla om engasjement og utforskartrøng. Eg er likevel ikkje sikker på om ein har nådd målet om skaparglede, sjølv om noko av datamateriale kan vise tendensar til dette. Eg kan difor ikkje konkludere med at undervisningsopplegget har nådd alle tre måla.

Vidare vil eg no dele refleksjonar og tankar om vidare forskning.

6.1 Konklusjon

Ut i frå resultata i dette masterprosjektet vil eg kunne meine at å bruke analog programmering som eit alternativ undervisningsopplegg for programmering i matematikk er noko som bør sjåast på som svært nyttig i skulen. Elevane fremja engasjement og utforskartrøng som er ein viktig del av verdigrunlaget til skulen og som absolutt bør oppmuntrast vidare. Det å kunne skape seg ei forståing for kva programmering er, korleis det fungera på ein praktisk måte med fysiske objekt, kan sjåast på som eit veldig godt verktøy for å arbeide vidare rundt tematikken. Alt i alt vil eg kunne seie at analog programmering absolutt legg godt til rette for engasjement og utforskartrøng blant elevane.

For å utarbeida eit opplegg som ga lærar desse erfaringane, tok eg som forskar og lærar saman i bruk Benton sine fem E-ar, Liljedahl sitt tenkande klasserom og Sentance sin Primm-modellen. Ved å kombinere dei elementa vi meinte var mest attraktive frå kvar og ein av dei, utarbeida vi eit opplegg som viste seg å vere attraktivt for lærar. Lærar tykkja at

undervisningsopplegget var særst attraktivt og ga hen ein måte å undervise programmering på som han ville ta med seg vidare i arbeidskvardagen. Det at fleire av lærarane manglar naudsynt kompetanse gjer arbeidet med programmering utfordrande, og lærar i denne oppgåva var difor særst glad for å kunne undervise programmering på ein måte som involverte hen meir. I tillegg ga det lærar ei klarare forståing for programmering, noko som bidrog og til at lærar kunne vere meir aktiv i si rolle og hjelpe elevane på ein god måte.

6.2 Refleksjon og vegen vidare

Det er viktig å poengtere at denne masteroppgåva berre representerer ein enkelt lærar sine erfaringar og ein klasse sin aktivitet rundt analog programmering, og resultatata er dermed ikkje representative for alle. På den andre sida kan naturalistisk generalisering gjere til at resultat frå denne masteroppgåva kan overførast til andre situasjonar. Studiet er presentert så transparent som mogleg, så lesarar av oppgåva sjølv kan vurdere om resultatata er gyldige for dei og deira situasjon.

Programmering er eit tidsrelevant tema som og blir meir relevant i skulen. Difor er det viktig å utforske korleis lærarar underviser og arbeider med kompetansemåla i programmering. I denne oppgåva har analog programmering blitt nytta som ein inngang til tematikken rundt programmering. I framtidig forskning kan det vere interessant å sjå vidare på korleis ein kan kople analog programmering som ein overgang mot den meir tradisjonelle tekstbaserte programmeringa ,og korleis det kan hjelpe elevar til å lære programmering på ein betre, raskare og eller smartare måte. Det er framleis behov for fleire studiar som forska på bruken av analog programmering, og korleis elevar opplever denne forma for undervisning. Det kan bidra til å forbetre undervisningsmetodane og hjelpe elevane til å utvikle naudsynte ferdigheitar for framtida.

Litteraturliste

- Anderson, T. and J. Shattuck (2012). "*Design-Based Research: A Decade of Progress in Education Research?*" *Educational Researcher* 41(1): 16-25.
- Aranda, George & Ferguson, Joseph. (2018). *Unplugged Programming: The future of teaching computational thinking?* *Pedagogika*. 68. 10.14712/23362189.2018.859.
- 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.
- Berg, T. K. (2021). *Analog programming*. *Tangenten - Tidsskrift for matematikkundervisning*, 32(3), 42-52.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. MIT media lab, ss. 1-25.
- Benton, L., Hoyles, C., Noss, R., & Kalas, I. (2016). *Building mathematical knowledge with programming: insights from the ScratchMaths project*. *Proceedings of Constructionism 2016*, s. 25-32.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). *Bridging primary programming and mathematics: Some findings of design research in England*. *Digital Experiences in Mathematics Education*, 3(2), 115-138. Henta frå <https://link.springer.com/content/pdf/10.1007/s40751-017-0028-x.pdf>
- B. A. Bradley and D. Reinking (2011). *Enhancing research and practice in early childhood through formative and design experiments*. *Early child development and care* 2011 Vol. 181 Issue 3 Pages 305-319.
- Chibas, Åsa & Nouri, Jalal & Norén, Eva & Zhang, Lechen. (2018). *Didactical strategies and challenges when teaching programming in pre-school*. 10.21125/edulearn.2018.0875.
- Cobb, P., et al. (2003). "*Design Experiments in Educational Research*." *Educational Researcher* 32(1): 9-13. Collective, T. D.-B. R. (2003). "*Design-Based Research: An Emerging Paradigm for Educational Inquiry*." *Educational Researcher* 32(1): 5-8.

- Cortina, T. J. (2015). *Broadening participation: Reaching a broader population of students through «unplugged» activities*. *Communication of the ACM*, 58 (3), 25–27.
<https://dl.acm.org/doi/pdf/10.1145/2723671>
- diSessa, A. A., & Cobb, P. (2004). *Ontological Innovation and the Role of Theory in Design Experiments*. *The Journal of the Learning Sciences*, 13(1), 77–103.
<http://www.jstor.org/stable/1466933>
- Dolonen, J. A., Kluge, A., Litherland, K., Mørch, A. I. (2019) *Litteraturgjennomgang av programmering i skolen*. Universitetet i Oslo
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., Solomon, C. (1969). *Programming-languages as a Conceptual Framework for Teaching Mathematics. Final report of the first fifteen months of the Logo project*. Washington, D.C, Bolt, Beranek and Newman. R-1889.
- Giannakos, M. N., Jaccheri, L., & Proto, R. (2013). *Teaching computer science to young children through creativity: Lessons learned from the case of Norway*. *Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research*, 103-111.
- Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Heikkilä, Mia & Mannila, Linda. (2018). *Debugging in Programming as a Multimodal Practice in Early Childhood Education Settings*. *Multimodal Technologies and Interaction*. 2. 42. <https://doi.org/10.3390/mti2030042>
- Hermans, F. Swidan, A. and Aivaloglou, E. 2018. *Code phonology: an exploration into the vocalization of code*. In *Proceedings of the 26th Conference on Program Comprehension (ICPC '18)*. Association for Computing Machinery, New York, NY, USA, 308–311. <https://doi.org/10.1145/3196321.3196355>
- Hiim, H., & Hippe, E. (2022). *Undervisningsplanlegging for yrkesfaglærere* (4. utgave. ed.). Oslo: Gyldendal.
- Holo, O.E., Kveim, E. N., Lysne, M.S., Taraldsen, L. H. & Haara, F.O. (2022) *A review of research on teaching of computer programming in primary school mathematics: moving towards sustainable classroom action*, *Education Inquiry*, <https://doi.org/10.1080/20004508.2022.2072575>

- Hovde, Kjell-Olav; Grønmo, Sigmund: *algoritme* i *Store norske leksikon* på snl.no. Henta 23. oktober 2022 frå <https://snl.no/algoritme>
- Hoyles. (1985). *What Is the Point of Group Discussion in Mathematics? Educational Studies in Mathematics*, 16(2), 205–214. <https://doi.org/10.1007/BF02400938>
- Humble, N., Mozelius, P., & Sällvin, L. (2020). *The introduction of programming in K-12 technology and mathematics teacher choice of programming tools and their perceptions of challenges and opportunities*. *Education Applications & Developments: Advances in Education and Educational Trends Series*, 117–126. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1456930>
- Humstad, K & Kalinina, K. (2022, desember 12). *Her snakkar dei I heile mattetimen – gir veldig gode resultat*. NRK. Henta 11.05.2023 frå <https://www.nrk.no/tromsogfinnmark/brukar-annleis-metodar-for-a-undervise-i-matte-far-overraskande-gode-resultat-1.16199767>
- Høgheim, S. (2020). *Masteroppgaven i GLU*. Bergen, Fagbokforlaget.
- Jacobsen, Dag Ingvar. 2005. *Hvordan gjennomføre undersøkelser?* 2. utg. Kristiansand: Høgskoleforlaget
- Johannessen, A., Christoffersen, L. & Tufte, P. A. (2021). *Introduksjon til samfunnsvitenskapelig metode* (6. utg.). Abstrakt forlag.
- Jørgensen, G. (2023, mars 20). *Meiner tavler i klasserommet gjer dei betre i matte*. NRK. Henta 11.05.2023 frå <https://www.nrk.no/rogaland/meiner-det-tenkende-klasserom-og-tavler-i-klasserommet-gjer-elevane-betre-i-matte-1.16333802>
- 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.
- Kjällander, S., Åkerfeldt, A., Mannila, L. & Parnes, P. (2018). *Makerspaces across settings: didactic design for programming in formal and informal teacher education in the Nordic countries*. *Journal of Digital Learning in Teacher Education*, 34(1), 18–30. <https://doi.org/10.1080/21532974.2017.1387831>
- Kluge, A., Litherland, K. T., Borgen, P. H. & Langslet, G. O. (In press). *Speaking in Codes: Combining programming with audio explanations*.

- Koehler, M., & Mishra, P. (2009). *What is Technological Pedagogical Content Knowledge (TPACK)?* Contemporary Issues in Technology and Teacher Education, 9(1), 60–70.
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju*. Gyldendal akademisk.
- Kölling, M. (2015). *Lessons from the Design of Three educational programming environments: Blue, BlueJ and Greenfoot*. International Journal of people-oriented programming, ss. 5-32.
- Larsen, A. K. (2017). *En enklere metode : veiledning i samfunnsvitenskapelig forskningsmetode*. Bergen, Fagbokforl.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., et al. (2011). *Computational thinking for youth in practice*. ACM Inroads, 2(1), 32–37.
- Liljedahl, P. (2016). *Building Thinking Classrooms: Conditions for Problem-Solving*. I P. Felmer, E. Pehkonen & J. Kilpatrick (Red.), *Posing and Solving Mathematical Problems: Advances and New Perspectives* (s. 361-386). Cham: Springer International Publishing. Hentet frå https://doi.org/10.1007/978-3-319-28023-3_21
- Liljedahl, P., et al. (2021). *Building thinking classrooms in mathematics, grades K-12 : 14 teaching practices for enhancing learning*. Thousand Oaks, California, Corwin.
- Lucas, Bill & Hanson, Janet & Bianchi, Lynne & Chippindall, Jon. (2017). *Learning to be an Engineer: Implications for the education system*.
<https://doi.org/10.13140/RG.2.2.23103.30888>
- Lye, S. Y., & Koh, J. H. L. (2014). *Review on teaching and learning of computational thinking through programming: What is next for K-12?* Computers in Human Behavior, 41, 51–61.
- Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). *Learning to program and learning to think: What's the connection?* Communications of the ACM, 29(7), 605–610.
<https://doi.org/10.1145/6138.6142>
- Mckenney, Susan & Reeves, T.. (2013). Educational Design Research.
https://doi.org/10.1007/978-1-4614-3185-5_11

- Moreau, H. N. (2021, desember 21). *Skal lære elevene koding, men forstår det ikke selv*. NRK. Henta 13.03.2023 frå <https://www.nrk.no/innlandet/laerere-trenger-hjelp-til-a-knekke-koden-pa-koding-1.15781343>
- Nardelli, E. (2019). *Do we really need computational thinking?* Commun. ACM 62, 2 (February 2019), 32-35. <https://doi.org/10.1145/3231587>
- NIFU rapport (2019:13) *Profesjonsfaglig digital kompetanse i lærerutdanningene*. Henta 30.09.2022 frå <https://nifu.brage.unit.no/nifu-xmlui/bitstream/handle/11250/2602702/NIFU-rapport2019-%20tf13rev.pdf?sequence=6&isAllowed=y>
- Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). *Primary Mathematics Teachers' Understanding of Computational Thinking*. *KI-Künstliche Intelligenz*, 1-12.
- NOU 2013:2. (2013). *Hindre for digital verdiskaping*. Henta 15.12.2022 frå <https://www.regjeringen.no/no/dokumenter/nou-2013-2/id711002/>
- NOU 2014:7. (2014) *Elevenes læring i fremtidens skole – Et kunnskapsgrunnlag*. Henta 10.05.2023 frå <https://www.regjeringen.no/no/dokumenter/NOU-2014-7/id766593/?ch=9>
- NOU 2015:8. (2015). *Fremtidens skole : Fornyelse av fag og kompetanser*. Henta 02.09.2022 frå <https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001/>
- Nouri, J., Zhang, L., Mannila, L. & Norén, E. (2020). *Development of computational thinking, digital competence and 21 st century skills when learning programming in K9*. *Education Inquiry*, 11(1), 1–17. <https://doi.org/10.1080/20004508.2019.1627844>
- Nygård, K. (2018). *Programmering i skolen*. Pedlex.
- OECD (2020), *Global Teaching InSights: Observation Tools*, OECD Publishing, Paris.
- Opplæringslova. (1998). *Lov om grunnskolen og den vidaregåande opplæringa* (LOV-1998 02-08). Oslo: Kunnskapsdepartementet. I. Hentet 11.05.2023 frå <https://lovdata.no/dokument/NL/lov/1998-07-17-61>
- Otterborn, Anna & Schönborn, Konrad & Hultén, Magnus. (2020). *Investigating Preschool Educators' Implementation of Computer Programming in Their Teaching Practice*. *Early Childhood Education Journal*. 48. 10.1007/s10643-019-00976-y.

- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Pea, R. D., & Kurland, D. M. (1984). *On the cognitive effects of learning computer programming*. *New Ideas in Psychology*, 2(2), 137–168.
[https://doi.org/10.1016/0732118X\(84\)90018-7](https://doi.org/10.1016/0732118X(84)90018-7)
- P. Plaza *et al.*, "Visual block programming languages and their use in educational robotics," 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 2020, pp. 457-464, <https://doi.org/10.1109/EDUCON45650.2020.9125219>
- Patton, M. Q. (1987). *How to Use Qualitative Methods in Evaluation*. California, USA: SAGE Publications, Inc.
- Postholm, M. B. (2005). *Kvalitativ metode: en innføring med fokus på fenomenologi, etnografi og kasusstudier*. Oslo: Universitetsforlaget AS.
- Postholm, M., Jacobsen, D., & Søbstad, R. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Oslo: Cappelen Damm akademisk.
- PrimmPortal. (2022). Henta 10.05.2023 frå <https://primmportal.com>
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009, November). *Scratch: Programming for all*. *Communications*, ss. 60-67. <https://doi.org/10.1145/1592761.1592779>
- Resnick, M. (2017). *Lifelong Kindergarten: Cultivating creativity through projects, passion, peers, and play*. Cambridge, MA: MIT Press.
- Scott, E. E., Wenderoth, M. P and Doherty, J. H. (2020) *Design-Based Research: A Methodology to Extend and Enrich Biology Education Research*. *CBE—Life Sciences Education* 2020 19:3
- Sevik, K. (2016). *Programmering i skolen*. Senter for IKT i utdanningen.
https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Sentance, S., Waite, J. & Kallia, M. (2019) *Teaching computer programming with PRIMM: a sociocultural perspective*, *Computer Science Education*, 29:2-3, 136-176,
<https://doi.org/10.1080/08993408.2019.1608781>

- StatPed. (2021a, 17.03.). *Språkvansker og begrepsavklaring*. Hentet 12.10.2022 frå <https://www.statped.no/laringsressurser/sprak-og-tale/sprakvansker-og-begrepsavklaring/>
- The Design-Based Research Collective. (2003). *Design-based research: An emerging paradigm for educational inquiry*. Educational Researcher, 5-8.
- Tjora, A. (2012). *Kvalitative forskningsmetoder i praksis*. Oslo: Gyldendal Norsk Forlag AS.
- Utdanningsdirektoratet (2018). *Kjerneelement*. Henta 11.05.2023, frå <https://www.udir.no/lk20/mat01-05/om-faget/kjerneelementer?lang=nno>
- Utdanningsdirektoratet (2019). *Algoritmisk tenkning*. Henta 06.05.2023, frå <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2020). *Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. Henta 20.04.2021, frå <https://www.udir.no/lk20/mat01-05?lang=nno>
- Utdanningsdirektoratet. (2021). *Rammeverk for lærerens profesjonsfaglige digitale kompetanse (PfdK)*. Henta 14.05.2023, frå <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/rammeverk-larerens-profesjonsfaglige-digitale-komp/>
- Van de Walle, John & Karp, Karen & Bay-Williams, Jennifer. (2014). *Elementary and middle school mathematics: Teaching developmentally* (4th Canadian Edition).
- Vihovde, E. H. (2019). *Programmeringsspråk*. I Store norske leksikon. Hentet fra <http://snl.no/programmeringsspr%C3%A5k>
- Waite (2018). *Pedagogy in teaching Computer Science in Schools: A Literature Review*. Henta frå <https://royalsociety.org/computing-education>
- Wohl, B., Porter, B., Clinch, S., 2015. *Teaching Computer Science to 5-7 year-olds*, in: .. <https://doi.org/10.1145/2818314.2818340>
- Wing, J. (2010, 11. 17). *Computational thinking: What and Why?* ss. 1-6.
- Zhang, L., & Nouri, J. (2019). *A systematic review of learning computational thinking through Scratch in K-9*. Department of Computer and Systems Sciences, Stockholm University.

Vedlegg

Vedlegg 1: Intervjuguide 1

Intervju guide 1

- Kva kan du om programmering?
- Kva har elevane lært frå før?
- Korleis underviser du programmering?
- Kva meina du skal til for å lage eit attraktivt undervisningsopplegg i matematikk?

- Kva synes du om dette opplegget? (Viser tabell 2)
 - Kva tenker du kan endrast for å gjere det meir attraktivt for din eigen del?
 - Andre innspel?

Vedlegg 2: Intervjuguide 2

Intervju guide 2 – Etter økt 1

Gjennomgang av timen -

Introduksjon / Inndeling av grupper / Utforsking og prøving av koding / Oppsummering

- Korleis gjekk denne delen for din eigen del?
- Korleis synes du elevane var i denne delen?
 - Bidrog det til skaparglede?
 - I så fall kva/korleis?
 - Bidrog det til engasjement?
 - I så fall kva/korleis?
 - Bidrog det til utforskartrøng?
 - I så fall kva/korleis?
- Noko som ikkje fungerte bra?
 - Kva og kvifor trur du det?
 - Kva kan vi endre til neste økt?
- Kva heilhetsinntrykk sit du igjen med frå økta?
- Andre tanker/erfaringar/innspeil?

Vedlegg 3: Intervjuguide 3

Intervju guide 3 – Etter økt 2

Gjennomgang av timen

Introduksjon / Inndeling av grupper / Utforsking og prøving av koding / Oppsummering

- Korleis gjekk denne delen for din eigen del?
- Korleis synes du elevane var i denne delen?
 - Bidrog det til skaparglede?
 - I så fall kva/korleis?
 - Bidrog det til engasjement?
 - I så fall kva/korleis?
 - Bidrog det til utforskartrøng?
 - I så fall kva/korleis?
- Noko som ikkje fungerte bra?
 - Kva og kvifor trur du det?
 - Kva kan vi endre til ei hypotetisk neste økt?
- Kva heilheitsinntrykk sit du igjen med frå økta?
- Andre tankar/erfaringar/innspeil?

- No når vi har gjennomført prosjektet
 - Kva erfaringar sitt du igjen med?
 - Var dette attraktivt for deg som lærar?
 - Nådde vi dei 3 prinsippa vi sett oss?

- Følte du vi nådde målet om skaparglede, engasjement og utforskartrøng hjå elevane?
 - Trur du elevane synes dette var kjekt?

Kan dette vere ønskeleg å gjere meir av?

Vedlegg 4: Godkjenningbrev frå NSD

Vurdering av behandling av personopplysninger

Skriv ut 02.02.2023

Referansenummer
455296

Vurderingstype
Standard

Dato
02.02.2023

Prosjekttittel

Masterprosjekt - Analog programmering

Behandlingsansvarlig institusjon

Høgskulen på Vestlandet / Fakultet for lærerutdanning, kultur og idrett / Institutt for språk, litteratur, matematikk og tolkning

Prosjektansvarlig

Karin Elisabeth Sørli Street

Student

Øystein Riksheim Østvik

Prosjektperiode

01.09.2022 - 31.05.2023

Kategorier personopplysninger

Alminnelige

Lovlig grunnlag

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

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 31.05.2023.

[Meldeskjema](#)

Kommentar

OM VURDERINGEN

Sikt har en avtale med institusjonen du forsker eller studerer ved. Denne avtalen innebærer at vi skal gi deg råd slik at behandlingen av personopplysninger i prosjektet ditt er lovlig etter personvernregelverket.

KOMMENTAR TIL GJENNOMFØRINGEN ELLER INFORMASJONSSKRIVET

Taushetsplikt

Forskningsdeltagerne har yrkesmessig taushetsplikt. De kan ikke dele taushetsbelagte opplysninger med forskningsprosjektet. Vi anbefaler at du minner dem på taushetsplikten.

Merk at det ikke er nok å utelate navn ved omtale av elever, pasienter el. Vær forsiktig med bruk av eksempler og bakgrunnsopplysninger som tid, sted, kjønn og alder.

FØLG DIN INSTITUSJONS RETNINGSLINJER

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

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

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til oss ved å oppdatere meldeskjemaet. Se våre nettsider om hvilke endringer du må melde: <https://sikt.no/melde-endringer-i-meldeskjema>

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

Vedlegg 5: Informasjonsskriv og samtykkjeskjema

Vil du delta i forskingsprosjektet ”Analog Programmering”?

Dette er eit spørsmål til deg om å delta i eit forskingsprosjekt der føremålet er å sjå på bruken av analog programmering i matematikkundervisning. I dette skrivet gjev vi deg informasjon om måla for prosjektet og om kva deltaking vil innebere for deg.

Føremål

Føremålet med prosjektet vil vere å sjå på om analog programmering kan vere ein alternativ måte å nå måla for skaparglede, engasjement og utforskartrong i grunnskulen. Her vil eg og sjå på korleis ein lærar oppfatar undervisningsopplegget og om det er mogleg å gjere dette gjennomførbart i ein hektisk kvardag.

Undervisningsopplegget tar utgangspunkt i relevant teori samt erfaringar frå samtale med kontaktlærar.

Dette er ei masteroppgåve av Øystein Riksheim Østvik ved Høgskulen på Vestlandet Campus Sogndal.

Kven er ansvarleg for forskingsprosjektet?

Høgskulen på Vestlandet campus Sogndal og meg sjølv Øystein Riksheim Østvik vil vere ansvarleg for prosjektet.

Kvifor får du spørsmål om å delta?

Eg ønskje å observere to undervisningstimar der ein brukar analog programmering i matematikk og etter samtalar med lærarar falt valet på Halbrend Skule og 6c til denne oppgåva.

Kva inneber det for deg å delta?

For å samle datamateriale til mi oppgåve treng eg å gjere samtala med deg som informant i tillegg til meir formelle intervju etter undervisningsopplegg.

Dersom du vel å delta i prosjektet, inneber det ingenting anna enn at du gjennomføra to undervisningstimar i matematikk som i ein normal skulekvardag samt deltek i intervju og samtalar undervegs

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Dersom du vel å delta, kan du når som helst trekkje samtykket tilbake utan å

gje nokon grunn. Alle opplysningane dine vil då bli sletta. Det vil ikkje føre til nokon negative konsekvensar for deg dersom du ikkje vil delta eller seinare vel å trekkje deg.

Ditt personvern – korleis vi oppbevarer og bruker opplysningane dine

Vi vil berre bruke opplysningane om deg til føremåla vi har fortalt om i dette skrivet. Vi behandlar opplysningane konfidensielt og i samsvar med personvernregelverket.

- *Det vil kun vere meg sjølv og Høgskulen på Vestlandet som vil ha tilgang til opplysningane eg samlar*
- *Opplysningane vil bli laga på ein sikker server tilhøyrande høgskulen på vestlandet utilgjengeleg for alle andre enn meg sjølv og vegleiarar for prosjektet*

Alt vil forgå anonymt så ingen av opplysningane vil gjere det mogleg å kjenne att enkelt individ

Kva skjer med opplysningane dine når vi avsluttar forskingsprosjektet?

Opplysningane blir anonymiserte når prosjektet er avslutta/oppgåva er godkjend, noko som etter planen er mai 2023.

Etter prosjektet er fullført vil all innsamla informasjon bli slettet

Kva gjev oss rett til å behandle personopplysingar om deg?

Vi behandlar opplysningar om deg basert på samtykket ditt.

På oppdrag frå *Høgskulen på vestlandet* har Personverntjenester vurdert at behandlinga av personopplysingar i dette prosjektet er i samsvar med personvernregelverket.

Dine rettar

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

- innsyn i kva opplysingar vi behandlar om deg, og å få utlevert ein kopi av opplysningane,
- å få retta opplysingar om deg som er feil eller misvisande,
- å få sletta personopplysingar om deg,
- å sende klage til Datatilsynet om behandlinga av personopplysingane dine.

Dersom du har spørsmål til studien, eller om du ønskjer å vite meir eller utøve rettane dine, ta kontakt med:

- *Høgskulen på Vestlandet campus Sogndal ved Øystein Riksheim Østvik: 573937@stud.hvl.no*
- *Høgskulen på Vestlandet campus Sogndal ved Odd-Eivind Holo: odd-eivind.holo@hvl.no*
- *Høgskulen på Vestlandet campus Sogndal ved Karin Elisabeth Sørlie Street: karin.street@hvl.no*
- *Vårt personvernombod: Trine Anikken Larsen, Trine.Anikken.Larsenvl.no hjå Høgskulen på Vestlandet].*

Dersom du har spørsmål knytt til Personverntjenester si vurdering av prosjektet kan du ta kontakt med:

- Personverntjenester, på e-post (personverntjenester@sikt.no) eller på telefon: 53 21 15 00.

Venleg helsing

Øystein Riksheim Østvik

Odd-Eivind Holo / Karin Street (Forskar/rettleiar)

Samtykkeerklæring

Samtykke kan innhentast skriftleg (også elektronisk) eller munnleg. NB! Du må kunne dokumentere at du har gjeve informasjon og innhenta samtykke frå dei du registrerer opplysingar om. Vi tilrår som hovudregel skriftleg informasjon og skriftleg samtykke.

- *Ved skriftleg samtykke på papir kan du bruke denne malen.*
- *Ved skriftleg samtykke som blir innhenta elektronisk må du velje ein framgangsmåte som gjer at du kan dokumentere at du har fått samtykke frå rett person.*
- *Dersom konteksten tilseier at du bør gje munnleg informasjon og innhente munnlege samtykke (t.d. ved forskning i munnlege kulturar eller mellom analfabetar), tilrår vi at du gjer lydopptak av informasjon og samtykke.*

Dersom foreldre/verje samtykker på vegner av born eller andre utan samtykkekompetanse, må du tilpasse formuleringane. Hugs at namnet til deltakaren må kome fram.

Tilpass avkryssingsboksane etter det som er aktuelt i ditt prosjekt. Det er mogeleg å bruke punkt i staden for avkryssingsboksar. Men viss du skal behandle særskilte kategoriar personopplysingar og/eller dei fire siste punkta er aktuelle, tilrår vi avkryssingsboksar pga. krav om eksplisitt samtykke.

Eg har motteke og forstått informasjon om prosjektet *Analog Programmering* og har fått høve til å stille spørsmål. Eg samtykker til:

- å delta i [*Undervisningsopplegg med analog programmering*]

Eg samtykker til at opplysingane mine kan behandlast fram til prosjektet er avslutta.

(Signert av prosjektdeltakar, dato)

Vedlegg 6: Symbol til undervisningsopplegg



Plukk opp koppen



Sett ned koppen



Eit skritt fram



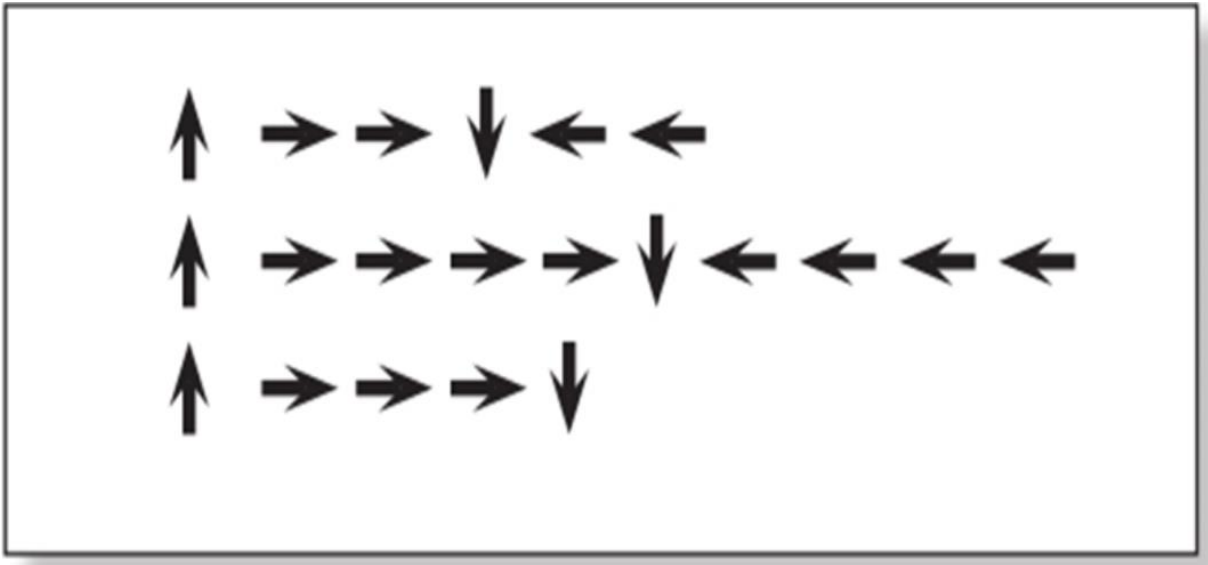
Eit skritt tilbake



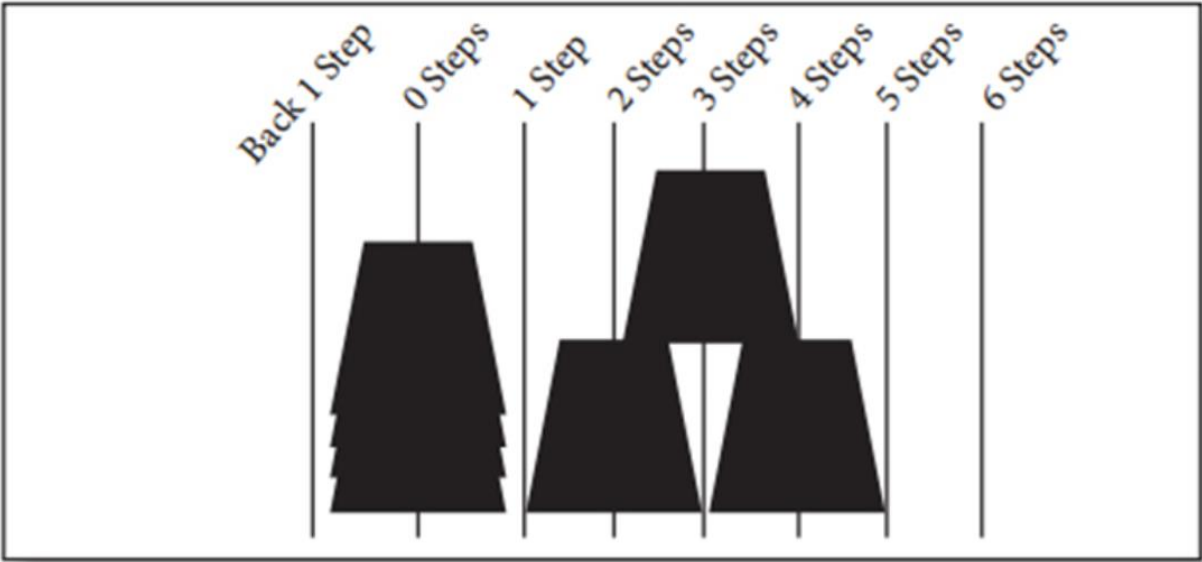
**Snu koppen 90°
til høgre**



**Snu koppen 90°
til venstre**



Fasit til kode



Oppgåve: Programmere ein «Robot»

1. Vel ein i gruppa som skal vere «Robot» først.
2. Send «Roboten» til «Robot Biblioteket», mens programmerarane (resten av gruppa) lagar **kode**.
3. De som koder skal programmere «Roboten» til å gjere oppgåve B.
4. Når koden er ferdig kan de hente «Roboten» som skal utføre oppgåva.
 - a. Har de programmert feil må «Roboten» sendes tilbake og de må «debugge» (Finne og fikse feil i koden).
5. Når «Roboten» klarer oppgåva kan de bytte på kven som er «Robot» og prøve på neste oppgåve.

