Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for samfunns- og utdanningsvitenskap
Institutt for lærerutdanning

Joakim Høyland

Algebra fra Scratch

En studie av elevers arbeid med programmering i algebraundervisning, med fokus på generalisering og variabler

Masteroppgave i Master i matematikkdidaktikk 5.-10. trinn
Veileder: Øistein Gjøvik
Medveileder: Berit Bungum
Mai 2021

NTNU
Kunnskap for en bedre verden

Joakim Høyland

Algebra fra Scratch

En studie av elevers arbeid med programmering i algebraundervisning, med fokus på generalisering og variabler

Masteroppgave i Master i matematikkdidaktikk 5.-10. trinn
Veileder: Øistein Gjøvik
Medveileder: Berit Bungum
Mai 2021

Norges teknisk-naturvitenskapelige universitet
Fakultet for samfunns- og utdanningsvitenskap
Institutt for lærerutdanning

NTNU
Kunnskap for en bedre verden

Sammendrag

Programmering krever at elever generaliserer og bruker variabler, noe som gir dem økt forståelse av algebra. I denne studien har jeg brukt designforskning (Design-Based Research) til å utvikle og prøve ut et undervisningsdesign for å undersøke om programmering kan være et godt verktøy ved oppstart av algebraundervisning. Bakgrunnen for studien er at programmering ble tatt inn i matematikkfaget i den nye læreplanen LK20, som ble iverksatt i den norske skolen august 2020. Med et ønske om et helhetlig fokus, ble det gjort undersøkelser i lys av tre ulike teoriplattformer: elevers generaliseringsnivåer, elevers beskrivelse av variabler og kognitiv belastningsteori. Undervisningsdesignet fasiliterer overføringslæring mellom fagområdene programmering og algebra, gjennom fellestrekkene *generalisering* og *variabler*. Undervisningsdesignets hensikt var å bidra til høy grad av generalisering hos elevene, utvikle elevers tolkning av variabelbegrepet, samt legge til rette for læring i algebra. På samme tid var det nødvendig at designet tok hensyn til at det var krevende for elevene å jobbe med både programmering og algebra samtidig.

Utprøvingen av designet ble gjennomført i en 8. klasse på en ungdomsskole i en norsk storby. Elevene brukte programvaren Scratch (https://scratch.mit.edu/) i sitt arbeid med oppgavene. Scratch ble i sin tid utviklet av MIT (Massachusetts Institute of Technology), rettet mot barn og ungdom som ville lære seg programmering. I studien ble det samlet inn datamateriale fra tilsammen 31 elever. Datamaterialet består av skjerm- og lydopptak av elevenes arbeid i par, elevenes Scratch-program, elevenes skriftlige notater og intervju av seks elever. Det ble gjennomført tematisk analyse, med hovedsakelig deduktiv tilnærming, der kategoriene var gitt fra det teoretiske grunnlaget. Et stort og bredt datamateriale bidro til god helhetsoversikt.

Resultatene viser at kombinasjonen av fagområdene programmering og algebra, og bruk av fellestrekkene generalisering og variabler, kan føre til et algebraisk faglig utbytte hos elevene. Samtlige elever generaliserte, men på varierende nivåer. Flere av elevene utviklet sine tolkninger av variabelbegrepet, men trolig har de resterende elevene behov for mer tid til å la de abstrakte idéene modnes. Kognitiv belastningsteori belyste en utfordring da elevene møtte på unødvendige programmeringstekniske utfordringer, som kunne vært unngått. Resultatene gir grunnlag for flere endringer av undervisningsdesignet, der den viktigste endringen er å tydeliggjøre hvilke forkunnskaper elevene bør ha i programmering, før de gjennomfører dette undervisningsdesignet.

Abstract

In programming students have to generalize and use variables, which will develop their understanding of algebra. Through the use of Design-Based Research, this study developed, and tested, an educational design to investigate if programming can be a proper tool in the beginning of algebra instruction. The background of this study is the implementation of programming in the subject of mathematics in the new curriculum, LK20, that was put into effect in Norway in august 2020. The necessity of a holistic view lead to investigation based on three different theoretical scaffolds: students' level of *generalization*, development of the students' notion of *variable* and *cognitive load*. The educational design facilitates transfer learning between the subject areas programming and algebra. The purpose of the educational design is to contribute, to a high level of generalization for the students, and develop the notion of the phenomenon variable. At the same time the educational design needed to considerate the demanding situation where the students needed to split their focus between both programming and algebra at the same time.

The testing and investigation of the educational design was carried out in an 8th grade (12-13 years old) at a lower secondary school in a large Norwegian city. The students used the software Scratch (https://scratch.mit.edu/) for their programming, that was developed by MIT (Massachusetts Institute of Technology) for educational purpose. Data material was collected from a total of 31 students. The data consists of screen- and sound recordings of students' work in pairs, students' final Scratch program, students' written notes and interview of the total of six students. Thematic analysis was carried out, with a mainly deductive approach, where the categories were given from the theoretical basis. A large and broad data material contributed to a good overview in the analysis.

The results shows that the combination of the subject areas programming and algebra, and the use of the common features generalization and variables, can lead to student learning in algebra. All students generalized, but at different levels. Several students did develop their notion of variable, but some still need more time to overcome the advanced abstract ideas. The use of cognitive load theory managed to point out a challenge when the students had to split their attention between programming and algebra at the same time. The students had to overcome unnecessary challenges in programming, that could have been avoided with better planning of the educational design. The results suggest changes of the interventions in the educational design, and the most important suggestion of change is the students prerequisite knowledge in programming.

Forord

Denne masteroppgaven er et resultat av to år med videreutdanning – en mastergrad i matematikkdidaktikk 5-10 på NTNU i Trondheim. Det har vært to år med mye læring, mange diskusjoner, gode samarbeid og samtaler om skole, matematikkundervisning og forskning. Samtidig har perioden vært preget av koronapandemien, som førte til en annerledes hverdag, med etter hvert færre møtepunkt, faglige diskusjoner og sosiale sammenkomster. Heldigvis ble min datainnsamling i liten grad påvirket av koronasituasjonen.

Masteroppgaven er også et produkt av 12 års erfaring som lærer på Charlottenlund ungdomsskole i Trondheim. Erfaringer fra disse 12 årene med praksis har vært viktige og gode å ha med i diskusjoner og arbeid som er gjort i forbindelse med studiene. Temaet for masteroppgaven er direkte inspirert av min egen praksis som matematikklærer.

Jeg har skrevet oppgaven alene, men har hatt mange støttespillere rundt meg som har vært til stor hjelp. For det første har jeg hatt to gode, konstruktive og kritiske veiledere, som har vært til stor hjelp i arbeidet. Øistein Gjøvik, ansatt ved Institutt for lærerutdanningen på NTNU har vært hovedveileder, og Berit Bungum, ansatt på Skolelaboratoriet ved Institutt for fysikk, var medveileder. I tillegg har Marte Morken Høyland lest korrektur og kommet med gode tilbakemeldinger som har vært til stor hjelp i skrivearbeidet.

Datainnsamlingen ble gjort i en 8. klasse der 31 elever sa seg villig til å bli forsket på. Jeg vil takke klassen og elevene som med åpne armer ønsket meg velkommen, og lot meg få prøve ut undervisningsdesignet. I tillegg vil jeg takke faglærer i klassen, som var en verdifull medspiller i flere diskusjoner og avklaringer, bisto med praktisk hjelp og fleksibilitet ved å la meg få nødvendig tid i forbindelse med både utprøving av undervisningsdesignet, og selve datainnsamlingen.

Å gjennomføre et masterløp ved siden av jobb er krevende, men jeg var så heldig å bli frikjøpt i en stor del av stillingen gjennom «Kompetanse for kvalitet». Min studiepermisjon har påvirket kolleger, så jeg må også rette en takk til ledelse og kolleger på Charlottenlund ungdomsskole, som har vært fleksible og lagt til rette for at jeg kunne gjennomføre masterstudiet.

Som familiefar, har jeg vært avhengig av mye tålmodighet og fleksibilitet fra mine nærmeste mens jeg har brukt mye tid på jobb og studier. Jeg vil takke dere for støtten jeg har fått av dere, og at dere gav meg tid og mulighet til å gjennomføre masterstudiet.

Trondheim, mai 2021

Joakim Høyland

Innhold

1 Innledning: programmering og algebra 13

1.1 Programmering og matematikk i skolen 14

1.2 Problemformulering 16

1.3 Studiens disposisjon 18

2 Teoretisk grunnlag og rammeverk 19

2.1 Kognitiv belastningsteori og lav ekstern belastning 19

2.1.1 Minnets arkitektur og læring 19

2.1.2 Kognitiv belastningsteori for forbedring av undervisningsdesign 20

2.2 Generalisering i algebra 21

2.2.1 Algebraisk tenkning 22

2.2.2 Figurtall 22

2.2.3 Rammeverk for generalisering 24

2.3 Variabler 25

3 Metodisk arkitektur og utvikling av undervisningsdesign 29

3.1 Designutvikling 29

3.1.1 Generelle kjennetegn i designutvikling 29

3.1.2 Metodisk rammeverk for designutvikling 30

3.1.3 Designutvikling i denne studien 32

3.2 Undervisningsdesignets konstruksjon og utprøving 32

3.2.1 Undervisningsdesignets konstruksjon og utprøving 32

3.2.2 Intervensjonen og læringshypoteser i undervisningsdesignet 36

3.3 Forskningens valg og utførelse 38

3.3.1 Utvalg av elever 39

3.3.2 Datainnsamling og -materiale 39

3.3.3 Transkripsjon 41

3.3.4 Tematisk analyse 41

3.3.5 Forskningens troverdighet 42

3.3.6 Etiske problemstillinger 44

4 Resultater 47

4.1 Høy ekstern kognitiv belastning 47

4.2 Elevenes generaliseringer 49

4.2.1 Oppgave 1 – forsmak på generalisering 49

4.2.2 Oppgave 3 – symbolsk algebraisk generalisering i Scratch 51

4.2.3 Oppgave 4 - varierende grad av generalisering på et «matematisk språk» 53

4.3 Variabler – elevenes beskrivelser av bruk av bokstaver 56

4.3.1 Elevenes forkunnskaper - *ukjent* 56

4.3.2 Elevenes beskrivelser av bokstavene *a* og *b* 58

5 Diskusjon 63

5.1 Undervisningsdesignet........ 63

5.2 Hva lærer elevene? 67

5.3 Metoderefleksjoner........ 70

6 Konklusjon og avsluttende kommentarer........ 73

Referanser 75

Vedlegg........ 81

Figurer

Figur 2.1: Kognitiv teori om minnet (oversatt og bearbeidet fra Mayer, 2009)...................... 20
Figur 2.2: To versjoner av samme trigonometrioppgave (inspirert av Sweller et al., 1998, s. 278, 279).. 21
Figur 2.3: Trekanttall, figur $T1$ til $T4$ 23
Figur 2.4: Trekanttall - eksempel 1 23
Figur 2.5: Trekanttall - eksempel 2 23
Figur 2.6: Eksempeloppgave (Tennant & Colloff, 2014, s. 40)... 26
Figur 3.1: Designutviklingens tre-delte prosess, basert på beskrivelser fra Anderson og Shattuck (2012), og The Design-Based Research Collective (2003)...................................... 30
Figur 3.2: Læringshypoteser (oversatt fra Sandoval, 2014)... 31
Figur 3.3: Sandovals (2004) rammeverk (modell konstruert ut i fra Sandovals beskrivelser) 32
Figur 3.4: Skjermbilde av Scratch når man programmerer.. 33
Figur 3.5: Oppgave 2.. 35
Figur 3.6: Oppgave 3.. 35
Figur 3.7: Eksempel på besvarelse av oppgave 3 35
Figur 3.8: Oppgave 4.. 36
Figur 3.9: Designets intervensjon og tiltak, og læringshypoteser 37
Figur 3.10: Datamateriale.. 40
Figur 3.11: Kriterier for analyse av opptak 42
Figur 4.1: Anne og Nils (gruppe 1) setter *a* og *b* inn i en addisjonsoperator......................... 52
Figur 5.1: Forventet utfall av undervisningsdesignet (kopi av Figur 3.9).............................. 64
Figur 5.2: Endring av bildet som viser hvilke blokker som trengs for å løse oppgave 3........ 65

Tabeller

Tabell 1.1: Algoritmisk tenkning (oversatt fra Shute et al., 2017, s. 153) 15
Tabell 2.1: Rammeverk for generalisering (oversatt fra Radford, 2006)............................... 24
Tabell 2.2: Teoretisk rammeverk knyttet til bruk av bokstaver i algebra (oversatt fra Küchemann, 1981) 25
Tabell 4.1: Elevenes besvarelser på oppgave 1, 3 og 4... 50
Tabell 4.2: Gruppenes generaliseringsnivå på oppgave 3 basert på besvarelsene i Tabell 4.1 51
Tabell 4.3: Gruppenes generaliseringsnivå på oppgave 4 basert på besvarelsene i Tabell 4.1 53
Tabell 4.4: «Hvorfor bruke bokstavene a og b i stedet for tall?» (opptak)............................. 58
Tabell 4.5: Endelig kategorisering av elevenes beskrivelser av bokstavene a og b etter triangulering av datakildene 60

1 Innledning: programmering og algebra

Hverdagen vår har blitt avhengig av den digitale teknologien vi har rundt oss. For eksempel ved bruk av cruisekontrollen i bilen, selvbetjente kasser på butikken, administrative verktøy på jobb, læringsplattform på skolen og loggføringsprogrammer på trening. Brukere av den digitale teknologien må forsøke å henge med på den raske teknologiske utviklingen. Det er noe av grunnen til at programmering ble innført i den norske skolen høsten 2020. Internasjonalt er det å fremme logisk tenkning og problemløsningskompetanse blant de viktigste argumentene for å integrere programmering i skolen (Sanne et al., 2016). Argumentene kan spores tilbake til Wings (2006) første artikkel om algoritmisk tenkning (computational thinking), som beskriver hvilke ferdigheter utviklere (programmerere) bruker, og argumenterer for hvordan ferdighetene kan være viktig å lære seg for alle, uansett beskjeftigelse. Blant annet handler algoritmisk tenkning om å tenke abstrakt og danne seg konseptuell forståelse om et fenomen. Algoritmisk tenkning har likhetstrekk med generalisering i algebra, som er kjernen i denne studien.

Sanne et al. anbefalte i sin rapport at teknologi og programmering burde være et eget fag i skolen (Sanne et al., 2016). Til tross for utvalgets anbefalinger, så man i grunnskolen at programmering heller ble implementert i fagene musikk, kunst og håndverk, naturfag og matematikk (Utdanningsdirektoratet, 2019b, 2019c, 2019d, 2019e), der matematikkfaget fikk et spesielt ansvar for opplæringen i programmering. Programmering er nå, på lik linje med for eksempel geometri og statistikk, et fagområde i matematikkfaget. Kompetansemålene i matematikk på 9. og 10. trinn viser til at programmering skal brukes for læring og utforsking i andre matematiske områder (Utdanningsdirektoratet, 2019c). En stor utfordring er at det fortsatt finnes lite forskning og få erfaringer å se til om hvordan dette kan gjøres, og hvilke konsekvenser det eventuelt vil gi (Sanne et al., 2016; Schanzer, Fisler, Krishnamurthi & Felleisen, 2015). Det er kjent at gode kunnskaper i matematikk kan øke motivasjonen til å gå videre med programmering (Shute, Sun & Asbell-Clarke, 2017), og at matematikk er viktig for å mestre å programmere (Sung, Ahn & Black, 2017). Samtidig finnes det for lite forskning om hvordan programmering kan bidra til læring i de andre fagområdene i matematikk (Schanzer et al., 2015). Endringen av læreplanen har ført til et behov for forskning på temaet, i tillegg til at lærere trenger mer kunnskap og hjelp til å finne ut hvordan dette kan gjøres i praksis. I denne undersøkelsen vil jeg se nærmere på hvordan man kan kombinere programmering med algebra, og jeg håper at den kan være et nyttig bidrag til både praksisfeltet, lærerutdannere, didaktikere og forskere.

For å knytte sammen studien med praksisfeltet er det valgt å bruke forskningsmetoden designforskning (Design-Based Research). Designforskning går ut på å utvikle, prøve ut, forske på og evaluere et undervisningsdesign. På den måten knytter metoden tettere bånd mellom empirien og praksisfeltet, som bidrar til systematisk kunnskap av mer generell relevans (Anderson & Shattuck, 2012). Designforskning har blitt kritisert for at forskningen er for sterkt forankret i de lokale kontekstene. Sandovals (2004) motargument er at utviklingen av undervisningsdesign i stor grad er en teoretisk aktivitet, som muliggjør generalisering av de lokale resultatene. I denne studien har jeg tatt utgangspunkt i Sandovals (2004, 2014) metodiske rammeverk for designforskning.

For å forstå undersøkelsens bakgrunn er det nødvendig å se nærmere på relasjonen mellom programmering og de andre områdene i matematikkfaget.

1.1 Programmering og matematikk i skolen

Papert (1980) blir sett på som opphavsmannen til bruk av programmering i matematikkundervisning. Han brukte programmeringsspråket LOGO til å tegne geometriske figurer. Selv om mange tok i bruk LOGO, ble det aldri etablert som arbeidsmetode i skolen (Benton, Hoyles, Kalas & Noss, 2016). Paperts ideer fikk etter hvert sin renessanse da Wing (2006) kom med sin artikkel om *algoritmisk tenkning* (computational thinking). Hennes oppfatning var at det å programmere innebar å utvikle ulike ferdigheter (algoritmisk tenkning), som *alle* ville ha nytte av, uansett beskjeftigelse. Wing rettet ikke sine tanker mot skolen, men hennes artikkel har inspirert andre til å se nærmere på dette feltet (blant annet Shute et al., 2017; Weintrop et al., 2016). Likevel er det mye som fortsatt er uklart om hvordan programmering kan implementeres i matematikkfaget (Schanzer et al., 2015). Sammenlignet med Paperts LOGO har vi i dag helt andre verktøy tilgjengelig, men samtidig er det få som har sett direkte på hvordan *algoritmisk tenkning* kan implementeres i matematikkfaget (Resnick et al., 2009; Sanne et al., 2016; Schanzer et al., 2015). Den dag i dag er det fortsatt et behov for en klargjøring av hva algoritmisk tenkning er i matematikkfaget.

Betydningen av begrepet algoritmisk tenkning er ikke det samme som *algorithmic thinking*. Algorithmic thinking har en mer snever betydning enn *computational thinking*. Derfor støtter denne oppgaven seg til Gjøvik og Torkildsens (2019) standpunkt, der de sier at begrepet vi bruker i Norge, *algoritmisk tenkning* (AT), har tilsvarende betydning som computational thinking.

Etter Wings (2006) artikkel er det kommet flere definisjoner av AT, der noen av dem spisser seg inn mot bruk i skole. Blant dem er Shute, Sun & Asbell-Clarke (2017) sitt rammeverk, som viser bredden av aspekter som kan inngå i AT. Rammeverket består av kategoriene og begrepene *dekomposisjon, abstraksjon, algoritmer, feilsøking, iterasjon* (gjentakelse) og *generalisering* (se forklaringer og underkategorier i Tabell **1.1**)[1].

I sin analyse av ulike definisjoner av både algoritmisk tenkning og algebraisk tenkning, fremhever Kilhamn og Bråting (2019) tre skjæringspunkt: struktur, symbolisering og generalisering og abstraksjon. I denne oppgaven har jeg valgt å undersøke hvordan man kan bruke programmering til å utforske algebra gjennom to av skjæringspunktene; symbolisering (variabler) og generalisering. Utfordringen er at forskningen på dette området er delt. På den

[1] Rammeverket har sine likheter med Utdanningsforbundets beskrivelse av AT, som inneholder begrepene *logikk, algoritmer, dekomposisjon, mønstre, abstraksjon* og *evaluering* (Utdanningsdirektoratet, 2019a).

Tabell 1.1: Algoritmisk tenkning (oversatt fra Shute et al., 2017, s. 153)

Begrep	**Definisjon**
Dekomposisjon	Dele opp et komplekst problem i håndterbare deler. Delene er ikke tilfeldige, men er funksjonelle elementer som kollektivt forenes i det hele systemet/problemet.
Abstraksjon	Trekker ut essensen i et (komplekst) system. Abstraksjon har tre underkategorier: (a) *Datainnsamling og analyse*: Samle inn viktig og mest relevant informasjon fra multiple kilder, og forstå relasjonene mellom ulike datasett *(b)* *Figurgjenkjenning*: Identifisere mønstre/regler som ligger under strukturen i dataene/informasjonen *(c)* *Modellering:* Bygge modeller eller simulasjoner som representerer hvordan et system opererer, og/eller hvordan et system vil fungere i fremtiden.
Algoritmer	Designe logiske og sekvenserte instruksjoner for å fremstille en løsning på et problem. Instruksjonene kan utføres av et menneske eller en maskin. Algoritmer har fire underkategorier: (a) *Algoritmisk design*: Konstruere sekvenser med koder, trinn for trinn, som kan løse problemet. (b) *Parallellisme*: Utføre flere sekvenser med koder samtidig. (c) *Effektivitet*: Fjerne overflødig og unødvendig kode slik at koden består av færrest mulig trinn. (d) *Automasjon*: Automatisere utførelsen slik at koden kan bli brukt til å løse lignende problem.
Feilsøking	Finne og identifisere feil, deretter fikse feilene, når løsningen ikke fungerer slik den skal.
Iterasjon	Gjenta designprosessen for å raffinere løsningene, helt til man har oppnådd ønsket resultat.
Generalisering	Overføre ferdighetene fra AT til andre situasjoner og temaer, for å effektiv og produktivt løse andre problemer.

ene siden ser man hvordan erfaringer med programmering er overførbart til algebra. Elevene opplever en annen representasjon av variabler, algebrauttrykk og algebraiske symboler i programmering, som kan gi økt forståelse av variabler og algebrauttrykk i algebra (Agatolio, Albanese & Moro, 2018; Bråting & Kilhamn, 2020; diSessa, 2018; Kilhamn & Bråting, 2019; Noss, 1986; Schanzer et al., 2015; Sengupta, Kinnebrew, Basu, Biswas & Clark, 2013; Usiskin, 1988). På den andre siden ser man at begreper har ulike betydninger i de to fagområdene, som kan føre til utfordringer og misoppfatninger (Agatolio et al., 2018; Bråting & Kilhamn, 2020; diSessa, 2018; Noss, 1986; Schanzer et al., 2015; Sengupta et al., 2013; Shute et al., 2017; Usiskin, 1988). Ett eksempel er begrepet *variabel,* som i matematikk står for én ukjent eller varierende mengder, mens variabelen i programmering har som formål å lagre informasjon som kan endres og brukes om igjen (Bråting & Kilhamn, 2020).

Oppsummert viser tilgjengelig forskning at programmering har flere fellestrekk med algebra, samtidig som det også kan by på flere mulige utfordringer.

1.2 Problemformulering

Resultatene fra TIMMS-undersøkelsene viser at norske ungdomsskoleelever, sammenlignet med de andre nordiske landene, skårer lavest i algebra (Kaarstein, Radišić, Lehre, Nilsen & Bergem, 2020; Utdanningsdirektoratet, 2016). I tillegg er algebra det området som trekker den norske gjennomsnittsskåren i matematikk ned. Det er interessant at de svenske forskerne Kilhamn og Bråting (2019) fremhever viktigheten av å undersøke skjæringspunktene mellom programmering og algebra. Det sammenfaller nemlig med Sveriges læreplaner, der programmering er innført i matematikk med vekt på fellestrekk i algoritmisk tenkning og algebra (Kilhamn & Bråting, 2019). Denne studien ser derfor nærmere på om elever kan utvikle algebraiske konseptuelle ideer ved å bruke programmering. Samtidig må man ta hensyn til at overgangen mellom programmering og algebra kan være krevende for elevene (Schanzer et al., 2015). Vi vet for lite om hvordan dette bør gjøres, og det er nettopp derfor at designutvikling er brukt som forskningsmetode. Det er utviklet, og prøvd ut, et undervisningsdesign som fasiliterer utviklingen av elevenes algebraiske konseptuelle forståelse ved bruk av programmering. Undersøkelsene, og utviklingen av undervisningsdesignet, har tre fokusområder, med hver sin tilhørende teoretiske plattform: *generalisering*, *variabelbegrepet* og *kognitiv belastning*.

Generalisering er et fellestrekk som inngår i både programmering (AT) og algebra. Grovt sett skiller man mellom *algebra* og *algebraisk tenkning* ved at algebra er det matematiske fagområdet der man lærer det algebraiske symbolspråket, som inkluderer de algebraiske manipulasjoner og operasjoner (Usiskin, 1988). Tradisjonelt har algebra i Norge blitt innført på ungdomstrinnet (Utdanningsdirektoratet, 2013). Derimot har algebraisk tenkning som mål å la elevene øve på abstraksjon og generalisering, helst i god tid før de lærer algebra (Kieran, 2004a). *Generalisering* er et skjæringspunkt mellom algebraisk tenkning og algoritmisk tenkning (Kilhamn & Bråting, 2019). Radford (2006) sitt rammeverk viser til ulike nivå av generalisering i forbindelse med algebraisk tenkning. Når elever mestrer å se det generelle og konseptuelle, samtidig som de navngir de ubestemte størrelsene med noe som er, eller tilnærmet er, et algebraisk symbolspråk, oppnår de Radfords øverste nivå: *symbolsk algebraisk generalisering*. Denne studien har derfor fokus på hvilket generaliseringsnivå elevene kan oppnå når man bruker programmering til å utforske algebra.

Variabel er et felles begrep i programmering og algebra. Når elever skal lære algebra har de utfordringer med å danne seg en god forståelse av *variabelbegrepet* (Bush & Karp, 2013). Forskning viser at variabler – «bokstaver i algebra» – opptrer ulikt, ut i fra hvilken kontekst de er i (Ely & Adams, 2012). Küchemann (1981) har kategorisert elevers tolkninger av bokstaver, og differensierer blant annet mellom tilfeller der elever tillegger bokstaven en ukorrekt tolkning (for eksempel at *e* står for objektet eple), elever ser bokstaven kun som *én ukjent* (bokstaven står for ett bestemt tall), og kategorien *variabel* der elever ser hvordan verdien bak bokstaven kan variere. Sistnevnte kategori har høyest abstraksjonsnivå. Ely og Adams (2012) identifiserer overgangene mellom ukorrekte tolkninger, kategorien *ukjent* og kategorien *variabel* som kritiske og utfordrende for elever. Ekstra utfordrende i denne studien er at begrepet variabel, som tidligere nevnt, defineres og brukes på ulike måter i både programmering og algebra. Undervisningsdesignet forsøker å legge til rette for at elevene møter variabler i programmering som varierer, og opererer derfor innenfor Küchemanns kategori *variabel.* Hensikten er at bruk av variabel i programmering skal føre til at elevene utvikler sin forståelse av variabelbegrepet i algebra.

Undervisningsdesignet har et faglig krevende innhold, spesielt når de må overføre kunnskap mellom fagområdene. Spørsmålet er hvordan designet kan legge til rette for at elevene kan få et algebraisk faglig utbytte. Sweller, Van Merrienboer og Paas (1998) mener at «all instructional designs should be analyzed from a cognitive load perspective» (s. 262). Kognitiv belastningsteori (psykologisk forskningsområde) viser til at mennesker har begrenset mulighet til å bearbeide informasjon i arbeidsminnet, slik at det blir lagret i langtidshukommelsen (Mayer, 2009). Hvis arbeidsminnet må håndtere for mye informasjon kan det føre til kognitiv overbelastning, som resulterer i lite læring. Når man analyserer instruksjonen fra den som formidler, skiller Sweller et al. (1998) mellom *iboende kognitiv belastning*, som er den iboende vanskegraden selve informasjonen har, og *ekstern kognitiv belastning*, som er vanskegraden på hvordan informasjonen blir presentert – instruksjonens vanskegrad. Kombinasjonen av programmering og algebra vil trolig føre til relativt høy iboende kognitiv belastning. Denne undersøkelsen har som mål å forme et undervisningsdesign som kombinerer programmering og algebra, på en slik måte at den eksterne kognitive belastningen blir så lav som mulig.

Studiens forskningsspørsmål er følgende:

Hvordan kan elever utvikle algebraiske konseptuelle ideer ved å bruke generalisering og variabel i både programmering og algebra?

I designutvikling konkretiserer man problemformuleringene ned til *teoretiske prinsipp*, som blir førende for utviklingen av undervisningsdesignet. De teoretiske prinsippene i denne studien er utledet fra forskningsspørsmålet og problemformuleringene fra de tre fokusområdene *kognitiv belastning, generalisering og variabler*, og er dermed følgende:

1) *Designet skal legge til rette for læring ved å redusere den eksterne kognitive belastningen.*
2) *Designet skal bidra til at elevene oppnår et høyt generaliseringsnivå, der de nærmer seg symbolsk algebraisk generalisering.*
3) *Designet skal stimulere til at elevene skal overkomme de kritiske tolkningsovergangene av variabelbegrepet, og beskrive bruk av bokstaver som noe som varierer.*

Prinsippene er grunnlag for valg av teoretiske rammeverk og analyse av datamaterialet, og er dermed en rød tråd gjennom hele studien og oppgaven. For å analysere undervisningsdesignets vanskegrad er det brukt Sweller et al. (1998) sine beskrivelser av iboende og ekstern kognitiv belastning. Radfords (2006) rammeverk for generalisering er valgt for å kunne differensiere elevenes ulike generaliseringsnivåer. Küchemanns (1981) rammeverk om elevers tolkning av variabler er valgt for å kunne kategorisere elevers beskrivelser av bokstavene de bruker i oppgavene.

1.3 Studiens disposisjon

Studien består av seks kapitler. Kapittel 2 presenterer studiens teoretiske grunnlag med de tre rammeverkene som bidrar til å belyse de tre teoretiske prinsippene – kognitiv belastning, generalisering og variabler. Kapittel 3 er tredelt og presenterer først designutvikling som forskningsmetode og Sandovals (2004, 2014) rammeverk. Videre presenteres utviklingen og utprøvingen av undervisningsdesignet. Det blir redegjort både for designutviklingen, hvilke oppgaver elevene arbeidet med, og hvordan utprøvingen ble gjennomført i en 8. klasse. I tillegg presenteres hvilke utfall man forventer i utprøvingen, noe som er en viktig del i Sandovals rammeverk. I siste del av kapittelet blir det redegjort for de forskningsmessige valgene som er tatt, ved å se nærmere på datainnsamling, analyse, forskningens troverdighet og etiske problemstillinger. I kapittel 4 presenteres resultater og analyser fra forskningen. Resultatene presenteres tematisk basert på de tre teoretiske prinsippene. Kapittel 5 starter med å sammenligne læringshypotesene med resultatene. Det gir grunnlag for endring og forbedring av undervisningsdesign. Deretter sees studien i lys av annen relevant forskning, før studiens metodologiske og resultatmessige forhold diskuteres. I kapittel 6 oppsummeres først studiens resultater, før oppgaven avsluttes med å presentere nye aktuelle forskningsspørsmål som kan være aktuelle å se på som et resultat av denne studien.

2 Teoretisk grunnlag og rammeverk

I kvalitativ forskning er det viktigste forskningsinstrumentet selve forskeren (Postholm, 2010). Forskerne har alltid med seg sine fortolkninger og sin subjektivitet. Teorier hjelper oss med å se virkeligheten objektivt – eller *mer* objektivt. Postholm (2010) skriver at teorier ikke er virkeligheten, «men et bilde på eller en oppfatning av hvordan forhold i det virkelige livet er» (s. 20). Teorier hjelper oss med å distansere oss – samtidig må vi ogs¨være bevisst på hvilke begrensninger teorien har.

Det teoretiske fundamentet i denne studien plasserer seg i *sosialkonstruktivismen,* som forutsetter at mennesket er aktivt og ansvarlig i sin læring, og at læring skjer i et samspill med omgivelsene (Lerman, 2014; Postholm, 2010). Læringssynet har likhetstrekk med Vygotskys *sosiokulturelle teori*, som baserer seg på at alle stimuli blir mediert eller tolket av individet. Individets utvikling skjer på to plan, det ytre (sosiale) planet, og det indre (individuelle) planet. Individet bygger bro mellom det ytre og det indre planet gjennom å bearbeide den sosiale aktiviteten (Postholm, 2010). Forskjellen er at i Vygotskys teori har læringen utgangspunkt i omgivelsene, mens i sosialkonstruktivismen har læringen utgangspunkt i individet.

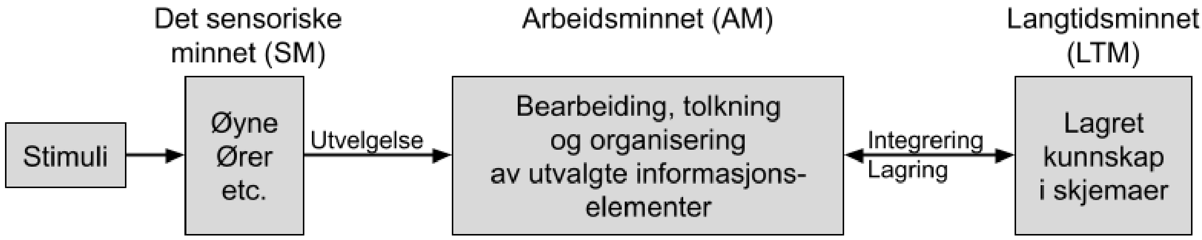
I dette kapittelet vil først teorien om kognitiv belastningsteori presenteres med beskrivelse av iboende og ekstern kognitiv belastning. Deretter er det redegjort for generalisering med et teoretisk rammeverk som kategoriserer hvilke ulike nivå av generaliseringer elevene gjør. Til slutt presenteres teori og rammeverk om variabler, som brukes til å kategorisere elevenes beskrivelser av bokstavene de har brukt i sitt arbeid i matematikk.

2.1 Kognitiv belastningsteori og lav ekstern belastning

I matematikkundervisning kan kognitiv belastningsteori bidra til å belyse faktorer som påvirker elevenes læring. Undervisningsdesignet skal la elevene bruke programmering til å utvikle algebraiske konseptuelle ideer, med utgangspunkt i at de ikke har forkunnskaper i algebra. Det gjør at det faglige innholdet kan være krevende. Kognitiv belastning sier noe om hvor mye informasjon mennesker kan behandle i arbeidsminnet samtidig. Den baserer seg på teoriene vi har om minnets oppbygning: det sensoriske minnet, arbeidsminnet og langtidsminnet. I psykologien defineres læring som når informasjon lagres i langtidsminnet (Mayer, 2009; Sweller et al., 1998). I første delkapittel forklares det hvordan psykologien deler inn minnet, hvordan delene opererer og samarbeider, og hva som skal til for at læring skal forekomme. Deretter redegjøres det for kognitiv belastningsteori, for å se nærmere på hvilke begrensninger som ligger i arbeidsminnet. Til slutt forklares *iboende* og *ekstern kognitiv belastning*.

2.1.1 Minnets arkitektur og læring

I psykologien deler man opp minnet i tre deler: det *sensorisk minnet* (SM), *arbeidsminnet* (AM) og *langtidsminnet* (LTM) (Mayer, 2009). I Figur **2.1** ser vi hvordan mennesket blir utsatt for stimuli, for eksempel gjennom en presentasjon, fortelling eller tekst. SM tar i mot



Figur 2.1: Kognitiv teori om minnet (oversatt og bearbeidet fra Mayer, 2009)

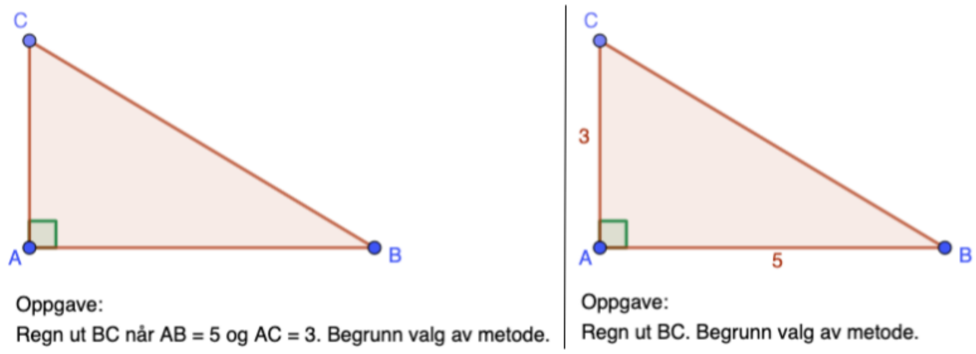
informasjonen og sender det videre til AM. I overgangen fra SM til AM skjer det en utvelgelse av noen få informasjonselementer, som videre bearbeides, tolkes og organiseres i AM. Videre ses informasjonselementene i AM i sammenheng med informasjonen som er lagret i LTM, som fører til ytterligere bearbeiding i AM. Til slutt lagres de bearbeidede informasjonselementene i LTM (Mayer, 2009).

I psykologien baserer man seg på teorien om at LTM er bygget opp av *skjemaer*, der informasjon blir organisert, strukturert og koblet sammen. Læring foregår når informasjonselementene i AM sees i sammenheng med informasjonen i skjemaene i LTM, som igjen fører til endring av skjemaene. Endringene skjer på ulike måter, blant annet ved at informasjonselementene integreres i skjemaer, utvider skjemaer, fører til koblinger mellom skjemaer og/eller bidrar til å danne nye skjemaer (Mayer, 2009; Sweller et al., 1998).

2.1.2 Kognitiv belastningsteori for forbedring av undervisningsdesign

Innenfor *kognitiv belastningsteori* har man funnet ut at det er begrenset hvor mange informasjonselementer AM maksimalt kan ha samtidig – ulik litteratur viser til mellom tre og syv elementer (Paas & Ayres, 2014; Sweller et al., 1998). Trolig er det også individuelle forskjeller. Jeg velger å illustrere med et eksempel der elever skal lære å multiplisere med divisoren i ligningen $\frac{a}{b} = c$. Oppgaven inneholder flere informasjonselementer, som blant annet variabler, divisjon, multiplikasjon, likhetstegn, ligning, alfanumerisk språk og rekkefølge av operasjoner. Hvis eleven ikke forstår ligninger vil antall informasjonselementer trolig være på, eller over, grensen av hva elevens AM kan behandle. Det er særlig to måter man kan senke belastningen til AM på: (1) Når flere av informasjonselementene er lagret i LTM fra før, økes kapasiteten i AM, og (2) når eleven har god trening i å behandle og lagre lignende elementer blir bearbeidelsesprosessen automatisert og tar mindre plass i AM (Mayer, 2009; Paas & Ayres, 2014; Sweller et al., 1998). Ved å se på eksempelet ovenfor vil en elev som har løst mange ligningsoppgaver tidligere, ikke oppleve at AM må behandle for mange informasjonselementer samtidig.

Split attention effect er en teori under kognitiv belastning, som mener at delt oppmerksomhet vil øke belastningen på AM. Trigonometrioppgaven i Figur **2.2** illustrerer split attention effect. I oppgaven til venstre er to av de nødvendige elementene for å løse oppgaven (AB = 5 og AC = 3) plassert i teksten under figuren, som fører til at elevene må skifte oppmerksomhet mellom figuren og teksten. I oppgaven til høyre er det derimot ikke behov for oppmerksomhetsskifte,



Figur 2.2: To versjoner av samme trigonometrioppgave (inspirert av Sweller et al., 1998, s. 278, 279)

da elementene er plassert i figuren. Sweller et al. (1998) viser til visuelle og auditive oppmerksomhetsskifter. Jeg mener at teorien er overførbar til denne studien da elevene må skifte oppmerksomhet mellom programmering i programmet Scratch og algebra. I så fall må man ta hensyn til at kombinasjonen av fagene trolig vil gi noe økt belastning.

Det er utfordrende å forske på aktiviteten i AM, men vi kan bruke det vi vet om kognitiv belastning til å analysere og forbedre undervisningsdesignet. Sweller et al. (1998) kommer derfor med to nye begreper som brukes i et slikt analysearbeid: *iboende* og *ekstern kognitiv belastning*. Iboende kognitiv belastning viser til belastningen som ligger i informasjonen elevene skal bearbeide og lære – det faglige innholdet. Ekstern kognitiv belastning viser til *hvordan* informasjonen blir presentert gjennom instruksjon, som også påvirker belastningen til elevene. Trigonometrioppgavene i Figur **2.2** har begge samme iboende kognitive belastning, men split attention effect gjør at den eksterne kognitive belastningen er større i oppgaven til venstre enn den til høyre. Ved å analysere undervisningsdesign i lys av både iboende og ekstern kognitiv belastning, kan man identifisere aspekter som potensielt kan gi unødvendige utfordringer for elever, og dermed gi grunnlag for utvikling og forbedring av undervisningsdesignet.

2.2 Generalisering i algebra

I programmering ønsker man å lage et mest mulig generelt program. Derfor er generalisering en viktig del av algoritmisk tenkning. Spørsmålet er om undervisningsdesignet kan legge til rette for at elevenes generalisering i programmering kan overføres til algebra. Det er nødvendig med et nærmere blikk på hva litteraturen sier om generalisering i algebra, og presentere et rammeverk som kan brukes til å analysere elevenes generaliseringer.

I følge Mason (1996) er *generalisering* selve hjertet i matematikken. Å generalisere handler om å legge merke til noe som er felles for flere tilfeller i en sekvens, og anta og begrunne for at det vil gjelde også for nye tilfeller (Radford, 2006). Overgangen fra å se diskrete tilfeller til å generalisere innebærer abstraksjon, noe som kan være utfordrende. Forskning på generalisering er sterkt knyttet til *algebraisk tenkning*, som har ført til at de fleste rammeverkene for generalisering har blitt utarbeidet innenfor dette fagområdet. Det passer bra i denne studien da en forutsetning i undervisningsdesignet var at elevene ikke skulle ha forkunnskaper i algebra.

Vi skal først se på hva algebraisk tenkning er. Deretter presenteres Radfords (2006) rammeverk som viser ulike nivå av generalisering i algebraisk tenkning.

2.2.1 Algebraisk tenkning

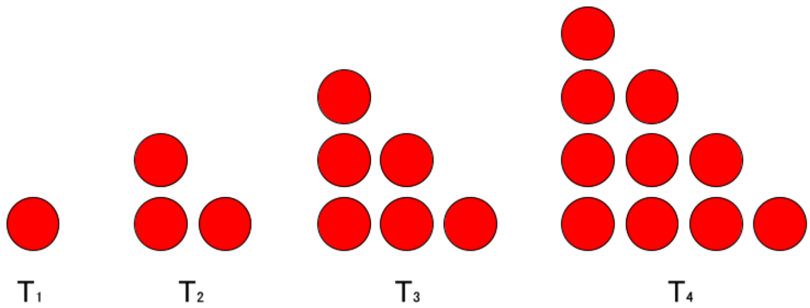
Algebra har tradisjonelt, i Norge og flere andre land, blitt introdusert på ungdomstrinnet (Booth, McGinn, Barbieri & Young, 2017; Utdanningsdirektoratet, 2013). På 90-tallet vokste det frem et nytt forskningsområde, *algebraisk tenkning* (også kalt tidlig algebra eller pre-algebra), hvor man ønsket å implementere generalisering allerede på barnetrinnet, men uten å knytte det direkte til algebra og det algebraiske symbolspråket – et alfanumerisk språk med en gitt syntaks og operasjonsregler (Kieran, 2006).

Algebraisk tenkning er forsøkt definert av mange, men fellestrekkene er at man skal oppdage matematiske relasjoner, endringer, strukturer og det generelle, mellom størrelser. Videre handler det om å uttrykke, begrunne og bevise den generelle sannheten. Algebraisk tenkning er en prosess der den generelle sannheten er målet (Blanton, 2008; Kieran, 2004a; Radford, 2018; Van de Walle, Karp, Bay-Williams, Wray & Brown, 2020). Med tiden har det utviklet seg ulike metoder og underkategorier av algebraisk tenkning, men arbeid med *strukturer*, *mønster* og *figurtall* er metodene og oppgavene som ofte er å se i forskningslitteraturen (blant annet hos Carraher, Martinez & Schliemann, 2008; Chimoni & Pitta-Pantazi, 2017; Kieran, 2004b; Lee, 1996; Mason, 2017; Radford, 2006, 2018; Van de Walle et al., 2020). Generalisering i programmering har tradisjonelt sett ikke blitt sett på i relasjon til algebraisk tenkning, men i denne studien vil elevenes generaliseringer operere i et skjæringspunkt mellom programmering, algebraisk tenkning og algebra. Det eksisterer ikke et rammeverk som passer denne situasjonen perfekt. Jeg har derfor sett til algebraisk tenkning og Radfords (2006) rammeverk for generalisering, som er tilpasset arbeid med figurtall. Rammeverket muliggjør differensiering av elevenes generaliseringsnivåer. For å lettere kunne forstå rammeverkets utgangspunkt er det nødvendig å vise et eksempel med figurtall, og hvilken type kunnskap og arbeid det inkluderer.

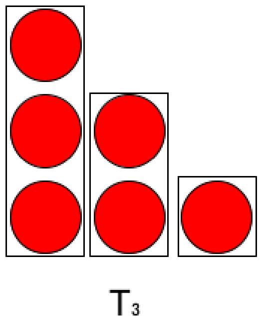
2.2.2 Figurtall

I arbeid med *figurtall* er det to ulike formler elever kan utvikle. Den *rekursive* formelen tar hensyn til det forrige leddet i sekvensen og krever at man legger merke til endringene mellom leddene. En rekursiv formel er dermed en lokal formel, der man kronologisk må regne ut hvert ledd i sekvensen (Mason, 1996). Ved å se på trekanttallene (se Figur **2.3**), vil den neste figuren alltid øke med tallet *n*. På et algebraisk språk kan den rekursive formelen uttrykkes som $T_n = T_{n-1} + n$. Arbeid med rekursive formler krever at elevene abstraherer, for å skille ut hvilke detaljer som er viktig å ta i bruk, for videre å kunne se hva som er felles i mønsterets utvikling. Elevene blir utfordret på å se endringer i et kontinuerlig mønster. Rekursive formler har dog sin begrensning, fordi man alltid må vite størrelsen på figuren i forkant, noe som medfører stort arbeid når man for eksempel skal regne ut figur nummer 20, 104 eller 3000.

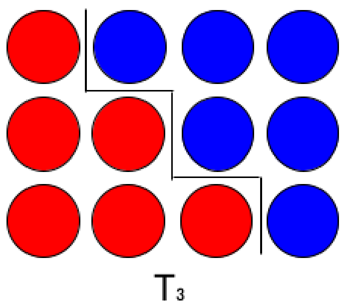
Ved å finne den *eksplisitte formelen* løfter man generaliseringen til et høyere nivå. En eksplisitt formel er generisk, som betyr at den viser det generelle som gjelder for alle figurer (elementer) – man ser det generelle i én figur og begrunner hvorfor det gjelder for alle figurer. Det gjør at



Figur 2.3: Trekanttall, figur T_1 til T_4



Figur 2.4: Trekanttall - eksempel 1



Figur 2.5: Trekanttall - eksempel 2

man kan forestille seg det generelle også utenfor det perseptuelle (det synlige) området (Mason, 1996). I et klasserom kan man oppleve flere ulike forklaringsmodeller – riktige og uriktige, og elegante og mindre elegante. Dette illustreres med eksemplene i Figur **2.4** og Figur **2.5**. I Figur **2.4** ser man et mønster ved å dele opp trekanttallet i kolonner, der man ser at den største kolonnen har samme høyde som n og at resten av kolonnene synker med 1. Mønsteret kan gi følgende utregning: $T_3 = 3 + 2 + 1 = 6$. Den eksplisitte formelen, og den generelle sannheten, blir dermed: $T_n = n + (n - 1) + (n - 2) + \ldots + (n - n)$.

En mer elegant løsning, er å se hvordan katetene i trekanten samsvaret med tallet n. Deretter kan man bruke formelen for areal av trekant som utgangspunkt for å regne ut antall prikker. Problemet er at formelen for areal av trekant innebærer halvering av prikkene i hypotenusen. Ved å snu på resonnementet, og spørre hva som skal til for å halvere, kan formelen videreutvikles: dupliser trekanten, snu den og plasser den slik at det blir et rektangel (Figur **2.5**).

T_3 vil da gi utregningen $T_3 = \frac{3 \cdot 4}{2} = 6$. Det kan lede frem til den mer elegante, eksplisitte og generaliserte formelen: $T_n = \frac{n \cdot (n+1)}{2}$.

Det finnes langt flere begrunnelsesmodeller enn disse to eksemplene. Ved å se på de to eksemplene som er vist, ser man hvordan strukturen på formelen også kan antyde hvilken generaliseringsprosess elevene har vært gjennom. Elever som arbeider med figurtall vil befinne seg på ulike generaliseringsnivåer. Radford (2006) presenterer et rammeverk som kan bidra til å analysere ulike generaliseringsnivåer.

2.2.3 Rammeverk for generalisering

Rammeverket til Radford (2006) er valgt i denne studien, fordi det kategoriserer og nivådeler generaliseringer ut i fra hvordan elever begrunner og utrykker sine generaliseringer. I undervisningsdesignet ble det lagt opp til at elevene først skulle generalisere i programmering og så overføre kunnskapen til algebra, men uten å ha forkunnskaper i algebra.

Radford (2006) studerte elevenes skriftlige arbeid og kommunikasjon seg i mellom mens de arbeidet med figurtall. Resultatet ble et rammeverk som viser en nivådeling av generalisering (se Tabell **2.1**). Det laveste nivået, *naiv induksjon*, er *ikke* generalisering. Naiv induksjon betyr at elever gjetter seg frem til en forklaring. Det skjer ofte ved prøving og feiling, der elevene *ikke* viser tegn til å se noe generelt. De kommer frem til en ubegrunnet, og ikke nødvendigvis riktig, hypotese.

De resterende kategoriene er alle en form for generalisering som betyr at elevene ser noe generelt, som gjelder for flere av eller alle elementene. Generalisering har to underkategorier og nivåer, som defineres ut i fra symbolisering. Ved *aritmetisk generalisering* ser elevene noe generelt, men overfører det kun til noen få diskrete eksempler, der de fortsatt velger å bruke tall. Illustrert med trekanttallene kan det være at elevene ser mønsteret i økningen av trekanttallene, men viser utregning med tall på for eksempel T_5 og T_6. De ser dermed noe generelt i et mønster, men viser kun til enkelttilfeller.

Ved *algebraisk generalisering* viser elevene til *ubestemte størrelser* i stedet for aritmetikk. I tillegg kan de se det generelle utenfor de perseptuelle elementene. Algebraisk generalisering har videre tre underkategorier og nivåer, som skilles etter hvordan de ubestemte størrelsene symboliseres og representeres: *faktuell*, *kontekstuell* og *symbolsk*.

Ved *faktuell algebraisk generalisering* har elevene ikke navngitt de ubestemte størrelsene. Det kan være at de fortsatt bruker tall i utregningene, men viser samtidig (for eksempel gjennom

Tabell 2.1: Rammeverk for generalisering (oversatt fra Radford, 2006)

Naiv induksjon	**Generalisering**			
	Aritmetisk	**Algebraisk**		
		Faktuell	Kontekstuell	Symbolsk
Gjetning Prøving og feiling	Delvis generalisering Aritmetikk	Generalisering Ubestemte størrelser er ikke navngitt.	Generalisering Kontekstuelle navn på ubestemte størrelser	Generalisering Algebraisk språk

verbal eller kroppslig kommunikasjon) evne til å se det generelle utenfor de perseptuelle elementene. Illustrert med trekanttall kan det være, i likhet med forrige eksempel, at de regner ut T_5 og T_6 med tall, men har noe i begrunnelsene sine som gjør at de kan vise til flere elementer i figursekvensen.

Kontekstuell algebraisk generalisering betyr at elevene har navngitt de ubestemte størrelsene, men navnene og/eller begrunnelsene er knyttet til formen eller konteksten. Illustrert med trekanttallene kan det være at de bruker ordene «lengde» og «bredde» i sin formel, som viser til begreper de kjenner fra utregning av areal av rektangel. For eksempel: $T_n = \frac{lengde \cdot bredde}{2}$.

Det øverste nivået er *symbolsk algebraisk generalisering*, som innebærer at elevene bruker det algebraiske symbolspråket for de ubestemte størrelsene. Radford (2006) poengterer at elevene ikke trenger å ha korrekt syntaks for å være i denne kategorien. For eksempel vil formelen $T_n = \frac{n \cdot n+1}{2}$ tilhøre denne kategorien, selv om syntaksen ikke er helt korrekt.

2.3 Variabler

Overgangen fra aritmetikk til algebra oppleves som vanskelig for mange elever, der det settes høyere krav til elevenes abstraksjonsnivå. I elevenes første møte med algebra har vi forskning som viser hvilke utfordringer elever får, hvilke forutsetninger som er viktig for å mestre overgangen til algebra og hvilke vanlige misoppfatninger som oppstår (Booth, Barbieri, Eyer & Paré-Blagoev, 2014; Bush & Karp, 2013; Jupri & Drijvers, 2016; Stacey & MacGregor, 1997). Blant annet ser man at elever får utfordringer med å skape forståelse av likhetstegnet, algebrauttrykk, likninger, funksjoner og variabler. Den teoretiske plattformen for det tredje prinsippet i denne oppgaven, som er knyttet til elevers beskrivelser og forståelse av variabler, tar utgangspunkt i Küchemanns (1981) rammeverk.

Küchemann (1981) var tidlig ute med et rammeverk der elevers tolkninger av bokstaver i algebra ble kategorisert. Til tross for at rammeverket ble publisert i 1981, er det ikke utdatert. De fleste andre som har skrevet om ulike tolkninger av variabler, baserer seg enten på Küchemanns rammeverk, eller har gjort egne induktive studier som i stor grad samsvarer med

Tabell 2.2: Teoretisk rammeverk knyttet til bruk av bokstaver i algebra (oversatt fra Küchemann, 1981)

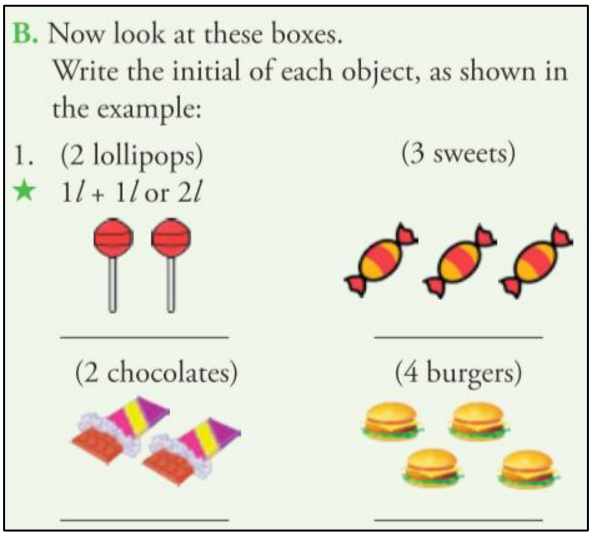
Kategori	Beskrivelse	Validering
Ignorert	Bokstaven blir ignorert	Ukorrekte tolkninger
Tillagt verdi	Bokstaven blir tillagt en verdi med feilaktig grunn	
Objekt	Bokstaven står for et objekt	
Ukjent	Bokstaven står for én ukjent verdi	Korrekte algebraiske tolkninger
Generalisert størrelse	Bokstaven står for en generell størrelse	
Variabel	Bokstaven står for en ubestemt kvantitet, som kan variere	

deler av Küchemanns funn (Booth et al., 2017; Ely & Adams, 2012; Küchemann, 1981; Philipp, 1992; Stacey & MacGregor, 1997; Tennant & Colloff, 2014; Usiskin, 1988). Det som skiller Küchemanns rammeverk fra de andre, er at der andre har analysert variabelens rolle i ulike matematiske kontekster, analyserte Küchemann elevenes tolkninger av bokstavene. Resultatet ble seks kategorier (se Tabell **2.2**), der tre er algebraisk ukorrekte, og tre er algebraiske korrekte. Rammeverket er valgt til denne studien fordi den ser på elevers beskrivelser, og dermed tolkninger, av variablene. Det betyr at det er mulig for elevene i denne studien å ha både korrekte og ukorrekte tolkninger.

I elevenes møte med algebra er det tilfeller der elevene ikke forstår bokstavenes hensikt, og velger å ikke forholde seg til dem. De velger å *ignorere* dem når de gjør utregninger: enten ved å fjerne bokstavene fra et oppsatt uttrykk, velge å ikke ha med bokstaver når man setter opp et uttrykk, eller ved å ignorere dem i utregning – slik vi kan se i følgende feilaktige forenkling: $2a + 4b + 6c = 12abc$ (Booth et al., 2017; Küchemann, 1981).

Booth et al. (2017) viser til at bruk av krypteringsoppgaver, der man skal bytte ut tall med bokstaver ut ifra hvilken posisjon de har i alfabetet, kan føre til misoppfatninger knyttet til variabler. Det er et eksempel på elevers tolkning som fører til at bokstavene får en feilaktig *tillagt verdi*. I dette tilfellet ignorerer de ikke bokstavene, men de bytter ut bokstavene med en verdi på feilaktig grunnlag (Booth et al., 2017; Küchemann, 1981; Stacey & MacGregor, 1997).

Tennant og Colloff (2014) viser hvordan oppgaver i lærebøker kan bidra til misoppfatninger av hva en variabel er, og viser eksempler på hvorfor elever kan bruke bokstavene som et *objekt* – i stedet for en verdi. Eksempelet i Figur **2.6** har de hentet fra en lærebok i India, som ble utgitt i 2010. I oppgavene skal elevene skrive et algebrauttrykk som representerer innholdet på bildet. Problemet er at elevene antagelig vil tolke bokstavene som forkortelser for lollipop, sweet, chocolate eller burger, og ikke oppfatte at bokstavene står for en verdi. Det vil skape negative ringvirkninger når de skal jobbe videre med algebra. Med et slikt grunnlag, hvordan kan de da senere skape mening om for eksempel $b \cdot l$ eller l^b? Feilaktig bruk av, og tolkning av, bokstaver



Figur 2.6: Eksempeloppgave (Tennant & Colloff, 2014, s. 40)

i algebra som et *objekt* (i stedet for en verdi) går igjen i mye forskningslitteratur (blant annet Bush & Karp, 2013; Ely & Adams, 2012; Küchemann, 1981; Philipp, 1992; Stacey & MacGregor, 1997; Tennant & Colloff, 2014).

Til nå er det presentert tre kategorier med algebraisk ukorrekte tolkninger: *ignorert*, *tillagt verdi* og *objekt*. Årsaker til elevers ukorrekte tolkninger kan være at de baserer tolkningen på andre erfaringer enn fra matematikken. Fra før er de vant med å bruke bokstaver i andre sammenhenger, men ikke som et symbol på verdier (Stacey & MacGregor, 1997, s. 110). *Ukjent*, *generalisert størrelse* og *variabel* (se Tabell **2.2**) er de tre neste kategoriene i rammeverket. Alle er algebraisk korrekte, og har stigende abstraksjonsnivå (Küchemann, 1981). Samtidig er hver og en av dem fullverdige i gitte algebraiske kontekster, der det ikke er mulig å gå videre til neste abstraksjonsnivå. Eksempler på dette blir gitt under.

Ukjent vil si når verdien bak bokstaven er ukjent, men det samtidig er gitt at den kun kan ha én verdi. Tolkningen av bokstaven i denne kategorien er helt riktig når den opptrer i ligninger, som av denne typen: $2x + 5 = 20 - x$. I dette tilfellet kan x *kun* være 5, og bare 5 (Ely & Adams, 2012; Küchemann, 1981; Philipp, 1992).

Når bokstaven opptrer som *generalisert størrelse*, klarer elevene å se at bokstaven står for mer enn bare én størrelse (Küchemann, 1981). Generalisert størrelse opptrer algebraisk riktig i følgende eksempel: $c + d = 10, der\ c\ og\ d\ er\ positive\ tall.$

Kategorien *variabel* er veldig lik generalisert størrelse, men legger vekt på at bokstaven ses som en verdi som kan *endres*, og har uendelige mange muligheter. Derfor er variabel på det høyeste abstraksjonsnivået i dette rammeverket. Variabel er for eksempel riktig kategori for bokstavene x og y i funksjoner – for eksempel: $y = 2x + 3$. Navnet på kategorien er en utfordring, fordi det er vanlig å bruke begrepet *variabel* for bruk av bokstaver i algebra. I resten av oppgaven er det forsøkt å gi tilstrekkelig informasjon og kontekst, slik at man til enhver tid forstår hvilken betydning det refereres til.

Det er krevende for elever å utvikle sin tolkning av bokstaver i algebra fra en ukorrekt tolkning til kategorien ukjent. I tillegg er det krevende med den videre utviklingen fra å tolke bokstavene som ukjent til variabel. Ely og Adams (2012) beskriver de to overgangene som utfordrende, og noe som vil kreve stor omstilling for elevene.

I denne studien møter elever på *variabler* i både algebra og programmering. Begrepet har ulik definisjon og bruk innenfor de to fagområdene. Der variabler står for en bestemt eller ubestemt størrelse i algebra, er formålet med variabel i programmering å kunne lagre verdi eller tekst, for deretter å bruke, hente og/eller manipulere dem underveis i programmet. Et eksempel er poengberegning i et spill: poengsummen vil være lagret i en variabel, som må endres (manipuleres) når spilleren får eller mister poeng. Bråting og Kilhamn (2020) mener ulikheten i begrepets definisjon og bruk kan skape utfordringer og eventuelle misoppfatninger i algebra for elever. Samtidig mente Usiskin (1988) at bruk av variabler i programmering likevel kan opptre i tilfeller som sammenfaller med blant annet de korrekte kategoriene vi ser i Küchemanns rammeverk. I denne studien møtte elevene på variabler i programmering, der de endres, og dermed sammenfaller med Küchemanns kategori *variabel* i algebra.

3 Metodisk arkitektur og utvikling av undervisningsdesign

Hensikten med denne studien var å gå i dybden på matematikkarbeid i én klasse for å identifisere flere faktorer som påvirker utprøvingen av et undervisningsdesign. For å ha mulighet til å tolke virkeligheten undervisningsdesignet er plassert i, ble kvalitativ forskningsmetode valgt. Studien ble derfor gjennomført i sin virkelige kontekst – i klasserommet, der teorier ble brukt til hjelp for å tolke virkeligheten (Postholm, 2010).

Som nevnt tidligere har denne studien plassert seg i sosialkonstruktivistisk forskningsteori, som muliggjør analyser av omgivelser, interaksjoner og kontekst for å kunne si noe om hvorvidt programmering kan bidra til å utvikle algebraiske konseptuelle ideer. Studien måtte kunne belyse hvilke faktorer som påvirket elevenes utfordringer og utvikling. Teori og praksis ble knyttet sammen ved å bruke designutvikling (Design-Based Research) som forskningsmetode – det ble utviklet et undervisningsdesign, som både møtte de teoretiske prinsippene, samtidig som det skulle passe i den lokale konteksten i utprøvingen.

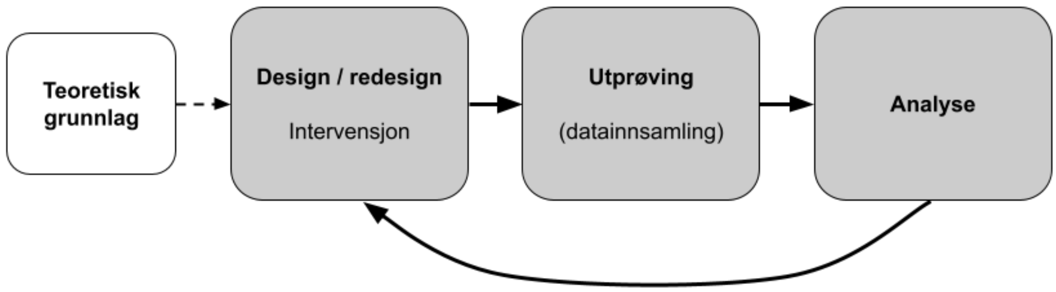
Metodekapittelet er delt i tre hoveddeler, der første del presenterer forskningsrammen for denne studien – designutvikling. Først blir designutvikling presentert generelt, før Sandovals (2004, 2014) rammeverk blir presentert i detalj, og første hoveddel avsluttes med hvordan rammeverket er tilpasset denne undersøkelsen. Andre hoveddel redegjør for utviklingen og gjennomføringen av undervisningsdesignet. Undervisningsdesignet blir presentert i lys av teori og kontekst. Siste hoveddel formidler hvordan forskningen er utført, ved å redegjøre for datainnsamling, analyse, forskningens troverdighet og etiske valg.

3.1 Designutvikling

Utviklingen av designutvikling hadde som mål å knytte tettere bånd mellom forskning og læreres praksis. Ved å utvikle undervisningsdesign og forske systematisk på utprøvingen, vil resultatene kunne utvikle teori, samt være relevante for forbedring av praksis (Anderson & Shattuck, 2012; The Design-Based Research Collective, 2003). Med min bakgrunn som både lærer og masterstudent var det et overordnet ønske at denne studien skulle gi kunnskap og innsikt til både lærerutdannere, didaktikere, forskere og lærere. Designutvikling legger til rette for dette, samt at det passer til denne studiens rammer. Denne studien tar i bruk Sandovals (2004, 2014) metodiske rammeverk som er noe kompleks. Derfor forklares først hva som kjennetegner designutvikling generelt, før det gis en av forklaring av Sandovals metodiske rammeverk. Til slutt vises hvilke tilpasninger som var nødvendig å gjøre i denne studien.

3.1.1 Generelle kjennetegn i designutvikling

Designutvikling har store variasjoner, men har samtidig noen viktige og sentrale felles aspekter. Store deler av prosessen foregår i et samarbeid mellom forsker og lærer. Videre baserer



Figur 3.1: Designutviklingens tre-delte prosess, basert på beskrivelser fra Anderson og Shattuck (2012), og The Design-Based Research Collective (2003)

undervisningsdesignet seg på et teorigrunnlag, som ofte leder ut i teoretiske prinsipper. Deretter følger en tredelt prosess (se Figur **3.1**), som består av design/redesign, utprøving og analyse, som gjerne gjentas flere ganger (Anderson & Shattuck, 2012; The Design-Based Research Collective, 2003).

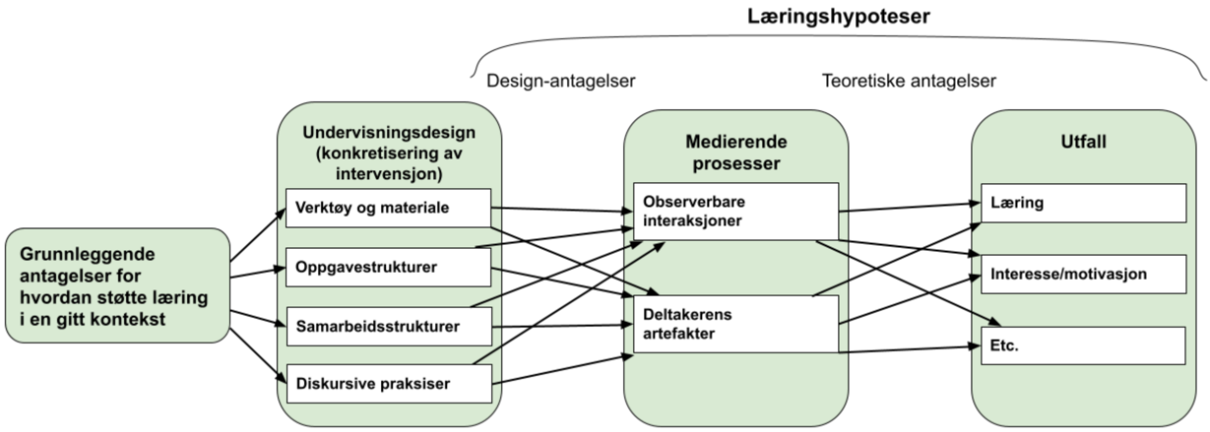
Design/redesign består av at lærer og forsker sammen planlegger en intervensjon i undervisningsdesignet. Intervensjonen består ofte av flere mindre konkrete tiltak som har en spesifikk hensikt – et ønske om å oppnå et mål, for eksempel et læringsmål. Intervensjonen er utviklet i samspill med eksisterende teori og den lokale konteksten. I andre del av prosessen gjennomfører læreren utprøvingen av designet i sin klasse, mens forskeren samler inn datamateriale. Tredje og siste del består av forskerens analyse av datamaterialet. Resultatene er grunnlag for endring og/eller forbedring av designet (redesign), som eventuelt følges av en ny gjentagelse av prosessen. Samspillet mellom empiri og lokal kontekst er til stede i alle de tre leddene. Det er ikke gitt hvor mange gjentagelser man bør ha i designutvikling, men designet blir gradvis forbedret for hver gang, samt at forskningens generaliseringer vil få mer støtte med flere utprøvinger i ulike kontekster (Anderson & Shattuck, 2012; The Design-Based Research Collective, 2003).

Forskeren bør ha en pragmatisk metodisk tilnærming. Det er viktig å kunne belyse de kontekstuelle faktorene, og derfor må valg av metode og forskningsverktøy begrunnes – ikke bare i teorien, men også i de lokale kontekstuelle rammene (Anderson & Shattuck, 2012).

Designutvikling har blitt kritisert for at resultatene mangler generell gyldighet ved å være for tilknyttet den lokale konteksten. Sandoval (2004, 2014) har svart på kritikken ved å beskrive et metodisk rammeverk som tydeliggjør relasjonen mellom teori og praksis, som presenteres i neste del.

3.1.2 Metodisk rammeverk for designutvikling

Sandoval (2004) mente at utvikling og forbedring av undervisningsdesign var en teoretisk aktivitet. Derfor innførte han begrepet *læringshypoteser* (samlet oversettelse av *embodied conjectures* (Sandoval, 2004) og *mapping conjectures* (Sandoval, 2014), men som i stor grad betyr det samme, men sett fra ulike perspektiv).



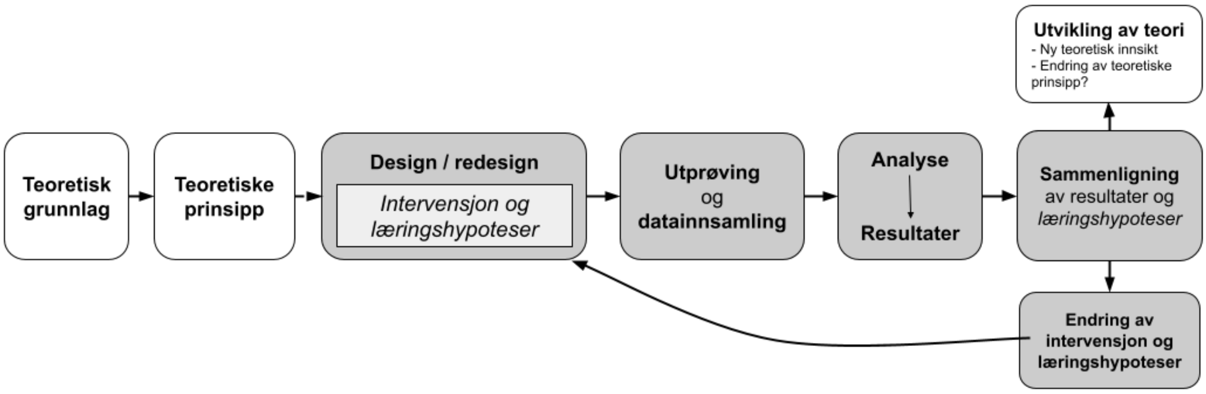
Figur 3.2: Læringshypoteser (oversatt fra Sandoval, 2014)

I følge Sandoval blir intervensjonen *konkretisert* gjennom et eller flere tiltak i undervisningsdesignets fire strukturer (se Figur **3.2**). Med *læringshypoteser* forsøker man å forutse en årsakssammenheng mellom tiltakene og utfallet. Læringshypotesene er basert på teori og lokal kontekst. Basert på tiltakene (designet) gjør man antagelser (design-antagelser) om hvilke *medierende prosesser* som vil skje. Medierende prosesser er observerbare og vil komme til uttrykk i interaksjoner og/eller deltakernes artefakter (for eksempel elevprodukt). Deretter antar man en videre årsakssammenheng (teoretiske antagelser) frem til *utfall*. Utfall, som for eksempel læring, er vanskeligere å observere enn medierende prosesser.

En forenklet illustrasjon av læringshypoteser, kan ses i et design der man ønsker at elevene skal lære egenskapene til en likesidet trekant. Et slikt design vil ha flere tiltak i intervensjonen, der vi ser for oss at én av dem er å bruke passer som verktøy. Deretter kan design-antagelsene anta en årsakssammenheng mellom bruk av passer, og medierende prosesser der elevene mestrer å bruke passer til å konstruere en likesidet trekant. Videre kan de teoretiske antagelsene være at elevene *forstår* sammenhengen mellom radiusen passeren gir, og sidelengdene i trekanten.

Bruk av læringshypoteser vil utvide prosessen fra tre til fire ledd, der siste ledd er en sammenligning av læringshypotesene og resultatene fra utprøvingen (se Figur **3.3**). Fordi læringshypotesene har tydelige årsakssammenhenger (pilene i Figur **3.2**), vil en sammenligning med resultatene gjøre det mulig å spore undervisningsdesignets effekt (positive og negative) tilbake til tiltakene, som igjen gjør det mulig å endre og forbedre undervisningsdesignet. Sammenligningen vil også gi bedre antagelser ved en eventuell ny utprøving – hvor man nok en gang kan forbedre og endre læringshypotesene. I tillegg hevdet Sandoval (2004) at bruk av læringshypoteser vil gi undersøkelsen et teoretisk fundament som muliggjør utvikling av teori.

Undersøkelsens teoretiske fundament ble også tydeliggjort i Sandovals (2004) rammeverk ved inkludering av *teoretiske prinsipp* (se Figur **3.3**). Teoretiske prinsipp er grunnlaget for utformingen av undervisningsdesignet og forteller, basert på teori, hva man ønsker å oppnå med et undervisningsdesign.



Figur 3.3: Sandovals (2004) rammeverk (modell konstruert ut i fra Sandovals beskrivelser)

Oppsummert hevdet Sandoval (2004, 2014) at bruk av *teoretiske prinsipp* og *læringshypoteser* tydeliggjorde designutvikling som en teoretisk aktivitet, som muliggjør generalisering fra resultatene, på tross av resultatenes nære relasjon til lokal kontekst.

3.1.3 Designutvikling i denne studien

Denne undersøkelsen har brukt Sandovals rammeverk, men med noen unntak. For det første var det et ønske fra elevenes faglærer at forskeren selv gjennomførte undervisningen, på grunn av programmeringstekniske ferdigheter. Faglærer var med som hjelpelærer i timen. For det andre ble det kun gjennomført én utprøving. Som nevnt er gjentagelser med flere utprøvinger et tiltak i designutvikling for å gi et bedre generaliseringsgrunnlag (Anderson & Shattuck, 2012). I denne undersøkelsen var det krevende å finne enda en klasse til ny utprøving av følgende årsaker: (1) Det var få klasser som hadde nødvendig forkunnskap i programmering til å gjennomføre oppgaven. (2) Undervisningsdesignet skulle være en inngang til temaet algebra, dermed var det nødvendig at elevene ikke hadde lært algebra tidligere. (3) En masteroppgave har sin tidsbegrensning, som gjør at utprøvingen måtte skje i et gitt tidsrom. Dermed var det ikke mulig å gjennomføre en ny utprøving.

Det faktum at det bare ble gjort én enkelt utprøving var dermed årsak til valget av Sandovals teoretiske rammeverk, for å gi studien en sterkere tilknytning til teori. De generaliseringene som er blitt gjort i denne studien må dermed sees i sammenheng med disse faktorene.

3.2 Undervisningsdesignets konstruksjon og utprøving

Dette delkapittelet starter med å presentere undervisningsdesignets konstruksjon og hvordan det ble prøvd ut i en 8. klasse, før det redegjøres og gis begrunnelse for valg av intervensjonen, med tilhørende tiltak, i designet.

3.2.1 Undervisningsdesignets konstruksjon og utprøving

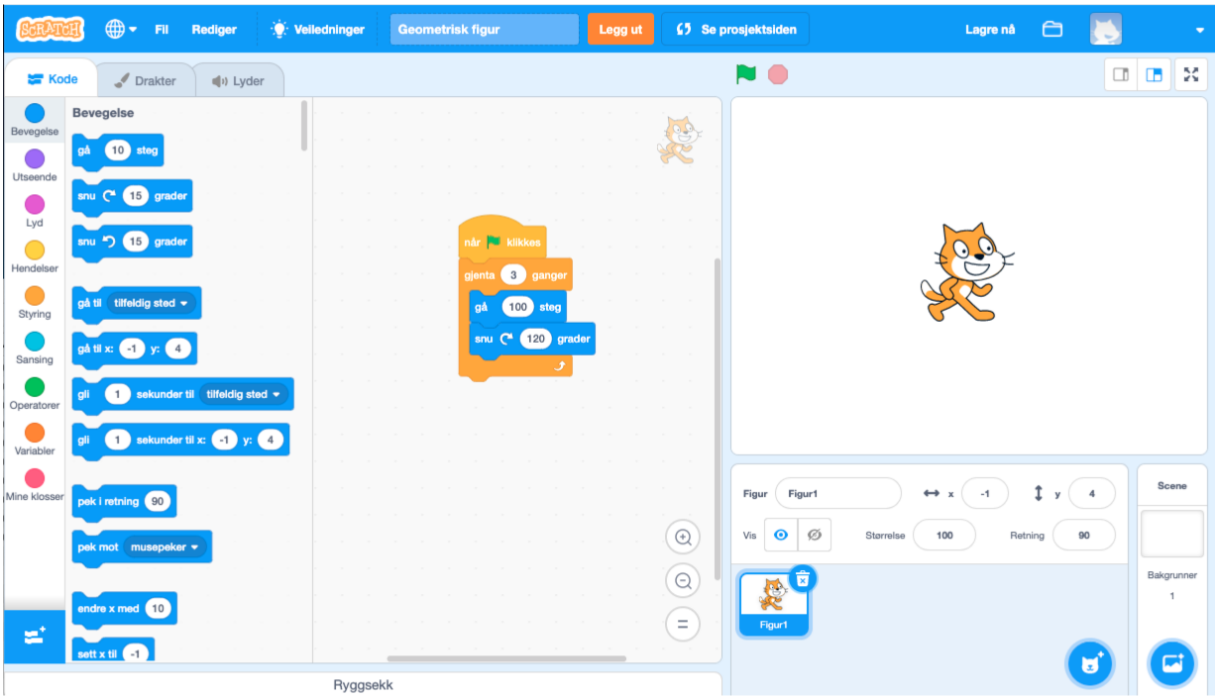
Utprøvingen foregikk i en 8. klasse bestående av 50 elever, der det ble samlet inn datamateriale fra 31 elever som hadde gitt samtykke. Klassen ble til vanlig delt inn i to undervisningsgrupper med hver sine undervisningsrom. Matematikktimene var på 90 minutter, der 15 minutter måtte settes av til oppstart og avslutning av datainnsamling. Utprøvingen ble gjennomført i begge undervisningsgruppene rett etter hverandre med 15 minutters pause mellom. Samtlige elever hadde tilgang på hver sin datamaskin, som de var godt vant med å bruke til skolearbeid, men designets samarbeidsstrukturer krevde at to og to elever arbeidet sammen på én datamaskin. Elevenes faglærer var med som hjelpelærer mens forskeren gjennomførte undervisningen.

Videre følger beskrivelse av undervisningsdesignet som først inneholder krav til elevenes forkunnskaper. Deretter presenteres designets kronologi i tre deler: del én er oppstart av undervisningsøkta, del to består av et oppgavesett elevene jobber med på egenhånd, og del tre er avslutning og oppsummering av undervisningsøkta.

3.2.1.1 Elevenes nødvendige forkunnskaper

Undervisningsdesignet krevde visse forkunnskaper i både matematikk og programmering. I matematikk måtte elevene ha kunnskap om rektangel, hva omkrets er og hvordan man regner ut omkrets på et rektangel. I programmering måtte elevene kunne lage, teste og forbedre enkle algoritmer, bruke løkker, opprette og endre variabler.

Elevene måtte også ha god kjennskap til programmeringsmiljøet Scratch (https://scratch.mit.edu/). Scratch er designet for å være en inngang til å lære seg programmering (Resnick et al., 2009). Det baserer seg på et programmeringsspråk som kalles blokkprogrammering, som er en forenkling av den tradisjonelle tekstbaserte programmeringen, og som heller har en grafisk tilnærming. Blokkprogrammering handler om å dra, flytte og koble sammen ulike ferdige blokker for å lage algoritmer. Algoritmer beskrives av Utdanningsdirektoratet (2019c) som et sett med trinnvise instruksjoner. Figur **3.4** viser et skjermbilde av Scratch med et eksempel på en algoritme som vil få katten til å bevege seg i en likesidet trekant. Selv om blokkprogrammering oppleves enklere enn tekstbasert programmering, inneholder det mange av de samme



Figur 3.4: Skjermbilde av Scratch når man programmerer

grunnprinsippene, som for eksempel løkker, tester, vilkår, variabler og funksjoner (Sáez-López, Román-González & Vázquez-Cano, 2016). For å unngå unødvendig begrepsforvirring omtales variabler i Scratch (programmering) heretter som «Scratch-variabel» i denne teksten. Derimot møtte elevene kun begrepet «variabel» i sine elevoppgaver, uten ulike differensieringer, noe som kommer til syne i beskrivelsen av elevoppgavene nedenfor.

Elevene i denne studien hadde de nødvendige forkunnskapene ved at de på forhånd hadde gjennomgått grunnleggende opplæring i programmering, gjennomført av faglærer i matematikk. De hadde i forkant gjennomført fire økter med ulike oppgaver der de hadde lært grunnleggende begreper som algoritme, løkke, betingelser, tester og Scratch-variabler, fått Scratch-figuren (for eksempel katten i Figur **3.4**) til å bevege seg, tegnet geometriske figurer og laget et enkelt spill der man får eller mister poeng ved å berøre andre Scratch-figurer.

3.2.1.2 Del 1: Oppstart av undervisningsøkta

Økta startet først med en kort introduksjon, før de ble delt inn i små grupper på to (med unntak av én gruppe som ble tre elever). De diskuterte i par hva en variabel er, og målet var at de skulle knytte det til det de hadde lært i forrige time om variabler i programmering. Noen av elevenes ideer ble tatt opp i plenum før læreren til slutt forklarte hvordan elevene skulle jobbe med oppgavesettet.

3.2.1.3 Del 2: Elevoppgaver

Elevene jobbet med oppgavene i par, der én programmerte, mens den andre instruerte. Det stod i oppgaveheftet (se vedlegg 1 for hele oppgavesettet) når de skulle bytte roller underveis.

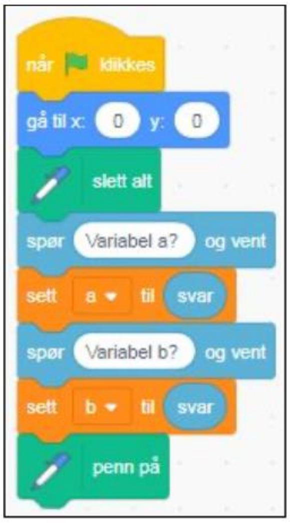
Oppgaveheftet bestod av tilsammen syv oppgaver. Oppgave 1, 5, 6 og 7 er kort forklart, da de ikke var viktige for analysearbeidet i denne studien. I tillegg var det få elever som rakk å komme noe særlig lengre enn oppgave 4. Oppgave 2, 3 og 4 inneholdt elevarbeid som var av særlig interesse med tanke på studiens forskningsspørsmål og de teoretiske prinsippene, og vil derfor få en grundigere beskrivelse.

Oppgave 1 hadde til hensikt å få elevene til å hente frem tidligere kunnskap om rektangel, rektangelets egenskaper og utregning av omkrets på rektangel. I tillegg fikk elevene en enkel oppstart til generalisering ved at de gjorde så få målinger som mulig, når de skulle måle og regne ut omkretsen av et tilfeldig rektangel de selv hadde tegnet.

Oppgave 2 (Figur **3.5**) ba elevene opprette Scratch-variablene a og b, og lot brukeren få skrive inn verdier for a og b hver gang programmet ble kjørt. Deretter ble elevene bedt om å lage en algoritme som tegnet et rektangel der sidelengdene var lik bokstavene a og b. Det førte til at rektangelet endret sidelengdene hver gang programmet ble kjørt.

I oppgave 3 (Figur **3.6**) ble elevene bedt om å regne ut omkretsen til rektangelet de programmerte i oppgave 2. Elevene måtte bruke operatorer (blokker med regnearter) som skulle settes inn i hverandre for å lage et algebrauttrykk som gav riktig utregning. Utfordringen var at rektangelet varierte i størrelse, noe som skapte et behov for å bruke Scratch-variablene a og b i utregningen. Navngivningen av Scratch-variablene til a og b var bevisst for å tilnærme seg et algebraisk symbolspråk i programmering for å legge til rette for en lettere overgang til algebra. Oppgaven hadde flere løsninger, der Figur **3.7** viser én mulighet.

Oppgave 2: Tegn et rektangel i Scratch (Figur 1)



Bilde 1: startkode

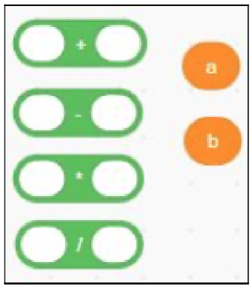
Bilde 2: Bevegelse

a) Opprett et nytt program i Scratch som du gir navnet "**Figurer med variabler**".
b) Velg kategori ***Variabler*** og lag to variabler med navn **a** og **b**.
c) Kopier koden i ***bilde 1***.
d) Bruk kombinasjoner av blokkene i ***bilde 2*** til å tegne et rektangel. Du kan også bruke **løkker** for å gjøre det lettere. Det er lov å bruke flere av hver kode.
e) Kjør programmet og forklar for hverandre hva utfallet blir.
f) Kjør programmet flere ganger der dere endrer på verdiene for ***variabel a*** og ***variabel b***. Forklar hverandre hva utfallet blir.

Figur 3.5: Oppgave 2

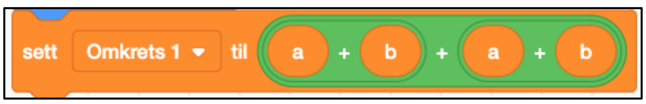
Oppgave 3: Regn ut omkrets

a) Bytt roller!
b) Lag en ny variabel med navn **Omkrets 1**.
c) Velg kategori ***Operatorer*** og ta utgangspunkt i kodene du ser på ***Bilde 3***. Lag en algoritme som regner ut omkretsen til rektangelet som lagres i variabelen **Omkrets 1**.
d) Test om dere har gjort det riktig:
   i) Kjør programmet flere ganger, men endre på verdiene på variablene **a** og **b**.
   ii) Bruk kalkulator for å sjekke om programmet dere har laget i Scratch er riktig.

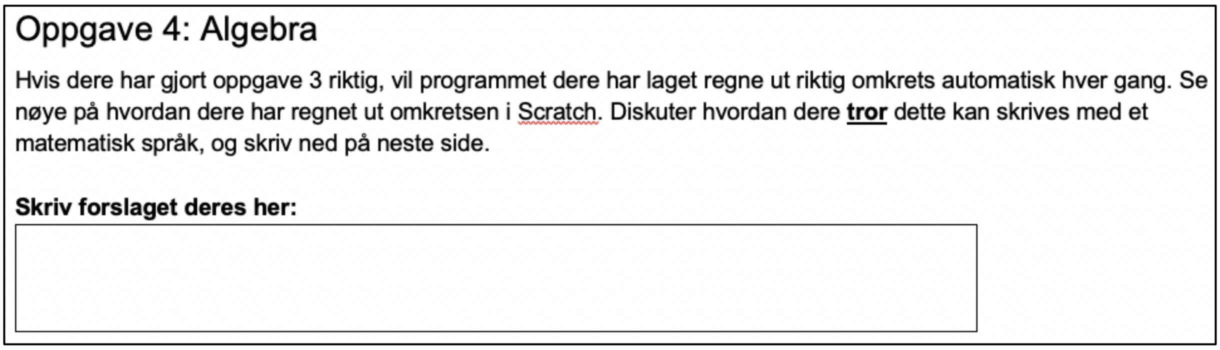


Bilde 3: Operatorer

Figur 3.6: Oppgave 3



Figur 3.7: Eksempel på besvarelse av oppgave 3



Oppgave 4: Algebra

Hvis dere har gjort oppgave 3 riktig, vil programmet dere har laget regne ut riktig omkrets automatisk hver gang. Se nøye på hvordan dere har regnet ut omkretsen i Scratch. Diskuter hvordan dere **tror** dette kan skrives med et matematisk språk, og skriv ned på neste side.

Skriv forslaget deres her:

Figur 3.8: Oppgave 4

Oppgave 4 (Figur **3.8**) ba elevene foreslå hvordan utregningen av omkrets fra oppgave 3, som inkluderte bruk av Scratch-variabler og dermed generalisering, kunne skrives på et matematisk språk – altså en overgang fra Scratch til algebraisk symbolspråk. Begrepet «matematisk språk» ble valgt, i stedet for algebraisk språk, fordi elevene ikke hadde forkunnskaper i algebra. Derfor var det også vektlagt at elevene skulle skrive det de trodde var riktig.

Oppgave 5, 6 og 7 repeterte aktivitetene tegning av figur, utregning av omkrets og overføring til algebra i hver sin oppgave, men ved å møte andre figurer og algebrauttrykk (vedlegg 1). Oppgave 5 hadde et rektangel med omkretsen $3a + 2b + 3a + 2b$. Oppgave 6 ba elevene lage en ny figur med samme omkrets som rektangelet i oppgave 5: $6b + 4b$. Oppgave 7 gav elevene større frihet, der de kunne tegne en vilkårlig figur, men fortsatt med å bruke variablene a og b. De skulle kladde figuren på et ruteark før de tegnet figuren i Scratch, regnet ut omkretsen og foreslo hvordan omkretsen kunne skrives på et matematisk språk.

3.2.1.4 Del 4: Avslutning og oppsummering

Uansett hvor langt elevene kom i oppgavene, ble det brukt fem til syv minutter til oppsummering. Oppsummeringen belyste ulike besvarelser på oppgave 3 og 4. Videre diskuterte de i par hvorfor de hadde brukt bokstavene a og b, og ikke tall, når de regnet ut omkrets. Til slutt oppsummerte læreren forskjellen på variabel i programmering og algebra.

3.2.2 Intervensjonen og læringshypoteser i undervisningsdesignet

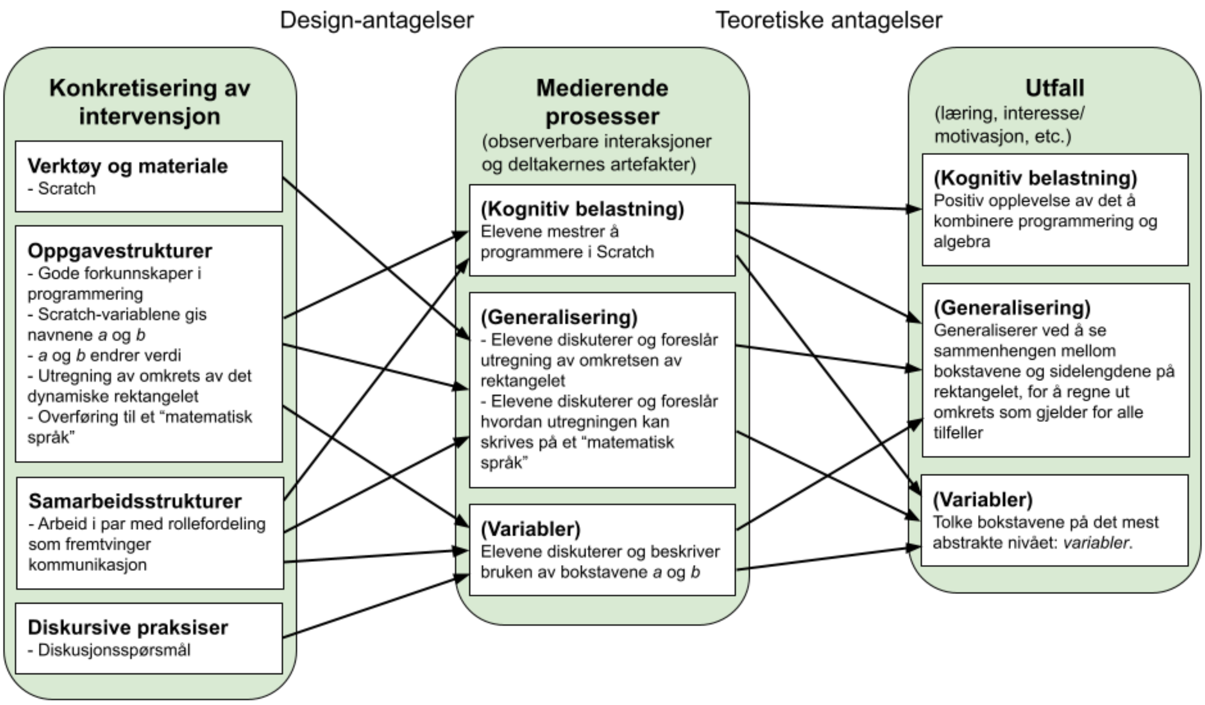
Dette delkapitlet vil se nærmere på intervensjonen med tilhørende tiltak og læringshypoteser som inngår i undervisningsdesignet.

Figur **3.9** oppsummerer og sammenfatter designets tiltak og læringshypoteser, der pilene viser antagelsene om hvilke årsakssammenhenger det er mellom tiltakene, medierende prosesser og utfall. Medierende prosesser inneholder observerbare interaksjoner og deltakernes artefakter som forventes i lys av, og strukturert etter, de tre teoretiske prinsippene. Utfall er strukturert på samme måte, men omhandler læring og motivasjon. Videre er det redegjort for tiltakene, strukturert etter de tre teoretiske prinsippene om kognitiv belastning, generalisering og variabler.

3.2.2.1 Kognitiv belastning

Det må først klargjøres hva i undervisningsdesignet det er som regnes som *iboende kognitiv belastning* og *ekstern kognitiv belastning*. Både det algebraiske innholdet (generalisering og variabler) og overføringen mellom Scratch og algebra inngår i iboende kognitiv belastning. Ettersom elevene ikke har forkunnskaper i algebra, og fordi overgangen mellom Scratch og algebra kan være krevende (Schanzer et al., 2015), vil den iboende kognitive belastningen trolig være relativt høy. Kombinasjonen av fagområdene programmering og algebra vil gi elevene flere utfordringer samtidig: de må løse programmeringsrelaterte problemer i Scratch, gjøre generelle utregninger, samt foreslår hvordan utregningene kan skrives på et «matematisk språk» (algebra). Sett i lys av split attention effect, vil elevene oppleve å måtte dele sin oppmerksomhet mellom problemer som tilhører to ulike fagområder, som kan gi ekstra belastning av arbeidsminnet. Derfor er det viktig med tiltak som kan begrense den eksterne kognitive belastningen.

En viktig forutsetning i denne studien var derfor at elevene måtte ha gode forkunnskaper om programmering i Scratch, slik at de skulle slippe å lære seg to fagområder samtidig. Forutsetningen er så viktig for designet at jeg velger å definere det som et tiltak. I tillegg arbeidet de i par slik at de sammen kunne utfylle hverandre i møte med eventuelle utfordringer. Det var dermed en forventning om at elevene ikke møtte på store programmeringsrelaterte utfordringer. Det ble antatt at den eksterne kognitive belastningen ble lav nok til at den totale kognitive belastningen i arbeidsminnet ble overkommelig for elevene. Videre ble det antatt at den lave eksterne kognitive belastningen la til rette for at elevene kunne mestre å utvikle konseptuell algebraisk forståelse, samt gi elevene en positiv opplevelse og økt motivasjon i timen.



Figur 3.9: Designets intervensjon og tiltak, og læringshypoteser

3.2.2.2 Generalisering

Radfords (2006) rammeverk om generalisering plasserer *symbolsk algebraisk generalisering* på øverste generaliseringsnivå – som innebærer å se det generelle utenfor det perseptuelle området og bruke et algebraisk symbolspråk for å navngi de ubestemte størrelsene. Via undervisningsdesignet forsøkte en å skape et behov for å generalisere, samt bruke algebraiske symboler, slik at elevene kunne nå nivået symbolsk algebraisk generalisering.

Første tiltak var i oppgave 2, der elevene måtte opprette Scratch-variabler med navnene *a* og *b*. Det gjorde at et algebraisk symbolspråk ble implementert i Scratch, som ville gjøre overgangen fra algebrauttrykket i Scratch til et matematisk språk enklere. Neste tiltak var at verdiene for *a* og *b* varierte for hver gang Scratch-programmet ble kjørt, og det kom til syne i det dynamiske rektangelet. Tredje tiltak kom i oppgave 3, der elevene måtte regne ut omkretsen av det dynamiske rektangelet. Det dynamiske rektangelet skapte et behov for at utregningen måtte utrykke en generalitet: utregningen gjelder for *alle* tilfeller av rektangelet. Hvis elevene har gitt Scratch-variablene navnene *a* og *b*, vil besvarelsen på oppgave 3 vise symbolsk algebraisk generalisering. Fjerde tiltak var når elevene ble utfordret i oppgave 4 til å overføre utregningen i oppgave 3 i Scratch til et matematisk språk. Femte tiltak var å la elevene arbeide i par slik at de kunne diskutere utfordringene de møtte på seg i mellom. Det var forventet at elevene fikk til å se noe generelt når de skulle regne ut omkretsen av rektangelet. På grunn av at elevene ikke hadde forkunnskaper i algebra, var det samtidig forventet at de kom til å vise ulike generaliseringsnivåer.

3.2.2.3 Variabler

Det var tre tiltak som forsøkte å legge til rette for at elevene skulle oppnå det høyeste nivået, *variabel*, i Küchemanns (1981) rammeverk. De to første tiltakene er sammenfallende med to nevnt under generalisering: Scratch-variablene navngis *a* og *b*, og verdiene for *a* og *b* endres. Det siste tiltaket sørget for at elevene i oppsummeringen av timen diskuterte hvorfor de måtte bruke bokstavene *a* og *b*, i stedet for tall, i utregningen av omkretsen. Det ble antatt at elevene observerte hvordan endringer i verdiene til *a* og *b* kom til syne i det dynamiske rektangelet, og i omkretsen (etter at den var regnet ut). Videre var det forventet at elevene, gjennom diskusjon i par, kunne beskrive bokstavene *a* og *b* som noe som endres, varierer, eller lignende, og dermed oppnå det høyeste, og mest abstrakte nivået i Küchemanns rammeverk: *variabel*. I så fall ville elevene ha utviklet sin konseptuelle algebraiske forståelse.

3.3 Forskningens valg og utførelse

I studien var det behov for å få god dybdekunnskap gjennom å få en helhetlig forståelse av én spesifikk kontekst, derfor ble kvalitativ forskningsmetode valgt (Grønmo, 2021). Forskeren var selv et forskningsinstrument, som førte til store forventninger til forskerens objektivitet, transparens og kvalitet i sitt forskningsarbeid.

Dette delkapittelet redegjør for de metodiske valgene. Først presenteres hvordan datamaterialet er samlet inn og hvilke typer datakilder det er brukt i undersøkelsen. Deretter redegjøres det for hvordan datamaterialet er bearbeidet gjennom transkripsjon og tematisk analyse. Avslutningsvis diskuteres forskningens troverdighet, og hvilke etiske problemstillinger det er tatt hensyn til i undersøkelsen.

3.3.1 Utvalg av elever

Studiens- og undervisningsdesignets krav og rammer skapte, som nevnt i kapittel 3.1.3, begrensninger for utvalget. Det var ønskelig å gjennomføre utprøvingen på en 8. klasse, som hadde visse forkunnskaper i programmering, og som ikke tidligere hadde hatt undervisning i algebra.

Det ble brukt etablerte kontakter for å finne en klasse som passet kriteriene for studien. Klassens faglærer i matematikk var positiv til studien, men ønsket at forskeren selv skulle gjennomføre utprøvingen av undervisningsdesignet.

Elevene i denne studien gikk på en relativt stor ungdomsskole i en norsk storby. Skolen var i stor grad etnisk og sosialt homogen, men med noen variasjoner. Foreldregruppa var generelt ressurssterke. Samtlige realfagslærere på skolen hadde godkjent undervisningskompetanse i sine fag. På nasjonale prøver i regning lå skolens, og klassens, gjennomsnitt i nærheten av det nasjonale gjennomsnittet, men med store variasjoner innad i klassen, som normalt er.

Av 50 elever (28 gutter og 22 jenter) gav 31 elever (18 gutter og 13 jenter) samtykke om datainnsamling. De 31 elevene var fordelt på 15 mindre grupper med gruppestørrelse på 2 elever, unntatt en gruppe som var 3 elever. Sammensetningene av gruppene var tilfeldig ved at de ble satt sammen ut ifra deres opprinnelige plassering i rommet.

3.3.2 Datainnsamling og -materiale

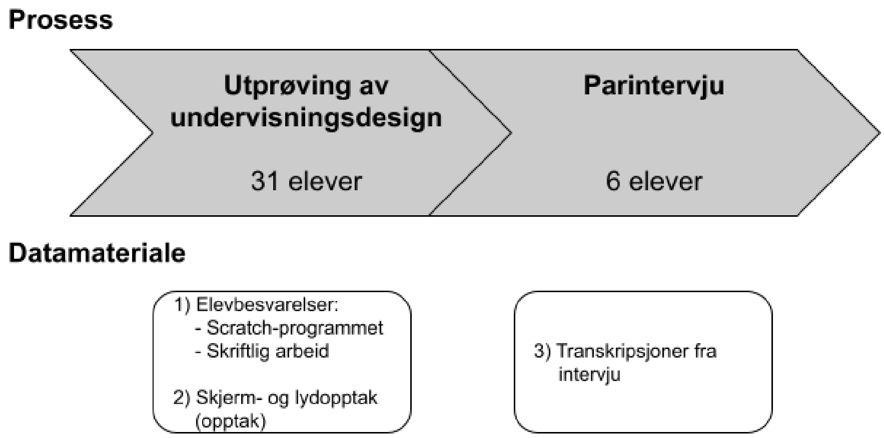
For å gi et best mulig helhetsbilde ble det valgt å bruke datatriangulering med datamateriale fra tre ulike kilder (se Figur **3.10**). Under selve utprøving av undervisningsdesignet ble det produsert to ulike typer datamateriale, som ble levert til forsker i etterkant av økten: elevbesvarelser og skjerm- og lydopptak. På samme dag, kort tid etter utprøvingen, ble det gjennomført tre parintervju.

3.3.2.1 Elevbesvarelser

Elevbesvarelsene bestod av elevenes endelige program i Scratch, og elevenes skriftlige arbeid i oppgaveheftet – løsningen på oppgave 1, og elevenes forslag til hvordan omkretsen kan skrives på et matematisk språk i oppgave 4, 5, 6 og 7. Elevbesvarelsene gav innblikk i elevenes endelige produkt og svar, men tok samtidig ikke hensyn til deres prosess. Selv om oppgaveheftet bestod av 7 oppgaver (vedlegg 1), var det oppgave 1 til 4 som var av interesse i forskningen, da oppgave 5, 6 og 7 var gjentagelse av de samme prosessene hos elevene.

3.3.2.2 Skjerm- og lydopptak

Det ble tatt et kombinert opptak av skjermen og elevenes samtaler gjennom datamaskinens mikrofon (i det følgende omtalt som opptak). Selv om elevene hadde tilgang på hver sin datamaskin, arbeidet de på én datamaskin. Tekniske problemer førte til at kun 11 av 15 grupper



Figur 3.10: Datamateriale

fikk fullført opptaket. Til forskjell fra elevbesvarelsene gav opptakene et godt innblikk i elevenes naturlige sosiale samtale, diskusjoner, beskrivelser og prosess (Cohen, Manion & Morrison, 2011, s. 530). For å kunne utfordre elevenes beskrivelser ytterligere, ble det i tillegg gjennomført et påfølgende intervju av tre av gruppene, samme dag som utprøvingen.

3.3.2.3 Intervju

Intervju produserer en stor mengde datamateriale, som gjør det nødvendig å begrense antall deltakere (Kvale & Brinkmann, 2015). Derfor ble det gjennomført tre gruppeintervju, der gruppene var de samme som i undervisningsøkta. Tilsammen ble det intervjuet seks elever. Elevenes faglærer fikk ansvaret for utvelgelsen av elever til intervju med følgende kriterier: elever som er trygge nok til å fortelle om sine opplevelser, og som samlet har et heterogent ferdighetsnivå i både matematikk og programmering. Intervjuene hadde hver en varighet på cirka 30 minutter. Det var forskeren selv som gjennomførte intervjuene, og de ble tatt opp med lydopptaker. På de tre gruppene som ble intervjuet var det mulig å gjennomføre kildetriangulering mellom intervju, opptak og besvarelser.

Det var nødvendig å være bevisst flere aspekter ved intervjuet. En viktig del av intervjuet inneholdt elevers beskrivelser av bokstaver i ulike algebrauttrykk, der elevene ikke hadde kunnskap om Küchemanns kategorier. Det gjorde intervjuet utfordrende for både elever og intervjueren. For å minske det allerede eksisterende skjeve maktforholdet mellom forsker og elever, var det viktig for meg i undervisningsøkta å jobbe med å skape gode relasjoner til elevene, ved å snakke mye med dem, samt å bruke god tid til å forberede dem på intervjuet (Kvale & Brinkmann, 2015).

Hensikten med å gjennomføre intervju, var å utfordre elevenes generaliseringer, beskrivelser av variabler og opplevelser av undervisningen. Tilnærmingen var hovedsakelig deduktiv, der kategoriene var gitt på forhånd, samtidig som den var induktiv, med åpenhet for eventuelle nye kategorier eller andre relevante opplevelser. En intervjuguide var utgangspunktet, men målet var å få flyt i samtalen og spesielt få tak i elevenes beskrivelser – intervjuformen var dermed semi-strukturert (Kvale & Brinkmann, 2015). Intervjuguiden kan sees i sin helhet i vedlegg 2, og hadde følgende hovedtemaer:

1) Briefing om intervjuet før opptaket starter
2) Elevenes opplevelse av undervisningsopplegget
3) Elevenes forkunnskaper og opplevelser med å jobbe med programmering
4) Elevenes forkunnskaper om variabler
5) Elevenes beskrivelser av algebrauttrykk, utregningen de har gjort og i hvor stor grad de mestrer å abstrahere og generalisere
6) Motivasjon og arbeid med to fag samtidig
7) Refleksjon over intervjuet (hva lærte de underveis i intervjuet)
8) Elevenes avsluttende spørsmål og kommentarer
9) Debriefing etter at opptaket er avsluttet

3.3.3 Transkripsjon

I denne studien ble intervjuene transkribert i sin helhet. I analysen av opptakene var det nødvendig å høre elevenes samtale i sammenheng med deres arbeid på dataskjermen, som førte til at de ikke ble transkribert i sin helhet. For å gjøre analyse- og kodearbeidet lettere, ble alle opptakene sett gjennom, mens det ble notert stikkord. Relevante enkeltsamtaler ble transkribert.

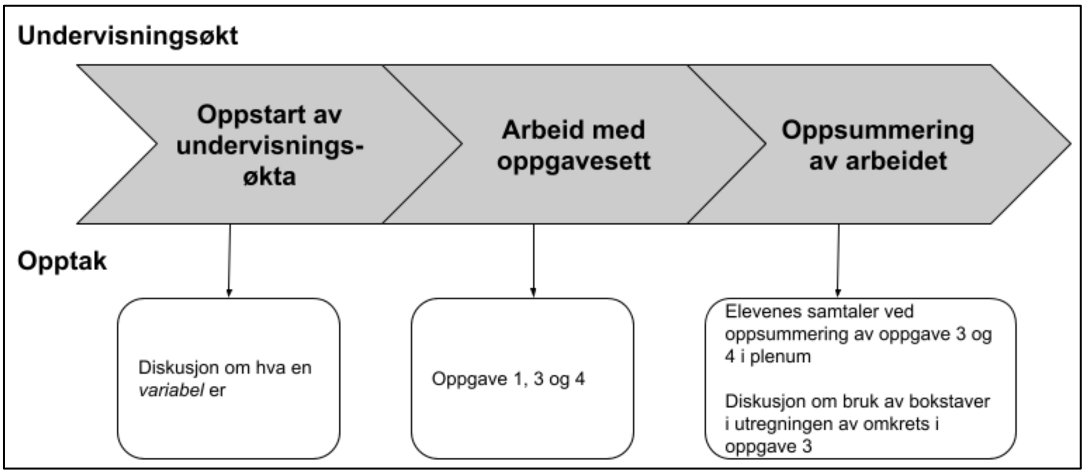
Det ble gjort mange tolkninger i transkripsjonsarbeidet, da mye datamateriale forsvinner i transformasjonen fra en samtale til tekst (Cohen et al., 2011, s. 426; Kvale & Brinkmann, 2015, s. 125-126). Transkripsjonsteksten taler ofte for seg selv i denne sammenhengen, men det ble likevel notert ned noen detaljer – hvem intervjuers spørsmål er rettet mot, om lyder som «mhm» og «mm» har bekreftende eller avkreftende tonefall, tenkepauser som ikke fremkommer i konteksten, og naturlige pauser som ikke kommer frem av seg selv i en tekst.

I intervjuet snakket deltakerne ofte rundt eksempler fra elevbesvarelsene og algebraeksempler som var notert på et ark. Notatene fra intervjuet ble tatt vare på for å kunne få riktige detaljer i transkripsjonen. Av de delene av opptakene som ble transkribert var det flere ganger nødvendig å notere ned sammenhengen mellom det elevene sa og hva elevene gjorde på datamaskinen.

3.3.4 Tematisk analyse

I denne studien valgte jeg å identifisere, analysere og finne mønster i datamaterialet ved å bruke tematisk analyse (Castleberry & Nolen, 2018). Fordi kategoriene allerede var gitt ble det valgt en deduktiv tilnærming, som gav en detaljert analyse av spesifikke aspekter i datamaterialet (Braun & Clarke, 2006). Samtidig var det ønskelig med en induktiv tilnærming for å være åpen for nye kategorier, motbevis og andre relevante opplevelser. Analysearbeidet kombinerte derfor deduktiv og induktiv tilnærming.

Inspirert av Braun og Clarkes (2006) forslag til analyseprosess, bestod analysen av fire faser: bli kjent med datamaterialet, kode, tolke og skrive rapport. Ved deduktiv tilnærming kan det bli forvirring omkring begrepene *kode* og *kategori* - koder er egentlig mindre bestanddeler av kategorier, men ved bruk av eksisterende teoretiske rammeverk er det kun kategoriene vi har innblikk i. Paradoksalt nok er det i den deduktive analyseprosessen likevel naturlig å bruke verbet *kode*. Kodeprosessen ble gjort i analyseverktøyet NVivo (QSR International, 2021), som bidro til mer oversikt i tolkningsprosessen. Prosessen bestod av å gå frem og tilbake mellom fasene gjentatte ganger, som var spesielt nødvendig for å sikre en fullstendig analyse av alle tre



Figur 3.11: Kriterier for analyse av opptak

rammeverkene (kognitiv belastning, generalisering og variabler). Samtidig valgte jeg å ha en induktiv tilnærming ved å være åpen for informasjon som kan skape nye koder eller kategorier, eller informasjon som ikke stemmer med de øvrige resultatene. Dette resulterte likevel ikke i noen nye kategorier.

Kategoriene var allerede gitt i de teoretiske rammeverkene, og de som ble funnet i datamaterialet var situasjoner som kan knyttes til iboende- og ekstern kognitiv belastning, ulike grader av generalisering og tolkningene ukjent og variabel. Det ble ikke opprettet noen nye kategorier, men det ble gjennomført egne gjennomlesninger som hadde til hensikt å finne interne motsigelser og motbevis, med mål om å få en helhetlig forståelse (Postholm, 2010, s. 105).

Opptakene inneholdt mye informasjon og var derfor en krevende datakilde. Innholdet var komplekst, med mange detaljer og mye overflødig informasjon, som var krevende å ta stilling til i analysearbeidet. Det var viktig med en bevissthet rundt hva opptakene inkluderer og hva de faktisk utelater (Cohen et al., 2011). For å begrense datamengden etablerte jeg forhåndsbestemte kriterier for hva jeg ville se etter (Cohen et al., 2011). Det var ønskelig å se på de hendelsene som inneholdt elevenes prosess og utfordringer i forbindelse med generalisering og bruk av bokstaver, samt deres refleksjoner, diskusjoner og beskrivelser i oppstarten og i oppsummeringen av undervisningsøkta. Figur **3.11** viser hvilke viktige hendelser som ble valgt og som var av interesse for forskningen.

3.3.5 Forskningens troverdighet

For å sikre god troverdighet i forskningen er det i denne studien gjennomført seks velbegrunnede grep som kan kategoriseres i en eller flere av Gubas fire kriterier for god og troverdig forskning: *kredibilitet, overførbarhet, reliabilitet* og *bekreftbarhet*. Neste avsnitt forklarer de fire kriteriene før hver av de seks grepene blir redegjort for.

Kredibilitet handler om at leseren kan være trygg på at den helhetlige beskrivelsen har støtte i alle deltakernes virkeligheter. *Overførbarhet* vil si å gjennomføre og presentere forskningen slik at den er overførbar til andre kontekster. *Reliabilitet* handler om bevissthet om hvordan forskningsinstrumentet (som er forskeren selv) er en faktor for resultatenes variasjon. Man må bruke ulike overlappende metoder, synliggjøre forskerens ståsted og forskningsprosessen, og legge fram nok informasjon til at det er mulig for andre å gjennomføre replikasjoner av studien. *Bekreftbarhet* handler om bevissthet om forskerens nærhet til forskningsfeltet, og forskerens bias (Guba, 1981).

Det er brukt to typer *triangulering* (1), både ved bruk av tre ulike teorier og tre ulike datakilder, som bidrar til forskningens kredibilitet, reliabilitet og bekreftbarhet. Det teoretiske grunnlaget, spesielt ved de matematikkdidaktiske temaene algebra og variabler, bidrar til distansering av den lokale og kontekstuelle forskningen. I tillegg bidrar teori om kognitiv belastning til god triangulering ved å få et helt annet perspektiv på undervisningsdesignet. Datakildene gir et bredt bilde av konteksten, der intervjuet var den datakilden som kunne åpne for størst kontekstuell påvirkning. Intervjuet er en formell setting med kontekstuelle rammer. Selv om intervjueren ytret ønske om at elevene var så ærlige som mulig, må man likevel se på elevenes besvarelser med et kritisk blikk. Det kan være at elevene har et ønske om å bidra til forskningen, med å gi svar de tror intervjueren ønsker å få. Derfor stilles det ekstra store krav til intervjuerens spørsmål, slik at de er verdinøytralt (Kvale & Brinkmann, 2015). Første halvdel av intervjuet var i stor grad verdinøytral, men halvveis ut i intervjuet ble elevene forklart kategoriene *ukjent* og *variabel*. Intervjueren påvirket elevene for mye og elevenes beskrivelser kan ikke bli brukt til å evaluere undervisningsdesignet. Derfor er kun første halvdel av intervjuene brukt som datamateriale i forskningen.

Det andre og tredje grepet er *tykke beskrivelser* (2) og *transparens* (3) som har styrket forskningens overførbarhet, reliabilitet og bekreftbarhet. Oppgaven presenterer tykke beskrivelser av de kontekstuelle faktorene ved utvikling av designet, utprøvingen og forskningen. Teksten er også transparent på to nivåer: leseren kan følge argumentasjonen gjennom datamateriale, resultat og konklusjon, samt forskningsprosessens detaljer.

Det fjerde grepet var tydeliggjøring av, og bevissthet rundt, egen bias og meg selv som forskningsinstrument (4), som har bidratt til reliabilitet og bekreftbarhet i forskningen. Forskningen har blitt påvirket av forskerens bias (Postholm, 2010), og en bevissthet rundt dette var spesielt viktig i analysen, der datamaterialet skulle tolkes og kortes ned. Samtidig har bias også vært en styrke, fordi det har gitt forskeren større innsikt i de kontekstuelle faktorene. Som Anderson og Shattuck (2012, s. 18) sier: «A certain wisdom is needed to walk this narrow line between objectivity and bias».

Det femte grepet var *utvalget* (5) av skole, klasse og elever som har bidratt til overførbarhet til andre lignende kontekster. Dette grepet var, som tidligere beskrevet, noe begrenset av forskningens rammer, men fra beskrivelsen av klassen under kapittel 3.3.1, er det likevel grunnlag for å si at utvalget har relevans for andre 8.-klasser i Norge. Utvalget muliggjør dermed overførbarhet til andre kontekster.

Det siste grepet har vært direkte rettet mot analyseprosessen (6) og har bidratt til kredibilitet i forskningen. Postholm (2010) skriver at bruk av teori vil «hjelpe alle kvalitative forskere som undersøker kjente farvann, til å distansere seg og dermed betrakte og analysere de ulike handlingsprosessene som utspiller seg» (s. 100). Den deduktive tilnærmingen i analysen har bidratt til distansering. I tillegg har jeg i analysen vært opptatt av å studere datamaterialets detaljer, lese over gjentatte ganger, og se etter motbevis og interne motsigelser. Målet har vært å fremstille resultater som er i tråd med detaljene i datamaterialet. Et eksempel på interne motsigelser finnes for eksempel mellom de ulike datakildene: Ella og Lars (gruppe 3) har gitt to svar på oppgave 4: $x + x + 40 + 40$ og $(a + b) + (a + b)$. Intervjuet av Ella og Lars bygget på intervjuerens antagelse om at de hadde diskutert $(a + b) + (a + b)$. I stedet viser opptaket at de kun diskuterte $x + x + 40 + 40$ når de arbeidet med oppgave 4. Trolig skrev de på $(a + b) + (a + b)$ når eksemplet ble vist på tavla under oppsummeringen av undervisningsøkta.

Samtidig med de seks grepene er det viktig å være bevisst på hvilke svakheter forskningen har. For det første er det flere kontekstuelle faktorer som har påvirket resultatene. For eksempel vil elevene alltid bli påvirket av endringer i hverdagen. Undervisningsøkta bestod av mange elementer som var annerledes enn i en vanlig matematikktime for elevene. De hadde en annen lærer, det ble gjennomført datainnsamling, og de visste at de var med på et forskningsarbeid. Alt dette kan ha bidratt til at elevene hadde en annen motivasjon og oppførsel enn til vanlig (Cohen et al., 2011). For det andre var det begrenset med klasser som passet utvalgskriteriene, men det er grunnlag for å hevde at utvalget muliggjør overførbarhet til andre kontekster. For det tredje ble det ikke gjennomført flere utprøvinger av undervisningsdesignet, og det er dermed ikke prøvd ut i andre kontekster. Som nevnt var det årsaken til valget av Sandovals rammeverk, for å få et tydeligere teoretisk fundament på forskningen, slik at generalisering og utvikling av teoretisk innsikt likevel ble mulig. Guba (1981) trekker frem «member check» som et av de viktigste tiltakene man kan gjøre i kvalitativ forskning. Det ble ansett som lite hensiktsmessig å gjennomføre, da deltakerne er i alderen 12-13 år. Trolig ville de fått lite ut av å lese forskningsteksten, da de har lite erfaring å vurdere det opp mot. I tillegg ville de trolig ha glemt en stor del av opplevelsene sine innen en gjennomlesning ville latt seg gjøre 5 måneder senere.

Oppsummert har forskningen, sett i lys av Gubas (1981) rammeverk, noen svakheter, men gjennomføringen av de seks grepene ovenfor forsøker å veie opp for manglene. Derfor vil forskningen likevel ha relevans, både for lærerutdannere, didaktikere, forskere og praksisfeltet.

3.3.6 Etiske problemstillinger

Deltakerne i forskning har rett til ivaretagelse av privatliv og personvern, gjennom å opprettholde anonymitet og konfidensialitet. Det skal ikke være mulig å identifisere deltakerne fra forskningsteksten. Derfor ble det gitt pseudonymer til alle deltakerne i denne studien. Krav til konfidensialitet ble også overholdt i denne studien, gjennom at opplysninger gitt i forskningen ikke skal gjøre det mulig å spore opp skolen, klassen eller elevene (Cohen et al., 2011). Fordi undertegnede selv også var lærer hadde det vært nærliggende å bruke egne elever til forskningen, men av hensyn til forskerens habilitet, og deltakernes rett til privatliv og personvern, ble det ikke tilfelle.

Privatliv og personvern handler også om at deltakerne selv skal kunne velge om de ønsker å ta del i forskningen eller ikke. Det ble det tatt hensyn til ved at deltakerne signerte et frivillig skriftlig informert samtykke, som inneholdt tilstrekkelig informasjon om studien og deltakernes rettigheter (Postholm, 2010). Anonymisert samtykkeskjema er vedlagt oppgaven som vedlegg nummer 3. Deltakerne i studien var ungdom i 12-13-årsalderen. For ungdom under 16 år er det krav om at foresatte gir samtykke, men deltakerne har samtidig selv rett til å bli hørt (Postholm, 2010). Derfor var det foresatte som gav skriftlig samtykke, samtidig som elevene gav et muntlig samtykke. Foresatte og elever var informert om at de hadde mulighet til å trekke hele eller deler av samtykket før, under og etter datainnsamlingen. Deltakerne i intervjuet ble spesielt informert i den foregående briefingen om at de hadde rett til å utelate hele eller deler av opplysningene de kom med.

4 Resultater

I dette kapittelet presenteres resultatene fra analysen av datamaterialet i studien. Presentasjonen er strukturert etter de tre teoretiske prinsippene. Første delkapittel presenterer resultatene som omhandler kognitiv belastning, andre delkapittel tar for seg elevenes generalisering, og siste delkapittel presenterer elevenes arbeid med variabler, i dette tilfellet elevenes beskrivelser av bruk av bokstaver i utregningen av omkrets. Innenfor hvert av områdene vil det henvises til de ulike datakildene for å bruke kilde-triangulering.

4.1 Høy ekstern kognitiv belastning

Elevene i denne studien fikk sitt første møte med algebra gjennom å bruke programmering. I intervjuene etter utprøvingen kom det frem at samtlige elever hadde hatt *litt* programmering på barneskolen, i tillegg til den grunnleggende opplæringen rett i forkant av utprøvingen.

Kildetrianguleringen viste noen motstridigheter, men det endelige resultatet er likevel tydelig. I intervjuene ble elevene spurt direkte om hvordan de syntes det var å jobbe med programmering og matematikk samtidig. Alle stilte seg positive til å kombinere de to fagområdene. Samtidig fremkommer det gjennom andre spørsmål i intervjuet at flere opplevde programmeringsdelen som utfordrende. Opptakene støtter sistnevnte observasjon: flere grupper hadde utfordringer med aktivitetene som var knyttet til programmering, og den eksterne kognitive belastningen i undervisningsdesignet var dermed høyere enn forventet.

Et eksempel på motstridigheter i kildene kan vi se hos Ella og Lars (gruppe 3), som svarer ulikt på to spørsmål i intervjuet. De svarer følgende på spørsmål om hvordan det var å koble matematikk og programmering (Joakim er intervjuer):

(Intervju)

Ella:	Jeg synes det har vært lettere å gjort det sammen, for da kan man skjønne det litt bedre, på en måte. Kan skjønne litt hva begge to betyr og hva de har til felles.
Joakim:	Ja, mhm... Har du noen konkrete eksempler på hva du følte var felles, her da?
Ella:	Mmm... Det var jo litt den der pluss-operatorene der, for eksempel. Det er jo som en parentes, bare litt annerledes.

Ellas konkretisering gjør utsagnet troverdig. Samtidig kommer det frem en motstridende opplevelse i et annet spørsmål, der de skal fortelle om sin umiddelbare opplevelse av økta:

(Intervju)

Lars:	Den var litt vanskelig, men...
Ella:	Jeg synes det var gøy jeg, men jeg synes det var litt vanskelig å forstå, men det var gøy når man fikk det til.
Joakim:	Ja...
Ella:	På en måte.
Joakim:	Vanskelig å forstå, hva var vanskelig, da?
Ella:	Nei... Jeg bare skjønte ikke helt hvordan alt fungerer, på en måte, med programmering.

Uttalelsene gir et mer nyansert bilde – Ella opplevde økta som «gøy», men viser samtidig til at hun opplevde programmeringen som utfordrende, noe opptakene støtter. I opptaket kan vi se at Ella og Lars måtte løse to problemer samtidig da de konstruerte et algebrauttrykk for omkretsen av rektangelet (oppgave 3). Det ene problemet (algebraproblemet) går under iboende kognitiv belastning: å konstruere utregningen ved bruk av Scratch-variabler. Det andre problemet (programmeringsproblemet) går under ekstern kognitiv belastning: å skape algoritmen i Scratch, kombinere operatorer og lagre utregningen i Scratch-variabelen *Omkrets 1*. Når Ella og Lars startet på oppgave 3, startet de med å diskutere algebraproblemet:

(Opptak)

Lars:	Da tar jeg a, da blir det $a + a$. Men skal vi bruke noen flere?
Ella:	Nei.
Lars:	Okey, $a + a$. Også $b + b$.

De ser sammenhenger, og tenker helt riktig. Likevel stoppet det opp her, fordi de ikke visste hvordan de skulle angripe programmeringsproblemet. Videre leser de oppgaveteksten på nytt og prøver litt til, før Lars uttrykker sin observasjon: «Det blir jo ikke lagra inni omkrets», der han sikter til at de ikke vet hvordan de skal lagre utregningen i Scratch-variabelen «Omkrets 1». Programmeringsproblemet var utfordrende, og de ble nødt til å sette algebraproblemet på pause. De brukte en del tid på å løse programmeringsproblemet, før de til slutt kunne fortsette på algebraproblemet igjen. Etter mye tid løste de oppgave 3 med følgende utregning i Scratch: $b + a + b + a$.

Det er ikke mulig for oss å analysere hvordan Ella og Lars behandler informasjonen i arbeidsminnet sitt, men vi kan observere hva de diskuterer og jobber med. Det blir tydelig at de ikke hadde nok forkunnskaper i programmering, noe som forstyrret deres arbeid med algebraproblemet. De manglet nok kunnskap om å bruke og kombinere operatorer, og hvordan de skulle lagre utregningen i Scratch-variabelen «Omkrets 1». Dermed ble den eksterne kognitive belastningen i undervisningsdesignet for høy for Ella og Lars.

Ella og Lars sine utfordringer gjenspeiles i stor grad hos de andre elevene i klassen, men med tre unntak: (A) Det fremkommer i både opptak og intervju at Mia (gruppe 2) opplevde hele undervisningsøkten som krevende. I intervjuet uttrykte hun at hun *verken* har interesse for programmering, eller opplever å mestre matematikkfaget. (B) På den andre siden var derimot Gustav og Axel (gruppe 5) de eneste som mestret å løse både algebra- og programmeringsproblemet samtidig. Opptaket viser at de hadde nok kunnskap om det programmeringstekniske i oppgaven til å klare programmeringsproblemet uten utfordringer. (C) På to grupper var det ikke datamateriale nok til å analysere utfordringene. Dea og Gry (gruppe 11) brukte lang tid, og rakk derfor aldri å starte på oppgave 3. Video fra gruppa ble av tekniske årsaker ikke lagret og det er derfor vanskelig å si noe om årsaken til tidsbruken. Liam og Magne (gruppe 12) fikk verken levert skjermopptak eller besvarelse, som gjør det umulig å si noe om deres progresjon.

Oppsummert kan vi si at de fleste elevene manglet viktige forkunnskaper i programmering som trengtes i oppgave 3, og dermed medførte undervisningsdesignet for høy ekstern kognitiv belastning.

4.2 Elevenes generaliseringer

Resultatene viser at samtlige elever har generalisert, men på ulike nivå. Elevenes diskusjoner og besvarelser på oppgave 1, 3 og 4 har vært grunnlag for analyse av elevenes generaliseringer. I oppgave 1 fikk elevene en forsmak på generalisering, mens i oppgave 3 måtte de lage en utregning for omkretsen av et rektangel der sidekantene varierte, ved å bruke Scratch-variablene *a* og *b*. I oppgave 4 skulle de foreslå hvordan utregningen av omkretsen kunne skrives på et matematisk språk – uten forkunnskaper i algebra. Resultatene er strukturert etter oppgave 1, 3 og 4, og presenterer elevenes generaliseringer i hver av dem. Tabell **4.1** viser en oversikt over gruppenes besvarelser på oppgave 1, 3 og 4.

4.2.1 Oppgave 1 – forsmak på generalisering

I oppgave 1 har samtlige grupper vist abstraksjon og generalisering på et enkelt nivå. Alle gruppene har enten markert riktig hvilke sider som er like lange, og/eller skrevet at de tok færre målinger enn fire.

Ved å se nærmere på Tabell **4.1** kommer det frem at fire grupper ikke har markert hvilke sider som er like lange. Gruppe 12 har markert kun to av fire sider. Videre ser vi at gruppe 9, 10 og 11 tok fire målinger. Gruppe 14 tegnet et kvadrat og skrev dermed at de gjorde én måling, mens resten skrev at de gjorde to målinger. Årsakene til at så mange grupper hadde små feil her kan være flere. Opptakene viser at samtlige grupper viet denne deloppgaven lite tid og oppmerksomhet. I opptakene fremkommer det at enkelte ikke har forstått oppgaven, har oversett oppgaven eller har vært usikker på hvordan de skulle markere sidene. Likevel viser alle gruppene enkel generalisering, i minst én av deloppgavene.

Tabell 4.1: Elevenes besvarelser på oppgave 1, 3 og 4.
R = riktig, DR = delvis riktig, IM = ikke markert

Gruppe	Oppgave 1 Markert like lange sider	 Antall målinger	Oppgave 3	Oppgave 4
1	R	2	sett Omkrets 1 til 2 * a + b	$2 \cdot (a+b) = omkrets\ 1$
2	IM	2	endre Omkrets 1 med a + b * 2	$a + b \cdot 2 = Omkrets$
3	IM	2	endre Omkrets 1 med b + a + b + a + Omkrets 1	$x + x + 40 + 40 = omkrets$ endret deretter til $(a+b) + (a+b) = O$
4	R	2	sett Omkrets 1 til a + b + a + b	$X + Y + X + Y = O$
5	R	2	sett omkrets1 til A + B * 2	$(a+b) \cdot 2$
6	R	2	sett Omkrets 1 til a * 2 + b * 2	$A \cdot 2 + B \cdot 2$
7	R	2	sett Omkrets 1 til a + b + a + b	$a \cdot 2 + b \cdot 2$
8	R	2	sett Omkrets 1 til 2 * A + B	$2 \cdot (L + H)$
9	R	4	sett Omkrets 1 til a + a + b + b	$(a+b) + (a+b) = Omkrets\ 1$
10	R	4	sett Omkrets 1 til a + b + a + b	$4 + 4 + 7 + 7 = Omkrets\ altså\ 22$
11	R	4	Ikke besvart	Ikke besvart
12	DR	2	Ikke levert	Ikke besvart
13	IM	2	sett omkrets 1 til a * 2 + b * 2	$2a + 2B$
14	IM	1 (tegnet et kvadrat)	hvis a + b > 85 si a+b er større enn 85 vent 1 sekunder si Så omkretsen er større enn 170. hvis a + b < 85 si a+b er mindre enn 85 vent 1 sekunder si Så omkretsen er mindre enn 170. hvis a + b = 85 si a+b er 85 vent 1 sekunder si Så omkretsen er 170.	Ikke besvart, men har regnet ut Omkrets 2 i oppgave 5 riktig der de kom frem til: $O = (a \cdot 3) \cdot (b \cdot 2) \cdot 2$ $= (15 \cdot 3) \cdot (60 \cdot 2) \cdot 2$ $= 45 \cdot 120 \cdot 2$ $= 10800cm$
15	R	2	sett Omkrets 1 til A + B + A + B	Ikke besvart

Tabell 4.2: Gruppenes generaliseringsnivå på oppgave 3 basert på besvarelsene i Tabell **4.1**

Elev-gruppe	**Naiv induksjon**	**Generalisering**				**Vanskelig å kategorisere**	**Ikke besvart / levert**
		Aritmetisk	**Algebraisk**				
			Faktuell	Kontekstuell	Symbolsk		
1					X		
2					X		
3					X		
4					X		
5					X		
6					X		
7					X		
8					X		
9					X		
10					X		
11							X
12							X
13					X		
14						X[2]	
15							X

4.2.2 Oppgave 3 – symbolsk algebraisk generalisering i Scratch

Samtlige grupper som har levert oppgave 3 (13 av 15 grupper), har brukt variablene a og b i sin besvarelse når de har regnet ut omkretsen av rektangelet (se Tabell **4.1**). Dermed har de konstruert et algebrauttrykk som gjelder for alle tilfeller av rektangelet, og de tretten gruppene viser i sine besvarelser i oppgave 3 at de har gjennomført en *symbolsk algebraisk generalisering* – den øverste kategorien i Radford (2006) rammeverk (se Tabell **4.2**).

Besvarelsene på oppgave 3 i Tabell **4.1** kan deles inn i tre kategorier. (1) Algebrauttrykket gjenspeiler at hver av sidene adderes, slik som gruppe 4 har løst det med $a + b + a + b$. (2) Algebrauttrykket gjenspeiler at elevene har brukt rektangelets egenskaper, der to og to sider er like lange, som utgangspunkt for sitt algebrauttrykk. Et eksempel ser vi i gruppe 1 sin besvarelse: $2(a + b)$. (3) Gruppe 14 skiller seg ut ved å ikke ha en direkte utregning, men en test på om omkretsen er mindre, lik eller større enn 170. De generaliserer likevel ved å bruke bokstavene a og b, og algoritmen viser at de bruker figurens egenskaper i sin løsning, fordi de dobler summen av $a + b$.

Opptakene viser samtidig at ikke alle kom frem til algebrauttrykket problemfritt. De fleste gruppene kom raskt frem til at de måtte bruke variablene a og b i sin utregning for å få det til å bli riktig, men diskusjonene viser ulike tilnærminger, der flere knyttet variablene til sidene på rektangelet. Anne og Nils (gruppe 1) er et godt eksempel på det, når de her begynner å jobbe på oppgave 3:

[2] Bruker bokstavene a og b og er dermed nær algebraisk symbolsk generalisering

(Opptak)

Nils: Ehh... Hvis du tar $a + b$ (Anne setter a og b inn i en addisjonsoperator, se Figur **4.1**).

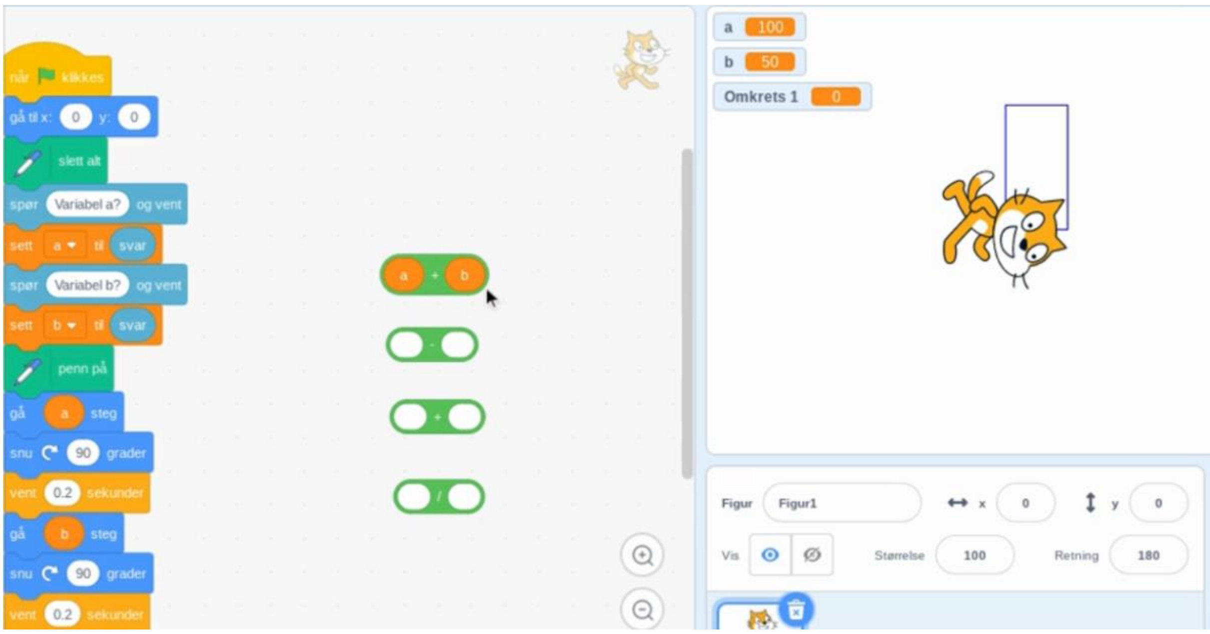
Anne: Er liksom dem her a... $a + b$? (Anne peker på to av sidene i rektangelet de har tegnet)

Nils: a er den lengste og b er den korteste.

Anne: Ja, så det blir $a + b$, ja...

Oppgaven skapte et behov for at elevene måtte generalisere ved å bruke bokstavene a og b: Anne og Nils forstår at de må bruke variablene for at utregningen skal bli riktig for alle tilfeller av rektangelet. Diskusjonen viser samtidig at de knytter a og b til konteksten – sidene i rektangelet. Hvis de kunne valgt navn på variablene selv, er det godt mulig at de ville endt opp på *kontekstuell algebraisk generalisering*, men fordi bokstavene, a og b, er gitt i oppgaven, blir de brukt i deres endelige utregning. Undervisningsdesignets tiltak om å navngi Scratch-variablene a og b, gjør at Anne og Nils oppnår *symbolsk algebraisk generalisering*.

Eksempelet med Anne og Nils gjenspeiler hvordan prosessen har vært for de fleste gruppene. Flere grupper har diskusjoner der navngivningen kan føre til at de passer inn i andre generaliseringsnivåer, men deres endelige svar i Scratch plasserer dem i den øverste kategorien: *symbolsk algebraisk generalisering*. Årsaken til det kan være at oppgaven skaper et behov for å konstruere en utregning som blir riktig i alle tilfeller – altså en generalisering. I tillegg ligger det i oppgavens rammer at de må bruke Scratch-variablene a og b, som nærmest tvinger elevene til å bruke navngivning som tilhører kategorien *symbolsk algebraisk generalisering*.



Figur 4.1: Anne og Nils (gruppe 1) setter a og b inn i en addisjonsoperator

4.2.2.1 Andre kommentarer til besvarelsene i oppgave 3

Besvarelsene i oppgave 3 (se Tabell **4.1**) viser noen variasjoner som må kommenteres. Både gruppe 2 og 3 brukte Scratch-blokkene «endre» i stedet for «sett». Ser man bort fra kontekst kan den erfarne bruker av Scratch se at det ikke blir riktig når man kjører programmet flere ganger, hvor *Omkrets 1* vil bli bare større og større for hver gang. Begge gruppene omgikk det problemet ved å nullstille variabelen *Omkrets 1* for hver gang programmet startet – koden ble dermed riktig likevel. Ella og Lars i gruppe 3 strevde med å få lagret utregningen i Scratch-variabelen *Omkrets 1*. I sin utforskning av problemet forsøkte de å addere *Omkrets 1* til slutt i algebrauttrykket. De fjernet aldri den addisjonen, men fordi *Omkrets 1* ble nullstilt i starten av programmet, hadde det ingen praktisk betydning – utregningen ble likevel riktig. Av tekniske årsaker manglet det store deler av opptaket til Dea og Gry i gruppe 11, men besvarelsene deres viser at de har gjennomført oppgave 2, men ikke rakk å begynne på oppgave 3. Trolig opplevde de oppgave 2 som krevende.

4.2.3 Oppgave 4 - varierende grad av generalisering på et «matematisk språk»

Besvarelsene i oppgave 4 viste en større variasjon enn i oppgave 3, men de fleste gruppene mestret også her å generalisere. Tabell **4.1** viste at seks av gruppene (gruppe 1, 2, 5, 6, 9 og 13) hadde besvarelser på oppgave 4 som gjenspeilet symboliseringen og strukturen de brukte i oppgave 3, noe som gjør at de fortsatt plasseres i kategorien *symbolsk algebraisk generalisering*. Gruppe 4 og 7 gjorde endringer i algebrauttrykket, men viste fortsatt symbolsk algebraisk generalisering. Gruppe 3, 8 og 10 endret på symboliseringen og/eller strukturen på algebrauttrykket sitt, noe som plasserte dem i andre kategorier. Gruppe 10 viser *aritmetisk*

Tabell 4.3: Gruppenes generaliseringsnivå på oppgave 4 basert på besvarelsene i Tabell 4.1

Gruppe	Naiv induksjon	Generalisering				Vanskelig å kategorisere	Ikke besvart / levert
		Aritmetisk	Algebraisk				
			Faktuell	Kontekstuell	Symbolsk		
1					X		
2					X		
3						X	
4					X		
5					X		
6					X		
7					X		
8				X			
9					X		
10		X					
11							X
12							X
13					X		
14					(X)		
15							X

generalisering, mens gruppe 8 viser *kontekstuell algebraisk generalisering*. Gruppe 3 er vanskelig å plassere i en kategori, men de er nærmest *symbolsk algebraisk generalisering* (se oversikt over gruppenes generaliseringsnivåer i Tabell **4.3**).

Resultatene på oppgave 4 viste generaliseringsnivåer som var mye bedre enn først forventet. Med tanke på at elevene ikke hadde forkunnskaper i algebra, er det imponerende at samtlige generaliserer, og at hele 9 grupper er på øverste nivå – symbolsk algebraisk generalisering. Noen av gruppene byttet både navngivning og type symbolisering underveis, som førte til at de viste en annen generaliseringskategori i oppgave 4 enn i oppgave 3.

Alva, Bea og Catrin (gruppe 10) sin besvarelse av oppgave 4 plasseres i kategorien *aritmetisk generalisering* (se Tabell **4.1**). Opptaket støtter også den kategoriseringen – de opplevde oppgaven som krevende og var ukomfortable med å bruke bokstaver. I oppsummeringen av timen, der de skal forklare hvorfor de brukte bokstaver, svarte alle tre «vet ikke». Alva, Bea og Catrin praktiserer dermed aritmetisk generalisering.

Bård og David, i gruppe 8, byttet ut *a* og *b* med *L* og *H*. Opptaket viser at de tenkte *L* for lengde og *H* for høyde. Med andre ord valgte de bokstaver som har tilknytning til rektangelet, og navngivningen nærmer seg en *kontekstuell algebraisk generalisering*.

Alvin og Lovise (gruppe 4), og Sivert og Jonas (gruppe 7) gjorde endringer i algebrauttrykket, men viste fortsatt *symbolsk algebraisk generalisering*. Sivert og Jonas endret strukturen på algebrauttrykket, der de i oppgave 4 også brukte rektangelets egenskaper for å se hvordan $a + b$ kunne multipliseres med 2.

Det var utfordrende å plassere Ella og Lars (gruppe 3) i én bestemt kategori. De gav to ulike svar (se Tabell **4.1**), men opptaket viste at deres egentlig svar var $x + x + 40 + 40 = omkrets$. Tidspunktet for når de skrev på $(a + b) + (a + b) = O$ kom ikke frem i noen av datakildene, men sannsynligvis ble det gjort når det eksemplet ble vist på tavla i oppsummeringen av økten. Sammenligning av svarene på oppgave 3 og 4 viser at de gjorde tre endringer (se Tabell **4.1**). De endret strukturen i algebrauttrykket, de endret den ene bokstaven til *x*, og de endret den andre til *40*. Utdrag fra opptaket viser at Ella og Lars har en innholdsrik diskusjon:

(Opptak)

1	Lars:	$a + b$...
2	Ella:	Nei, nei, nei... Omkrets er lik $x + x + y + y$.
3	Lars:	Nei, det gir jo ikke mening
4	Ella:	Hvorfor det? Men det er jo det.
5	Lars:	Det er $xx + xy + yx + yy + yy$.
6	Ella:	Hvorfor yy og yy?
7	Lars:	Fordi det er to variabler. yy, xx, xy, yx, yy, yy. Ehm. Da kan vi også bare skrive $a + b + a + b + omkrets$ 1.

8	Ella:	Men det er jo akkurat det samme.
9	Lars:	Ja.
10	Ella:	Men vi skal skrive det i algebra.
11	Lars:	Oja, i algebra, ja.
12	Ella:	Nja, det er derfor jeg har foreslått *x* og *y* og sånt.
13	Lars:	Jammen, x er jo det som er her. Vi kan ta $x + x + x + x = omkrets.$
14	Ella:	Nei, nei, nei. Fordi $x + x$ er jo... x står for ett tall, og hvis det er to x-er, så betyr det at de har samme verdi, så da kan det ikke være fire x-er, for de har ikke samme lengde, i et rektangel
15	Lars:	Men da kan jeg ta $x + x\ +\ 40\ + 40\ =\ omkrets$, som er lik x.
16	Ella:	Nei, det må være lik p for eksempel. X-en er jo...
17	Lars:	Okey. Da er det. Jeg kan... $x + x + 40 + 40 = omkrets$
18	Ella:	Men hvorfor 40?
19	Lars:	Fordi det er jo...
20	Ella:	Jaja, ok, greit

I diskusjonen er det mye som fortjener en kommentar. For det første startet Lars med å vise til bokstavene a og b. Senere, i linje 10, begrunnet Ella hvorfor de ikke kunne bruke bokstavene a og b. I intervjuene av gruppe 1, 2 og 3 kom det frem at samtlige elever hadde lært om enkle ligninger tidlig på barneskolen. Eksemplene de nevnte hadde alltid *x* som den ukjente. Det kan trolig være årsaken til at Ella og Lars hadde en forestilling om at algebra måtte inneholde bokstavene *x* og *y*.

Det er uvisst hvorfor Lars foreslo $xx\ +\ xy\ +\ yx\ +\ yy\ +\ yy$ i linje 5, og det er også vanskelig å se sammenhengen med $a + b + a + b + omkrets$ 1, bortsett fra at antall ledd stemmer overens.

Diskusjonen deres viste også viktige aspekter i generalisering. De visste at de måtte skrive et uttrykk som gjaldt for flere tilfeller, og at de dermed måtte bruke bokstaver. Samtidig hadde de en formening om at algebra betydde at de måtte bruke *x* og *y*, men det kom ikke frem hvorfor de til slutt valgte å bruke tallet 40 i stedet for *y* (linje 17). Til slutt viste de også forståelse for at det å kun bruke *x* ikke passet med et rektangel, da alle sidene i så fall ville få samme verdi: $x + x + x + x = omkrets$ (linje 14).

Prosessen til Ella og Lars viste at de var nære en *symbolsk algebraisk generalisering*, men de endte opp med å bruke tallet 40, og det ble derfor vanskelig å plassere dem i én bestemt kategori. Likevel hadde de en god diskusjon der de utfordret hverandre på forståelser av algebrauttrykk og bruk av bokstaver. Det kan komme godt med i deres videre arbeid med algebra.

Oppsummert viste samtlige grupper evne til å generalisere i oppgave 3 og 4. Selv om nivåene varierte noe mer i oppgave 4, har elevene generalisert på et høyere nivå enn man kanskje kunne forvente, med tanke på at de ikke hadde forkunnskaper i algebra.

4.3 Variabler – elevenes beskrivelser av bruk av bokstaver

I intervjuene kom det frem at elevene tidligere hadde lært enkle ligninger, og dermed hadde de allerede etablert en tolkning av bokstaver i kategorien *ukjent*. Scratch-variablene i undervisningsdesignet varierte i verdi og tilhørte derfor kategorien *variabel*. Seks av ti grupper mestret å utvikle sin tolkning fra *ukjent* til *variabel* i løpet av undervisningsøkta. Tre av gruppene hadde beskrivelser som var vage eller mangelfulle, og dermed vanskelig å plassere i en kategori, og én gruppe hadde en beskrivelse som lå et sted mellom kategoriene *ukjent* og *variabel*.

Elevenes beskrivelser ble hentet fra intervjuene og fra opptakene, der elevene i slutten av undervisningsøkta forklarte hvorfor de valgte å bruke variablene *a* og *b*, da de skulle regne ut omkrets 1. I tillegg var det noen tilfeller av beskrivelser på andre tidspunkt i opptakene som ble inkludert i analysearbeidet.

Analysearbeidet om beskrivelse av bokstaver var krevende, fordi elevene ikke hadde kunnskap om rammeverket som ble brukt (det var heller ikke forventet). Det var flere eksempler på at elever brukte ordet *ukjent*, mens de egentlig beskrev kategorien *variabel*. Dette kan de selvfølgelig ikke klandres for. Begrepsblandingen gjorde det også vanskelig å vite om de egentlig tenkte på noe som endres eller ikke når de brukte ordet *ukjent*, og da særlig kortfattede beskrivelser. Begrepsblandingen hos elevene stilte ekstra krav til analysens grundighet.

I neste del vil elevenes forkunnskaper knyttet til bruk av bokstaver i matematikk bli presentert. Deretter følger elevenes beskrivelser av bokstavene de brukte i undervisningsøkta.

4.3.1 Elevenes forkunnskaper - *ukjent*

I intervjuene kom det frem at elevene hadde erfaring med *ukjent* fra tidligere arbeid med enkle ligninger. Dette gjenspeiles delvis i opptakene, fordi mange brukte ordet *ukjent* når de beskrev bokstavene. I intervjuet med Ella og Lars (gruppe 3) forteller de om sitt tidligere arbeid med ligninger og *ukjent*:

(Intervju)

Lars:	Vi hadde x og sånt, også skulle vi finne ut hva x-en var, med... eh... et gangestykke, også skal vi finne ut hva x-en var, og svaret og sånt.
Ella:	Ja, vi hadde og det – ukjent. Og, liksom, ligninger.
Joakim:	Ja... Ukjent, hva mener du med det?
Ella:	Ukjent tall, at man skal finne ut hva tallet er.

I alle de tre intervjuene ble begrepet *ukjent* nevnt. Ellas beskrivelse viser til ordet «tallet», som er i entall, og kan dermed antyde at det er snakk om én verdi. Hennes beskrivelse passer derfor i kategorien *ukjent*. Ola i gruppe 2 ble utfordret mer i sitt intervju, og kom med en fyldigere beskrivelse:

(Intervju)

Joakim:	(...) Hva var på en måte poenget med den her, da? En sånn oppgave? (peker på ligningen $3 + x = 5$ som står på arket foran dem)
Ola:	Det er jo å definere *x*, da.
Joakim:	Mhm. Og hva er *x* her, i tilfellet, da?
Ola:	Det er to.
Joakim:	Ja. Kan *x* ha noen annen verdi enn to her?
Ola:	Nei.
Joakim:	Hvorfor ikke? Nå stiller jeg et vanskelig spørsmål, altså.
Ola:	Fordi at man er nødt til å... fordi at 3+2 er jo fem, men det er ikke noe annet svar. Så da må jo tre, nei *x* være to.

I intervjuene brukte samtlige bokstaven x i sine gjengivelser, og de viste til enkle eksempler, slik som $2 + x = 8$ og $3x = 21$. Det er derfor naturlig å anta at alle elevene tidligere har lært om enkle ligninger, og at de trolig har en etablert tolkning som tilhører kategorien *ukjent*. Dette støttes i opptakene da flere elever bruker ordet *ukjent* underveis. For eksempel viser Alvin og Lovise (gruppe 4) at de er kjent med å bruke ordet *ukjent* i algebraisk sammenheng allerede i oppstarten av timen:

(Opptak)

Alvin:	Er ikke variabel i matte *x* og sånn da?
Lovise:	Det er liksom sånn som er ukjent, som vi ikke vet hva er, da
Alvin:	Ja, nei *x*...
Lovise:	Ja, ukjent

Beskrivelsen er ikke så tydelig at vi kan konkludere med at den peker mot kategorien *ukjent* eller *variabel*. Men – det viser at de har hørt om ordet «ukjent» og at bokstaven «x» er vanlig å bruke i matematikk.

Tabell 4.4: «Hvorfor bruke bokstavene a og b i stedet for tall?» (opptak)

Beskrivelse (kategori)	**Gruppe**
Variabel	2, 6, 7 og 11
Ukjent	1 og 9
Mellom *ukjent* og *variabel*	3 og 5
Ingen kategori	8 og 10

4.3.2 Elevenes beskrivelser av bokstavene *a* og *b*

I oppsummeringen av timen fikk elevene i oppgave å forklare hvorfor de brukte bokstavene *a* og *b* i stedet for tall når de regnet ut omkretsen i oppgave 3. Tabell **4.4** viser hvilken kategori gruppens beskrivelser plasserer seg i (det mangler opptak av gruppe 4 fra denne delen).

Videre presenteres eksempler på beskrivelser fra oppsummeringen i timen, men også om de har kommet med beskrivelser på andre tidspunkt i timen eller i intervjuet, som gjør at de kan plasseres i en annen kategori. Tabell **4.5** viser elevenes endelige kategorisering.

De som beskrev bokstavene i kategorien *variabel* viste til noe som «endres» eller «byttes», eller knyttet bokstavene til «mange» tall. Mia og Ola (gruppe 2) diskuterte følgende:

(Opptak)

50 Ola: Fordi at tallene er ukjent, på en måte. Det er liksom ikke noe bestemt tall.

51 Mia: Okay...?

52 Ola: Man kan jo endre på *a* og *b*, og da kan det jo ikke stå 1, uansett hva.

53 Mia: Nei, nei, det er sant.

54 Ola: Det er ukjent.

Det er vanskelig å fastslå om beskrivelsen i linje 52 var tiltenkt Scratch-variabler eller bruk av bokstaver i algebra, men uansett hadde de en forståelse av at verdiene for *a* og *b* kunne endres. Beskrivelsen på linje 50 er ikke like tydelig, men kan tolkes til en beskrivelse som henviser til ubestemte kvantiteter, eller en mengde. Mia og Olas tolkning og beskrivelse passer til kategorien *variabel*.

Gruppe 1 og 9 ble plassert i kategorien *ukjent*, men begge konkluderer raskt, uten noen form for beskrivelse. Anne og Nils (gruppe 1) «diskuterte» på følgende vis:

(Opptak)

Anne: Hvorfor bruke bokstaver og sånn?

Nils: Fordi tallet er ukjent.

Anne: Det var akkurat det jeg skulle si.

På grunn av elevenes mange eksempler på begrepsblanding er det umulig å si om Anne og Nils egentlig mente kategorien *variabel* eller ikke, når de bruker ordet *ukjent*. I intervjuet kom de med en mer utdypende beskrivelse som passer til kategorien *variabel*:

(Intervju)

Nils: Fordi det på en måte er en ukjent og ikke.. Eller det… man vet ik… eller det varierer fra hver gang.

Joakim: Ja. Hva varierer da, i så fall?

Nils: På en måte hvor langt den går, og sånt. Du skriver jo inn.

Fredrik og Emil (gruppe 9) brukte, i likhet med Anne og Nils, ordet *ukjent* i sin samtale i opptaket, men beskrivelsen de gav var for vag. Dermed ble de ikke plassert i noen av kategoriene.

Gruppe 3 og 5 hadde beskrivelser som er vanskeligere å kategorisere. Axel og Gustav (gruppe 5) hadde for eksempel følgende diskusjon:

(Opptak)

170 Axel: Fordi det er ukjent.

(...)

173 Gustav: Hvis vi bruker et bestemt tall i stedet for et ubestemt tall, så vil vi ikke få det riktige svaret.

Axel og Gustav ble ikke intervjuet, og dermed var det vanskelig å tolke hva de legger i *ubestemt tall* – og om beskrivelsen tilhører kategorien *ukjent* eller *variabel*. I starten av timen brukte de likevel samme ord når de beskrev begrepet *variabel* i en programmeringskontekst:

(Opptak)

40 Axel: Det er noe som viser for eksempel poeng eller gamespeeden.

41 Gustav: Et ubestemt tall som du kan endre til hva du vil for å få de resultatene du vil ha, da. For eksempel. Så det er jo ikke noe annet enn et tall. Et tall som du bruker for å få resultatene du vil ha.

42 Axel: Ja

Dermed kan man anta, ut fra beskrivelsen på linje 41, at de tillegger en grad av «endring» i deres forståelse av *ubestemt tall*. I så fall kan man konkludere at beskrivelsen på linje 173 tilhører kategorien *variabel*.

Ella og Lars, i gruppe 3, ble utfordret videre på sin beskrivelse i intervjuet. Deres beskrivelser er sterkt knyttet til Scratch-programmet og at brukeren velger nye tall for a og b:

(Intervju)

Joakim:	Hvorfor brukte dere bokstaver i stedet for tall?
Ella:	Det er jo fordi at, hvis det hadde stått 1+2+1+2, så hadde det jo blitt 6, uten at det var det som egentlig var svaret.
Joakim:	Ja.
Ella:	Det der (peker på utregningen i Scratch: $b + a + b + a = Omkrets$ 1) er på en måte, det der er det man plusser sammen av dem tallene man selv velger, enn hvis det står ett tall, så er det selve tallet, i stedet for det vi vil at det skal være.
Joakim:	Mhm
Ella:	På en måte

Beskrivelsen er vag, og ble ikke mye tydeligere utover i intervjuet. Mellom linjene kan man tolke det Ella sa til at bokstavenes verdi endrer seg, men det er vanskelig å vite om hun relaterte det kun til programmering eller om hun også tenkte algebra. Beskrivelsen passer heller ikke til kategorien ukjent, og sjekk av andre datakilder fører ikke til at man kan konkludere om hvorvidt deres beskrivelser kan plasseres i kategorien *ukjent* eller *variabel* – de plasseres «mellom» ukjent og variabel.

Gruppe 8 og 10 hadde samtaler seg i mellom som ikke viser noen beskrivelse av bokstavene a og b. Bård og David (gruppe 8) forsøkte å diskutere det, men endte opp med å bruke tall i stedet. Det er overraskende, fordi de mestret å konstruere algebrauttrykket for omkretsen i oppgave 3, og viste at de så sammenhengen mellom bokstavene a og b med sidene i rektangelet (l og h) i oppgave 4. Man kan dermed tenke seg at de kunne ha kommet med en bedre beskrivelse, hvis de hadde blitt ytterligere utfordret på dette i et påfølgende intervju. Alva, Bea og Catrin i gruppe 10, konkluderte raskt med «jeg vet ikke», som samsvarer med det de viste ellers i undervisningsøkta. De kan dermed ikke plasseres i noen av kategoriene.

Tabell 4.5: Endelig kategorisering av elevenes beskrivelser av bokstavene a og b etter triangulering av datakildene

Beskrivelse (kategori)	**Gruppe**
Variabel	1, 2, 5, 6, 7 og 11
Mellom *ukjent* og *variabel*	3
Mangelfulle/vage beskrivelser	8, 9 og 10

Oppsummert har seks av ti grupper vist en beskrivelse i løpet av undervisningsøkta eller intervjuet som sammenfaller med kategorien variabel (se Tabell **4.5**). En gruppe er vanskelig å plassere i en kategori, og ender opp «mellom» ukjent og variabel. De tre resterende gruppene har manglende eller vage beskrivelser. I tillegg er det verdt å merke seg at gruppe 11, som ikke kom lengre enn til oppgave 2, likevel klarte å komme med en beskrivelse som tilhørte kategorien *variabel*.

5 Diskusjon

I denne studien har jeg utviklet, prøvd ut og studert et undervisningsdesign som kombinerer fagområdene programmering og algebra. Studien har vært styrt av følgende forskningsspørsmål:

Hvordan kan elever utvikle algebraiske konseptuelle ideer ved å bruke generalisering og variabel i både programmering og algebra?

De teoretiske prinsippene i denne studien leder ut fra forskningsspørsmålet og problemformuleringene fra de tre fokusområdene *kognitiv belastning, generalisering og variabler*, og prinsippene lyder som følger:

1) *Designet skal legge til rette for læring ved å redusere den eksterne kognitive belastningen.*
2) *Designet skal bidra til at elevene oppnår et høyt generaliseringsnivå, der de nærmer seg symbolsk algebraisk generalisering.*
3) *Designet skal stimulere til at elevene skal overkomme de kritiske tolkningsovergangene av variabelbegrepet, og beskrive bruk av bokstaver som noe som varierer.*

Resultatene viser at undervisningsdesignet hadde høyere ekstern kognitiv belastning enn det som var tenkt, noe som førte til at elevene møtte på unødvendige utfordringer knyttet til programmering. Undervisningsdesignet bidro til at alle elevene klarte å generalisere. I Scratch (oppgave 3) viste alle gruppene et høyt generaliseringsnivå, mens det i oppgave 4 fremkom mer variasjon i nivåene når elevene skulle bruke et «matematisk» språk. Undervisningsdesignet klarte til en viss grad å stimulere elevene til å utvikle sine tolkninger forbi de kritiske tolkningsovergangene av variabelbegrepet. Seks av ti grupper utviklet sine tolkninger forbi den øverste kritiske overgangen til kategorien *variabel*. Tre av gruppene hadde vage eller mangelfulle beskrivelser, som gjorde det umulig å kategorisere dem. En gruppe viste en beskrivelse som var vanskelig å plassere i en av kategoriene, og ble derfor plassert «mellom» ukjent og variabel. Resultatene tyder på at elevene har utviklet algebraiske konseptuelle ideer ved å bruke generalisering og variabel i både programmering og algebra.

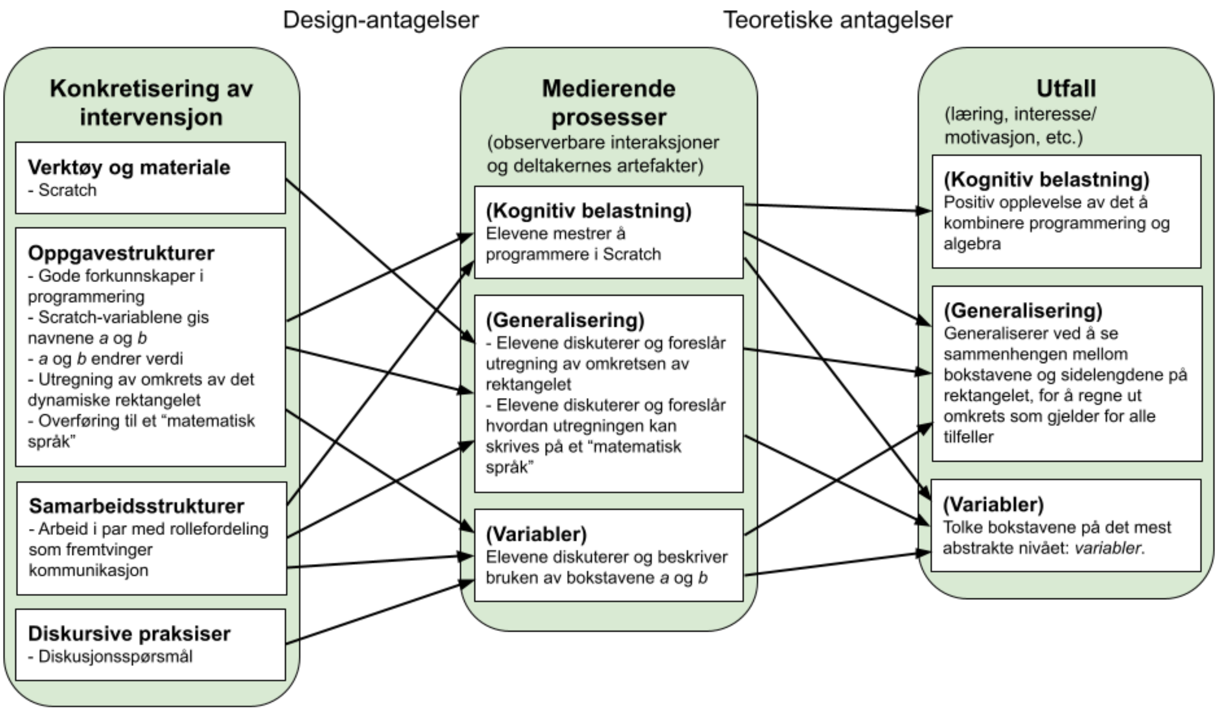
Kapittelet har tre deler der første del ser på hvilke forbedringer man kan gjøre i undervisningsdesignet. I andre del generaliseres resultatene ved å diskutere hva elevene lærte, i lys av annen relevant forskning. I siste del diskuteres studiens begrensninger.

5.1 Undervisningsdesignet

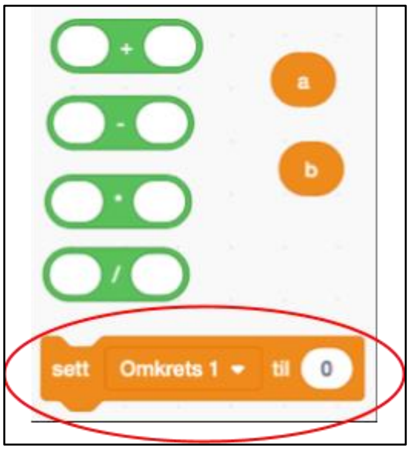
Endring av intervensjonen og forbedring av undervisningsdesignet gjøres på grunnlag av en sammenligning mellom læringshypotesene og resultatene. Studiens resultater er i stor grad sammenfallende med læringshypotesene, men med noen unntak. Videre følger en sammenligning av læringshypoteser og resultater innenfor hvert av de tre teoretiske prinsippene, etterfulgt av forslag til eventuelle forbedringer.

Den tydeligste forskjellen mellom læringshypotesene og resultatene kan knyttes til den eksterne kognitive belastningen i undervisningsdesignet, som viste seg å være for høy og dermed bør reduseres ytterligere. I oppgave 3 møtte elevene på to problemer som jeg navnga «algebraproblemet» og «programmeringsproblemet». Programmeringsproblemet innebar hvordan elevene måtte konstruere utregningen av omkretsen av rektangelet i Scratch. Informasjon som allerede er lagret i langtidsminnet vil være enklere for arbeidsminnet å bearbeide, enn informasjon man møter for første gang (Sweller, van Merriënboer & Paas, 2019). Hensikten med undervisningsdesignet var at all informasjonen som elevene måtte håndtere, knyttet til programmering, skulle være lagret i langtidsminnet fra tidligere. Det viste seg å ikke være tilfelle. Trolig innebar programmeringsproblemet ny informasjon som belastet arbeidsminnet unødvendig mye når elevene samtidig stod ovenfor algebraproblemet. Utfordringene kan gjennom årsakssammenhengene spores tilbake til tiltaket om elevenes *forkunnskaper i programmering* (se Figur **5.1**). Det foreslås dermed to forbedringer: Undervisningsdesignet må sørge for at elevene på forhånd har kjennskap til hvordan de kan kombinere Scratch-blokker med operatorer (regnearter), og hvordan de skal lagre utregningen i Scratch-variabelen «Omkrets 1». I tillegg bør et bilde knyttet til oppgave 3 i oppgavesettet endres. Den viktige blokken som trengs for å lagre utregningen i Scratch-variabelen vises ikke på bildet, og bør dermed inkluderes, slik som vist i Figur **5.2**.

Ved å sammenligne resultatene og læringshypotesene i lys av elevenes generaliseringer, mestret samtlige elever å generalisere ved å bruke bokstaver i sin utregning av omkrets i Scratch



Figur 5.1: Forventet utfall av undervisningsdesignet (kopi av Figur **3.9**)



Figur 5.2: Endring av bildet som viser hvilke blokker som trengs for å løse oppgave 3

(oppgave 3) – alle var på det øverste nivået; *symbolsk algebraisk generalisering*. I overgangen til et «matematisk språk» viste fortsatt alle generalisering, men generaliseringsnivåene var noe mer variert (oppgave 4). Siden elevene ikke hadde forkunnskaper i algebra, var det ikke forventet at alle skulle komme med generaliserte forslag (dette kommer ikke tydelig frem i Figur **5.1**, men er nevnt i kapittel 3.2.2.2). Elever som ikke har lært algebra enda, har ikke kunnskap om hvordan man kan representere matematiske objekter ved bruk av algebraisk notasjon. Derfor bygger algebraisk tenkning på at generaliseringsaktiviteter bør basere seg på andre representasjoner som elevene har tilgjengelig i sitt repertoar (Carraher et al., 2008). I undervisningsdesignet har elevene allerede kjennskap til bruk av Scratch-variabler. Undervisningsdesignet klarer derfor å legge til rette for at generaliseringer bruker Scratch-variabler som representasjon. Jeg mener at dette undervisningsdesignet skiller seg fra andre aktiviteter under algebraisk tenkning, ved at Scratch-variablene navngis i relasjon til det algebraiske symbolspråket. På denne måten fikk elevene hjelp til å mestre overgangen fra utregningen i Scratch til et algebrauttrykk (referert til som «matematisk» språk i oppgaveheftet). Sett i lys av Radfords rammeverk for generaliseringsnivå, var det ikke forventet at elevene generaliserte på et så høyt nivå som resultatene viser. Det foreslås derfor ingen endringer av tiltakene knyttet til generalisering.

Sammenligning av resultatene og læringshypotesene viser ulikheter omkring elevenes beskrivelser av variablene de brukte. Det var forventet at elevene skulle komme over den kritiske og utfordrende overgangen fra *ukjent* til *variabel*. Av de ti gruppene det var datamateriale på, var det kun seks av disse som viste beskrivelser knyttet til kategorien *variabel*. Resultatene bekrefter det Ely og Adam (2012) hevdet: å mestre overgangen fra *ukjent* til *variabel* er krevende for elever. Figur **5.1** viser at forventet utfall knyttet til bokstaver har årsakssammenhenger (piler) til flere av tiltakene, som gjør det vanskelig å peke på et konkret tiltak som bør endres. Det blir derfor ikke foreslått endringer av tiltak i denne sammenhengen, men det foreslås i stedet å endre forventningene i læringshypotesene. Trolig er ikke 75 minutter nok tid for elevene til å kunne bearbeide og utvikle de abstrakte tankene knyttet til bruk av bokstaver. Det at over halvparten klarte overgangen, og at samtlige var på et høyere generaliseringsnivå enn forventet, viser dog at undervisningsdesignet har et potensiale til å gi elevene det nødvendige grunnlaget for å mestre denne overgangen. Trolig trenger en del elever mer tid på seg for å nå det krevende abstrakte nivået i kategorien *variabel*. Det foreslås å senke forventningene i læringshypotesene som vises i Figur **5.1**, og endre forventet utfall til «utfordrer sin tolkning i bruk av bokstaver i algebra». I tillegg bør det legges til en anbefaling til lærere om å fortsette å utfordre elevene på beskrivelser av bokstaver i algebra i tiden etter gjennomføringen av undervisningsdesignet, for å sørge for at alle elevene til slutt oppnår kategorien *variabel*.

Basert på resultatene foreslås det en siste endring av undervisningsdesignet, som ikke kan plasseres under noen av de tre teoretiske prinsippene. Datamaterialet viser at Dea og Gry i gruppe 11 ikke kom lengre enn til oppgave 2. Opptaket inneholder kun slutten av undervisningsøkta, og de var ikke blant dem som ble intervjuet, så man vet ingenting om deres prosess. Likevel viser opptaket på slutten av undervisningsøkta at de mestret å beskrive bruk av bokstaver i kategorien *variabel*. Med andre ord viser de evne til å forstå hvorfor man bør bruke bokstaver, samtidig som de trolig har strevd med fremdriften i oppgaven. Her bør det legges til et tiltak som hjelper alle elevene med å få en god fremdrift og utvikling i sitt arbeid. Inspirert av Boaler (2002), som i sin artikkel skrev om rettferdig og tilpasset undervisning (equity), kan man bruke mer tid i starten på å la elevene diskutere problemet og målet med aktiviteten, slik at alle i klassen får en felles forståelse før de starter arbeidet. I tillegg kan man også vurdere å ha en eller flere mindre oppsummeringer underveis, der enten lærer og/eller elever kan diskutere hvordan de har løst et mindre problem. Det vil trolig kunne hjelpe dem som står fast med videre fremdrift. Slike tiltak kan trolig være fornuftig å gjennomføre i alle typer klasser, uavhengig om de er sterkere eller svakere enn elevene i denne studien.

Oppsummert foreslås det følgende endringer av intervensjonen og læringshypotesene:

- Tydeliggjøre nødvendige forkunnskaper for elevene slik at de har kjennskap til hvordan de kan kombinere Scratch-blokker med operatorer (regnearter), og hvordan de skal lagre utregningen i Scratch-variabelen «Omkrets 1».
- Endre bildet til oppgave 3 i oppgavesettet (se Figur **5.2**).
- Endre forventet utfall i læringshypotesen fra «Tolke bokstavene på det mest abstrakte nivået: *variabler*» til «Utfordrer sin tolkning i bruk av bokstaver i algebra».
- Legge til en anbefaling om å fortsette å utfordre elevenes beskrivelser av bokstaver i algebra i tiden etter at undervisningsdesignet er gjennomført.
- Legge til rette for mer rettferdig og tilpasset undervisning ved å la elevene diskutere aspekter ved oppgaven før og eventuelt underveis i arbeidet.

Intervensjonen med tilhørende tiltak viste seg å være relativt god sett i lys av de tre teoretiske prinsippene. Undervisningsdesignet har bidratt til algebraisk konseptuell utvikling hos elevene. Som nevnt kan man ikke peke på ett av tiltakene som den klare suksessfaktoren, det er heller den helhetlige intervensjonen med de underliggende og komplementære tiltakene som har ført frem. Pilene som viser årsakssammenhengene i Figur **5.1** viser hvordan flere av tiltakene er viktige for de forventede utfallene i læringshypotesene. For eksempel viser årsakssammenhengene i Figur **5.1** nødvendigheten av at elevene samarbeidet i par for at de kunne diskutere seg i mellom, for å kunne komme med beskrivelser av bokstavene *a* og *b*. Samtidig ville ikke diskusjonene ha hatt samme relevans uten at elevene opplevde at variablene endret verdi i

Scratch. Med forbedringene som er foreslått, vil jeg hevde at undervisningsdesignet og intervensjonen med de komplementære tiltakene, vil møte de tre teoretiske prinsippene relativt godt.

Tiltakene i undervisningsdesignet er formulert slik at de ikke er direkte avhengige av den lokale konteksten som tilhører denne studien. Jeg vil mene at elever på nivået som tilsvarer en norsk gjennomsnittlig 8. klasse, uten forkunnskaper i algebra, vil ha mulighet til å gjennomføre samme undervisningsdesign med gode erfaringer. Et viktig premiss vil være at elevene har tilgang til digitale verktøy som gjør det mulig å bruke Scratch (nettbrett er dårlig egnet til å programmere i Scratch). Undervisningsdesignet kan derfor sies å ha relevans i andre lignende kontekster.

5.2 Hva lærer elevene?

Litteraturgjennomgangen viser at det er få andre som systematisk har prøvd ut, og forsket på, det å kombinere programmering og algebra. Da denne studien ikke er en litteraturstudie har jeg ikke gjennomført et systematisk litteratursøk, men etter mange søk, lesing av mange artikler og sjekk av referanselister, fant jeg kun tre artikler som har sett nærmere på relasjonen mellom programmering og algebra (Agatolio et al., 2018; Bråting & Kilhamn, 2020; Kilhamn & Bråting, 2019). Sammenligningsgrunnlaget er derfor lite, og det skaper behov for å se mot annen relevant forskning. Dette delkapittelet forsøker å løfte resultatene fra denne studien inn i en bredere kontekst, både ved å se det opp mot det teoretiske grunnlaget for studien, de tre nevnte artiklene som handler om programmering og algebra, og annen relevant litteratur.

Sweller, Van Merrienboer og Paas (1998) mener at «all instructional designs should be analyzed from a cognitive load perspective» (s. 262). Kognitiv belastningsteori gav studien et teoretisk grunnlag til å analysere undervisningsdesignets belastningsgrad, både det iboende og eksterne (ytre), der målet var å senke den eksterne kognitive belastningen til et minimum. I tillegg ble det antatt at teorien om split attention effect (Sweller et al., 1998) var overførbar til denne konteksten, som i så fall innebar at elevene ville oppleve økt belastning av arbeidsminnet når de måtte rette sin oppmerksomhet mot både algebra og programmering samtidig. Scratch er konstruert for å la barn leke med blokker, og prøve og feile, med mål om å la dem utforske egne problemstillinger (Resnick et al., 2009), og stimulerer dermed til å bruke åpne oppgaver i undervisningen. Benton, Saunders, Kalas, Hoyles og Noss (2018) observerte utprøving av ett og samme undervisningsdesign i to påfølgende undervisningstimer, der elevene programmerte med Scratch. Læreren i deres studie opplevde at elevene møtte på flere utfordringer i første time, og så dermed behovet for å redusere åpenheten i oppgaven i neste time, med tydeligere instruksjoner underveis, noe hen lyktes med. Observasjonen sammenfaller med det Kirschner, Sweller og Clark (2006) skrev i sin artikkel, om at man må ta hensyn til elevenes kognitive belastning når man planlegger åpne oppgaver. Elevene i denne studien møtte ikke en like åpen oppgave, men måtte prøve og feile, og utforske problemstillinger som dukket opp underveis. De opplevde lignende utfordringer som beskrevet hos Benton et al. (2018) når de måtte løse et problem fra hvert fagområde samtidig. I denne undersøkelsen viste bruk av kognitiv belastningsteori seg å være nyttig ved analyse av undervisningsdesignet og resultatene. Slik kunne man identifisere aspekter som bør endres for at elevene skal ha mulighet til å rette sin oppmerksomhet mot det som var hensikten i undervisningsdesignet – å utvikle algebraiske konseptuelle ideer.

Undervisningsdesignet i denne studien opererer i overgangen mellom aritmetikk og algebra. Overgangen krever en abstraksjonsutvikling hos elevene, og kan føre til en rekke misoppfatninger knyttet til flere aspekter i algebra, deriblant forståelse av variabler og variabelbegrepet (Bush & Karp, 2013). En av misoppfatningene er feilaktige tolkninger omkring bruk av bokstaver i algebra (Küchemann, 1981). Bush og Karp (2013) nevner blant annet at elever så sent som på 10. trinn fortsatt kan ha feilaktige tolkninger av bokstavene. Resultatene i denne studien viser at elever gjennom å bruke programmering, kan utvikle sin forståelse av variabelbegrepet, og dermed unngå noen av de vanlige misoppfatningene Bush og Karp, og Küchemann nevner. Ingen av elevene i denne studien viste tolkninger av bokstavene *a* og *b* som var ukorrekte. Intervensjonen i undervisningsdesignet kan være en av årsakene til at det ble unngått, men samtidig viste intervjuene at elevene hadde møtt på enkle ligninger på barneskolen. Vi kan derfor anta at elevene allerede hadde en tolkning som var algebraisk korrekt til bruk i ligninger, tilhørende kategorien *ukjent*. Denne studien har dermed kun sett på om elevenes beskrivelser fortsatt tilhører kategorien *ukjent*, eller om de har utviklet sine tolkninger til kategorien *variabel*.

Etter at programmering ble implementert i matematikkfaget i Sverige poengtert Bråting og Kilhamn (2020) at variabelbegrepets ulike definisjon i programmering og algebra, kan føre til forvirring og misoppfatninger av begrepet. Tidsbegrensningen i denne studien gjør at det er umulig å si noe om eventuelle misoppfatninger senere i algebraundervisningen, men i det tilgjengelige datamaterialet har det ikke vært mulig å identifisere noen misoppfatninger omkring definisjonsforskjellene i programmering og algebra. Agatolio, Albanese og Moro (2018) belyste også samme problem, men fremhevet samtidig i sin diskusjon muligheten for å la variabler i programmering opptre på samme måte som bokstaver i algebra, slik at det kan bidra til elevenes utvikling av algebraiske konseptuelle ideer – noe utprøvingen av undervisningsdesignet i denne studien kan bekrefte. Det er interessant at Usiskin (1988) allerede i 1988 hevdet at elever utviklet en god forståelse av variabelbegrepet raskere når de brukte variabler i programmering, enn ved tradisjonell innlæring i algebra. En av suksessfaktorene mente han var at variabelen i programmering kan dekke de fleste fasettene variabelen i algebra kan ha. Denne studien gir et eksempel på Usiskins påstand der vi her kan se hvordan en variabel i programmering (Scratch) kan opptre slik at den passer med Küchemanns (1981) kategori *variabel*.

Det finnes en del forskning knyttet til ulike tolkninger av variabelen, og meninger variabelen får i ulike kontekster (blant annet Booth et al., 2017; Bush & Karp, 2013; Ely & Adams, 2012; Kieran, 2006; Küchemann, 1981; Partanen & Tolvanen, 2019; Stacey & MacGregor, 1997). En god forståelse av variabelens mange fasetter, i de ulike kontekstene den opptrer i algebra, er viktig for å senere kunne skape god forståelse av mer komplekse algebraiske problemer (Hewitt, 2014). Derfor mener jeg at lærere bør skaffe seg kunnskap om variabelens fasetter og didaktiske utfordringer. Elevene i denne studien ble utfordret til å muntlig diskutere og beskrive variabelen. Det kan ses i sammenheng med Philipp (1992) sine erfaringer der diskusjon omkring variabelens ulike bruksområder bidrar til bevisstgjøring hos elevene.

Et av tiltakene i undervisningsdesignet var samarbeidsstrukturen der elevene arbeidet i par med tydelige gitte roller. En slik struktur kan slå feil ut ved at én elev gjør all jobben, mens den andre ser på. Opptakene viste her at de fleste gruppene kommuniserte godt sammen, og opplevde trolig rollefordelingen og samarbeidet som positivt. Det ble ikke spurt direkte om dette i intervjuene, men i to av tre intervjuer bringer de selv frem samarbeidet som noe positivt med undervisningsøkta. Megan Franke skrev i boken «Intentional Talk: How to structure and lead productive mathematical discussions» at samtaler og diskusjoner er avgjørende for læring i matematikk (Franke, 2014). Elevene i denne studien snakket sammen om matematiske problem og programmeringsproblem gjennom hele undervisningsøkta. I tillegg ble de utfordret til å diskutere, argumentere og beskrive matematiske ideer i oppstarten og avslutningen av timen. Tiltaket med samarbeidsstrukturen var trolig viktig for elevenes utvikling av algebraiske konseptuelle ideer.

Carraher et al. (2008) mente at den *eksplisitte formelen* uttrykker et figurtalls «sanne generalitet». De fant i sin studie om elevers generalisering av figurtall, at det var utfordrende for elevene å utvikle sin generalisering fra en rekursiv formel til en eksplisitt formel. I utprøvingen i denne studien har elevene generalisert på et vanskeligere nivå enn trekanttallet som er nevnt i kapittel 2.2.2. Problemet med omkretsen av et rektangel bestod av to variabler (a og b), i motsetning til én variabel (n) i trekanttallet. I tillegg var den rekursive formelen uaktuell slik undervisningsdesignet var utformet, og elevene var nødt til å gå direkte til et uttrykk som kan sammenlignes med den eksplisitte formelen. Elevene måtte kunne se for seg hvordan algebrauttrykket deres skulle kunne gjelde for alle tilfeller av sidelengdene på rektangelet. Bush og Karp (2013) fant i sin litteraturstudie at det var utfordrende for elevene å utvikle forståelse for hvordan bokstavene i det algebraiske symbolspråket er nødvendige for å løse problemet nøyaktig og effektivt. I denne studien gjorde elevene beskrivelser av bokstavene a og b, men det er vanskelig å vite om de er nært knyttet til deres opplevelse av Scratch-variablene, eller om de heller relaterer beskrivelsene til de algebraiske variablene. I tillegg ble elevene kun utfordret på å sammenligne bruk av tall og bokstavene a og b, ikke andre eventuelle representasjoner. Likevel viser elevenes beskrivelser at de innenfor disse rammene ser nytten og formålet med å bruke bokstaver i sin generaliserte utregning.

Dersom andre vil gjennomføre dette undervisningsdesignet, er det flere forhold man må være klar over. Læreren må ha nødvendig kompetanse i programmering, for å kunne gjennomføre en slik økt. I denne studien ønsket elevenes faglærer at jeg, med min kompetanse i programmering, skulle gjennomføre undervisningen. Faglæreren hadde nok klart å gjennomføre undervisningen med sin kompetanse, men ønsket ikke undervisningsansvaret når det var avgjørende for datainnsamlingen til min studie. Elevenes forkunnskaper er en annen viktig forutsetning, såpass viktig at jeg har valgt å definere det som et av tiltakene i intervensjonen. Undervisningsdesignet baserer seg på bruk av programmet Scratch, og elevene må derfor ha tilgang på en datamaskin. Nettbrett vil trolig ikke fungere så godt med Scratch. Det er ikke avgjørende for gjennomføringen, men hvis faglærer ønsker å gjøre best mulige tilpasninger for sine elever, bør hen ha tilstrekkelig kunnskap om variabelbegrepet og variabelens mange fasetter i algebra. I tillegg bør man være bevisst på hvilke utfordringer de ulike definisjonene variabelbegrepet har i algebra og programmering. Til slutt vil det være nyttig for læreren å ha kunnskap om generaliseringsnivåer for å kunne analysere elevers besvarelser.

Resultatene i denne studien viser noe variasjon med tanke på elevenes generaliseringer og beskrivelse av variabler. Det er tydelig at flere av elevene hadde hatt behov for mer tid for å beherske de utfordrende abstrakte ideene innenfor både generalisering og variabler, men sammenlignet med hva annen forskning sier om elevers utvikling i algebra, mener jeg elevene i denne studien har hatt en formidabel utvikling i løpet av 75 minutter. Datamaterialet hadde en tydelig tidsbegrensning, og viste ikke noe om elevenes videre utvikling i algebra. Jeg tør likevel å hevde at bruk av programmering kan føre til at elevene utvikler algebraiske konseptuelle ideer, samtidig som man unngår noen av de vanlige misoppfatningene som fremkommer i andre studier. Undervisningsdesignet kan også være et godt utgangspunkt for elevenes videre arbeid i algebra. Jeg mener at undervisningsdesignet, med de forbedringene som er foreslått, kan og bør tas i bruk hos lærere når de skal starte opp undervisning i algebra på 8. trinn. Diskusjonen i dette kapittelet viser også hvordan denne studien har bidratt til ny teoretisk innsikt om kombinasjonen av fagområdene programmering og algebra i matematikk.

5.3 Metoderefleksjoner

Denne studien har en tydelig ekstern begrensning fordi det finnes såpass lite forskning på hvordan programmering bør implementeres i matematikkfaget (Sanne et al., 2016; Schanzer et al., 2015). Sammenligningsgrunnlaget med lignende forskning er begrenset, noe som gjør at man i stor grad må sammenligne med andre forskningsområder innenfor matematikkdidaktikk. Det er behov for mer forskning om programmering i matematikk, både for å få mer teoretisk innsikt, og for at lærere i skolen skal få mer kunnskap om hvordan man kan drive god undervisning innenfor dette temaet.

Studien har også sine interne begrensninger. Studiens nokså strenge rammer førte til at det kun ble gjennomført én utprøving. Det er poengtert i litteratur om designutvikling at man bør prøve ut undervisningsdesignet i flere kontekster for å kunne komme med resultater som kan generaliseres (Anderson & Shattuck, 2012; The Design-Based Research Collective, 2003). Et forsøk på å veie opp for studiens svakhet er bruk av Sandovals (2004, 2014) forskningsrammeverk, der det teoretiske fundamentet er viktig i hele forskningsprosessen.

Et annet viktig aspekt i designutvikling er hvordan samspillet mellom lærer og forsker skaper et møte mellom praksisfeltet og empiri (Anderson & Shattuck, 2012; The Design-Based Research Collective, 2003). Læreren og forskeren er i denne studien samme person, som kan ha ført til mindre diversitet enn det man ønsker. Jeg mener at min bakgrunn som lærer og masterstudent er tilstrekkelig til at jeg klarer å fylle begge rollene. På denne måten skjer det fortsatt et møte mellom teori og praksis.

Studien er nokså begrenset i tid, da datamaterialet er hentet fra én undervisningsøkt, med påfølgende intervju samme dag. Elevenes faglige utvikling er dermed begrenset til dette tidsrommet og kan ikke fortelle noe om elevenes videre beskrivelser, opplevelser og utvikling i algebra.

Teorier kan aldri beskrive virkeligheten fullt ut, men kun belyse deler av den. Kognitiv belastningsteori, rammeverket om generaliseringsnivåer og tolkning omkring bruk av bokstaver i algebra, har sine begrensninger. Teorien om kognitiv belastning er godt begrunnet i forskning, men er samtidig basert på en rekke antagelser om hvordan menneskets minne er bygget opp og hvordan vi lærer. Rammeverket om generaliseringsnivåene er utviklet fra arbeid med figurtall, som er annerledes enn generaliseringene elevene i denne studien utfører. For det første er det vanlig å utfordre elevenes begrunnelser i arbeid med figurtall, noe som ikke ble gjort i dette undervisningsdesignet – kun med et mindre utvalg elever i intervju. For det andre er generaliseringer i arbeid med figurtall basert på et én-dimensjonalt mønster, der kun én variabel (n) endrer seg. I denne studien endrer begge bokstavene a og b seg (sidelengdene på rektangelet) – rekken er to-dimensjonal, og det abstrakte nivået er dermed høyere. For det tredje foregår elevenes generaliseringer i et skille mellom programmering og algebra, og det kan være vanskelig å vite om elevene assosierer sine generaliseringer med programmering, algebra, rektangelet de tegner, eller noe helt annet. Det er vanskelig å vite hva rammeverket ikke viser i en slik kontekst, og ideelt sett burde man hatt et rammeverk tilpasset situasjonen i denne studien. Elevenes generaliseringsnivåer er således vist innenfor rammeverkets begrensninger. Studiens svakheter til tross, er den bygget på et solid og bredt teoretisk grunnlag, og forskningsspørsmålet er belyst ved to ulike trianguleringer: gjennom tre ulike teoretiske prinsipp med medfølgende teorier, og ved bruk av tre ulike datakilder.

Datatriangulering var en stor styrke i undersøkelsen. I forkant av datainnsamlingen antok jeg at intervjuene skulle bli det datamaterialet som kunne gi mest relevant informasjon til forskningen, med støtte i de to andre datakildene. Det har vist seg at både besvarelsene og opptakene (skjerm- og lydopptak) har vært mye viktigere enn først antatt. Besvarelsene gav gode konkrete sammenligningsgrunnlag i analysen av elevenes generaliseringer. Opptakene var trolig den datakilden som bidro mest til analysearbeidet. Suksessfaktoren til opptakenes store bidrag var samarbeidsstrukturen i undervisningsdesignet som førte til at elevene kommuniserte verbalt om oppgavene og utfordringene de stod i, og som gav et veldig godt innblikk i elevenes refleksjoner underveis. Elevenes interaksjon i sammenheng med skjermopptaket, som viste hva de gjorde på skjermen, gav verdifull informasjon i analysearbeidet. I tillegg var de planlagte diskusjonsspørsmålene i oppstarten og avslutningen av undervisningsøkta viktig å få innblikk i. Tydelige kriterier for hvilke deler av opptakene som var av interesse, gjorde det overkommelig å se gjennom elleve opptak av undervisningsøktene. Datatrianguleringen fungerte dermed bedre enn først antatt.

Sandovals (2004, 2014) forskningsrammeverk i designutvikling har vist seg å fungere godt i denne undersøkelsen, fordi det bidro til god teoretisk forankring samtidig som forskningen lå nær praksisfeltet. Formulering av teoretiske prinsipp gav studien retning og tydelige føringer for hvordan undervisningsdesignet skulle utvikles med en intervensjon og flere konkrete tiltak. Utformingen av læringshypoteser var en god øvelse på å kombinere teori og lokal kontekst i forkant av selve gjennomføringen. Prosessen ved å utforme teoretiske prinsipp, undervisningsdesign og læringshypoteser var på ingen måte en lineær prosess. De ble heller utviklet parallelt, der de påvirket hverandre, og der man var nødt til å gå tilbake og gjøre endringer flere ganger. I sammenligningen av læringshypotesene og resultatene var det nyttig å bruke årsakssammenhengene i læringshypotesene til å se hvordan man burde forbedre intervensjonen og de konkrete tiltakene i undervisningsdesignet. Designutvikling viste seg å være et godt forskningsverktøy for å studere elevers utvikling av algebraiske konseptuelle ideer med programmering og programmet Scratch som verktøy.

6 Konklusjon og avsluttende kommentarer

I denne studien har jeg undersøkt hvordan elever kan utvikle algebraiske konseptuelle ideer ved å bruke generalisering og variabel i både programmering og algebra. Studien viser at målrettede didaktiske grep kan føre til at elever generaliserer og utvikler sine tolkninger av variabel i algebra. Velbegrunnede tiltak i undervisningsdesignet kan legge til rette for at elevene overfører konseptuelle ideer mellom de to fagområdene, for eksempel ved å navngi Scratch-variablene *a* og *b*, for å tilnærme seg det algebraisk symbolspråk. Elevene hadde møtt på enkle ligninger tidlig på barneskolen, men ellers hadde de ikke fått tidligere undervisning i algebra. Likevel mestret elevene å generalisere i programmering, og å overføre det til et algebraisk språk. Samtlige elever mestret generalisering, men på ulike nivåer. I tillegg utviklet flere av elevene sine tolkninger om bruk av bokstaver. Samtidig viser resultatene at elevene, som følge av kombinasjonen av programmering og algebra, møtte på utfordringer underveis. Utfordringene ble grunnlag til forslag for forbedring av undervisningsdesignet.

Studien gir flere implikasjoner for praksisfeltet. Lærerutdannere og lærere bør være bevisst på hvordan det å kombinere to fagområder kan føre til høy kognitiv belastning. Forhåpentligvis kan dette undervisningsdesignet, med forslag til forbedring, være et godt eksempel på hvordan man kan bidra til å senke denne belastningen hos elevene. Studien har sett på to av fellestrekkene mellom programmering og algebra, generalisering og variabler, men man bør også utforske andre fellestrekk mellom programmering og de andre fagområdene i matematikk. Det å bruke *løkker* (gjentagende operasjoner) i programmering, kan for eksempel være et aktuelt verktøy for utforskning av den store talls lov i sannsynlighetsregning. Studien viser også hvordan programmering kan brukes til å skape et behov for å utforske, bruke verktøy i, og/eller lære i andre fagområder.

Det fremkommer også implikasjoner for det teoretiske feltet – for lærerutdannere, didaktikere og forskere. Det er utviklet teoretisk innsikt og avdekket behov for videre forskning. Studien viser at bruk av fellestrekk i to fagområder kan føre til overgang og læring hos elever. Vi bør derfor ha mer forskning om hvilke fellestrekk programmering kan ha med andre fagområder i matematikk, og hvordan de kan være utgangspunkt for læring. Eksempler kan være bruk av funksjoner i programmering mot funksjonslære, løkker (gjentagende algoritme) med rekursive formler, differensialligninger eller utforskning av store talls lov, eller felles arbeidsmetoder som for eksempel problemløsning. Kombinasjonen av fagområdene gir dessuten utfordringer hos elevene. Denne studien viser behovet for å senke den eksterne kognitive belastningen, men det er behov for mer forskning på hvilke utfordringer kombinasjonen av fagområdene kan gi, og hvordan de bør håndteres. Studien viser også at det å kombinere kognitiv belastningsteori med matematikkdidaktisk teori bidrar til å gi et mer helhetlig bilde, som var viktig for å belyse flere av aspektene i undervisningsdesignet. Vi ser også et behov for et rammeverk som er bedre tilpasset elevenes generaliseringer, som skjer i skillet mellom programmering og algebra. Det fremstår helt klart at det bør gjennomføres replikasjon av denne studien, med flere utprøvinger i flere kontekster, for å få et bedre generaliseringsgrunnlag av studiens resultat. Til sist ville det vært nyttig å ha et datamateriale som spenner over lengre tid, for å se om undervisningsdesignet har noen langvarige konsekvenser i elevenes arbeid med algebra. Studiens funn åpner opp for flere nye spørsmål. Jeg foreslår derfor følgende forskningsspørsmål som utgangspunkt for videre forskning:

- Hvordan generaliserer elevene i arbeid med programmering og algebra?
- Hvilke algebraiske konsekvenser, positive og negative, vil elever som har prøvd ut denne studiens undervisningsdesign få på lengre sikt?
- Hvilke fellestrekk har programmering med andre fagområder i matematikk, og hvilke muligheter og utfordringer ligger det i å bruke disse fellestrekkene i elevenes læringsarbeid?
- Hvordan kan fellestrekk mellom to ulike fagområder i matematikk bidra til faglig utvikling hos elevene?

Bråting og Kilhamn (2020; 2019) har i sine to artikler stilt flere spørsmål om hvilke negative konsekvenser kombinasjonen av programmering og algebra kan gi. Jeg er enig i at vi må få mer forskning og kunnskap om utfordringer implementeringen av programmering i matematikkfaget gir, og hvordan lærere og elever bør håndtere dette. Denne studien har bidratt til feltet ved å belyse en utfordring knyttet til kognitiv belastning. Samtidig bør vi også få mer kunnskap om hvilke positive effekter som kan utvinnes ved å implementere programmering i matematikkfaget. Studien har belyst hvordan abstrakte algebraiske ideer kan utvikles ved å bruke programmering. Generalisering er for eksempel en veldig viktig del i programmerings-prosessen, som kan legge til rette for generalisering i algebra. Wing (2006) hevdet at algoritmisk tenkning var ferdigheter alle, uansett beskjeftigelse, bør utvikle. Basert på erfaringer fra denne oppgaven, vil jeg driste meg til å påstå at alle elever bør utvikle sine abstrakte ideer på tvers av fag og fagområder, for slik å bedre kunne forstå og bidra til det samfunnet vi lever i.

Referanser

Agatolio, F., Albanese, F. & Moro, M. (2018). How an Ambitious Informatics Curriculum Can Influence Algebraic Thinking of Primary School Children. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (s. 354-365): Springer, Cham. https://doi.org/10.1007/978-3-030-02750-6_27

Anderson, T. & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational researcher, 41*(1), 16-25. https://doi.org/10.3102/0013189X11428813

Benton, L., Hoyles, C., Kalas, I. & Noss, R. (2016). Building mathematical knowledge with programming: insights from the ScratchMaths project. *Constructionism in Action 2016: Conference Proceedings*: Suksapattana Foundation. Hentet fra https://discovery.ucl.ac.uk/id/eprint/1475523

Benton, L., Saunders, P., Kalas, I., Hoyles, C. & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International journal of child-computer interaction, 16*, 68-76. https://doi.org/10.1016/j.ijcci.2017.12.004

Blanton, M. L. (2008). *Algebra and the elementary classroom : transforming thinking, transforming practice*. Portsmouth, N.H: Heinemann.

Boaler, J. (2002). Learning from teaching: Exploring the relationship between reform curriculum and equity. *Journal for Research in Mathematics Education, 33*(4), 239-258. https://doi.org/10.2307/749740

Booth, J. L., Barbieri, C., Eyer, F. & Paré-Blagoev, E. J. (2014). Persistent and pernicious errors in algebraic problem solving. *The Journal of Problem Solving, 7*(1), 3. https://doi.org/10.7771/1932-6246.1161

Booth, J. L., McGinn, K. M., Barbieri, C. & Young, L. K. (2017). Misconceptions and learning algebra. I S. Stewart (Red.), *And the rest is just algebra* (s. 63-78). Springer.

Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology, 3*(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Bråting, K. & Kilhamn, C. (2020). Exploring the intersection of algebraic and computational thinking. *Mathematical thinking and learning*, 1-16. https://doi.org/10.1080/10986065.2020.1779012

Bush, S. B. & Karp, K. S. (2013). Prerequisite algebra skills and associated misconceptions of middle grade students: A review. *The Journal of Mathematical Behavior, 32*(3), 613-632. https://doi.org/10.1016/j.jmathb.2013.07.002

Carraher, D. W., Martinez, M. V. & Schliemann, A. D. (2008). Early algebra and mathematical generalization. *Zdm, 40*(1), 3-22. https://doi.org/10.1007/s11858-007-0067-7

Castleberry, A. & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds? *Currents in Pharmacy Teaching and Learning, 10*(6), 807-815. https://doi.org/10.1016/j.cptl.2018.03.019

Chimoni, M. & Pitta-Pantazi, D. (2017). The effect of two intervention courses on students' early algebraic thinking. *Tenth Congress of the European Society for Research in Mathematics Education CERME 10*. Hentet fra https://hal.archives-ouvertes.fr/hal-01914682

Cohen, L., Manion, L. & Morrison, K. (2011). *Research methods in education* (7. utg.) Routledge.

diSessa, A. A. (2018). Computational literacy and "the big picture" concerning computers in mathematics education. *Mathematical thinking and learning, 20*(1), 3-31. https://doi.org/10.1080/10986065.2018.1403544

Ely, R. & Adams, A. E. (2012). Unknown, placeholder, or variable: what is x? *Mathematics Education Research Journal, 24*(1), 19-38. https://doi.org/10.1007/s13394-011-0029-9

Franke, M. (2014). Foreword. I E. Kazemi & A. Hintz (Red.), *Intentional talk: How to structure and lead productive mathematical discussions*. Stenhouse Publishers.

Gjøvik, Ø. & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten – tidsskrift for matematikkundervisning, 30*(3), 31-37.

Grønmo, S. (2021, 20. april). kvalitativ metode. I *Store norske leksikon*. Hentet 3. desember 2020 fra https://snl.no/kvalitativ_metode

Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. *Ectj, 29*(2), 75-91. https://doi.org/10.1007/BF02766777

Hewitt, D. (2014). The Space between the Unknown and a Variable. *North American Chapter of the International Group for the Psychology of Mathematics Education*.

Jupri, A. & Drijvers, P. (2016). Student difficulties in mathematizing word problems in algebra. *Eurasia Journal of Mathematics, Science and Technology Education, 12*(9), 2481-2502. https://doi.org/10.12973/eurasia.2016.1299a

Kieran, C. (2004a). Algebraic thinking in the early grades: What is it. *The Mathematics Educator, 8*(1), 139-151.

Kieran, C. (2004b). The core of algebra: Reflections on its main activities. I K. Stacey, H. Chick & M. Kendal (Red.), *The Future of the Teaching and Learning of Algebra The 12thICMI Study* (bd. 8, s. 21-33). Dordrecht: Springer.

Kieran, C. (2006). Research on the learning and teaching of algebra: A broadening of sources of meaning. I Á. Gutiérrez & P. Boero (Red.), *Handbook of research on the psychology of mathematics education* (s. 11-49). Brill Sense.

Kilhamn, C. & Bråting, K. (2019). Algebraic thinking in the shadow of programming. *Eleventh Congress of the European Society for Research in Mathematics Education*: Freudenthal Group; Freudenthal Institute; ERME. Hentet fra https://hal.archives-ouvertes.fr/hal-02429028/

Kirschner, P., Sweller, J. & Clark, R. E. (2006). Why unguided learning does not work: An analysis of the failure of discovery learning, problem-based learning, experiential learning and inquiry-based learning. *Educational psychologist, 41*(2), 75-86.

Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg.). Oslo: Gyldendal.

Küchemann, D. (1981). Algebra. I K. M. Hart (Red.), *Children's understanding of mathematics: 11-16* (s. 102-119). London: John Murray.

Kaarstein, H., Radišić, J., Lehre, A. C., Nilsen, T. & Bergem, O. K. (2020). *TIMMS 2019. Kortrapport*. Institutt for lærerutdanning og skoleforskning, Universitetet i Oslo. Hentet fra https://www.udir.no/contentassets/37a3d93be4464299a8998258ba1ae814/timss-2019-kortrapport---nettversjon.pdf

Lee, L. (1996). An initiation into algebraic culture through generalization activities. I *Approaches to algebra* (s. 87-106). Springer.

Lerman, S. (2014). Learning and knowing mathematics. I *Masterclass in mathematics education: International perspectives on teaching and learning* (bd. 1, s. 15-26). Beaverton: Beaverton: Ringgold, Inc.

Mason, J. (1996). Expressing generality and roots of algebra. I N. Bednarz, C. Kieran & L. Lesley (Red.), *Approaches to algebra* (1. utg., s. 65-86). Dordrecht: Kluwer Academic Publishers.

Mason, J. (2017). Overcoming the algebra barrier: Being particular about the general, and generally looking beyond the particular, in homage to Mary Boole. I S. Stewart (Red.), *And the rest is just algebra* (s. 97-117). Springer International Publishing.

Mayer, R. E. (2009). *Multimedia learning* (2. utg.). Cambridge: Cambridge University Press.

Noss, R. (1986). Constructing a conceptual framework for elementary algebra through Logo programming. *Educational studies in Mathematics, 17*(4), 335-357. https://doi.org/10.1007/BF00311324

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Partanen, A.-M. & Tolvanen, P. (2019). Developing a frame for analysing different meaning of the concept of variable mediated by tasks in elemantary-school mathematics textbooks. *Nordic Studies in Mathematics Education, 24*(3-4), 59-79.

Philipp, R. A. (1992). The many uses of algebraic variables. *The Mathematics Teacher, 85*(7), 557-561. https://doi.org/10.5951/MT.85.7.0557

Postholm, M. B. (2010). *Kvalitativ metode: en innføring med fokus på fenomenologi, etnografi og kasusstudier* (2. utg.)Universitetsforlaget.

Paas, F. & Ayres, P. (2014). Cognitive load theory: A broader view on the role of memory in learning and education. *Educational Psychology Review, 26*(2), 191-195. https://doi.org/10.1007/s10648-014-9263-5

QSR International. (2021). Nvivo (Versjon 1.4.1). Hentet fra https://www.qsrinternational.com/

Radford, L. (2006). Algebraic thinking and the generalization of patterns: A semiotic perspective. *Proceedings of the 28th conference of the international group for the psychology of mathematics education, North American chapter* (s. 2-21).

Radford, L. (2018). The emergence of symbolic algebraic thinking in primary school. I *Teaching and learning algebraic thinking with 5-to 12-year-olds* (s. 3-25). Springer, Cham.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Silverman, B. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11), 60-67. https://doi.org/10.1145/1592761.1592779

Sáez-López, J.-M., Román-González, M. & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education, 97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Sandoval, W. (2004). Developing learning theory by refining conjectures embodied in educational designs. *Educational psychologist, 39*(4), 213-223. https://doi.org/10.1207/s15326985ep3904_3

Sandoval, W. (2014). Conjecture mapping: An approach to systematic educational design research. *Journal of the learning sciences, 23*(1), 18-36. https://doi.org/10.1080/10508406.2013.778204

Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., … Voll, L. O. (2016). *Teknologi og programmering for alle. En faggjennomgang med forslag til endringer i grunnopplæringen*. Oslo: Utdanningsdirektoratet.

Schanzer, E., Fisler, K., Krishnamurthi, S. & Felleisen, M. (2015). Transferring skills at solving word problems from computing to algebra through Bootstrap. *Proceedings of the 46th ACM Technical symposium on computer science education* (s. 616-621). https://doi.org/10.1145/2676723.2677238

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G. & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351-380. https://doi.org/10.1007/s10639-012-9240-x

Shute, V. J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Stacey, K. & MacGregor, M. (1997). Ideas about symbolism that students bring to algebra. *The Mathematics Teacher*, *90*(2), 110-113. https://doi.org/10.5951/MT.90.2.0110

Sung, W., Ahn, J. & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning, 22*(3), 443-463. https://doi.org/10.1007/s10758-017-9328-x

Sweller, J., van Merriënboer, J. J. & Paas, F. (2019). Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review, 31*(2), 261-292.

Sweller, J., Van Merrienboer, J. J. & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10*(3), 251-296. https://doi.org/10.1023/A:1022193728205

Tennant, G. & Colloff, K. (2014). Fruit salad algebra: alive and kicking! *Mathematics Teaching,* (239), 40. Hentet fra https://ecommons.aku.edu/eastafrica_ied/5/

The Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational researcher, 32*(1), 5-8. https://doi.org/10.3102/0013189X032001005

Usiskin, Z. (1988). Conceptions of school algebra and uses of variables. *The ideas of algebra, K-12, 8*, 19.

Utdanningsdirektoratet. (2013). *Læreplan i matematikk fellesfag* (MAT1-04). Hentet fra http://data.udir.no/kl06/MAT1-04.pdf?lang=http://data.udir.no/kl06/nob

Utdanningsdirektoratet. (2016, 29. november). Hovedresultater fra TIMMS 2015. Hentet 15. januar 2021 fra https://www.udir.no/contentassets/7b41d7e958ad41cc8596f58dfd4838d1/timss_2015_hovedresultater.pdf

Utdanningsdirektoratet. (2019a, 27.03.2019). Algoritmisk tenkning. Hentet fra https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/

Utdanningsdirektoratet. (2019b). *Læreplan i kunst og håndverk 1.-10. trinn* (KHV01-02). Hentet fra https://www.udir.no/lk20/khv01-02

Utdanningsdirektoratet. (2019c). *Læreplan i matematikk 1.-10. trinn* (MAT01-05). Hentet fra https://www.udir.no/lk20/mat01-05

Utdanningsdirektoratet. (2019d). *Læreplan i musikk 1.-10. trinn* (MUS01-02). Hentet fra https://www.udir.no/lk20/mus01-02

Utdanningsdirektoratet. (2019e). *Læreplan i naturfag 1.-10. trinn* (NAT01-04). Hentet fra https://www.udir.no/lk20/nat01-04

Van de Walle, J. A., Karp, K. S., Bay-Williams, J. M., Wray, J. A. & Brown, E. T. (2020). *Elementary and middle school mathematics : teaching developmentally* (10. utg.). Boston: Pearson.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Vedlegg

Vedlegg 1: Oppgavesett gitt til elevene i utprøvingen av undervisningsdesignet

Vedlegg 2: Intervjuguide

Vedlegg 3: Samtykkeskjema (anonymisert)

Vedlegg 1: Oppgavesett gitt til elevene i utprøvingen av undervisningsdesignet

Tegne figurer med variabler

Arbeid i par:

I denne oppgaven skal dere jobbe i par. Den ene skal være **programmerer**, mens den andre er **instruktør**. Bytt på rollene for hver oppgave dere programmerer (oppgave 2, oppgave 3, oppgave 5, oppgave 6 og oppgave 7).

Oppgave 1: Tegn et rektangel

a) Tegn et rektangel på arket dere har fått utdelt.
b) Marker på figuren hvilke sider som er like lange.
c) Regn ut omkretsen av rektangelet, men gjør så få målinger (med linjal) som mulig.

Tegn rektangelet her:	
Hvor lang er omkretsen av rektangelet?	
Hvor mange målinger måtte dere ta?	

Oppgave 2: Tegn et rektangel i Scratch (Figur 1)

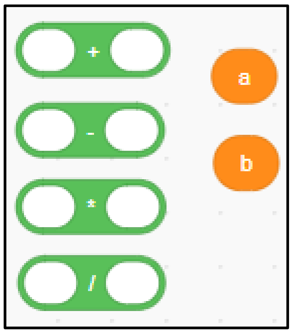
Bilde 1: startkode

Bilde 2: Bevegelse

a) Opprett et nytt program i Scratch som du gir navnet "**Figurer med variabler**".
b) Velg kategori ***Variabler*** og lag to variabler med navn **a** og **b**.
c) Kopier koden i ***bilde 1***.
d) Bruk kombinasjoner av blokkene i ***bilde 2*** til å tegne et rektangel. Du kan også bruke **løkker** for å gjøre det lettere. Det er lov å bruke flere av hver kode.
e) Kjør programmet og forklar for hverandre hva utfallet blir.
f) Kjør programmet flere ganger der dere endrer på verdiene for ***variabel a*** og ***variabel b***. Forklar hverandre hva utfallet blir.

Oppgave 3: Regn ut omkrets

a) Bytt roller!
b) Lag en ny variabel med navn **Omkrets 1**.
c) Velg kategori ***Operatorer*** og ta utgangspunkt i kodene du ser på ***Bilde 3***. Lag en algoritme som regner ut omkretsen til rektangelet som lagres i variabelen **Omkrets 1**.
d) Test om dere har gjort det riktig:
   i) Kjør programmet flere ganger, men endre på verdiene på variablene **a** og **b**.
   ii) Bruk kalkulator for å sjekke om programmet dere har laget i Scratch er riktig.



Bilde 3: Operatorer

Oppgave 4: Algebra

Hvis dere har gjort oppgave 3 riktig, vil programmet dere har laget regne ut riktig omkrets automatisk hver gang. Se nøye på hvordan dere har regnet ut omkretsen i Scratch. Diskuter hvordan dere <u>**tror**</u> dette kan skrives med et matematisk språk, og skriv ned på neste side.

Skriv forslaget deres her:

Oppgave 5: Tegn et annet rektangel (Figur 2)

a) Bytt roller!

b) Tegn et nytt rektangel som har omkretsen
   $3a + 2b + 3a + 2b.$
   Tegn det nye rektangelet ved siden av det forrige.
   Husk at du kan bruke **løkker** for å forenkle algoritmen.

c) Regn ut omkretsen som du lagrer i en ny variabel **Omkrets 2**.

d) Skriv ned hvordan dere tror omkretsen kan skrives på et matematisk språk.

Skriv forslaget deres her:

Oppgave 6: Tegn en annen figur med samme omkrets (Figur 3)

Bytt roller!
Det algebraiske uttrykket fra oppgave tre kan forenkles til følgende uttrykk

$$6a + 4b$$

Hvordan kan du finne en annen figur som har samme omkrets?

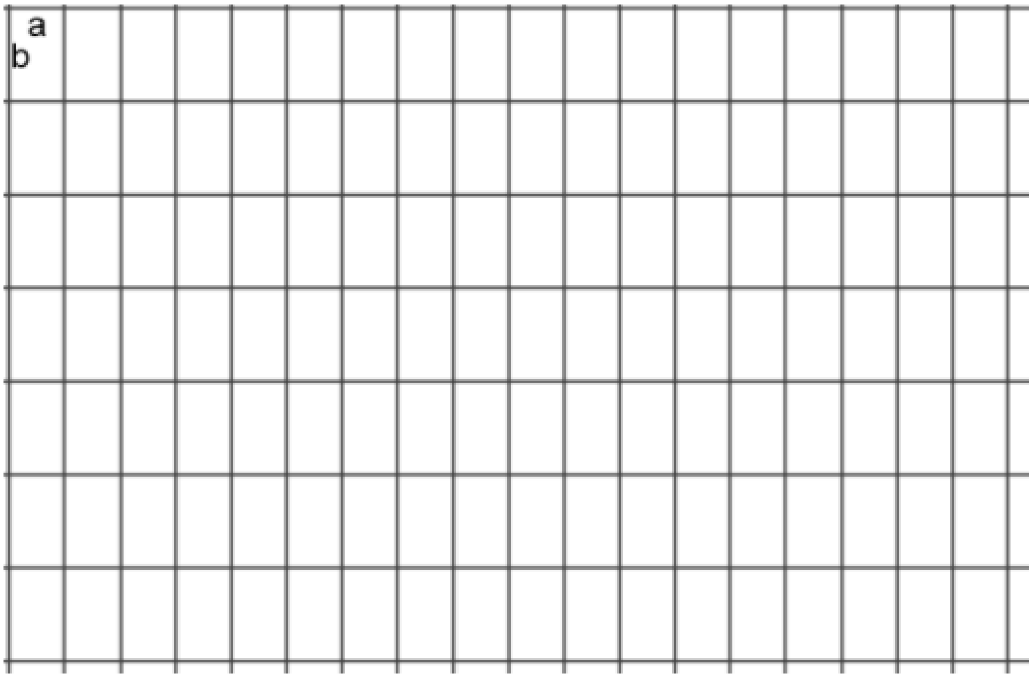
a) Kladd en figur med samme omkrets.

Kladd figuren her:

b) Tegn figuren i Scratch.

Oppgave 7: Tegn en vilkårlig figur (Figur 4)

a) Tegn en ny vilkårlig figur på rutenettet nedenfor, der bredden på hver rute tilsvarer **a** og lengden tilsvarer **b**.



b) Bytt roller!

c) Tegn figuren i Scratch.

d) Regn ut omkretsen som du lagrer i en ny variabel: **Omkrets 3**

e) Skriv ned hvordan dere tror omkretsen kan skrives på et matematisk språk.

Skriv forslaget deres her:

Vedlegg 2: Intervjuguide

Intervjuguide

Brifing

- Interessert i deres tanker og beskrivelser
    - Ingen fasit
    - Si det man tenker selv, selv om det er noe annet enn det den andre tenker
- Alt dere sier kan trekkes tilbake underveis eller i ettertid
- Lydopptaker: prøv og glem at den er der
- Har dere noen spørsmål før vi starter?

Intervjuguide

Tema	Spørsmål
1) Elevenes opplevelse av undervisningsopplegget	1) Hva synes dere om timen (og oppgaven)? 2) Hva lærte dere i løpet av timen?
2) Elevenes forkunnskaper og opplevelser ved å jobbe med programmering	**1) Har dere programmert tidligere?** a. Hva gjorde dere da? b. Barneskole/fritid? c. Likt eller annerledes enn det dere har hatt de siste ukene? 2) Hvordan synes dere det er å jobbe med programmering?
3) Elevenes forkunnskaper om variabler	**1) Har dere hørt om variabler før?** a. Når møtte dere det? b. Programmering? Hva har dere brukt det til? Hvordan? c. Matematikk? Hva har dere brukt det til? Hvordan?
4) Elevenes beskrivelser av begrepet variabler	Vise til noe de har gjort i Scratch **1) Hva brukte dere variabler til?** a. **Hvorfor** brukte dere variabler? 2) **Kan dere forsøke å beskrive hva en variabel er**? a. I lys av rammeverket b. Er det noen **forskjell i programmering og i matematikk**? 3) Kan variabler ha **forskjellige betydninger**? 4) Vise flere eksempler?
5) Elevenes beskrivelse av algebrauttrykk, de utregningene de har gjort og i hvor stor grad de mestrer å abstrahere og generalisere	Vise til noe de har gjort i Scratch **1) Hvordan regnet dere ut omkretsen av figurene?** **2) Hvorfor brukte dere variabler/tall i utregningen?** 3) **Hva betyr variablene** / beskrivelse? a. Hver enkelt i uttrykket (Er omkrets en variabel?) 4) Er dere **sikker på at utregningen er riktig**? a. Hvordan kan dere være det? 5) Algebra versus likning (funksjon) Hva er likhetene og forskjellene på disse to? 6) Vise flere eksempler?

6) Motivasjon	**1) Hva tenker dere om å bruke programmering i matematikktimene?** 2) Hva synes dere var artig i timen? 3) Klarte dere å holde fokus på oppgaven? 4) **Var det utfordrende å jobbe med både programmering og matematikk samtidig?** 5) Hva ville dere følt, og hvordan ville dere reagert hvis jeg sa at dere skal ha mer/mindre programmering i matematikktimene fremover? a. Hvorfor?
7) Refleksjon over intervjuet	**1) Har dere lært noe nytt underveis i intervjuet?** **2) Hvordan har intervjuet påvirket hva dere kan om** a. Variabler? b. Algebrauttrykk? c. Algebra? d. Programmering?
8) Sluttkommentar	1) Er det noe mer dere vil legge til, si eller kommentere helt til slutt? 2) Har dere noen spørsmål?

Debrifing

- Hvordan synes dere det gikk?
- Anonymisert

Eksempeloppgaver som kan vises under spørsmål 5-6

$$a + b$$

$$2a + 2b$$

$$2l + 2b$$

$$A = l \cdot b$$

$$O = s + s + s + s$$

$$3a + 2b - a + b =$$

Vedlegg 3: Samtykkeskjema

Forespørsel om deltagelse i studie om programmering i matematikk: Informasjonsskriv til foresatte

Jeg, Joakim Høyland, er lærer, men skriver for tiden masteroppgave i matematikkdidaktikk ved Institutt for lærerutdanning ved NTNU.

Masteroppgaven har temaet **programmering i matematikk**. I august 2020 fikk vi ny læreplan der programmering inngår som en del av opplæringen i matematikkfaget. Denne studien vil se nærmere på fordeler og utfordringer knyttet til denne endringen. Det vil bli gjennomført et undervisningsopplegg der målet er at elevene skal lære matematikk ved å bruke programmering som verktøy.

Som en del av studien gjør jeg forskning som skal bidra til ny kunnskap om elevers læring når man bruker programmering i matematikk. Studiet vil innebære at jeg tar skjermopptak av elevenes arbeid, og lydopptak av elevenes samtaler. Elevarbeid i timene, både skriftlig og digitalt vil bli samlet inn. Til slutt vil noen tilfeldig utvalgte elever gjennomføre et intervju i par, Intervjuet vil bli tatt opp på lydopptaker.

Alt datamateriale vil bli anonymisert, slik at det ikke er mulig å spore det tilbake til elevene. Datamaterialet vil bli brukt til denne studien, men i dag ønsker NSD (Norsk senter for forskningsdata) at anonymiserte og ikke-sporbare datamateriale, kan bli delt til annen forskning. Det skjer kun gjennom samtykke fra foresatte.

Jeg håper at dere som foresatte vil støtte opp om prosjektet ved å tillate at deres sønn/datter kan inngå i prosjektet som beskrevet ovenfor.

Vi ber derfor om at dere fyller ut vedlagte skjema og returnerer til elevens faglærer i matematikk.

På baksiden finner dere nærmere informasjon om formaliteter og rettigheter som angår personvern, samt aktuell kontaktinformasjon.

Med hilsen,

Joakim Høyland,
Masterstudent ved NTNU

Kontaktinformasjon:

E-post: xxx@xxx.xx

Telefon: xx xx xx xx

Informasjon om personvern og rettigheter forskningsprosjektet

Reservasjonsrett:
Det er mulig å reservere seg mot hele eller deler av datainnsamlingen vi planlegger. Eleven vil da få et likeverdig undervisningsopplegg. Det er mulig å gi tillatelse til bare deler av datainnsamlingen.

Rett til å trekke tillatelsen:
Dere har rett til å trekke tilbake tillatelse som er gitt, når som helst og uten begrunnelse. Ta i så fall kontakt med Joakim Høyland.

Innsyn og klagerett:
Dere kan når som helst få innsyn i hvilken dokumentasjon prosjektet har om deres sønn/datter, dere kan få kopi av materialet og dere kan be om å få det slettet. Hvis deres sønn/datter skal delta i intervju vil dere få beskjed om dette på forhånd. Dere har rett til å påklage manglende oppfølging til Datatilsynet.

Databehandling:
Ansvarlig for behandling av data er Joakim Høyland og Øistein Gjøvik (veileder) ved NTNU. Opptak og annet materiale som inneholder informasjon om elevene vil oppbevares i sikre lagringsmedier ved NTNU. Det er kun Joakim Høyland (masterstudent), Øistein Gjøvik (veileder) og Berit Bungum (med-veileder) som vil ha adgang til materialet, men hvis dere gir tillatelse, kan anonymisert datamateriale bli gjort tilgjengelig for andre forskningsprosjekter.
Det lagres ikke informasjon med fullt navn på elevene, men bruk av fornavn vil forekomme. I intervjuer som skrives ut (transkriberes) vil elever og lærere anonymiseres ved å gis fiktive navn.
Materialet vil lagres til masteroppgaven er ferdig og levert, som er planlagt juni i 2021. Når oppgaven er ferdig slettes og makuleres alle sporbare datamaterialer. Anonymisert datamateriale vil bli tatt vare på ved eventuell videreføring av prosjektet.

Rolleavklaring:
I forkant av datainnsamlingen vil jeg være til stede som en hjelpelærer. Det er faglærer som fortsatt vil ha ansvaret for undervisningen og vurderingen av elevene. Når undervisningsopplegget er gjennomført vil alt arbeid jeg gjør med datamateriale være som forsker, og ingenting av det jeg finner vil bli formidlet tilbake til faglærer.

Kontaktinformasjon for personvern:
Personvernombud ved NTNU:
Thomas Helgesen, e-post: thomas.helgesen@ntnu.no , tlf 93079038

Datatilsynet:
Se https://www.datatilsynet.no/om-datatilsynet/kontakt-oss/

Norsk senter for forskningsdata:
Se https://nsd.no/om/ E-post: personverntjenester@nsd.no, telefon: 55 58 21 17.

Ansvarlige for studien:
Masterstudent: Joakim Høyland, e-post: joakim.hoyland@ou.trondheim.kommune.no, tlf xx
Veileder ved NTNU: Øistein Gjøvik, e-post: oistein.gjovik@ntnu.no, tlf xx
Medveileder ved NTNU: Berit Bungum, e-post: berit.bungum@ntnu.no , tlf xx.

Samtykke til deltagelse i studien

Elevens navn:____

Foresattes navn: ____

Jeg samtykker i at eleven kan delta i (sett kryss):

- ☐ Skjerm- og lydopptak av undervisning i klasserommet.
  Dersom foresatte ikke gir samtykke til skjerm- eller lydopptak av undervisningen vil eleven få et likeverdig undervisningsopplegg, om nødvendig i et annet rom.

- ☐ Intervju i par (intervjuet vil innebære at elevene forteller om produkter de har laget, hva de synes de har lært og hvordan de har opplevd undervisningen).
- ☐ At produkter eleven har laget i forbindelse med undervisningen blir analysert.
- ☐ At anonymisert, og ikke sporbart, datamateriale kan brukes i andre forskningsprosjekter.

Eller

- ☐ Jeg samtykker <u>ikke</u> i at eleven kan delta i studien.

____

Dato og underskrift

NTNU

Kunnskap for en bedre verden