

Fakultetet for humaniora, idretts- og utdanningsvitenskap

Mastergradsavhandling

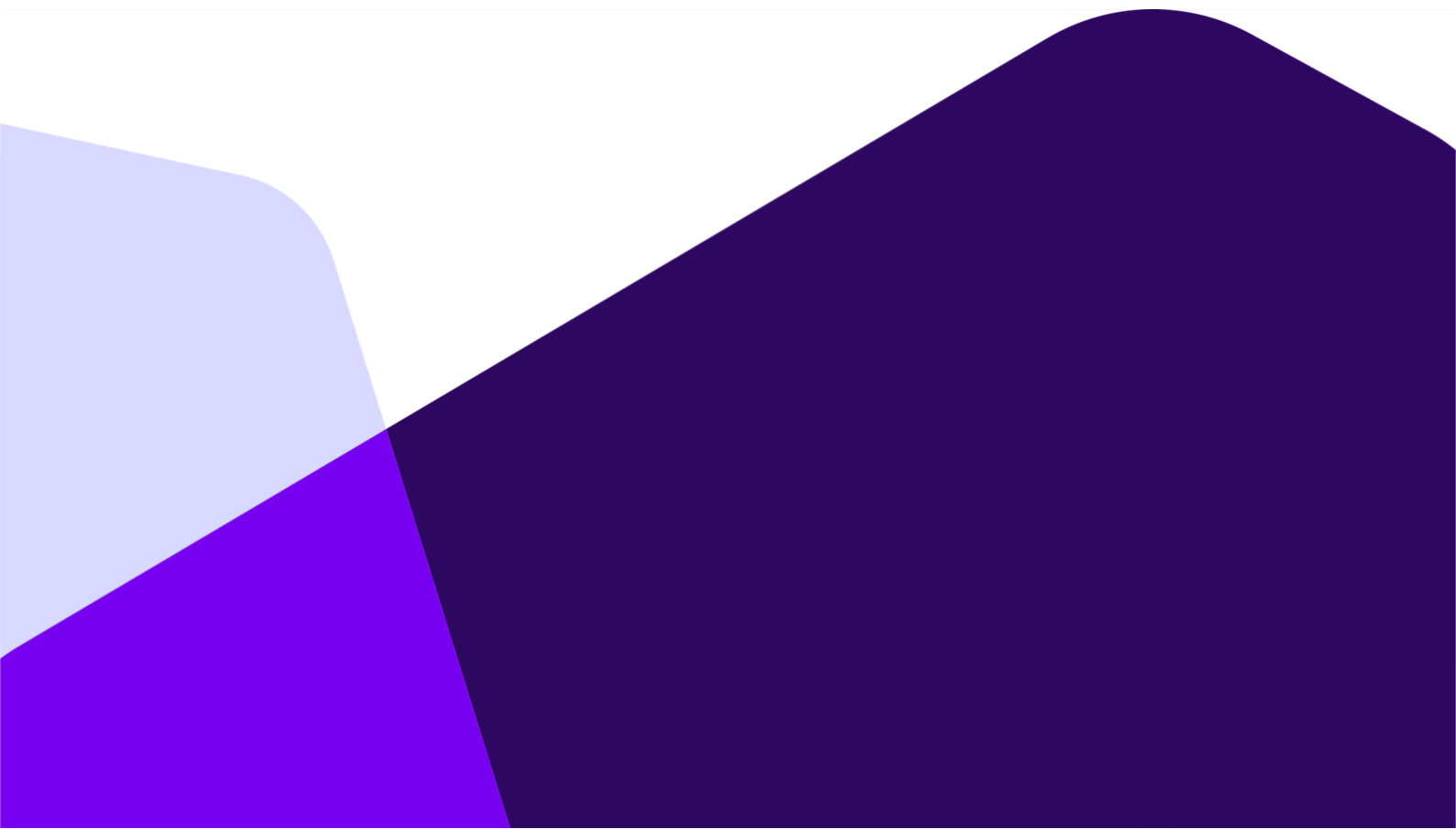
Matematikk 7 /MG2MA7

Vår 2024

Cornelia Viken

# Tegning av geometriske figurer ved hjelp av blokkprogrammering

En kvalitativ studie om elevers bruk av Scratch som verktøy i arbeid med  
algoritmisk tenkning og geometri



**Universitetet i Sørøst-Norge**

Fakultet for humaniora, idretts- og utdanningsvitenskap

Institutt for matematikk og naturfag

Postboks 4

3199 Borre

<http://www.usn.no>

© 2024 Cornelia Viken

Denne avhandlingen representerer 45 studiepoeng

# Sammendrag

I denne masteroppgaven utforsker jeg hvordan arbeid med geometrioppgaver i Scratch kan bidra til utvikling av barneskoleelevers algoritmisk tenkning, og forståelse av egenskapene til utvalgte geometriske figurer i planet. I henhold til gjeldende læreplanverk (LK20) er programmering i samspill med algoritmisk tenkning og geometri et sentralt moment for matematikkundervisningen på mellomtrinnet.

For å avgrense min studie, har jeg formulert to forskningsspørsmål. Det første forskningsspørsmålet tar for seg hvilke kjerneferdigheter innenfor algoritmisk tenkning, som er mest fremtredene under elevenes arbeid. Det andre forskningsspørsmålet tar for seg hvordan elevene uttrykker forståelse for geometriske figurer under pararbeid i Scratch. For å besvare disse forskningsspørsmålene, har jeg anvendt en kvalitativ metodisk tilnærming av casestudie og deltagende observasjon, supplert med skjerm- og lydopptak av to elevpar på 7. trinn. Kombinasjonen av disse metodene gav et detaljert bilde av gangen i oppgaveløsningen til elevene og refleksjoner som ble gjort underveis.

Datainnsamlingen fra observasjon og skjerm- og lydopptak ble analysert i henholdt til Braun og Clark (2006) tematisk analyse. Resultatene fra analysen viser at elevene tok i bruk de fem kjerneelementene feilsøking, algoritmebehandling, generalisering, dekomponering og automatisering i deres oppgaveløsning. Deres geometriske forståelse ble uttrykt gjennom usikkerhet knyttet til flere av egenskapene ved de geometriske figurene, gjennom et upresist språk, feil bruk av begreper og stor grad av eksperimentering. Funnene indikerer at elevene i stor grad befant seg mellom van Hieles nivå 0 og 1.

Masteroppgaven konkluderer med en oppfordring til lærer om å være bevisst på eget matematikkspråk og bruk av begreper i undervisning. I tillegg understrekes viktigheten av å gi elevene en grundig opplæring i analog programmering som en introduksjon til digital programmering.

# Abstract

In this master's thesis, I explore how working with geometry tasks in Scratch can contribute to developing elementary school students algorithmic thinking and understanding of the properties of selected geometric figures in the plane. According to the current curriculum (LK20), programming in conjunction with algorithmic thinking and geometry is a central aspect of mathematics education at the intermediate level.

To narrow the focus of my study, I have formulated two research questions. The first research question addresses which core skills in algorithmic thinking are most prominent during student's work. The second research question explores how students express their understanding of geometric figures during pair work in Scratch. To answer these research questions, I employed a qualitative methodological approach involving case studies and participant observation, supplemented with screen- and audio recordings of two pairs of 7<sup>th</sup> grade students. The combination of these methods provided me with a detailed picture of the students' problem-solving process and the reflections made along the way.

The data collection from observation and screen- and audio recordings was analyzed according to Braun and Clark's (2006) thematic analysis. The results of the analysis show that the students utilized the five core elements of debugging, algorithm processing, generalization, decomposition, and automation were prominent in their problem-solving. Their geometric understanding was expressed through uncertainty related to several properties of the geometric figures, imprecise language, incorrect use of terms, and a high degree of experimentation. The findings indicate that the students largely operated between van Hiele's levels 0 and 1.

The master's thesis concludes with a call for teachers to be conscious of their mathematical language and use of terms in teachings. Additionally, it emphasizes the importance of providing students with thorough training in analog programming as an introduction to digital programming.

# Innholdsfortegnelse

<b>Sammendrag</b> .....	<b>ii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Innholdsfortegnelse</b> .....	<b>iv</b>
<b>Forord</b> .....	<b>vii</b>
<b>1 Innledning</b> .....	<b>1</b>
1.1 Bakgrunn for valg av tema .....	2
1.2 Formål, problemstilling og forskningsspørsmål .....	3
1.3 Avgrensninger.....	5
1.4 Begrepsavklaring .....	5
1.4.1 Programmering.....	5
1.4.2 Scratch- et blokkbasert programmeringsverktøy .....	5
1.5 Struktur .....	6
<b>2 Teori og tidligere forskning</b> .....	<b>7</b>
2.1 Sosiokulturell læringsteori.....	7
2.1.1 Språket som psykologisk redskap .....	8
2.1.2 Den proksimale utviklingssonen.....	9
2.2 Papert og konstruksjonismen .....	10
2.3 Algoritmisk tenkning .....	12
2.3.1 Internasjonal forskning på den algoritmiske tenkeren .....	13
2.3.2 Bocconi et al. (2016) rammeverk.....	16
2.3.3 Brennan og Resnicks (2012) tre dimensjoner .....	19
2.4 Forståelse av geometri .....	20
2.4.1 Ord, begreper og definisjoner .....	21
2.5 Utvikling av geometrisk tenkning .....	22
2.5.1 Van Hieles nivåer.....	23

2.6	Oppsummering av teori og tidligere forskning.....	25
<b>3</b>	<b>Metodisk tilnærming.....</b>	<b>26</b>
3.1	Forskningsdesign .....	26
3.1.1	Datainnsamling .....	27
3.1.2	Casestudie .....	28
3.1.3	Observasjon.....	28
3.2	Utvalg .....	30
3.2.1	Presentasjon av informanter.....	31
3.3	Pilotering.....	31
3.4	Utforming av programmeringsaktiviteter i Scratch.....	33
3.5	Tematisk analyse .....	35
3.5.1	Fremgangsmåte for analyse av observasjoner .....	35
3.6	Reliabilitet og validitet .....	38
3.7	Forskningsetiske vurderinger.....	39
3.8	Oppsummering av metodisk tilnærming .....	41
<b>4</b>	<b>Resultater .....</b>	<b>42</b>
4.1	Hvilke kjerneferdigheter innenfor algoritmisk tenkning er mest fremtredene i elevenes arbeid med Scratch? .....	42
4.1.1	Feilsøking.....	43
4.1.2	Algoritmebehandling .....	46
4.1.3	Generalisering .....	49
4.1.4	Dekomponering.....	50
4.1.5	Automatisering.....	52
4.2	Hvordan uttrykker elevene sin forståelse for geometriske figurer gjennom arbeidet med Scratch?.....	53
4.2.1	Geometriforståelse i praksis.....	53
4.2.2	Ord, begreper og definisjoner i geometrien .....	57
4.3	Oppsummering av resultater.....	59
<b>5</b>	<b>Diskusjon.....</b>	<b>60</b>

5.1	Utvikling av algoritmisk tenkning gjennom oppgaveløsning i Scratch.....	60
5.2	Utvikling av geometrisk kompetanse gjennom programmering i Scratch .....	66
5.3	Utvikling av geometrisk tenking gjennom van Hiele nivåer .....	69
5.4	Oppsummering av diskusjon .....	72
5.5	Studiens begrensninger .....	73
<b>6</b>	<b>Avslutning .....</b>	<b>75</b>
6.1	Forslag til videre forskning.....	77
	<b>Litteraturliste.....</b>	<b>79</b>
	<b>Oversikt over figurer og tabeller .....</b>	<b>85</b>
	<b>Vedlegg .....</b>	<b>86</b>
	Vedlegg I: Godkjenning fra Sikt.....	86
	Vedlegg II: Informasjonsskriv til foresatte med samtykkeerklæring .....	87
	Vedlegg III: Oppgavesett brukt ved datainnsamling .....	90
	Vedlegg IV: Transkripsjonsnøkkel inspirert av Du Bois- system .....	91

# Forord

Med denne masteravhandlingen markerer jeg slutten på et inspirerende master år, og fem lærerike og flotte år på grunnskolelærerutdanningen ved USN. De siste fem årene har vært beriket med mye glede, nye vennskap, ny kunnskap og verdifulle erfaringer som jeg ikke ville vært foruten. Nå ser jeg frem til å neste kapittel, og jobb i skolen fra høsten.

Først ønsker jeg å rette en stor takk til min veileder, Pål Erik Eidsvig for verdifulle samtaler, veiledning og inspirasjonen du har gitt meg. Ditt sterke engasjement har vært til stor hjelp og motivasjon gjennom dette året. Videre vil jeg takke skolen, lærerne og elevene som bidrog til datainnsamlingen til min masteroppgave. Uten deres hjelp ville det ikke blitt det samme.

Jeg vil også takke mine medstudenter for hyggelige lunsjpauser og avbrekk under et hektisk år, og ikke mist for fantastiske vennskap som jeg vil ta med meg etter studietiden. Til slutt vil jeg rette en stor takk til familie, venner og min samboer Lars for tålmodighet, støtte, motivasjon og ikke minst, flere runder med korrekturlesning.

Drammen, juni 2024

Cornelia Viken



# 1 Innledning

På 1980- tallet var det nødvendig å tilegne seg teknologiske ferdigheter, tilknyttet databruk og programmering gjennom «datalære» (NOU 2014: 7, 2014). I dagens sammenheng utgjør de teknologiske ferdighetene en sentral del av vårt dagligliv. Den kontinuerlige teknologiske utviklingen har bidratt til å skape nye arbeidsoppgaver og yrker, samtidig som den har ført til et stadig økende krav til befolkningens ferdigheter og kunnskap, med sikte på å sikre optimal utnyttelse av ressursene (NOU 2020:2, 2020).

I lys av den teknologiske utviklingen har skolen en sentral rolle i å utvikle relevant kompetanse og ferdigheter hos elevene, slik at de er forberedt på en ukjent fremtid, der de skal lære, arbeide og leve i det 21. århundre (Sevik, 2016, s. 10). Relevant kompetanse og ferdigheter vil være viktig i fremtidens arbeidsmarked, og blir sett på som grunnleggende for elever i den moderne skolen. Ifølge Sevik (2016) beskrives ferdighetene med følgende begreper: kreativitet, innovasjon, kritisk tenkning, metakognisjon, kommunikasjon, samarbeid, digital kompetanse, digital dannelse, medborgerskap, karriere og arbeidsliv (s. 10). I arbeidet med å forberede elevene på den ukjente fremtiden og det 21. århundre, er det avgjørende å anerkjenne at digital kompetanse utgjør en grunnleggende forutsetning for å utnytte seg av de mulighetene teknologien gir til samfunnet (NOU 2020:2, 2020). Av den grunn er det vesentlig at skolen utvikler og integrerer strategier som fremmer elevens digitale kompetanse i undervisningen.

I tråd med den økende betydning av teknologi i utdanningssystemet, har digital kompetanse og bruk av digitale verktøy fått en tydelig plass i elevenes grunnutdanning (NOU 2020:2, 2020). I 2017 presenterte regjeringen (Kunnskapsdepartementet, 2017) en ny digitaliseringsstrategi for grunnskolen og videregående opplæring. Formålet med digitaliseringsstrategien var å forbedre elevenes digitale kompetanse gjennom opplæring i teknologi, og bidra til økt forståelse og mestring av algoritmisk tenkemåte og programmering blant elevene (Kunnskapsdepartementet, 2017, s. 18). I tråd med innføringen av digitaliseringsstrategien ble det innført en ny læreplan i Norge, LK20.

LK20 inneholdt flere endringer, og et tydelig fokus rettet mot samfunnets utviklingen (Kunnskapsdepartementet, 2017, s. 16). I LK20 understreker Kunnskapsdepartementet (2017) viktigheten av at elever utvikler forståelse for samspillet mellom teknologien og ulike programmer, i tillegg til å ha kjennskap til de samfunnsmessige konsekvensene digitaliseringen

fører til (s. 18). Kjennskap til forståelse og konsekvenser vil ikke bare bidra til å forberede elevene til å delta aktivt i den digitale verdenen, men også forstå at deres utdanningen har betydning for deres daglige liv og fremtid. I min studie vil jeg utforske hvordan dette kan gjøres i praksis.

## 1.1 Bakgrunn for valg av tema

I min studie vil jeg utforske hvordan arbeid med teknologi i skolen, kan bidra til å gi elevene kunnskap tilpasset det moderne samfunnet. Basert på tidligere argumenter fra Kunnskapsdepartementet og NOU, er det viktig at elevene oppfatter sammenhengen mellom anvendelsen av programmering og kjente matematiske prinsipper, tilpasset kravene i det 21. århundre. I den sammenheng ønsker jeg å utforske hvordan arbeid med geometrioppgaver i Scratch kan være med på å fremme algoritmisk tenkning og styrke elevenes forståelse av egenskaper til utvalgte geometriske figurer i planet.

Sammenfattende kan algoritmisk tenkning beskrives som en problemløsningsmetode, der elevene tilegner seg kunnskap om ulike problemer på en systematisk måte (Selvik, 2016, s. 13) [se kapittel 2.3]. Ifølge Kunnskapsdepartementet (2017) innebærer algoritmisk tenkning å identifisere problemer på en slik måte at datamaskinen kan hjelpe til med å løse dem. Dette forutsetter at man har en evne til å bryte ned problemet i logiske steg, kjent som algoritmer (s. 18). Målet med å integrere algoritmisk tenkning i LK20, var å utvikle strategier og fremgangsmåter for å løse problemer (Kunnskapsdepartementet, 2019a, s. 2), gjennom kjerneelementet *utforskning og problemløsning*.

I henhold til Kunnskapsdepartementet (2019b) defineres *kjerneelementene* i matematikk som «det viktigste faglige innholdet elevene skal arbeide med i opplæringen» (s.1). I kjerneelementet *utforskning og problemløsning* er elevenes oppgave å lete etter mønstre, oppdage sammenhenger og diskutere seg frem til en felles forståelse (Kunnskapsdepartementet, 2019c). Algoritmisk tenkning utgjør en viktig del av denne prosessen, og bidrar til å utvikle strategier og fremgangsmåter for problemløsning (Utdanningsdirektoratet, 2019).

I tillegg til sin sentrale rolle i matematikken, påvirker algoritmisk tenkning vårt dagligliv. Algoritmer muliggjør effektiv løsning av oppgaver, automatiserer flere av våre prosesser, og reduserer risikoen for feilaktig valg (Lund, 2023). Basert på algoritmenes sentrale plass i LK20 og vårt dagligliv, ble algoritmisk tenkning et av temaene for masteroppgaven min.

Det andre temaet for min masteroppgave er geometri. Geometri representerer en av de seks kunnskapsområdene i matematikken, og er avgjørende for elevenes utvikling av en god romforståelse (Kunnskapsdepartementet, 2019c). Geometri, sammen med kunnskapsområdene tall og tallforståelse, algebra, funksjoner, statistikk og sannsynlighet utgjør grunnlaget som elevene trenger for å utvikle sin matematikkforståelse. Ifølge Kunnskapsdepartementet (2019c) og LK20, utvikles matematikkforståelse gjennom utforskning av sammenhenger både innenfor og mellom de seks kunnskapsområdene.

I LK20, har geometri fått en større rolle i kombinasjon med programmering. Geometriens økende betydning førte til større mulighet for utforskning av geometriske figurer og mønstre, i sammenheng med praktiske situasjoner innen programmering (Kunnskapsdepartementet, 2019a). Målet med dette var at elevene skulle legge vekt på utvikling av strategier og fremgangsmåter, fremfor selve løsningen (Kunnskapsdepartementet, 2019a). Ettersom elevenes individuelle ferdighetsnivå varierer, kan det være hensiktsmessig å anvende oppgaver som tar for seg geometrisk tankemåte (van Hiele, 1984) og strategiutvikling, for å støtte utvikling av algoritmisk tankegang og geometriforståelse blant elevene. Basert på dette ble geometri mitt andre tema for oppgaven.

På bakgrunn av observasjoner og egne erfaringer fra praksis og arbeid i skolen, har jeg observert at programmering fortsatt er utfordrende for flere lærere og elever, til tross for at dette har vært integrert i den nye læreplanen i fire år. Elevene har ofte en manglende oppfatning av programmering som en del av matematikkfaget, og oppfatter dette som en spillbasert aktivitet. På bakgrunn av dette ønsker jeg som kommende pedagog å lære mer om programmering, forberede elevene på det moderne samfunnet og bevisstgjøre de på matematikken som læres gjennom programmering.

## **1.2 Formål, problemstilling og forskningsspørsmål**

Under mitt litteratursøk var det utfordrende å finne utdypende teori om Utdanningsdirektoratets (2019) fremstilling av den algoritmiske tenkeren. Til tross for at den algoritmiske tenkeren har en sentrale plass i LK20, gjennom kjerneelementet *utforskning og problemløsning*, er Utdanningsdirektoratets (2019) fremstilling av den algoritmiske tenkeren i stor grad påvirket av internasjonale studier. På bakgrunn av dette ønsker jeg å bidra med min forskning.

Formålet med denne masteroppgaven er å få økt innsikt i hvordan programmeringsoppgaver i Scratch, fungerer som et verktøy for å utvikle algoritmisk tenkning og geometriforståelse hos elever på mellomtrinnet. Med utgangspunkt i den fremtredende posisjonen til algoritmisk tenkning, programmering og geometri i LK20, i tillegg til min personlige interesse, har jeg formulert følgende problemstillingen:

*«På hvilke måter kan arbeid med geometrioppgaver i Scratch bidra til å utvikle barneskoleelevers algoritmiske tenkning, og forståelse av egenskapene til utvalgte geometriske figurer i planet?»*

For å besvare min problemstillingen, har jeg valgt å dele den opp i to forskningsspørsmål. Den første delen av problemstillingen fokuserer på elevenes utvikling av algoritmisk tenkning gjennom arbeid med geometrioppgaver i Scratch. Ettersom både norske og internasjonale studier om algoritmisk tenkning omfatter flere nøkkelbegreper, arbeidsmåter og kjerneferdigheter, vil jeg undersøke hvilke av disse som er mest fremtredende i elevens arbeid.

På bakgrunn av utfordringen med å finne utdypende teori om Utdanningsdirektoratets (2019) fremstilling av den algoritmiske tenkeren, har jeg valgt å i å belyse dette gjennom internasjonale studiers fremstilling av kjerneelementene i algoritmisk tenkning. På denne måten ønsker jeg å supplere den norske litteraturen med internasjonale studier. Basert på dette har jeg utformer følgende forskningsspørsmål:

*1. Hvilke kjerneferdigheter innenfor algoritmisk tenkning er mest fremtredene i elevenes arbeid med Scratch?*

Den andre delen av problemstillingen fokuserer på elevenes forståelse av utvalgte geometriske figurer i planet. For å belyse dette har valgt å undersøke hvordan elevene praktiserer sin forståelse av geometri i praksis, og anvender matematikkspråket under oppgaveløsningen. Det andre forskningsspørsmålet lyder derfor som følgende:

*2. Hvordan uttrykker elevene sin forståelse for geometriske figurer gjennom arbeid med Scratch?*

## 1.3 Avgrensninger

I min masteroppgave har jeg valgt å forske på et elevperspektiv. Datainnsamlingen ble avgrenset til én skole på 7. trinn, høsten 2023. Hadde jeg valgt å samle inn data på ungdomsskolen, et annet mellomtrinn eller en annen skole, er det grunn til å tro at resultatene mine hadde hatt et annet utfall.

## 1.4 Begrepsavklaring

### 1.4.1 Programmering

Ifølge Forsström & Kaufmann (2018) er programmering prosessen «knyttet til utvikling og implementeringen av instruksjoner for dataprogrammer, slik at datamaskinen kan utføre spesifikke oppgaver, løse problemer og større menneskelige interaksjoner» (s. 19). Dette involverer å skrive koder i et programmeringsspråk for å beskrive hva programmet skal gjøre (Selvik, 2016, s. 9).

### 1.4.2 Scratch- et blokkbasert programmeringsverktøy

Ved valg av programmeringsspråk for mitt studie, ville jeg benytte et programmeringsspråk som var tilpasset elever på mellomtrinnet. Et av hovedkriteriene for mitt valg, var å sikre at programvaren var lett tilgjengelig for elevene, uten at det krevde forkunnskaper innen programmering. Basert på disse kriteriene, valgt jeg å bruke Scratch som programmeringsspråk i min oppgave.

Scratch er et blokkprogrammeringsverktøy som er enkelt utformet med flere bruksområder (Resnick et al., 2009, s. 60). Når man programmerer i Scratch setter man sammen flere blokker av kommandoer, som kjøres etter hverandre, ovenfra og ned. Blokkene inneholder ulike farger og er formkodet etter hvilken funksjon blokken har (Gjøvik & Høyland, 2022, s. 17). Når blokkene er satt sammen, og koden blir kjørt, er det en spriter som vil gjennomføre handlingen til kommandoen (Gjøvik & Høyland, 2022, s. 20). I denne oppgaven vil spriteren illustreres av en katt med navnet Felix.

## 1.5 Struktur

Masteroppgaven er organisert i seks kapitler. **Kapittel 1** fokuserer på formålet med oppgaven, bakgrunn for valg av tema, oppgavens hensikt og fremstillingen av oppgavens problemstilling og forskningsspørsmål. I **Kapittel 2** blir relevant teori som har betydning for oppgaven presentert. I dette kapitlet vil jeg ta for meg sosiokulturell læringsteori, innholdet i den første og andre bølgen av programmering og forståelse av geometri. Videre i **kapittel 3** blir den metodiske tilnærmingen presentert. Valg av metode, utvalg, pilotering, utforming av oppgavesett, beskrivelse av analyseprosess, validitet og relabilitet, og etiske betraktninger blir belyst i dette kapitlet. Deretter følger en fremstilling av resultatene mine i **Kapittel 4**. Dette kapitlet er strukturert i to delkapitler, basert på mine to forskningsspørsmål. Det første delkapitlet tar for seg elevenes bruk av algoritmisk tenkning i Scratch, med fokus på kjerneferdighetene innenfor algoritmisk tenkning. I tillegg til dette vil jeg belyse hvilke nøkkelbegreper og arbeidsmetoder som er fremtredene i elevenes arbeid. I det andre delkapitlet vil jeg trekke frem hvordan elevene uttrykker sin forståelse for geometriske figurer under arbeidet i Scratch, med fokus på figural forståelse og bruk av ord, uttrykk og definisjoner. I **kapittel 5** diskuteres funnene fremstilt i kapittel 4, opp mot teori fremstilt i kapittel 2. Avslutningsvis i kapittel 6 vil jeg konkludere masteroppgave min og komme med forslag til videre forskning.

## 2 Teori og tidligere forskning

I dette kapitlet presenteres teori og tidligere forskning som er av betydning for min studie. Innledningsvis i dette kapitlet vil jeg ta for meg teorien om sosial læring, både som et perspektiv på læring og på grunn av dens relevans for min studie. Deretter vil jeg belyse innholdet i den første og andre bølgen av programmering, med særlig fokus på konstruksjonismen og fremstillingen av algoritmisk tenkning i norsk og internasjonal forskningslitteratur. Videre vil jeg undersøke hvordan elever tilegner seg geometrisk forståelse gjennom arbeid med programmering, og hvordan språket kan påvirke deres forståelse av geometri. Avslutningsvis vil jeg undersøke utviklingen av geometrisk tenkning med utgangspunkt i van Hieles teori.

### 2.1 Sosiokulturell læringsteori

I min studie har jeg valgt å utforske sosiokulturell læringsteori av to hovedgrunner. Den første grunnen til at jeg tar for meg sosiokulturell læringsteori i denne studien, er at denne teorien samsvarer med min oppfatning av at læring er en prosess som ikke foregår i en isolert situasjon, men i interaksjon med andre individer (Vygotsky, 1978; Utdanningsdirektoratet, 2019). På bakgrunn av dette vil jeg ikke gå i dybden på sosiokulturell læringsteori, men heller presentere teorien som et overordnet perspektiv for min masteroppgave. Den andre grunnen til relevansen av sosiokulturell læringsteori, ligger i observasjonen av at informantene i min datainnsamling jobbet i par, og samarbeidet om å løse gitte oppgaver. Basert på disse to hovedgrunnene, anser jeg det som hensiktsmessig å inkludere sosiokulturell læringsteori i min studie.

Den russiske psykologen *Lev Vygotsky (1896-1934)* var med på å danne grunntankene for sosiokulturell læringsteori (Imsen, 2020, s. 46). Vygotsky forstod læring og utvikling som «grunnleggende sosiale prosesser», hvor læring oppstod via andre i en sosial kontekst (Vygotsky, 1978; Wittek & Brandmo, 2014, s. 123). Han argumenterte for at utvikling som «grunnleggende sosiale prosesser», ikke kan forstås som et resultat av et utviklingsprinsipp alene, men at de ulike prinsippene vil gi deg styrke til ulike deler av livet (Imsen, 2020, s. 195-196).

På bakgrunn av dette sier Vygotsky at den menneskelige utviklingen kjennetegnes som et resultat av samspillet mellom modningen og forholdet i miljøet (Imsen, 2020, s. 195- 196). I et

klasserom blir samspillet tydelig, gjennom interaksjoner med andre medelever, der læring ikke foregår i en isolert situasjon. På samme måte som læring blir formet gjennom interaksjon med andre elever, innehar også læreren en betydningsfull rolle i å fremme elevers utvikling på flere nivåer (Vygotsky, 1978). I min oppgave vil jeg fokusere på elevperspektivet og læringen som oppstår i interaksjon med hverandre.

Säljö (2001) skriver at «kunnskap først lever i samspill mellom mennesker, og blir deretter en del av det enkelte individet og hans eller hennes tenkning / handling» (s. 9). Ut i fra dette kan vi si at kunnskap oppstår i interaksjon mellom mennesker, før det blir overført til individets tanker. Sett i lys av miljøet man deltar i, vil man tilegne seg kunnskaper på andre steder enn skolen. Samfunnet og miljøet rundt spiller en sentral rolle i individets forming, samtidig som det fungerer som en veiledning for hva som anses som hensiktsmessig å tilegne seg av kunnskap (Säljö, 2001). Et eksempel på dette er viktigheten av å tilegne seg kunnskap innenfor programmering og teknologi, med hensyn til nødvendig kompetanse og ferdigheter som kreves for fremtiden og det 21. århundre.

### **2.1.1 Språket som psykologisk redskap**

I samspillet mellom modning og forhold i miljøet, er det vesentlig å vurdere hvordan man bruker språket som et redskap til å mestre omgivelsene rundt oss. Vygotsky (1978) benyttet idéen om *redskaper*, for å beskrive hvordan individet tilegner seg kultur og felles kunnskaper (Imsen, 2020, s. 196). *Redskap* er et nøkkelbegrep innenfor sosiokulturell læringsteori. Nøkkelbegrepet omfatter de ressursene, både språklige og fysiske, som vi benytter for å forstå hva som foregår rundt oss. Vi sier at redskapene *medierer* handlingene våre (Säljö, 2016, s. 108). Ifølge Vygotsky (1978) kan vi oppnå flere presentasjoner ved å benytte redskapene til å tilegne oss nye ferdigheter. Et eksempel på *mediering* er når elever i samspill med hverandre skal løse en felles oppgave i Scratch. Dette krever at elevene aktivt må bruke språket til å kommunisere og reflektere over ulike løsningsstrategier og valg. Gjennom samspillet vil elevene lykkes med oppgaveløsningen i fellesskap med hverandre.

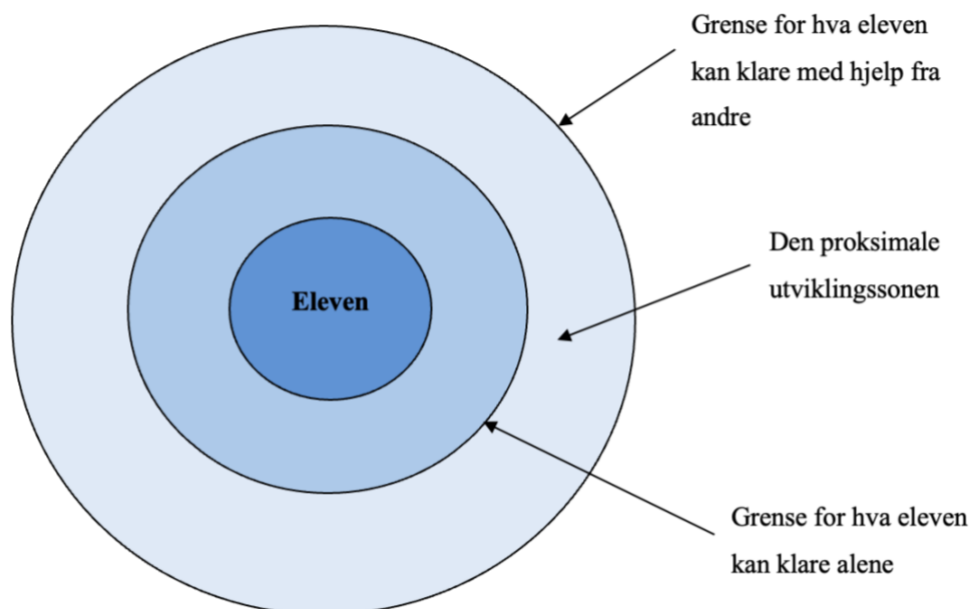
Vygotsky (1978) påpeker at vårt viktigste redskap i prosessen med å tilegne oss kultur og felles kunnskap er *språket*. Ifølge Vygotskys teori utgjør disse byggesteinene vår tenkning (Imsen, 2020, s. 197). Når barn i tidlig alder har samtaler med seg selv, bruker de språket som et redskap for å stimulere tankeprosessen.



Säljö (2001) hevder at mennesket er lærende. Med andre ord hevder han at vi tilegner oss «erfaringer og bruker dem i fremtidige sammenhenger» (s. 13). Han understreker også vår evne til å internalisere kunnskaper og ferdigheter fra andre (Säljö, 2016, s. 114). Dette blir tydelig når elever samarbeider i par for å løse en felles oppgave, og deler ulike strategier og løsningsmetoder. Gjennom interaksjon lærer elevene av hverandre, samtidig som de utvikler sine perspektiver og tilnærminger til matematikken.

### 2.1.2 Den proksimale utviklingssonen

Et annet viktig perspektiv i Vygotskys (1978) teori er *den proksimale utviklingssonen*. *Den proksimale utviklingssonen* er en modell som har til hensikt å illustrere hva eleven er i stand til å utføre på egenhånd, og med hjelp fra andre (Vygotsky, 1978). Den proksimale utviklingssonen, også kalt *den nærmeste utviklingssonen* (Imsen, 2020, s. 200), er det området mellom grensen for hva eleven klarer å utføre på egenhånd, og grensen for hva eleven kan klare å utføre med hjelp fra en annen (Vygotsky, 1978). Den pedagogiske utfordringen som oppstår i denne sonen, er å utnytte utviklingssonen til å skape et læringsmiljø som stimulerer barnet til å jobbe aktivt med andre, samtidig som du veileder det (Imsen, 2020, s. 199- 200).



Figur 2.1 Min fremstilling av Vygotskys (1978) proksimale utviklingszone

Den «andre» som hjelper til, og muliggjør at individet får til mer enn det den ellers hadde klart å utføre alene, blir på engelsk referert til som «scaffolding». På norsk blir begrepet ofte omtalt som *stillasbyggere* eller *støttende undervisning* (Imsen, 2020, s. 202). Videre i min oppgave vil jeg omtale dette som stillasbygger. En *stillasbygger* er en mer kompetent person, som kan hjelpe den som vil lære, gjennom løsningen av et problem som ellers ville vært utfordrende å løse på egenhånd (Vygotsky, 1978). Når en *stillasbygger* underviser, eller prøver å hjelpe til med å løse et problem, oppstår dette i *den proksimale utviklingssonen*. Jo vanskeligere oppgaven eller problemet er, desto mer hjelp skal eleven få (Imsen, 2020, s. 202). I et klasserom blir ofte læreren sett på som en viktig *stillasbygger* for elevene, gjennom veiledning. Stillasbygging pågår frem til eleven er i stand til å løse problemet på egenhånd (Vygotsky, 1978). Medelever kan også opptre som stillasbyggere gjennom samarbeid i grupper eller par, hvor de hjelper hverandre til å mestre mer. Medelever som *stillasbyggere* vil være sentralt for denne oppgaven, da elevene arbeider i par.

Ved å se hvordan medelever kan støtte hverandre som stillasbyggere i løsningen av geometrioppgaver i Scratch, er det tydelig at hjelp fra en *stillasbygger* er med på å utvikle *den proksimale utviklingssonen* til eleven (Vygotsky, 1978). Gjennom samarbeide og hjelp, legger elevene grunnlaget for en dypere forståelse av geometriprinsippet og hvordan de skal mestre programmeringen i Scratch.

## 2.2 Papert og konstruksjonismen

Oppgavesettet elevene jobber med, tar for seg flere programmeringsoppgaver med søkelys på algoritmisk tenkning og geometri. Før jeg går videre til det første temaet for oppgavesettet, algoritmisk tenkning, også kjent som *den andre bølgen av programmering* (Litherland, 2021), er det hensiktsmessig å først ta for seg *den første bølgen av programmering*.

I USA, på 1960- tallet ble *den første bølgen av programmering* introdusert for skolen gjennom LOGO- programmering (Litherland, 2021). LOGO-programmeringen var ment som et mulig rammeverk for undervisning i matematikk, og var i stor grad basert på Seymour Paper idéer (Lye & Koh, 2014, s. 52). Papert argumenterte for at programmering ville gjøre elevene rustet for problemløsning (Selvik, 2016, s. 8). Med sin idé, var han motstander av at elever skulle «programmeres» av et spesielt designet programvare, en lærer eller et læreplanverk. Papert mente at elevene skulle være aktive skapere av ulike programmer, ikke passive brukere av et

program (Litherland, 2021). Elevene ville lære best når de var aktive og engasjerte, slik at de kunne utforske, skape og tenke på egenhånd.

Idéen om konstruksjonismen ble først introdusert av Seymour Papert og Idit Harel's i 1991, gjennom deres artikkel «Constructionism». I «Constructionism» fremhevet Papert og Harel (1991) konstruksjonismen som et tilnærming til å tenke på konstruksjon av kunnskap gjennom «learning- by- making». «Learning- by- making» vektlegger læring som å «bygge kunnskapsstrukturer», uavhengig av omstendighetene ved læringen (Papert, 1991). Gjennom kreativitet, vil man få innsikt i hvordan ulike idéer blir utviklet og formet gjennom ulike situasjoner, før det videre blir bearbeidet til individets sinn. Denne tilnærmingen skifter fokus fra generelle prinsipper til elevens interaksjoner, hvor elevene i samspill velger sine personlige favoritter blant representasjoner, objekter og verktøy (Ackermann, 2001, s. 4). Dette er en tilnærming som gir elevene mulighet til å utforske og forme kunnskap gjennom praktisk utforskning og kreativitet.

I sin bok «Mindstorms: Children, computers and powerful ideas» belyser Papert (1993) hvordan datamaskinen ble en større del av menneskenes hverdag. Med utgangspunkt i datamaskinens sentrale plass, ønsket han å forklare hvordan teknologien kunne påvirke menneskets tanker og måter å lære på (Papert, 1993, s. 3). Med sin forklaring ble Papert ansett som konstruksjonismen far, hvor han betraktet læring gjennom skapelse av objekter som et sentralt perspektiv (Litherland, 2021). Papert (1993) hevdet at programmeringsspråket LOGO ikke bare ville lære elevene matematikk, men også bidra til å utvikle deres kognitive ferdigheter og tenkemåte (Litherland, 2021). Med dette understreket Papert viktigheten av å gi elevene mulighet til å aktivt delta i egen læring, slik at de kan utforske og anvende teknologien som et verktøy for problemløsning.

Papert (1991) hevdet at læring oppstår underveis i en prosess. Det å nå frem til en løsning er ikke nødvendigvis det viktigste, men prosessen med å arbeide mot målet som bidrar til læring og utvikling hos den enkelte (Papert, 1991). Et eksempel på dette er når elever arbeider med programmering i Scratch. Her får de muligheten til å utforske forskjellige kombinasjoner, fremfor å følge en bestemt oppskrift for å nå ønsket resultat. Gjennom refleksjon over ulike løsninger og justering underveis, jobber elevene med problemløsning og en «learning- by making» tilnærming.

## 2.3 Algoritmisk tenkning

Den andre bølgen av programmering tar for seg *algoritmisk tenkning* (Litherland, 2021). Før jeg utdyper begrepets utvikling, vil det være passende å se hvordan begrepet blir presentert i norsk og intensjonal forskningslitteratur. Utdanningsdirektoratet (2019) påpeker i sin artikkel «Algoritmisk tenkning» at det er flere ulike definisjoner av algoritmisk tenkning, men at hovedtrekkene i definisjonene er sammenfallende. I sin artikkel definerer de *algoritmisk tenkning* slik:

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

*Algoritmisk tenkning* er en problemløsningsmetode, som krever en systematisk tilnærming til problemer, både ved formulering og ved utvikling av nye løsninger. Dette innebærer at elevene skal lære seg å bli kjent med ulike problemer på en systematisk måte. Når problemet er identifisert og brutt ned til mindre og mer håndterlige delproblemer, kan elevene foreslå løsninger og bruke sin teknologiske kompetanse til å løse problemene ved hjelp av en datamaskin (Selvik, 2016, s. 13).

Utdanningsforbundet (2019) har utviklet en figur som illustrerer den algoritmisk tenkeren. Denne figuren presenterer viktige nøkkelbegreper som omfatter *algoritmisk tenking*, sammen med typiske arbeidsmåter som anvendes for å løse problemer innenfor *algoritmisk tenkning*. Figurens fremstillingen baserer seg på den britiske forskningen fra Barefoot Computing (Utdanningsdirektoratet, 2019).



Figur 2.2 Utdanningsdirektoratets fremstilling av den algoritmiske tenkeren

Følgende nøkkelbegreper og arbeidsmetoder danner grunnlaget for utviklingen av *algoritmisk tenking* hos elever i den norske skolen. Disse er integrert i den norske lærerplanen for å fremme teknologisk kompetanse og problemløsning i utdanningen (Litherland 2021; Utdanningsdirektoratet, 2019).

Integreringen av *algoritmisk tenking* og programmering kom sammen med den nye læreplanen og Kunnskapsløftet 2020. På bakgrunn av begrepets nye plass i læreplanen, er det et begrenset omfang med litteratur og forskning som omhandler erfaring knyttet til denne tilnærmingen i skolen. I lys av dette er det relevant å utforske hvordan algoritmisk tenkning blir presentert i internasjonal forskning, for å oppnå et bredere og mer helhetlig perspektiv på utviklingen av begrepet og dets integrering i undervisningspraksis.

### 2.3.1 Internasjonal forskning på den algoritmiske tenkeren

*Algoritmisk tenkning*, også kjent som *Computational Thinking (CT)*, har blitt et sentralt tema i internasjonal forskningen de siste årene (Grover & Pea, 2013). For leservennlighet vil jeg bruke forkortelsen *CT* når jeg referer til *computational thinking* videre i teksten. Grover og Pea påpekte i 2013 den økende interessen for CT, etter Jeanette Wing (2006) publiseringen av artikkelen «Computational Thinking» (s. 38). Selv om Seymour Papert først introdusert begrepet CT, hadde Wing en visjon om å tilpasse begrepet, det 21. århundre. I sin artikkel definerte Wing CT slik:

“Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science” (Wing, 2006, s. 33).

Wing (2006) sitt ønske om å tilpasse begrepet, skyldes hennes tanker om at CT er en grunnleggende ferdighet, som det forventes at alle skal lære og anvende, ikke bare informatikere (s. 33). CT handler om å bryte ned et problem i mindre deler for å gjøre problemet mer håndterbart. Dette representerer en tilnærming som mennesker anvender for å løse problemer, og som ikke er begrenset til datamaskiner alene (Wing, 2006). CT er en tilnærming som ikke bare reflekteres av våre kognitive evner, men også vår evne til å tilpasse oss det digitale samfunnet (Wing, 2006, s. 35).

På bakgrunn av den stadig økende digitaliseringen i samfunnet, ønsket Wing (2008) å utvide forståelsen av begrepet, CT. I 2008, gjennom sin nye artikkel «Computational thinking and thinking about computing», utvidet hun sitt perspektiv på CT (s. 3717). I denne artikkelen argumenterte hun for at CT representerer en form for analytisk tankegang, hvor essensen i tankegangen er abstraksjon. Ifølge Wing (2008) er anvendelse av datamaskiner en nødvendig prosess, som inneholder abstraksjon. Når man bruker en datamaskin, skal man ikke bare forstå hvorfor man bruker datamaskinen, men også vite hvordan man kan tenke på problemløsning på en slik måte at datamaskinen kan bidra. Den abstrakte tankegangen gir oss evnen til å forestille oss hva en datamaskin kan gjøre, og tenke utover våre generelle abstraksjoner, til fysiske dimensjoner av tid og rom (Wing, 2008, s. 3717- 3718). Et eksempel på dette er når elever i et klasserom blir oppfordret til å anvende abstrakt tankegang for å løse geometrioppgaver i Scratch. I stedet for å kun bruke Scratch som et programmeringsverktøy til å tegne geometrisk figurer, oppfordres elevene til å tenke over hvordan de kan bruke datamaskinen til problemløsning, på en mer avansert måte.

Videre i sin artikkel «Computational thinking and thinking about computing», fremhever Wing (2008) viktighetene av å lære de grunnleggende prinsippene for CT i tidlig alder. Hun argumenterte for at tidlig eksponering av de grunnleggende prinsippene for CT, ville danne et solid grunnlag for videre utvikling. Wing (2008) fremhevet at læring bør begynne allerede i barnehagealder, hvor barn begynner å utforske tall og telling på en naturlig måte (s. 3721).

For å få diskusjonen om CT videre, foreslo Wing (2010) en ny definisjon for CT sammen med Jan Cuny og Larry Snyder i deres artikkel “Computational thinking: What and why?”. Bakgrunnen for at Wing (2010) nok en gang ønsket å presentere en ny definisjon, kom som et resultat av den nye utviklingen innenfor informatikkfeltet, og den økende kunnskapen om hvordan CT kunne integreres i utdanningen. Wing, Cuny og Snyder (2010) forklarte CT som en tankeprosess, hvor man formulerer problemer og deres løsninger (s. 1). Deretter vil de formulerte løsningene bli presentert på en effektiv måte, som kan løses av en informasjonsbehandler. CT beskriver den mentale prosessen som oppstår når du skal definere et problem, slik at det er mulig å finne løsninger via beregninger (Wing, 2010, s. 1).

### **Hvordan definere Computational thinking**

Etter flere ulike fremstillinger av CT er det uenighet om hva definisjonen skal innebære. Til tross for dette, har definisjonen til Wing (2010) blitt et anerkjent referansepunkt innenfor feltet (Bocconi et al., 2016, s. 7). Basert på denne definisjonen, ble det identifisert to verdifulle aspekter ved CT i studien til Bocconi et al. «Developing Computational Thinking in Compulsory Education» (2016) også kjent som *CompuThink- studien*. I sin studie tar Bocconi et al. (2016) for seg mangelen på en entydig og klar definisjon av CT, ved å presentere flere sentrale kjernebegreper, ferdigheter og aspekter.

Ifølge studien til Bocconi et al. (2016) er det første aspektet ved CT representerer av en tankeprosess som er uavhengig av teknologi. Videre argumenterer han for at det andre aspektet ved CT representerer en spesifikk type problemløsning som innebærer distinkte evne (s. 15). Et eksempel på dette er å utforme løsninger som kan utføres av et menneske, en datamaskin, eller gjennom en kombinasjon av menneskelig og maskinell innsats (Bocconi et al., 2016, s. 15).

Til tross for at definisjonen til Wing (2010) har blitt et anerkjent referansepunkt for diskusjonen om CT, har det dukket opp flere klare definisjoner i litteraturen. Blant de mest siterte er definisjonen til Royal Society fra 2012:

“Computational thinking is the process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes” (The Royal Society, 2012, s. 29).

Denne definisjonen understreker at beregninger ikke er en utelukkende menneskelig konstruksjon, men at det også finnes i naturen, for eksempel i DNA (Bocconi et al., 2016, s. 15). CT er en grunnleggende del av vår verden, og vi ønsker at våre barn skal forstå og spille en aktiv rolle i den digitale verdenen som omgir dem. Vi skal utvikle barna til å bli teknologiskapere, ikke kun brukere av teknologien (The Royal Society, 2012, s. 29). Ved å lære om datamaskinen, kan barna utvikle sentrale ferdigheter for logisk resonnement, modellering, abstraksjon og problemløsning. Når barna anvender datamaskinen, fremheves betydning av presisjon som en sentral faktor (Bocconi et al., 2016, s. 15). Mangelen på presisjon kan føre til at programmet mislykkes, og gir umiddelbar tilbakemelding på hva som er galt. Dette fungerer på samme måte som idéen om å bryte ned et problem i delproblemer, som kan løses hver for seg (The Royal Society, 2012, s. 29).

Etter å ha etablert betydningen av CT som en sentral del av vår hverdag, er det avgjørende å se hvordan dette kan integreres i utdanningen. Gjennom å integrere CT i elevenes undervisning, vil vi ikke bare styrke deres forståelse av digitale konsepter, men også forberede dem på et teknologidrevet og digitalt samfunn (Bocconi et al., 2016). Videre vil jeg presentere to forskjellige rammeverk som tar for seg algoritmisk tenkning innenfor forskningslandskapet.

### **2.3.2 Bocconi et al. (2016) rammeverk**

I 2016 utførte Bocconi et al. en analyse av forskning på CT, hvor han presenterte funnene fra forskningslandskapet i sin rapport fra *CompuThink- studien*. Forskningen på *CompuThink- studien* resulterte i et rammeverk bestående av seks ulike kjerneferdigheter (Bocconi et al., 2016). Kjerneferdighetene blir presentert etter store variasjoner og forslag på hva algoritmisk tenkning er. De seks kjerneferdighetene er et resultat av de mest fremtredende ferdigheten i fem ledende artikler (Bocconi et al., 2016). De foreslåtte ferdighetene er i tråd med de mest fremtredende ferdighetene i den øvrige litteraturen innenfor *CompuThink- studien*.



Barr & Stephenson, 2011	Lee et al., 2011	Grover & Pea, 2013	Selby & Woollard, 2013	Angeli et al., 2016
Abstraction	Abstraction	Abstractions and pattern generalizations	Abstraction	Abstraction
Algorithms & procedures		Algorithmic notions of flow of control	Algorithmic thinking	Algorithms (including Sequencing and Flow of control)
Automation	Automation			
	Analysis			
		Conditional logic		
Problem Decomposition		Structured problem decomposition (modularizing)	Decomposition	Decomposition
		Debugging and systematic error detection		Debugging
		Efficiency and performance constraints	Evaluation	
			Generalizations	Generalization
		Iterative, recursive, and parallel thinking		
Parallelization				
Simulation				
		Symbol systems and representations		
		Systematic processing of information		

Figur 2.3 Skjermdump fra CompuThink rapporten (Bocconi et al., 2016, s.17)

Flere av ferdighetene som er presentert i figur 2.3, blir også nevnt i Wing (2006, 2008 & 2010) sine artikler om CT. Med utgangspunkt i dette kan CT defineres som en kognitiv tankeprosessprosess. Først formuleres et problem, slik at det legges til rette for utledning av en løsning ved hjelp av beregning (Bocconi et al., 2016). De sentrale elementene i denne tankeprosessene inkluderer de seks kjerneelementene: *abstraksjon, algoritmisk tenkning, automatisering, dekomponering, feilsøking og generalisering* (Bocconi et al., 2016). Bocconi et al. (2016) hevder at dersom vi betrakter kompetanse som en kombinasjon av ferdigheter, kunnskaper og holdninger, vil de seks kjerneelementene være avgjørende for å vurdere om CT utgjør en slik kompetanse (s. 19).

### Algoritmisk tenkning i Norden

Med utgangspunkt i resultater og erfaringer samlet i CompuThink- studien, ble det utarbeidet en ny rapport om hvordan man underviser i CT i Danmark, Finland, Norge og Sverige (Bocconi et al., 2018). Rapporten «The Nordic approach to introducing computational thinking and programming in compulsory education», skulle gi en oversikt over daværende status for CT og programmering i de fire landenes læreplaner og nasjonale planer. Oversikten over status skulle

presenteres sammen med nøkkelfunn og nye trender, med idéer til politiske handlinger (Bocconi et al., 2018, s. 1- 2). I denne rapporten skriver Bocconi et al. (2018) at CT har blitt forfremmet av pedagogiske interesser. Selv om det eksisterer flere definisjoner av begrepet som er i omløp, er det mulig å identifisere CT med utgangspunkt i kjerneferdighetene abstraksjon, algoritmisk tenkning, automatisering, dekomponering og generalisering (s.17). Det er verdt å bemerke seg at det er en endring i det nye rapporten fra 2018. Antall kjerneelementer er blitt redusert fra seks til fem. Reduseringen av kjerneelementer blir tydelig reflektert i rapporten, hvor tilnærmingen er sentral for rammeverket.

Videre hevder Bocconi et al. (2018) at CT også har potensiale som et verktøy for kreative problemløsninger og innovasjon innen en rekke andre fagområder. Dette understreker betydningen av CT som en avgjørende komponent i den obligatoriske læreplanen (Bocconi et al, 2018, s. 7). I deres studie påpeker Bocconi et al. (2018) at Norge vektlegger problemløsningsprosessen og metodene for å skape løsninger i sin læreplan. Denne prioriteringen er et resultat av den økende vektleggingen av digitale kompetansen i LK20. Bocconi et al. (2018) argumenterer for at vi i Norge, ønsker at elevene skal være kompetente og trygge brukere av digitale verktøy, slik at elevene er i stand til å utvikle egne løsninger (s. 12)

I artikkelen til Gjøvik og Torkildsen (2019) «Algoritmisk tenkning» presenteres en oversettelse av de fem kjerneelementene som presenteres i Bocconi et al. (2018) sin *CompuThinkNordic-rapport*. Ifølge Gjøvik og Torkildsen (2019) bidrar de fem kjerneferdighetene til en mer fremtidsrettet tilnærming av CT. Dette blir sett på som en relevant måte å jobbe med matematikken på. Nedenfor i figur 2.4 fremstilles kjerneelementene fra Bocconi et al. (2016 & 2018) sitt rammeverk, sammen med oversettelsen til Gjøvik og Torkildsen (2019).

<b>Kjerneferdigheter i algoritmisk tenkning</b>	<b>Definisjon</b>
<b>Abstraksjon</b>	Prosesen som gjør artefakter mer forståelig, ved å fjerne unødvendige detaljer og se bort ifra irrelevante opplysninger. En essensiell del ved abstraksjon er å velge en hensiktsmessig representasjon av et system.
<b>Automatisering</b>	En effektiviseringsprosess hvor en datamaskin blir instruert til å utføre en rekke gjentakende oppgaver raskt og effektivt. Man ønsker å gjøre det menneskelige bidraget minimalt ved å implementere løsningen av problemer i programmeringsspråket.
<b>Algoritmebehandling</b>	Du følger en forklart og trinnvis instruksjon. Det er en evne til å tenke i termer av sekvenser og regler som en måte å løse problemer eller forstå situasjoner på.
<b>Generalisering</b>	Assosieres med å gjenkjenne mønstre, likheter og kombinasjoner, og lage allmenne regler og metoder som fungerer på flere eksempler. Basert på tidligere løsninger og erfaringer, er dette en måte som raskt vil løse nye problemer.
<b>Dekomponering</b>	Kunne bryte opp et problem i mindre deler, slik at hver komponent kan forstås, løses, utvikles og evalueres hver for seg.
<b>Feilsøke</b>	Systematisk bruk av analyse og evaluering. Man bruker ferdigheter som testing, sporing og logisk tenkning for å forutsi og verifisere resultatet.

Figur 2.4 Min fremstilling av Bocconi et al. (2016 & 2018) og Gjøvik & Torkildsen (2019) rammeverk for algoritmisk tenkning

### 2.3.3 Brennan og Resnicks (2012) tre dimensjoner

Et annet fremtredende rammeverk innenfor CT er Brennan og Resnick (2012) sitt rammeverk «New Frameworks for Studying and Assessing the Development of Computational Thinking». Dette rammeverket basert seg på forskning av programmeringsaktiviteter blant unge Scratch-brukere. Formålet med Brennan og Resnick (2012) sitt rammeverk er å fremme utvikling av CT hos elever.

Bakgrunnen foretableringen av Brennan og Resnick (2012) sitt rammeverket, var den økende oppmerksomheten rundt CT i nyere tid (s. 1). Til tross for den økende oppmerksomheten var det uenighet om hva CT omfattet, og hvilke strategier som eksisterte for å vurdere utvikling av CT hos elevene (Brennan & Resnick, 2012, s. 2).

Brennan og Resnick (2012) sitt rammeverket består av tre ulike dimensjoner, og skiller seg fra rammeverket til Bocconi et al. (2016). De tre dimensjonene er: *computational concepts*, *computational practices* og *computational perspectives*.

### **Computational thinking concepts**

Computational thinking concepts tar for seg syv ulike konsepter som er svært nyttige i et bredt spekter av Scratch- prosjekter. Alle syv konsepter kan overføres til andre programmeringskontekster. De syv konseptene er sekvenser, loop, parallellitet, hendelser, betinger, operatører og data. Alle konseptene kan knyttes til konkrete eksempler i Scratch (Brennan & Resnick, 2012, s. 3).

### **Computational thinking practices**

Den neste dimensjonen er computational thinking practices. Denne dimensjonen beskriver konstruksjonsprosessene med å tenke og lære. Kjernen i computational thinking practices ligger i læringsprosessen, ikke på hva man lærer. Denne dimensjonen tar for seg tenke- og læringsprosessen. Gjennom observasjoner og intervjuer, observerte Brennan og Resnick fire fremtredende praksiser: å være inkrementell og iterativ, teste og feilsøke, gjenbruke og remikse, og abstrahere og modulisere (Brennan & Resnick, 2012, s. 6- 7). Sett i lys av norsk forskning er dette den dimensjonen som ligner mest på Utdanningsforbundets fremstilling av den algoritmiske tenkeren.

### **Computational thinking perspectives**

Den siste dimensjonen tar for seg samspillet og interaksjonene som oppstår mellom elevene når de arbeidet i Scratch. Computational thinking perspectives legger vekt på hvordan man forstår seg selv. I denne dimensjonen presenteres følgende tre ulike perspektiver: hvordan uttrykker elevene seg, hvordan fungerer elevene i samspill med hverandre og hvordan stiller elevene spørsmål til seg selv og hverandre (Brennan & Resnick, 2012, s. 10- 11).

## **2.4 Forståelse av geometri**

Geometri defineres som et «nettverk av begreper, måter å resonnerer på og representasjonssystemer som brukes til å utforske og analysere form og rom» (Battista, 2007, s. 843). Geometri blir sett på som kjernen i tidlig matematikk. Gjennom historien har geometri blitt betraktet som en av de mest fundamentale grenene innen matematikkens verden. Helt siden

egypterne bygde sine første pyramider, har geometri vært sentralt i vår kultur og vitenskapelige utforskning (Smith, 1958, s. 16). Som Battista (2007) sier, presenterer geometri en tilnærming for å utforske og analysere form og rom. Dette innebærer å utvikle og anvende romlig sans (s. 843). Romlig sans er essensielt i matematiske studier, og blir beskrevet som evnen til å forstå og visualisere geometriske former og relasjoner. Det inkluderer evnen til å visualisere objekter og deres romlige forhold, samt kjenne til de geometriske beskrivelsene av objektet (Sarama & Clements, 2009; Van de Walle et al., 2020, s. 548). I denne oppgaven vil elevene arbeid med utvalgte geometriske figurer i planet. Dette inkludert todimensjonale figurer som firkanter, trekkanter og vinkler (Forsström og Kaufmann, 2018, s. 27).

### 2.4.1 Ord, begreper og definisjoner

I LK20 har Utdanningsdirektoratet (2019) satt seg som mål at elevene skal engasjere seg mer aktivt muntlig innen matematikk, med sikt på å styrke deres forståelse av faget (Jensen, 2020, s. 169). Økt muntlig deltakelse i matematikk vil ikke bare fremme elevenes tenkning og læring, men også bidra til å utvikle et eget matematikkspråk. Matematikkspråket skiller seg fra det daglige språket, ved sin evne av presisjon. Bruken av riktige ord og begreper er derfor avgjørende for å både lære og kommunisere matematisk, både muntlig og skriftlig (Jensen, 2020, s. 169- 170). På grunnlag av dette er det viktig at elevene lærer å skille mellom ord, begreper og definisjoner for å mestre det matematiske språket.

Elever med et godt matematikkspråk, har evnen til å identifisere ord og begreper for å forstå fagstoffet (Jensen, 2020, s. 171). I matematikken blir *ord* brukt for å formidle *begrepene*. Begrepet utgjør det abstrakte innholdet i ordet (Stengrundet & Valbekmo, 2019, s. 3). Det er viktig at elevene lærer begrepene godt, slik at de kan ta med seg dette i videre læring. Ofte begynner begrepslæring tidlig, hvor barn bruker begrepene før de kjenner til begrepens innhold. Når elevene introduseres for et matematisk begrep, er kontinuerlig arbeid en viktig del av prosessen, slik ta de kan ta med seg begrepene videre, for å meste avansert matematikk (Jensen, 2020, s. 171; Stengrundet & Valbekmo, 2019, s. 3). Et eksempel på et geometrisk begrep i matematikken er *rektangel*. Gjennom utvikling av en god begrepsforståelse hos elevene, kan de beskrive og kjenne til et rektangel, samtidig som de forstår når det er hensiktsmessig å anvende begrepet.

Når elevene skal forklare et begrep, men bruker andre egenskaper og ord enn det de har lært, kan de møte på prototypefenomenet (Solem et al., 2017, s. 137). Prototypefenomenet oppstår

når elevene lærer navnene på ulike figurer og kobler figurene til prototyper, der de assosierer prototypene med deres typiske visuelle fremstilling med navnet på figuren, uten å vurdere de ulike variasjonene (Solem et al., 2017, s. 137- 138). Som lærer er det derfor viktig å være oppmerksom på prototypefenomenet, slik at man kan hjelpe elevene til å utvide begrepsforståelse, og gi dem riktige definisjoner.

Gode definisjoner er avgjørende for at et begrep skal gi mening for elevene. Winicki- Landman og Leikin (2000) har foreslått noen prinsipper som bør gjelde for matematiske definisjoner, disse prinsippene inkluderer:

- Å definere er å gi navn til et begrep.
- For å definere nye begrep kan bare tidligere definerte begrep brukes.
- En definisjon har med nødvendige og tilstrekkelige betingelser for begrepet.
- Definisjonen bør inneholde så få betingelser som mulig (Hovtun, 2024, s. 35).

En matematisk definisjon kan også være inkluderende eller ekskluderende. En *inkluderende definisjon* avgrensner et begrep til en undergruppe av de mer generelle begrepene (De Villiers, 1994, s. 11). Et eksempel på dette er «Et trapes er en firkant med *minst* to parallelle sider» (Hovtun, 2024, s. 37). Dette er en *inkluderende definisjon*, fordi den inkluderer parallelogram, romber og kvadrater, som sammen grupperer firkantene i et hierarki. Dette gjør at de ulike firkantene blir spesialtilfeller av en firkant (Hovtun, 2024, s. 37). På den andre siden er *den ekskluderende definisjon* avhengig av å ha en klar evne til å skille mellom ulike undergrupper av begreper (De Villiers, 1994, s. 11). Et eksempel på dette er «Et trapes er en firkant med *nøyaktig* to parallelle sider» (Hovtun, 2024, s. 37). Bakgrunnen for at dette eksempelet er ekskluderende, er fordi parallelogram, rektangel, romber og kvadrat ikke blir definert som er trapes slik som den inkluderende definisjonen (Hovtun, 2024, s. 37). Figurene blir sett på som uavhengig av hverandre.

## 2.5 Utvikling av geometrisk tenkning

Alle elever er i stand til å utvikle evnen til å tenke og resonnerer i geometriske sammenhenger. Evnen til å tenke og resonnerer i geometriske sammenhenger krever kontinuerlig og betydelig erfaring på kryss av en utviklingsprosess. De to forskerne Pierre van Hiele og Dina van Hiele-Geldolf, har grundig utforsket teorien om utviklingen av elevers geometrisk tenkning gjennom

beskrivelser av ulike tankenivåer (Van de Walle et al., 2020, s. 548). I følge van Hieles (1984) teori handler forståelsen av matematikk om å identifisere sammenhengen mellom *teoremer*. Når man forstår meningen med teoremene, kjenner man også til deres forbindelser (s. 247). For å tilegne seg van Hieles teori må individet gjennom fem ulike nivåer for geometrisk tenkning. Hvert av nivåene er avhengig av hverandre og bygger videre på de tidligere nivåenes forståelse. Nivåene blir sett på som utviklingsstudier, hvor du beveger deg i nivåene gjennom erfaringer med geometriske idéer (Fuys et al., 1988. s. 5).

Van Hieles teori (1984) har betydelig innflytelse på geometrilæreplaner over hele verden, og kan hjelpe lærere å forstå neste trinn for elevens geometriundervisning. Målet for van Hiele-teori er å forklare hvorfor så mange elever syntes at konkret geometri er utfordrende (Van de Walle et al., 2020, s. 548).

Teorien til van Hiele (1984) er delt opp i tre ulike aspekter: *eksistensen av nivåer, egenskaper til nivåene og bevegelsen fra ett nivå til det neste*. Det først aspektet *eksistensen av nivåer*, tar for seg de ulike nivåene, og hvilken geometrisk forståelse hvert nivå omfatter (Usiskin, 1982, s. 14). Det andre aspektet tar for seg *egenskapene til de ulike nivåene*. For å forstå geometrien, må man følge nivåene i deres bestemte og kronologiske rekkefølge. For å gå videre i prosessen må man først ha fullført det tidligere nivået (Usiskin, 1982, s. 14). Det tredje og siste aspektet tar for seg *bevegelsen fra et nivå til det neste*. Det er viktig at en elev mestrer nivåene i kronologisk rekkefølge, da hvert nivå inneholder et bestemt språk og symboler tilpasset sitt nivå (Usiskin, 1982, s. 15). Selv om teorien til van Hiele er delt opp i tre ulike aspekter er det selve nivåene som vil være hovedfokus for min oppgave.

I følge van Hiel (1984) kan den kognitive utviklingen i geometri, bli framskynder ved hjelp av læreren (Usiskin, 1982, s. 3- 5). Læreren og eleven kommuniserer ofte på ulike måter. Av den grunn er det viktig at læreren gir elevene gode forklaringer, tilpasset elevens nivå (van Hiele, 1984).

### **2.5.1 Van Hieles nivåer**

Van Hieles teori bidrar til å utvikle individets geometriske tenkemåte gjennom en progressiv modell, bestående av fem nivåer. I denne oppgaven vil jeg presentere de fem nivåene fra 0 – 4 basert på van Hieles egen beskrivelse i *The Child's Thought and Geometry* (van Hiele, 1984) og med beskrivelsene funnet i Usikin (1982) og Van de Walle et al. (2020).

### **Nivå 0, visualisering:**

Nivå 0 blir ofte referert til som basisnivået for geometri. På dette nivået går elevene fra å vurdere former og hvordan formene «ser ut», til å utvikle klasser eller grupperinger av former som er «like». Elever på nivå 0 gjenkjenner og navngir figurer basert på deres visuelle egenskaper. Et eksempel på dette, er en firkant. Denne figuren blir definert som en firkant «fordi den ser ut som en firkant». Videre vil elever på dette nivået, gjenkjenne et rektangel på bakgrunn av sin form. Likevel vil ikke elevene se at et rektangel er samme figur som et kvadrat. Utseende dominerer elevens oppfatning, og overstyrer elevenes tenkning om egenskapene til en figur. Når elever på dette nivået skal sortere og klassifisere ulike figurer basert på deres utseende, vil de «sette sammen disse figurene fordi de er spisse». Målet for dette nivået er utforske like og ulike figurer, samt lage grupper av de ulike figurene.

### **Nivå 1, analyse:**

Nivå 1 tar for seg egenskapene til de geometriske figurene. På dette nivået går elevene fra å vurdere ulike grupper av figurer til de individuelle figurene, hvor de fokuserer på deres egenskaper. Egenskapene til de geometriske blir identifisert gjennom en uformell analyse, bestående av observasjon. Under observasjonen identifiserer elevene de ulike egenskapene som visualiseres. På dette nivået kan en elev gjenkjenne et rektangel, og identifisere hva som utgjør figurens karakteristikk. Ved å observere at figuren har fire vinkler, at diagonalene er like lange, og at motstående sider er like, kan elevene identifisere figuren som et rektangelet. Egenskapene som blir identifisert er ikke systematisert ennå, og eleven har foreløpig ikke lagt merke til sammenhengen mellom et kvadrat og et rektangel.

### **Nivå 2, abstraksjon:**

På nivå 2 systematiseres egenskapene til figurene. Elevene beveger seg fra å vurdere egenskapene til formene, til å identifisere sammenhengen mellom figurer og de individuelle egenskapene hos den enkelte figur. På dette nivået begynner elevene å tenke på egenskapene hos geometriske figurer, uten å begrense seg til å fokusere på en bestemt figur. Elevene utvikler relasjoner mellom egenskapene og begynner å se et mønster og sammenheng mellom figurene. De kan identifisere at et kvadrat også er et spesialtilfelle av et rektangel, fordi de oppdager at figurene har flere av de samme egenskapene. Elevene engasjere seg også i hvis- da- resonnementet når de systematisere egenskapene til figurene. På dette nivået lærer elevene viktigheten av nøyaktige definisjoner.



### **Nivå 3, deduksjon:**

En elev på nivå 3 mestrer å resonere formelt innenfor deres geometriske tankegang. Elevene går fra å vurdere sammenhengen mellom egenskapene hos de geometriske figurene, til å utvikle deduktive systemer for geometrien. Tankegangen til elevene blir styrt av deduksjon, og resonnementet går fra det generelle til det spesielle. På nivå 3, kan eleven ta i bruk postulater, teoremer og bevisføring i sitt geometriske resonnement.

### **Nivå 4, rigor:**

På det siste nivået kan elevene sammenligne forskjellige systemer basert på ulike aksiomer. Elevene går fra å vurdere deduktive aksiomatiske systemer for geometri, til å sammenligne og se kontrastene hos de ulike aksiomatiske geometrisystemene. På nivå 4, kan elevene studere ulike geometriske konsepter, uten å bruke konkrete modeller eller deduksjoner i et system.

## **2.6 Oppsummering av teori og tidligere forskning**

I dette kapitlet ble relevant teori og tidligere forskning for denne masteroppgaven presentert. Først i kapitlet ble sosiokulturell læringsteori som et grunnleggende perspektiv for læring beskrevet, og sentrale begreper som *redskaper*, *språk*, *stillasbygger* og *den proksimale utviklingssonen* ble redegjort. Videre i kapitlet ble innholdet i *den første bølgen av programmering* avklart. I dette delkapitlet ble det redegjort for Seymour Papert og konstruksjonismen, og hans prinsipp «learning-by-making». Deretter ble innholdet i *den andre bølgen av programmering* avklart. I dette delkapitlet ble norsk og internasjonal forskning på algoritmisk tenkning presentert, før det jeg presentert to rammeverk for algoritmisk tenkning: Bocconi et al. (2016) og Brennan og Resnick (2012). Deretter tok jeg for meg geometri og forståelsen av konkret geometri, hvor forskjellen på *et ord*, *et begrep* og *en definisjon* ble beskrevet, og sentrale begreper som *prototypefenomenet*, og *inkluderende* og *ekskluderende* definisjoner ble presentert. Avslutningsvis blir geometrisk tenkning i lys av van Hieles fem nivåer beskrevet.

## 3 Metodisk tilnærming

I dette kapittelet vil jeg presentere og begrunne den metodiske tilnærmingen som er benyttet i denne studien: observasjon, supplert med skjerm- og lydopptak. Først vil jeg presentere forskningsdesignet som danner grunnlaget for datainnsamlingen. Deretter vil jeg presentere sammensetningen av utvalg, etterfulgt av en beskrivelse av piloteringen og utformingen av oppgavesettet. Videre vil jeg ta for meg den analytiske tilnærmingen og fremgangsmåten for analyse av observasjoner. Avslutningsvis har jeg gjort rede for de forskningsetiske vurderingene som ligger til grunn for å sikre reliabilitet og validitet i min masteroppgave.

### 3.1 Forskningsdesign

Blikstad- Balas og Dalland (2021) påpeker at et forskningsdesign representerer «en overordnet metodisk plan for den forskningen som skal gjennomføres» (s. 21). Det betyr at forskningsdesignet binder sammen forskningsspørsmål, data og mulige konklusjoner på et overordnet og metodisk nivå (Blikstad- Balas & Dalland, 2021). I min oppgave har jeg valgt å anvende det kvalitative paradigmet for å samle inn data og besvare min problemstilling. Det kvalitative paradigmet har som hensikt «å beskrive og forstå menneskers handlinger og meningsskaping i deres naturlige kontekst, hvor observasjonene foregår i naturlige og autentiske situasjoner» (Postholm & Jacobsen, 2018, s. 113). I følge Postholm og Jacobsen (2018) basere det kvalitative paradigmet seg på å innhente detaljert og omfattende informasjon for å fange opp det som ikke er tydelig definert, eller har klare rammer for å bli forstått og utforsket (s. 113). Det kvalitative paradigme skiller seg fra kvantitative studier, hvor man har som mål å samle inn målbare data (Nyeng, 2012, s. 79). I motsetning til kvantitative studier søker kvalitative studier etter å identifisere og beskrive kvaliteter ved sosiale fenomener på et mer dyptgående og detaljert nivå. Ofte inneholder disse studiene et mindre utvalgt av deltakere (Nyeng, 2012, s. 71; Høgheim, 2020, s. 64).

I denne studien har jeg valgt å anvende observasjon som metode, supplert med skjerm- og lydopptak, for å kunne besvare min problemstillingen: «*På hvilke måter kan arbeid med geometrioppgaver i Scratch bidra til å utvikle barneskoleelevers algoritmiske tenkning, og forståelse av egenskapene til utvalgte geometriske figurer i planet?*»

Min problemstilling etterspør hvordan arbeid med geometrioppgaver i Scratch, kan bidra til utvikling av algoritmisk tenkning og økt forståelse av egenskapene til utvalgte geometriske figurer i planet. Anker (2020) hevder at den aktuelle problemstillingen vil utvikle seg underveis i arbeidet, og gjennomgå flere runder med revisjon underveis i prosessen (s. 28). Bakgrunnen for revidering av problemstillingen underveis i prosessen, er at problemstillingen ble spisset underveis i prosessen (Anker, 2020, s. 28), og at resultat fra datainnsamlingen min viste til andre funn enn antatt.

### **3.1.1 Datainnsamling**

Datainnsamlingen til min studie ble gjennomført over flere faser. I den første fasen etablerte jeg kontakt med en skole jeg allerede hadde kjennskap til. Deretter tok jeg direkte kontakt med matematikklærerne på 6. trinn og 7. trinn. Faglærerne med det fiktive navnet Oda og Randi hadde ansvar for begge matematikkllassene på 6. trinn, og Pål hadde ansvar for begge matematikkllassene på 7. trinn. Etter dialog ble det besluttet å gjennomføre en pilotering på 6. trinn, og selve datainnsamlingen på 7. trinn. Bakgrunnen for at dette ble gjennomført på to ulike trinn, skyldes at klassene på 6. trinn har matematikk parallelt på timeplanen. I tillegg til dette uttrykte Oda og Randi et ønske om at begge klassene på 6. trinn skulle gjennomføre undervisningsopplegget på samme tid, slik at det ble likhet i undervisningen på trinnet.

Piloteringen foregikk over én klokke time i en helklasse situasjon. Alle informantene var inndelt i par på to og to elever, som samarbeidet i fellesskap på en datamaskin. For leservennlighet vil jeg heretter omtale informantene som elever. To elevpar ble tildelt mikrofoner og verktøy for skjermopptak. I tillegg til dette ble de utvalgte elevparene tildelt diktafoner for å sikre lydopptaket, ved eventuell teknisk svikt hos mikrofonene. Etter piloteringen besluttet jeg å bytte ut programvaren for skjermopptak [se kapittel 3.3]. Før datainnsamlingen ble gjennomført fikk jeg godkjenning fra Sikt [se vedlegg I], og distribuerte samtykkeskjemaer til de foresatte [se vedlegg II]. Bakgrunnen for at jeg måtte søke Sikt, er fordi jeg skulle samle inn personopplysninger (Sikt, u.å).

Datainnsamlingen startet med at jeg var innom den aktuelle klassen på 7. trinn, hvor datainnsamlingen skulle gjennomføres. Jeg fortalte elevene om hva som skulle foregå, hvorfor jeg skulle gjennomføre datainnsamling og at deltagelsen var frivillig uten negative konsekvenser for de som ikke ønsket å delta. Avslutningsvis distribuerte jeg samtykkeskjema til elevene, hvor de elevene som ønsket å delta i studien måtte levere tilbake signert

samtykkeskjema fra foresatte til kontaktlærere, innen kort tid. Ettersom elevene ikke har samtykkekompetanse, krevdes det at en foresatt signerte på deres vegne (Slotfeldt- Ellingsen, 2020, s. 209).

Datainnsamlingen skjedde i et eget klasserom, over én matematikktime høsten 2023. Datainnsamlingen tok for seg to elevpar, som jobbet på hver sin datamaskin med tilhørende utstyr for skjerm- og lydopptak. Elevene var ivrige og jobbet godt. Begrunnelse for valg av metode og beskrivelse, kommer nedenfor [se kapittel 3.1.2 og 3.1.3].

### **3.1.2 Casestudie**

I min masteroppgave ble det benyttet en allerede eksisterende grense for hva og hvem mine data skulle inkludere og ekskludere (Tjora, 2021, s. 47). For å besvare problemstilling valgte jeg å gjennomføre en casestudie. Postholm og Jacobsen (2018) hevder at «En casestudie er en samlebetegnelse for en rekke forskningsdesign med enkle variasjoner» (s. 63). Begrepet *case* har sin opprinnelse fra det latinske ordet *casus*, og betyr tilfelle. Case er et begrep som har til hensikt å forske på et tilfelle av noe (Andersen, 2013, s. 14). Ifølge Andersen (2013) er casestudier, intensive kvalitative studier av en eller noen få undersøkelsesenheter (s.14). Følgende studier er nyttig for de som ønsker en mer omfattende forståelse og forklaring på ulike handlinger og prosesser som oppstår. Dette er en tilnærming som gjør datamaterialet detaljert og dyptgående, på bakgrunn av dens avgrensning til en definert kontekst (Andersen, 2013, s. 14). På bakgrunn av at jeg ønsket å gå i dybden på, og observere to strategisk utvalgte elevpar, er casestudie særlig relevant for mitt masteroppgave [se kapittel 3.2].

### **3.1.3 Observasjon**

I denne studien ble observasjon som metode benyttet. Sammen med observasjon ble notasjon av feltnotater underveis og produksjon av skjerm- og lydopptak benyttet for dokumentasjon. Det er verdt å nevne at skjerm- og lydopptaket kun ble benyttet når elevene arbeidet sammen i par. Ikke under gjennomgangen av oppgavesettet, eller ved praktisk informasjon. Valg av observasjon som metode, begrunnes i metodens evne til gi gode beskrivelser av samarbeidet mellom elevparene (Andersson- Bakken et al., 2021, s. 126). I tillegg til metodens evne til å gi gode beskrivelser, er dette en metode som er lite tidkrevende for elevene (Tjora, 2021, s. 63).

Observasjonen foregikk over én klokke. Jeg og de to elevparene som skulle delta som informanter, gjennomførte datainnsamlingen i et eget klasserom. Bakgrunnen for valget om å

gjennomføre dette i et eget klasserom, er et resultat av observasjoner og erfaringer som oppstod under piloteringen i en helklassesituasjon. Skjerm- og lydopptak hjalp meg med å sikre at dataene som ble samlet inn, var nøyaktige og detaljerte. Gjennomføringen av skjerm- og lydopptak tilrettela for at jeg i senere tid kunne gå tilbake og se informantenes arbeid i Scratch steg for steg. I tillegg til å se igjennom arbeidet til elevene steg for steg, kunne jeg høre hvordan de kommuniserte og reflektere med hverandre under oppgaveløsningen. Dette gjorde at jeg kunne observere elevene uten å notere ned alt de sa. Ofte oppstår det tilfeller hvor observatøren skriver ned for mye, slik at mening og innhold kan forsvinne (Høgheim, 2020 s. 134). I mitt tilfelle kunne jeg skrive feltnotater under observasjonen og gå tilbake til skjerm- og lydopptaket for detaljerte og utfyllende funn.

Feltnotatene som ble skrevet ned, dokumenterte det som ikke ble fanget opp av skjerm- eller lydopptaket. Feltnotateter er registreringer man gjør imens man er i feltet som observeres, underveis eller etterpå (Høgheim, 2020, s.137). Dersom det oppstod interessante observasjonen, noterte jeg tidspunktet, slik at jeg kunne bruke feltnotatene til finne igjen observasjonene i skjerm- og lydopptaket.

Mitt fokus under observasjonen var kroppsspråk, samspill mellom elevene, evnen til problemløsning og anvendelse av geometrisk kompetanse. Selv om kombinasjonen av skjerm- og lydopptak gav meg et detaljert og godt bilde av elevenes arbeidsprosess, er det viktig å ta i betraktning at elevenes adferd kan ha blitt påvirket av bevisstheten om at de ble observert, i tillegg til at programmeringsprosessen og samtalen deres ble tatt opp gjennom mikrofon og skjermopptak (Brinkmann et al., 2012, s. 91). En annet moment som kan ha påvirket elevenes adferd, er valget om å gjennomføre undervisningsopplegget med fire elever i et eget klasserom, på bakgrunn av at dette ikke presenterer en naturlig situasjon for elevene.

Under datainnsamlingen valgte jeg å ta rollen som *fullt deltagende observatør*. Følgende valg ble gjort på bakgrunn av erfaringer fra piloteringen, i tillegg til at jeg var den eneste læreren som var tilstede i klasserommet under datainnsamlingen. Andersson- Bakken et al. (2021) beskriver rollen som *fullt deltagende observatør*, som en rolle hvor du er en del av miljøet du observerer og at du tar del i aktiviteten selv (s. 137). I likhet med Andersson- Bakken et al. (2021) beskrivelse av rollen som *fullt deltagende observatør* er Tjora (2021) sin kategori *observerende deltaker* også passende for å beskrive min observatørrolle. Bakgrunnen for dette er fordi Tjora (2021) sin kategori *observerende deltaker* beskriver at elevene fikk informasjon

om hva som skulle skje, hvem jeg var og at datainnsamling skulle bli brukt til forskning (s. 68). Dette samsvarer med mine valg og handlinger som ble gjennomført. For leservennlighet brukes *observasjon* videre i oppgaveteksten.

Under observasjonen jobbet elevene i par, på hver sin side av klasserommet. Dette gav elevene mulighet til å reflektere høyt i eget par, uten å være til sjenanse for hverandre. Under observasjonen var det viktig at jeg som forsker var bevisst på min egen rolle og deltakelse, slik at jeg ikke påvirket det som ble observert eller resultatene av studien (Gold, 1958, s. 221). Ved spørsmål fra elevene var jeg opptatt av å fatte meg i korthet, slik at jeg ikke påvirket eller la føringer for hvordan elevene skulle løse oppgavene.

## 3.2 Utvalg

Først i prosessen med å finne elever som kunne stille som informanter, tok jeg kontakt med matematikklærerne på 6.- og 7. trinn ved en skole jeg allerede har kjennskap til. Videre i prosessen gjennomførte vi et møte i fellesskap, hvor ble vi enige om å gjennomføre datainnsamlingen på 7. trinn. Deretter ble det distribuert samtykkeskjemaer til begge klassen. Etter at elevene hadde levert tilbake samtykkeskjemaet med signatur fra en foresatt, gjennomførte Pål og jeg et nytt møte for å diskutere hvilke fire elever som skulle observeres. I tillegg til å diskutere hvilke elever som skulle delta i min studie, diskuterte vi elevenes erfaringer med programmering, og anvendelse av Scratch for å kartlegge deres forkunnskaper.

Under utvelgelsen av de fire elevene som skulle observeres, vurderte vi flere faktorer. En vesentlig faktor var at jeg ønsket meg elever med middels til høy måloppnåelse i matematikkfaget i min forskning, slik at de matematiske kunnskapene ikke ville være til hinder for deres evne til å programmere. En annen viktig faktor, var at jeg ønsket meg elever som var muntlig aktive. Ettersom jeg samlet inn data gjennom skjerm- og lydopptak, var det fordelaktig med elever som var komfortable med å kommunisere høyt, slik at jeg kunne få innsikt i deres algoritmiske tenkning. Imidlertid er det ikke representativt å velge ut elever etter bestemte kriterier. Denne måten å velge ut elever på vil ikke være representativt for denne skolen, eller 7. trinns elever i Norge.

Årsaken til at elevene var organisert i par (to- og to), er strategisk valgt for å fremme refleksjon og kommunikasjon mellom samarbeidspartnerne. Dette skyldes at elevenes refleksjon ofte

starter med individuelle tanker. Gjennom interaksjon vil elevene uttrykke seg, og anvende språket som et *redskap* for å bygge videre på hverandres idéer (Vygotsky, 1978). I konteksten av casestudien jeg utfører, hvor jeg fokuserer på dyptgående og detaljerte observasjoner, vil dialogen mellom elevene gi meg et klarere bilde på deres tenkemåte og refleksjoner. Gjennom dialog mellom elevene vil jeg få det mest autentiske bildet av hva elevene tenker på, da deres tanker ikke kan observeres direkte. På bakgrunn av dette var det viktig at elevene som deltok i min studie var muntlig aktive og turte å uttrykke seg muntlig.

Både elevene og lærerne på 7. trinn hadde lite til ingen erfaring med bruk av programmeringsverktøyet Scratch. På bakgrunn av dette ble jeg nødt til å starte observasjonen med å introdusere elevene for «grønt flagg», «penn på», «slett alt» og «nullstilling av Felix», i tillegg til at oppgavesettet ble nøye gjennomgått.

### **3.2.1 Presentasjon av informanter**

Informantene i min studie besto av fire elever fordelt på to elevpar. Begge elevparene var tilstede under hele økten, og deltok på og gjennomførte alle oppgavene. For å ivareta elevenes personvern og anonymitet, har jeg valgt å bruke fiktive navn.

#### **Elevpar 1**

Elevpar 1 består av to gutter, som er gitt de fiktive navnene Jens og Ola. Både Jens og Ola går i klasse 7A, de har samarbeidet tidligere. Jens har erfaring med Scratch gjennom Kodeklubben ifra tidligere. Ola har aldri anvendt programmering eller Scratch tidligere. Det er verdt å merke seg at ingen av elevene har brukt Scratch for arbeid med matematikk tidligere.

#### **Elevpar 2**

Elevpar 2 består av to jenter, som er gitt de fiktive navnene Josefine og Sara. Josefine går i klasse 7A og Sara går i klasse 7B, de har ikke samarbeidet tidligere. Josefine og Sara har ingen erfaring med programmering fra tidligere, og har derfor aldri anvendt Scratch.

## **3.3 Pilotering**

I forbindelse med masteroppgaven ble datainnsamlingen først *pilotert*. *Pilotering* betyr utprøving og innebærer testing av datainnsamlingsprosessen, med en lignende populasjon før den faktiske datainnsamlingen (Høgheim, 2020, s. 164). På bakgrunn av at piloteringen ble

gjennomført på et annet trinn enn datainnsamlingen, vil det være sannsynlig at elevene hadde ulike forutsetninger for gjennomføring av oppgavesettet.

Piloteringen ble gjennomført over én klokke, i en helklassesituasjon i klasse 6A. Innledningsvis gjennomførte faglæreren til elevene en felles introduksjon og gjennomgang av oppgavesettet. Oppgavesettet som ble anvendt under pilotering inneholdt de samme retningslinjene og oppgavene som ble bruk under datainnsamlingen.

Mitt fokus under piloteringen var tidsbruk, vanskelighetsgrad på oppgavene, gruppestørrelse, plassering av elevpar i klasserommet, plassering av observatør, observatørrolle, veiledning/hjelp fra læreren, bruk av mikrofoner, programvare for skjermopptak og samspill mellom lyd og bilde. I tillegg til dette observerte jeg hvordan elevene reagerte på oppgavesettet, og om de åpnet opp for samarbeid og refleksjon. Selv om dette er mange punkter, gav piloteringen meg et godt bilde, og tilbakemelding på flere av nevnte punkter.

Etter gjennomføringen av piloteringen konkluderte jeg med at én klokke var tilstrekkelig, da elevene akkurat rakk å fullføre oppgavesettet. Oppgavens ordlyd og vanskelighetsgrad var passende og elevene forstod og gjennomførte innholdet, samtidig som de ble utfordret underveis. Bakgrunnen for at dette ble positivt mottatt av elevene, kan være et resultat av den grundige gjennomgangen av oppgavesettet, hvor elevene fikk stille spørsmål og læreren gjentok viktige funksjoner i Scratch. Gruppestørrelsen bestående av to elever, opplevde jeg som positivt. I tillegg til dette var det ingen vesentlige forskjeller mellom rene jentepar, guttepar og par som var bestående av en jente og en gutt.

Elevparene som ble observert var plassert bakerst i klasserommet for å begrense støy og forstyrrelser fra resten av klassen. Selv om plassering av elevene viste seg og fungerte godt, registrerte mikrofonene for mye lyd fra omgivelsene i rommet. I tillegg til dette benyttet elevene som ble observert hviskestemme, for å unngå at resten av klassen skulle høre deres samtaler og løsninger. Videre observerte jeg at elevene klarte seg godt på egenhånd, og at det var lite behov for veiledning fra læreren. Ved gjennomgang av skjermopptaket etter piloteringen, ble det tydelig at programvaren som ble brukt for skjermopptaket ikke var gunstig.

På bakgrunn av dette var det noen aspekter som ble forandret etter piloteringen. Det viktigste var at elevene som skulle observeres ble flyttet til et eget rom slik at støy og lyd fra andre elever



ikke ble med på opptaket. Dette ville også gi elevene større frihet til å kommunisere høyt med hverandre i plenum. Videre bestemte jeg meg for å bytte programvare for skjermopptak. Gjennom piloteringen fikk jeg mulighet til å identifisere og løse eventuelle utfordringer som oppstod under datainnsamlingen, observere om oppgaven besvarte min problemstilling og mulighetene til å styrke den. Dette gjorde at jeg var godt forberedt til den faktiske datainnsamlingen.

### 3.4 Utforming av programmeringsaktiviteter i Scratch

Under utformingen av oppgavesettet, støtte jeg på utfordringer med å tilpasse oppgavene til elevenes ferdighetsnivå innen både matematikk og programmering. Særlig var det vanskelig å tilpasse oppgavene i henhold til elevenes programmeringsferdigheter, gitt deres begrensede erfaring med programmeringsverktøyet Scratch. På bakgrunn av dette valgte jeg å tilpasse oppgaven i samsvar med følgende kompetansemål, tilpasset 6. trinn:

- *beskrive egenskaper ved og minimumsdefinisjonar av to- og tredimensjonale figurar og forklare kva for egenskaper figurane har felles, og kva for egenskaper som skil dei frå kvarandre*
- *bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster* (Kunnskapsdepartementet, 2019a)

I denne tilpasningen lot jeg meg inspirere av Brennan og Resnicks (2012) rammeverk for utviklingen av algoritmisk tenkning i Scratch, «Espens klasserom» sitt undervisningsopplegg for geometri i Scratch, og Forsström og Kaufmann (2018) sin studie, som viser at visuelle programmeringsspråk som Scratch og LOGO ofte blir anvendt til å programmere figurer i planet [se vedlegg III]. Gjennom denne tilnærmingen bidro ikke oppgavesettet til å kun fremme utviklingen av algoritmisk tenkning, men også til utvikling av elevenes geometriske kompetanse. Formålet med oppgavesettet var å tilrettelegge for at elevene kunne anvende sin eksisterende kunnskap om geometri og overføre denne til programmering i Scratch, samt utvikle algoritmisk tenkning og forståelse av egenskaper til utvalgte geometriske figurer i planet. På bakgrunn av dette var det avgjørende at oppgavene ikke krevde spesifikke forkunnskaper innen programmering, slik at dette ville være til hinder for elevene under oppgaveløsingen.

Den første oppgaven, oppgave 1a, etterspør om elevene kan hjelpe Felix med å programmere de tre ulike geometriske figurene, kvadrat, rektangel og likesidet trekant. Denne oppgaven fungerer som en oppvarming til utforskning av geometri gjennom praktisk programmering. Hensikten med denne oppgaven er å styrke elevenes forståelse av geometri. I denne oppgaven ville elevene bli bevist på viktigheten av å ha kjennskap til de geometriske egenskapene til figurene, samt kjenne til og anvende de tilhørende begrepene riktig.

I oppgave 1b skal elevene analysere og sammenligne figurene de har programmert med hverandre og deres tilhørende koder på et eget svarark. Hensikten med denne oppgaven er at elevene skal kunne beskrive mønster, likheter og forskjeller som de har identifisert gjennom deres funn. Dette er en viktig faktor for å observere hvordan elev reflekterer over egenskapene som identifiserer hver av figurene, i tillegg til dette blir de fremstilt for valgene de har tatt under programmeinrigsprosessen.

I oppgave 2 blir elevene utfordret til å programmere en sekskant i Scratch. I denne oppgaven får elevene mulighet til å praktisere og anvende sin forståelse av geometriske figurer og deres egenskaper, gjennom programmering av en mangekant. Videre i oppgave 3a, skal elevene bruke informasjonene om sekskanten til å programmere en blomst. Her oppfordres elevene til å være kreative og utforske ulike tilnærminger for å programmere en blomst som kun består av sekskanter.

I den siste oppgaven, oppgave 3b, blir elevene bedt om å telle hvor mange sekskanter de har anvendt. Dette krevet at elevene gjennomgår koden sin, evaluere resultatet og vurdere om deres figuren faktisk ligner en blomst.

Målet med denne økten er at elevene skal utvikle og anvende geometriske forståelse og algoritmisk tenkning i praksis. Gjennom denne oppgaven ønsker jeg at elevene skal tilegne seg flere metoder for problemløsning. Jeg ønsker også å belyse viktigheten av å kjenne til de grunnleggende geometriske prinsippene, begreper og ulike funksjoner i Scratch, som vil oppgaveløsningen til elevene enklere og mer effektiv.

Oppgaven jeg har utarbeidet tar hensyn til læreplanens kjerneelementer, spesielt kjerneelementet *utforskning og problemløsning*. I min oppgave blir dette tydelig når elevene skal dele opp et større problem, til et mindre mer håndterbare deler (Kunnskapsdepartementet,

2019c). For å gjøre problemet mer forståelig, må elevene benytte opplysningene de får i de første oppgavene, for å mestre de senere oppgavene. Denne kunnskapen skal til slutt benyttes til å utforme en kode i Scratch, som programmerer en visuell representasjon av en blomst.

### **3.5 Tematisk analyse**

Malterud (Anker, 2020) påpeker at en analyse «.. består av å stille spørsmål til det empiriske materialet, lese og organisere data systematisk i lys av relevant teori og gjenfortelle svarene på en forståelig og relevant måte» (s. 17). For å analysere mitt datamateriale, har jeg valgt å benytte meg av tematisk analyse. Tematisk analyse er en metode for å identifisere, analysere og rapportere mønstre i kvalitative data, som vil gi deg et organisert og beskrivende datasett i detalj (Braun & Clarke 2006, s. 80). Dette er en oppskrift for hvordan du kan lete etter svar på spørsmål i ditt datamateriale (Johannesen et al., 2018, s. 282). I mitt prosjekt er datamaterialet delt opp i 4 lydopptak, 2 skjermopptak og feltnotater fra observasjon. Skjerm- og lydopptaket er delt i to, hvor hvert elevpar har 2 lydopptak og 1 skjermopptak hver.

For å analysere datamaterialet har valgt å benytte Virginia Braun og Victoria Clarke (2006) sin beskrivelse av de seks fasene i en tematisk analyse, som retningslinjer for min analyse (s. 87). Analyseprosessen kategoriseres som induktiv, på bakgrunn av at jeg ønsker å bygge kunnskap basert på innhentet datamaterie, og overføre dette til teori (Braun & Clarke, 2006, s. 83.). Basert på dette fulgte min analyseprosess følgende steg:

Fase 1: Bli kjent med datamaterialet gjennom transkripsjon av skjerm- og lydopptak

Fase 2: Inkludering av feltnotater i transkripsjonen, og innlede koder

Fase 3: Identifisere temaene og fargekoding av transkripsjoner

Fase 4: Gjennomgang av temaer og revidere og vurdere temaer

Fase 5: Definere og navngi temaer

Fase 6: Skrive ned funn

#### **3.5.1 Fremgangsmåte for analyse av observasjoner**

I den første fasen ble jeg bedre kjent med datamaterialet. Dette ble gjort gjennom transkripsjon, kort tid etter observasjonen. For å sikre nøyaktighet utviklet jeg et sett med transkripsjonsnøkler [se vedlegg IV] inspirert av Du Bois- systemet. Du Bois- systemet er et system som deler inn

transkripsjonen ved hjelp av intonasjonsenheter. Intonasjonsenhetene indikere tonefall, nøling osv. (Nygaard, 2022). Under transkripsjonsprosessen skrev jeg også ned forslag på temaer som kunne anvendes senere i prosessen. Da jeg var ferdig med transkripsjon prosessen satt jeg igjen med totalt 62 sider med transkripsjon fra både skjerm- og lydopptaket.

I den andre fasen ble eventuelle feltnotater fra observasjon inkludert i transkripsjonsnotatene. Feltnotatene var notater som ikke kom til syne under skjerm- eller lydopptaket. Videre i den andre fasen innledet jeg koder fra dataene. Hver av kodene som ble innledet inneholdt interessante trekk ved sitater eller observasjoner på en systematisk måte. Jeg benyttet fargekoder, der hver kode fikk en «merkelapp» med tilhørende farge. Dette skulle være med på å systematisere funnene mine (Anker, 2020, s. 76). Fargekodene var må å lage et tydeligere og mer detaljert bilde av de ulike operasjonene som ble gjennomført. Nedenfor er et utklipp ifra kodingen. Den første kolonnen inneholder nummeret på sitatet, kolonne to er inneholder tidspunkt om dette er relevant, kolonne tre inneholder hvem som snakker, kolonne fire inneholder sitat, og den siste kolonnen inneholder koden som ble brukt for å systematisere funnene mine.

Tabell 1 Utdrag fra koding

278		Ola	<i>Men da blir det jo, det der er jo en åttekant</i>	Geometriske egenskaper Feilsøke
279		Jens	<i>En, to, tre, fire, fem, ja det er en åttekant det. Vi, vii, hva gjør vi nå?</i>	

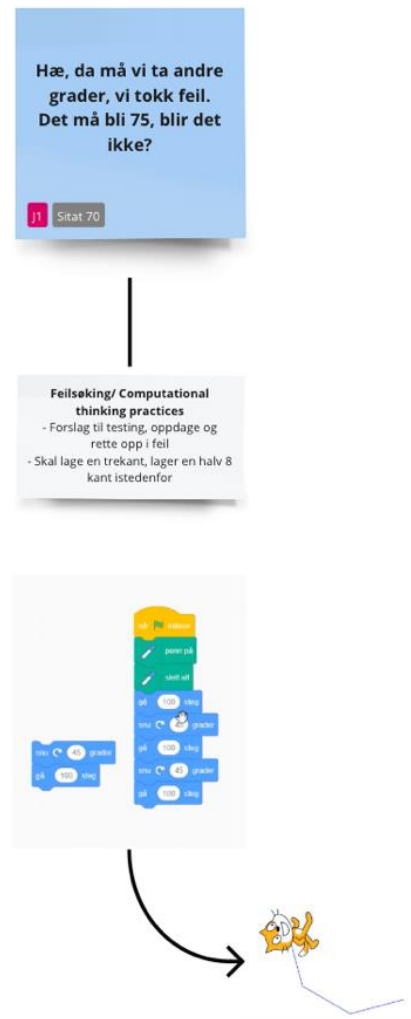
I fase tre skulle temaene identifiseres, med andre ord skulle jeg samle kodene under noe felles (Braun & Clark, 2006, s. 89). Identifisering av temaer ble iverksatt når alle dataene var kodet og sammenstilt. For å identifisere de temaene som passet min problemstilling og funn best, ble prosessen gjennomført tre ganger. Bakgrunnen for dette var at funnene jeg fikk, ikke var i tråd med problemstillingen, og jeg måtte derfor kategorisere funnene mine annerledes.

For å få et oversiktlig resultat, benyttet jeg programvaren Miro. Først fordelte jeg elevparene i to ulike grupper. Deretter kom jeg frem til følgende kategorier som jeg la inn i programmet: *programmeringsfunn, matematiske funn, samarbeidsfunn.*

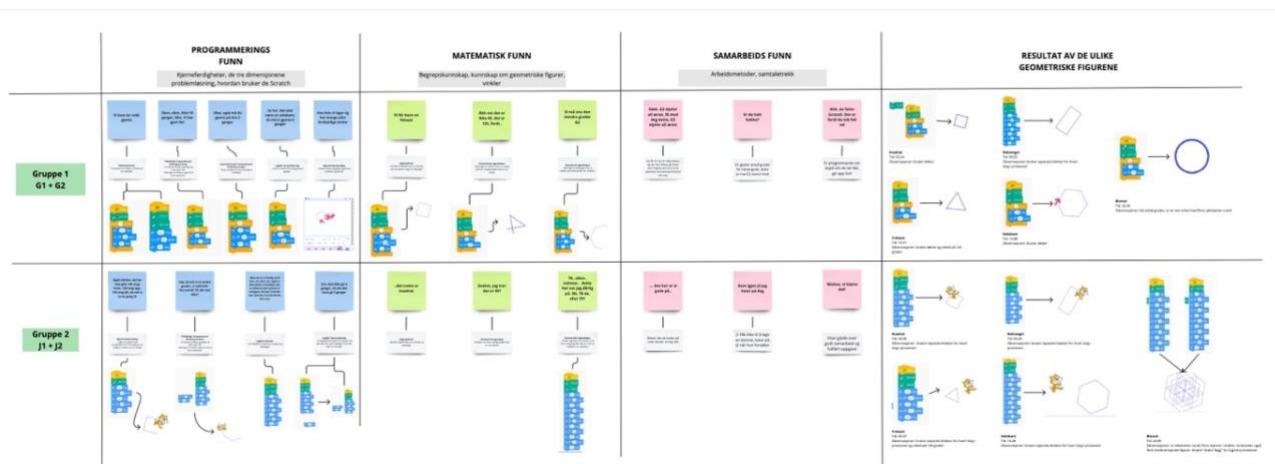
Kategori 1 *programmeringsfunn* består av: de algoritmiske kjerneferdigheter (Bocconi et al., 2016 & 2018), Brennan og Resnicks (2012) tre dimensjoner, problemløsning, Utanningsdirektoratet (2019) sin fremstilling av den algoritmiske tenkeren og bruk av Scratch som programmeringsverktøy. Kategori 2 *matematiske funn* er delt inn i figurale egenskaper, bruk av ord, begreper og definisjoner, og van Hieles nivåer (1984). Den siste kategorien *samarbeidsfunn* tar for seg samtalen mellom elevene, samspill, validering av svar og tidsbruk.

Etter at kategoriene var lagt inn i Miro, la jeg inn flere «post-it lapper» med sitat under de ulike kategoriene (se figur 3.1). Hver av «post-it lappene» inneholdt sitat nummer og «navn» på den som hadde sakt sitatet. Under sitatet la jeg til relevant teori og en kommentar, koden som var brukt og et skjermbilde av hva koden programmerte.

Videre i fase fire undersøkte jeg om alle temaene inneholdt nok data og at de samsvarte med de fremstilte kategoriene. Det endelige resultat ble seende slik ut:



Figur 3.1 Utklipp fra Miro



Figur 3.2 Min fremstilling av koder lagt inn under kategorier i Miro

Etter at alt av datamateriale var kodet og kategorisert, og eksempler var lagt inn i Miro, ble dataene mine svært strukturerte og oversiktlig. Dette forenklet også analyseprosessen min, ved at jeg enkelt kunne finne tilbake til sitater i transkripsjonen.

I fase fem skulle jeg navngi og definere temaene som var funnet. I denne fasen gjorde jeg ingen endringer. Den siste fasen, fase seks, bestod av å skrive ut funnene mine fra analysen. Dette er presentert i kapittel 4.1 og 4.2.

### 3.6 Reliabilitet og validitet

I sammenheng med forskningen til masteroppgaven min er det to viktige kriterier som diskuteres for å vurdere forskningens *reliabilitet* og *validitet*. På bakgrunn av dette er det viktig å være tydelig på hvilke prosedyrer som er gjennomført under forskningen og påpeke at funnene som blir presentert er begrenset til min forståelse og tolkning av de konkrete situasjonene som har blitt studert. *Reliabilitet* tar for seg hvor tillitvekkende dataene er, med andre ord om de er til å stole på (Nyeng, 2012, s. 105). Anker (2020) beskriver *reliabilitet* som et omdiskutert begrep, hvor det er vanlig å anvende *pålitelighet* i stedet for *reliabilitet* innen kvalitativ forskning (s. 108). For leservennlighet vil jeg videre i mitt studie bruke *pålitelighet*.

Pålitelighet er en fremstillingen av datamaterialet som er godt gjennomført og på en etterrettelig måte, hvor forskeren gjør rede for og forklarer valgene som er gjort i prosjektet (Anker, 2020, s. 108- 109). Forskningens pålitelighet styrkes når leseren ser at det er relevante koblinger mellom empiri, analyse og teori (Tjora, 2021, s. 163). Påliteligheten i mitt prosjekt blir tilknyttet hvordan jeg som forsker kan ha vært med på å påvirke de ulike funnene og resultatene som kommer frem i studien. For å sikre pålitelighet i mitt prosjekt har jeg forklart hvilken rolle jeg hadde som observatør. Etersom jeg inntok en rolle som *fullt deltagende observatørrolle* (Andersson- Bakken et al., 2021, s. 137) var jeg tilgjengelig for elevene til en hver tid, og den eneste læreren som var tilstede i rommet. Til tross for dette var jeg opptatt av å fatte meg i korthet ved spørsmål og dialog, slik at jeg ikke påvirket besvarelser eller gav elevene hint som kunne være med på å ødelegge og påvirke forskningen. Andre aspekter som er med på å påvirke påliteligheten i min oppgave, er beskrevet og gjort rede for i det neste delkapittelet.

*Validitet* handler om å måle det man har til hensikt å undersøke (Nyeng, 2012, s. 109). På samme måte som reliabilitet, påpeker Anker (2020) at man ofte bruker begrepet *gyldighet* i kvalitative studier (s. 109). I mitt studie vil jeg heretter bruke *gyldighet*. Gyldighet tar for seg kvalitetskriteriene, for å se om forskningsresultatene svarer på fenomenet som blir utforsket. For å se om resultatene av funnene er gyldige, må man først se om de svarer på problemstillingen, og om det er sammenheng mellom problemstilling og metode (Anker, 2020, s. 109). Med utgangspunkt i min problemstillingen ville tekstanalyse fungert dårlig som metode for å svare på problemstillingen.

I min problemstilling etterspør jeg hvordan arbeid med Scratch kan være med på å påvirke elevenes algoritmiske tenkning og geometriske kunnskaper. For å svare på dette er det naturlig å bruke en metode som gir forskeren tilgang til elevenes livsverden (Anker, 2020, s. 28). Som beskrevet tidligere i teksten har jeg valgt observasjon som metode. Dette er fordi observasjon er en metode som fungerer godt når man skal undersøke naturlige og sosiale settinger (Tjora, 2021, s. 62). På bakgrunn av dette er observasjon en passende metode for dette studiet. For å styrke min observasjon supplerte jeg med både skjerm- og lydopptak for å få med meg elevenes refleksjoner og tankegang under oppgaveløsningen. Bruk av skjerm- og lydopptak var med på å øke studiens transparens og overførbarhet. Gjennom beskrivelse av de ulike situasjonene, kan leseren overføre og tilpasse de ulike funnene og beskrivelsene, til egne situasjoner (Postholm & Jacobsen, 2018, s. 239). På bakgrunn av at denne oppgaven har et kvalitativt perspektiv, vil overføringen av resultater knyttes til det som er beskrevet og gjenkjennbart i studien.

Et annet tiltak som ble gjennomført for å sikre gyldighet i forskningen var å gjennomføre en pilotering (Høgheim, 2020, s. 165). Dette gav meg en indikasjon på om problemstillingen min var i tråd med den valgte metoden, og om oppgavene bidro til å gi meg funn som kunne svare på problemstillingen.

### **3.7 Forskningsetiske vurderinger**

Når man gjennomfører et forskningsprosjekt, er det alltid nødvendig å vurdere etiske hensyn og betrakte forskningen kritisk. Som forsker er det nødvendig å ha et bevisst forhold til de forskningsetiske retningslinjer, som utgjør et mangfold av regler, verdier og normer som regulerer forskningen (Høgheim, 2020, s. 86). På bakgrunn av dette vil jeg gi en beskrivelse av de etiske overveielser som ble gjort i forbindelse med denne masteroppgaven.

### **Fritt informert samtykke**

For å gjennomføre studien var jeg nødt til å innhente *fritt informert samtykke* fra elevene som skulle delta (Høgheim, 2020, s. 88). For å sikre fritt informert samtykke, ble utdelt samtykkeskjema, som inneholdt informasjon om forskningsprosjektet, at det var frivillig å delta, formålet med studien, konsekvenser ved å delta, og at det ikke ville være konsekvenser for de elevene som ikke valgte å delta. I tillegg til dette var jeg innom klassen for å informere om prosjektet og innholdet i samtykkeskjemaet. Under informasjonen la jeg vekt på å tilpasse beskrivelsene av forskningens formål, prosedyrer og lagring av data, slik at elevene hele tiden forstod innholdet. Ifølge Høgheim (2020) er det viktig at man som forsker kjenner til at barn har særlig krav på beskyttelse (s. 89). Derfor var det viktig for meg at elevene ble godt informert på forhånd. Samtykkeskjemaet ble selvsagt ikke utdelt før jeg hadde fått godkjenning fra Sikt.

### **Behandling av personopplysninger**

Ettersom mitt forskningsprosjekt tok for seg behandling av personopplysninger i forbindelse med et forskningsprosjekt måtte jeg søke om godkjenning før jeg innhentet personopplysninger (Anker, 2020, s. 105). I tillegg til dette faller mitt forskningsprosjekt under meldeplikten til Sikt, fordi ble det hentet inn både skjerm- og lydopptak. Dette er en form for behandling av personopplysninger ettersom stemmene på lydopptaket kan identifiseres til enkeltpersoner, og navn ble sagt høyt (Sikt, u.å). På bakgrunn av dette måtte både skjerm- og lydopptak lagres på «sikre og krypterte lagringsområder» (Anker, 2020, s. 107). En viktig faktor for å sikre at dette ble gjennomført var å overføre skjermopptakene fra elevenes datamaskin til en ekstern harddisk uten internettilkobling. Lydopptakene ble tatt opp på to ulike enheter, hvor den ene enheten tok opp lyd med UiO- nettskjema sin app. Her lagres ikke lydopptakene lokalt på mobiltelefonen. De blir istedenfor kryptert og lagres direkte i et eget nettskjema (Universitet i Oslo, 2017). Den andre enheten som tok opp lyd var en egen lydopptaker, hvor opptaket ble lagret på et eksternt SD- kort, før det ble overført på samme harddisk som skjermopptaket.

Under innhenting av data ble også elevsvarene notert på linjemark, og tatt bilde av med et digitalkamera uten tilgang til internett. For å sikre at anonymiteten til elevene ble ivaretatt ble deres ekte navn byttet ut med Sara, Josefine, Jens og Ola ved transkripsjon. Anonymisering av navn er for at den enkelte deltaker ikke skal kjennes igjen i data eller forskningsformidling, i tillegg til at alle innsamlede data er taushetsbelagt. Dette betyr at informasjonen ikke skal deles med andre enn det som er avtalt på forhånd (Høgheim, 2020, s. 90).



## Forforståelse

I min masteroppgave brukte jeg en lærer og elever som jeg allerede kjenner til, som informanter til min datainnsamling. I forbindelse med dette var det viktig at jeg som forsker hadde et bevist forhold til egen forforståelse, slik at jeg unngikk *bias*. For å unngå dette var det viktig at datainnsamlingen og analysen min ikke ble påvirket av min teori, verdier eller forforståelse (Maxwell, 2009, s. 243). På bakgrunn av dette var det ekstra viktig at jeg var ekstra tydelig på min rolle som forsker overfor elevene, samtidig som jeg var bevisst på og reflekterte rundt egen forforståelse.

Maxwell (2009) påpeker at det du tar med deg inn i forskningen fra din bakgrunn og identitet blir behandlet som *bias* (s. 224). På bakgrunn av dette er det viktig at jeg holder meg mest mulig nøytral og objektiv, slik at jeg unngår bias. Når datamaterialet ble transkribert var det viktig å ivare meningsinnholdet, ved gjennomgang av sitater for å forstå innholdet i det elevene sa. Det var også viktig at mine positive holdninger til programmering, ikke var med på å påvirke elevene negativt eller positivt, samtidig som jeg respekterte og verdsatte mangfoldet av tilnærminger og løsningsstrategier fra elevene.

## 3.8 Oppsummering av metodisk tilnærming

Denne oppgaven tar for seg forskning innenfor det kvalitative paradigme, der jeg benytter meg av rollen som fullt deltagende observatør, supplert med skjerm- og lydopptak. Dette er en hensiktsmessig metode for å få svar på min problemstillingen. Elevene arbeidet med tildelte geometrioppgaver i Scratch, mens jeg observerte arbeidet. Samtidig ble det gjennomført et skjerm- og lydopptak for å innhente detaljerte observasjoner. På bakgrunn av at dette nytt for meg, gjennomført jeg en pilotering for å styrke oppgaven og få tilbakemelding på viktig aspekter som jeg ikke hadde forutsett tidligere. Elevene som ble observert var strategisk utvalgt blant de som signerte samtykkeskjemaet. Jeg tok i bruk Virginia Braun og Victoria Clarke (2006) beskrivelse av de seks fasene i en tematisk analyse for å analysere dataene. For å sikre oppgavens pålitelighet har min rolle som forsker blitt nøye forklart. Videre har valget av observasjon som metode vært i tråd med problemstillingen, samtidig som det har bidratt til å sikre oppgavens gyldighet. Flere etiske betraktninger og forforståelse for å beskytte informantene i prosjektet ble også opplyst.

## 4 Resultater

I dette kapitlet vil jeg presentere resultatene fra min datainnsamling. For å besvare min problemstillingen har jeg valgt å strukturere kapitlet i to delkapitler, i henhold til forskningsspørsmålene som ble presentert i kapittel 1. Forskningsspørsmålene lyder som følgende:

1. *Hvilke kjerneferdigheter innenfor algoritmisk tenkning er mest fremtredene i elevenes arbeid med Scratch?*
2. *Hvordan uttrykker elevene sin forståelse for geometriske figurer gjennom arbeid med Scratch?*

I kapittel 4.1 vil jeg presentere hvilke kjerneelementer innen algoritmiske tenkning som er mest fremtredene i elevenes arbeid med Scratch. I dette delkapitlet vil jeg identifisere sentrale nøkkelbegreper og arbeidsmetoder i lys av kjerneelementene presentert i til Bocconi et al. (2016, 2018) og Gjøvik og Torkildsen (2019) studier. I Kapittel 4.2 vil jeg presentere hvordan elevene uttrykte forståelse for de utvalgte geometriske figurene i planen gjennom geometriforståelse i praksis og bruk av riktig ord, begreper og definisjoner.

Analysen av datamaterialet har blitt gjennomført gjennom koding og kategorisering, basert på en induktiv tematisk analyse [se kapittel 3.5]. Eventuelle handlinger eller hendelser som ikke er direkte angitt i transkripsjonen, men som er av betydning for datamaterialet, vil bli kommentert i parentes, for eksempel: (endrer koden flere ganger).

### **4.1 Hvilke kjerneferdigheter innenfor algoritmisk tenkning er mest fremtredene i elevenes arbeid med Scratch?**

I dette delkapitlet vil jeg utforske hvordan elevene uttrykte algoritmisk tenkning gjennom oppgaveløsningen med Scratch. Ettersom det var utfordrende å finne utdypende teori om nøkkelbegrepene og arbeidsmåtene til den algoritmisk tenkeren i norsk litteratur, vil jeg i hovedsak presentere den algoritmiske tenkningen til elevene i henhold til Bocconi et al. (2016, 2018) og Gjøvik og Torkildsen (2019) seks kjerneelementer under fremstillingen av mine resultater. Jeg vil inkludere observasjonsnotater, skjermbilde av programmeringsprosessen og

sitater fra transkripsjoner av skjerm- og lydopptaket. Identifisering av kjerneelementene vil være basert på teori og tidligere forskning presentert i kapittel 2.3.

Først vil jeg redegjøre for de tre kjerneelementene som var felles for begge elevparene: *feilsøking*, *algoritmebehandling* og *generalisering*. Deretter vil jeg redegjøre for de to kjerneelementene som kun var fremtredende i elevpar 1 sin oppgaveløsning: *automatisering* og *dekomponering*.

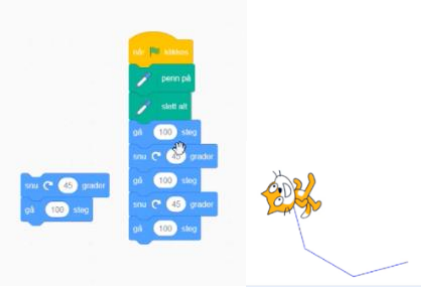
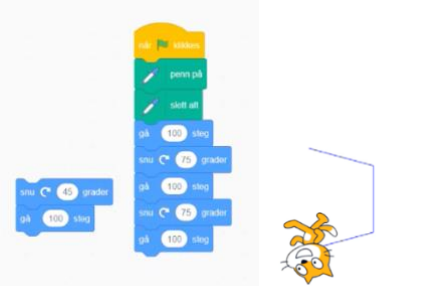
#### **4.1.1 Feilsøking**

*Feilsøking* er prosessen som oppstår når elevene skal identifisere, analysere og løse feil eller problemer (Bocconi et al, 2016). Feilsøking var det mest fremtredende kjerneelementet blant begge elevparene, særlig var det merkbart i oppgaveløsning til elevpar 1.

Etter at oppgavesettet var utdelt, viste elevene stor iver og entusiasme etter å sette i gang med oppgaveløsningen. I samsvar med piloteringen gikk vi grundig gjennom alle oppgavene i plenum, før elevene startet arbeidet. En felles gjennomgangen, som inkluderte detaljerte forklaringer og svar på spørsmål, resulterte i at elevene brukte mindre tid på å gjennomgå oppgaveteksten, og planlegge sin tilnærming til oppgavesettet. Dette førte til at elevene begynte å løse oppgavene med en gang, uten en klar plan. På bakgrunn av at elevene startet med å løse oppgavene uten en klar plan, ble deres tilnærming i stor grad påvirket av feilsøkingsprosessen. Feilsøkingsprosessen til elevene involverte mye prøving og feiling.

Tilnærmingen til elevene blir ifølge Utdanningsdirektoratets (2019) fremstilling av den algoritmiske tenkeren fremstilt som arbeidsmetodene fikle, feilsøke, holde ut og samarbeide. Følgende arbeidsmetoder tar for seg muligheten til å utforske og eksperimentere, oppdage og rette opp i feil, fortsette og prøve igjen, og dele og jobbe sammen med andre. Et eksempel på dette ble tydelig når Josefine og Sara i elevpar 2 skulle programmere en likesidet trekant.

Tabell 2 Utdrag fra transkripsjon, feilsøking, elevpar 2, sitat 70-71

70	Josefine	<i>Hæ, da må vi ta andre grader, vi tok feil. Det må bli 75, blir det ikke?</i>	
71	Sara	<i>Jo, prøv det da...2.0</i>	

I følgende utdrag viser Sara og Josefine en tendens til å raskt bli involvert i feilsøkingprosessen. I dette tilfellet befinner elevene seg i en tidlig fase, hvor de ønsker å finne ut av vinkelstørrelsen til en likesidet trekant. I sitat 70, har jentene testet ut en vinkelstørrelse på 45 grader. Dette gir elevene feil resultat, og Josefine reagerer med å si «hæ, da må vi ta andre grader, vi tok feil». Deretter tester elevene 75 grader, og får feil resultat igjen.

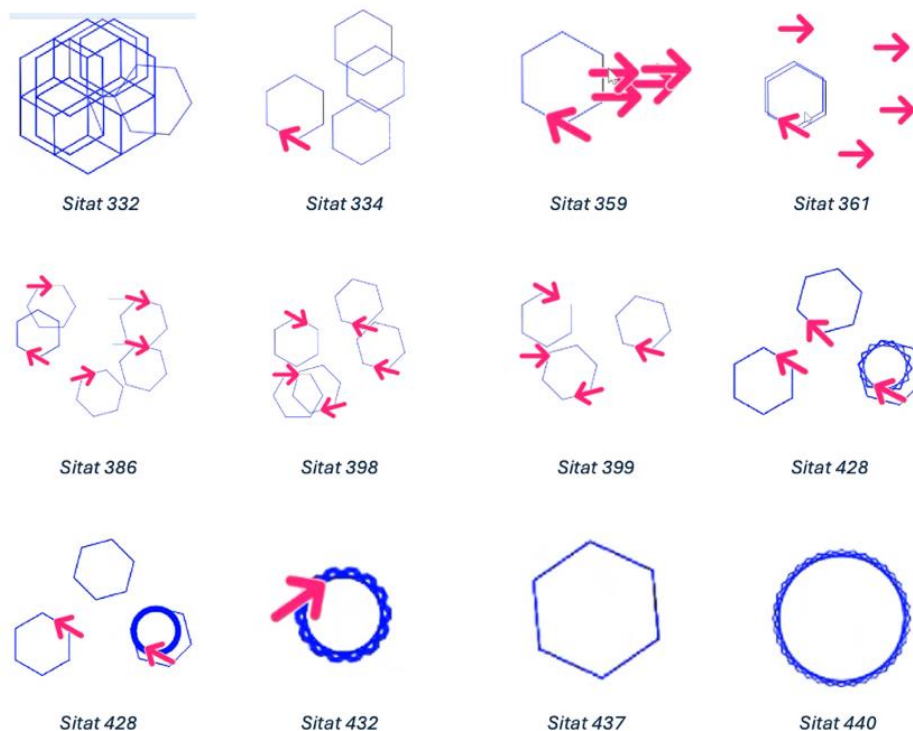
Videre i oppgaveløsningen fikler elevene mye, og stiller spørsmål til hvor stor vinkelen skal være. Bakgrunnen for at elevene eksperimenterer med ulike vinkelstørrelser, kommer som et resultat av at de ikke har en klar strategi, eller tilnærming til oppgaven. I tillegg til dette registrerer ikke elevene at trekanten skal være likesidet.

I løpet av feilsøkingprosessen diskuterer jentene ulike løsninger med hverandre og feilsøker, ved å gå igjennom hver og en av blokkene for å identifisere hvor i prosessen det ble feil. Da jentene oppdager at figuren de får med vinkelstørrelse 135 grader ligner på en likesidet trekant, forstår de at vinkelstørrelsen må reduseres. Dette viser hvordan elevene i elevpar 2 engasjerer seg i problemløsning, og samarbeidet for å finne en løsning på oppgaven gjennom feilsøkingprosessen.

Selv om Josefine og Sara gjentok feilsøkingprosessen videre i sin oppgaveløsning, viste det seg at de ofte hadde en strategisk tilnærming til oppgaven. Gjennom identifisering av separate

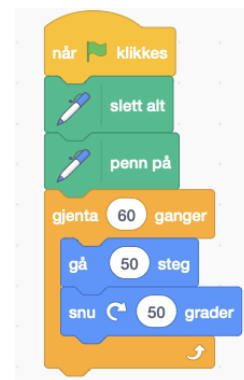
blokker som passet sammen, fant de koden som gav ønsket resultat og figur. For å validere resultatet, sammenlignet de figuren de hadde programmert i Scratch med forventet geometrisk figur. Utformingen av den geometriske figuren som ble programmert, spilte derfor en avgjørende rolle når feilsøkingen ble avsluttet. Dette kan sammenlignes med Utdanningsdirektoratets (2019) nøkkelbegrep evaluering. Evaluering går ut på at elevene må gjøre vurderinger underveis i oppgaveløsningen.

I likhet med elevpar 2, var feilsøking et sentralt element i problemløsningen til elevpar 1. Feilsøking var den tilnærmingen som ble mest anvendt. Et eksempel på den store mengden med feilsøking blir presentert i figuren under. I denne figuren presenteres skjermbildene fra da elevpar 1 skulle programmere en blomst bestående av sekskanter. Sitatene i denne figuren viser til skjermbildene ved prosessen:



Figur 4.1 Oversikt over prosessen frem til en blomst av sekskanter

I dette tilfellet bli det tydelig at elevene tester programmet gjentatte ganger. Dette blir gjort, på bakgrunn av at koden deres ikke gir elevene en blomst. Som et resultat av dette, gjennomfører elevene flere endringer i koden sin. Etterhvert som resultatet til elevene ikke blir som ønsket, blir de utfordret til å holde ut. Til slutt kommer de frem til deres resultat av en blomst bestående av sekskanter (sitat 440) (kode, se figur 4.2). For å komme frem til dette benytter elevene Utdanningsdirektoratets (2019) arbeidsmetode skape.



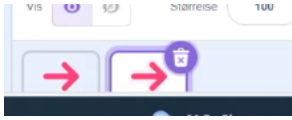
Figur 4.2 Kode tilhørende sitat 440


### 4.1.2 Algoritmebehandling

*Algoritmebehandling* er en nøyaktig beskrivelse av en serie operasjoner, som leder til en løsning gjennom en klar definisjon av trinnene som må utføres. Når man gjennomfører en algoritmebehandling, krever det en evnen til å tenke i termer av sekvenser. Det kan sammenlignes med en oppskrift, der en gitt sekvens av programmeringsinstrukser må utføres i en bestemt rekkefølge (Csizmadia et al., 2015, s. 7; Brennan & Resnick, 2012, s. 3). Algoritmebehandling kan sammenlignes med Utdanningsdirektoratets (2019) nøkkelbegrep algoritmer. Algoritmer går ut på å følge regler, steg for steg.

I Scratch kan elevene utvikle algoritmer ved å gi instruksjoner til Felix gjennom koder og kommandoer. Et eksempel på algoritmebehandling er når elevpar 1 skulle programmere en blomst.

Tabell 3 Utdrag fra transkripsjon, algoritmebehandling, elevpar 1, sitat 352- 360

353	Jens	Hva hvis vi lager og har mange piler forskjellige steder?	
354	Ola	Jaa	
355	Jens	Også får vi bare de til å gjøre det samme	
356	Ola	Jaa	
357	Jens	'Tree, 'fireee?	
358	Ola	Fem, ta fem da	

359	Jens	<i>Fire, fem. Også bare, hvor setter vi de da, hmm?</i>	
360	Ola	<i>Vi setter de på hvert sitt sted</i>	

I utdraget ovenfor ser vi at guttene benytter seg av algoritmebehandling gjennom variabler. Hver pil symboliserer en sekskant hver, og variablene fungerer som en lagringsenhet for dataene. Under utforskningen justerer elevene systematisk pilenes posisjon, og antall piler som trengs for å danne en blomst. Da elevene jobber systematisk med én og én pil om gangen, jobber de i ulike termer, som bidrar til å etablere en klar definisjon av hva som trengs for å danne en blomst, og hvilke endringer som må gjøres.

En annen observasjon er at elevparene hadde en variert tilnærming til algoritmebehandling. Elevpar 2 brukte dette på en annen måte enn elevpar 1. Allerede i den første oppgaven ble dette synlig når Josefine og Sara skulle programmere et kvadrat.

Tabell 4 Utdrag fra transkripsjon, algoritmebehandling, elevpar 2, sitat 27

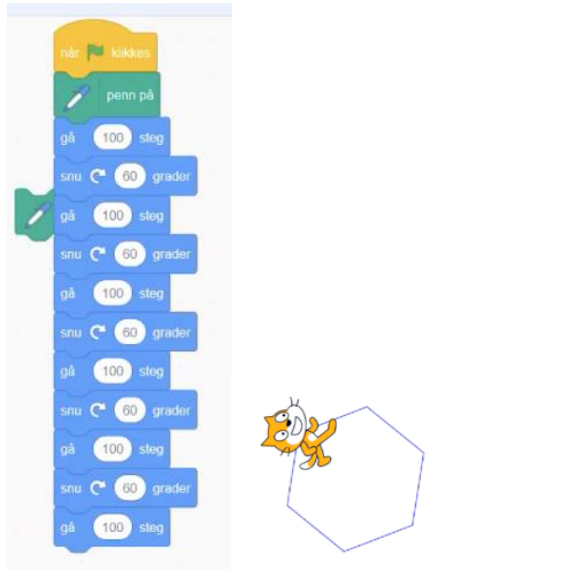
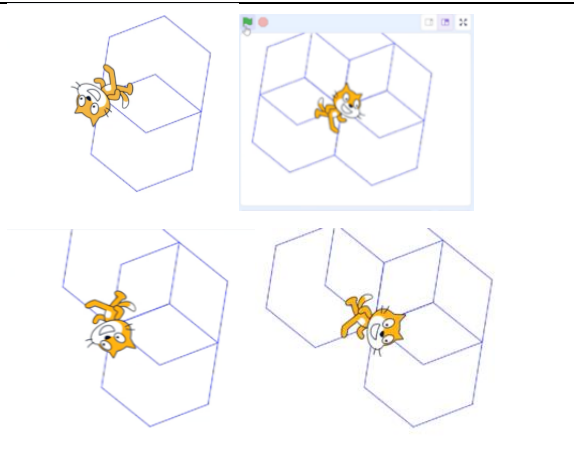
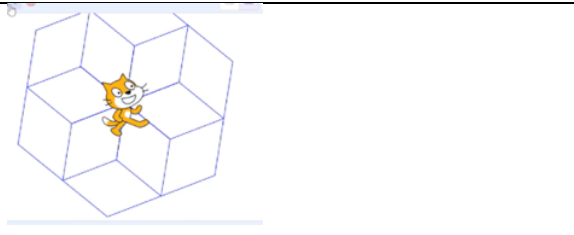
27	Josefine	<i>Også ehmm, da har den gått 100 steg frem, 100 steg opp, 100 steg dit, da må vi ta en gang til</i>	Lager et kvadrat med musepekeren for å visualisere de stegene i koden som er ferdig
----	----------	--	---

Når jentene skulle programmere et kvadrat, løste de dette ved å anvende separate blokker. Hver blokk representerte en side og en vinkel på figuren. Før de testet koden, brukte de musepekeren til å visualisere og gå igjennom hver av blokkene i koden. Under denne prosessen observerte Josefine at de kun hadde kodet blokker til tre av de fire sidene på figuren. Visualiseringen hjalp jentene med å systematisk gå igjennom hvert steg i prosessen. Dette var en tilnærming som elevpar 2 aktivt brukte gjennom resten av oppgaveløsningen.

Et funn som oppstod hos begge elevpar, var at de trykket mye på «grønt flagg» for å fullføre algoritmen sin. Når koden deres ikke ga ønsket resultat, fortsatte elevene å trykke på «grønt flagg», til de oppnådde ønsket resultat. Denne tilnærmingen skyldes at elevene hadde produsert en ufullstendig algoritme, og tok i bruk «grønt flagg» for å fullføre algoritmen. Denne

tilnærmingen ble synlig når elevpar 2 skulle programmere en blomst bestående av sekskanter.

Tabell 5 Utdrag fra transkripsjon, algoritmebehandling, elevpar 2, sitat 160- 161

160	Josefine	<p><i>Vent jeg har en ide, hvis vi bare tar bort den som er sletta alt.. Vent litt, se her nå, vi må bare sette han i midten. Jeg har en ide. Sånn, også...</i></p>	 <p>The image shows a Scratch script with the following blocks: 'når flagget klikkes' (when green flag clicked), 'penn på' (pen on), 'gå 100 steg' (move 100 steps), 'snu 60 grader' (turn 60 degrees), and another 'gå 100 steg' block. This sequence is repeated six times. To the right, a single blue hexagon is drawn with the character Felix the cat in the center.</p>
(Trykker på grønt flagg gjentatte ganger)			 <p>The image contains four screenshots showing the character Felix the cat moving to the center of a hexagon and then rotating 60 degrees to draw a new petal. This process is repeated four times, showing the growth of a six-petaled flower.</p>
161	Sara	<p><i>Sånn</i> <i>(Trykker på grønt flagg gjentatte ganger)</i></p>	 <p>The image shows a single screenshot of the completed Scratch program, displaying a full six-petaled flower with Felix the cat in the center.</p>

Den første gangen Josefine og Sara trykket på «grønt flagg», tegnet Felix en sekskant. Når «grønt flagg» blir trykket på igjen, roterer Felix 60 grader og tegnet en ny sekskant. Denne handlingen gjentok de seks ganger for å få resultatet i sitat 161, deres versjon av en blomst av sekskanter.



Blant jentene i elevpar 2 fungerte denne tilnærmingen som en suksessfull metode, for å løse flere av oppgavene. Dette var en tilnærming som elevene ofte tok i bruk, der «grønt flagg» erstattet løkker og «gjenta- blokken». Den hyppige bruken av «grønt flagg» var med på å effektivisere oppgaveløsningen flere ganger. Samtidig at det viktig å påpeke at selv om «grønt flagg» effektiviserte oppgaveløsningen til elevene, var prosessen frem til riktig resultat tidkrevende, og preget av mye feilsøking.

### 4.1.3 Generalisering

De siste kjerneelementet som var felles for begge elevparen er *generalisering*. Generalisering assosieres med identifisering av mønster, likheter og kombinasjon, og utnyttelsen av disse. Dette er en måte som raskt vil løse nye problemer, basert på tidligere løsninger og erfaringer (Csizmadia et al., 2015, s. 8). Generalisering kan sammenlignes med Utdanningsdirektoratets (2019) nøkkelbegreper logikk, hvor man skal analysere og forutse, og nøkkelbegrepet mønstre, hvor man skal finne og bruke likheter. Generalisering ble tydelig da elevpar 1 skulle programmere en sekskant. Etter at Ole har lest oppgaveteksten sier han:

Tabell 6 Utdrag fra transkripsjon, generalisering, elevpar 1, sitat 287

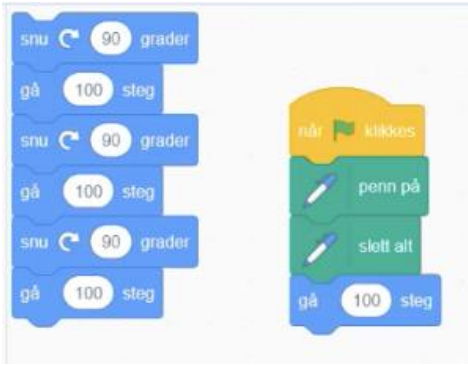
287	Ola	Se her, det skal være en sekskant, da må vi gjenta 6 ganger
-----	-----	---



I følgende utdrag anvender guttene «gjenta-blokken» for å effektivisere programmeringsprosessen. Da Ola observerer at de skal programmere en sekskant, foreslår han å gjenta handlingen seks ganger. Denne tilnærmingen viser Olas evne til å identifisere et mønster i figuren, der antall gjentakelser, karakteriserer hver av vinklene i sekskanten. Ved å anvende «gjenta- blokken» ønsker elevparet å gjenta handling seks ganger for å fullføre tegningen av sekskanten på en mer strukturert og effektiv måte. Dette viser til elevenes evne til å bruke løkker, og anvendelse av mønster og systemer i figuren.

I oppgaveløsningen til Sara og Josefine oppsto det flere eksempler av generalisering. Et eksempel på dette oppsto i den første oppgaven, hvor elevparet skulle programmere et rektangel. Under programmeringen, observerer Josefine at det er likhet mellom figurene kvadrat og rektangel. Hun sier:

Tabell 7 Utdrag fra transkripsjon, generalisering, elevpar 2, sitat 41

41	Josefine	<p>Sånn da er vi ferdig med den, ok, slett alt, også er det neste er kvadrat, det er akkurat det samme vi må gjøre, da kan vi bruke den samme. Du må skrive den ned.</p>	
----	----------	--	--

I dette utdraget sier Josefine «...det er akkurat det samme..» når hun observerer at de kan anvende den samme koden som ble brukte under programmeringen av et kvadrat, til å programmere et rektangel. Dette viser til Josefines evne til å se likheter mellom figurer og oppdage et mønster. Josefine oppdager at figurene inneholder flere av de samme egenskapene, som fire rette vinkler og parallelle sider.

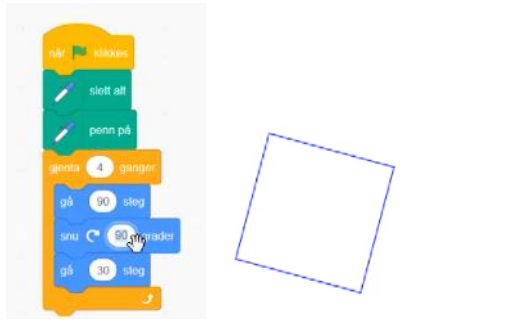

Ved programmering av et rektangel trenger elevene kun å endre lengden på to av sidene i figuren. Når Josefine og Sara identifiserer sammenhengen mellom figurene, løser de oppgaven på en rask og effektiv måte. Denne observasjonen fører til en generalisering av egenskapene til både et kvadrat og et rektangel. Dette viser at elevene har en dypere forståelse for geometriske former og deres egenskaper, samtidig som det bidrar til å styrke programmeringsferdighetene til elevene.

#### 4.1.4 Dekomponering

I tillegg til de tre nevnte kjerneferdighetene, inkluderte elevpar 2 dekomponering og automatisering i sin oppgaveløsning. *Dekomponering* er å bryte opp et problem i mindre deler, slik at hver komponent kan forstås, løses, utvikles og evalueres hver for seg (Bocconi et al., 2016). Dette kan sammenlignes med Utdanningsdirektoratets (2019) nøkkelbegrep dekomposisjon. Et eksempel på dette er når elevpar 1 skulle løse oppgave 1a. Jens og Ola har

nettopp blitt ferdig med å programmere et kvadrat, og skal nå programmere et rektangel:

Tabell 8 Utdrag fra transkripsjon, dekomponering, elevpar 1, sitat 105- 127

105	Ola	<i>Vi får bare firkant vi!</i>	
(endrer koden flere ganger)			
118	Jens	<i>'Hmmm, det gikk jo kjempe bra</i>	
119	Ola	<i>Gå 120 steg, snu</i>	
(endrer koden flere ganger)			
127	Ola	<i>Vi bare tar vekk gjenta!</i>	


Da elevene begynte å programmere rektangelet, valgte de å beholde den samme koden som de hadde anvendt under programmeringen av et kvadrat. Elevene endret kun på lengden til to av de parallelle sidene. Etter gjentatte forsøk med feilsøking påker Ola, at resultatene deres bare blir en «firkant». I stedet for å legge en plan, fortsatte guttene med å endre på koden sin, ved å justere lengden på sidene og vinkelstørrelsen. Til tross for endringene fikk de ikke forventet resultat. I sitat 127 oppdager Ola at de kan eliminere «gjenta- blokken», og bruke separate blokker for å programmere rektangelet.

I stedet for å identifisere de geometriske egenskapene hos et rektangel, deler elevene opp koden sin i flere og mindre håndterbare sekvenser. Dette representerer dekomponering, fordi Ola anvender en tilnærming hvor han bryter ned problemet med «gjenta- blokken», og anvende mer håndterlige deler gjennom separate blokker.

### 4.1.5 Automatisering

*Automatisering* er en effektiviseringsprosess hvor man instruerer en datamaskin til å utføre flere oppgaver raskt og effektivt (Lee et al., 2011, s. 33). Gjennom programmering til elevpar 1, kan vi se at denne effektiviseringsprosessen kommer frem gjennom ulike gjentatt sekvenser og bruk av løkker. Når Jens og Ola skulle begynne å løse oppgavene, var det merkbart at Jens hadde brukt Scratch tidligere. Dette er fordi han hadde kjennskap til å bruke «gjenta- blokken».

Tabell 9 Utdrag fra transkripsjon, automatisering, elevpar 1, sitat 46- 51

46	Jens	<i>Også loop?</i>	
47	Ola	<i>Også ja...2.0 Gjenta 4 ganger</i>	
48	Jens	<i>Her?</i>	
49	Ola	<i>Ja</i>	
50	Jens	<i>Sånn</i>	
51	Ola	<i>Også prøver vi</i>	

I dette utdraget utforskere guttene programmering av et kvadrat. I stedet for å bruke separate blokker til å representere hver av sidene, bruker de «gjenta- blokken» for å effektivisere prosessen. Ved å anvende «gjenta- blokken» og etablere løkker, får de en mer konsis og effektiv sekvensen av instruksjoner. I stedet for å gjenta de samme kommandoene åtte ganger, reduserer de kommandoene til kun tre blokker: gjenta 4 ganger, gå 40 steg, snu 90 grader. Som en følge av dette anvender ikke elevene lang tid på å finne riktig sammensetning av blokker, og prosessen deres blir effektiv.

## 4.2 Hvordan uttrykker elevene sin forståelse for geometriske figurer gjennom arbeidet med Scratch?

Funnen i dette delkapittelet presenterer elevenes geometriske forståelse under arbeidet med Scratch. Funnene er delt inn i to hovedtemaer: *geometriforståelse i praksis og ord, begreper og definisjoner i geometrien*. For å identifisere disse temaene vil jeg inkludere sitater fra transkripsjoner av skjerm- og lydopptaket.

### 4.2.1 Geometriforståelse i praksis

Da elevene begynte å programmere de ulike geometriske figurene, oppdaget de raskt at dette var annerledes fra å tegne med penn og papir. Allerede i løpet av den første oppgaven oppdaget elevene at det var avgjørende å ha kjennskap til de ulike egenskapene, til de utvalgte geometriske figurene i planet.

#### Konstruksjon av kvadrat, rektangel og likesidet trekant

I den første oppgaven observerte jeg at begge elevparene viste stor entusiasme for programmering, og de mestret programmeringen av et kvadratet uten store vanskeligheter. Imidlertid møtte elevpar 1 på utfordring i den påfølgende programmeringsoppgaven, da de skulle programmere et rektangel. I utdraget under kommer dette tydelig frem når Jens spør Ola om hvordan de skal gå frem i prosessen for å programmere et rektangel:

Tabell 10 Utdrag fra transkripsjon, elevpar 1, sitat 80- 103

- 80 Jens (...) Hvordan skal vi, hvordan skal vi lage et rektangel liksom?  
(diskuterer hvem som skal notere ned koden ifra forrige oppgave)
- 84 Jens Ja nå kan du ta den, den ser jo bra ut, se!
- 85 Ola Ja, okai. Også må du, ta, du må ta, gå 90, snu 90, også må du ta gå, ehh, gå for eksempel 30  
(prøver ulike lenger)
- 101 Ola Hva skjer a?
- 102 Jens Det som skjer, er at alt blir sånn ompala lompa
- 103 Ola Vi får bare en firkant!

Under forberedelsen til programmeringsprosessen, uttrykte Jens usikkerhet angående fremgangsmåten for å programmere et rektangel. Til tross for at de løste den første oppgaven på kort tid, hadde elevene i elevpar 1 en kritisk innstilling til programmeringen av et rektangel.

I sitat 85 påpeker Ola at rektangelet har vinkler med en vinkelstørrelse på 90 grader, og at sidene må ha ulik lengde. Selv om elevene gir uttrykk for å kjenne til vinkelstørrelsen og at to og to parallelle sider skal ha lik lengde, oppnår de ikke ønsket figur.

Bakgrunnen for dette er at koden de anvender tar for seg «gjenta 4 ganger». På bakgrunn av at et kvadrat har to og to motstående like sider, er de nødt til å endre «gjenta- blokken» til «gjenta 2 ganger». Da vil de kun repetere handlingen to ganger, slik at de motstående sidene er like. Selv om elevene kjenner til de geometriske egenskapene, klarer de ikke å tilpasse koden til å programmere et rektangel. På denne måten oppdager elevene at programmering i Scratch skiller seg fra konstruksjon og tegning.

Når elevpar 2 skulle løse den samme oppgaven brukte elevene kortere tid på oppgaveløsningen. Dette skyldes at Josefine tidlig ser muligheten til å gjenbruke store deler av koden fra kvadratet til å lage et rektangel:

*Tabell 11 Utdrag fra transkripsjon, elevpar 2, sitat 41- 53*

41    Josefine    Sånn da er vi ferdig med den, ok, slett alt, også er det neste er kvadrat, det er akkurat det samme vi må gjøre, da kan vi bruke den samme. Du må skrive den ned.

(diskuterer hvem som skal notere ned koden for kvadratet)

51    Josefine    Ok, også, nå tar vi, fordi at nå tar vi bare og sier at den går, 200 steg, der, også går den 100 som er oppover, også går den 200 skritt som er

52    Sara        Ta han der

53    Josefine    200, også er det 100

I innledningen av oppgaveløsningen blander Josefine begrepene kvadrat og rektangel. Til tross for dette observerer hun at begge figurene deler flere av de samme geometriske egenskaper, og at det er en sammenheng mellom figurene. Dette viser til god forståelse av grunnleggende geometriske prinsipper. Selv om elevene blander begrepene, viser Josefine og Sara evne til å forstå sammenhengen mellom de geometrisk figurene.

## Hva skal vinkelstørrelsen til figuren være?

Da elevene skulle programmere en likesidet trekant og sekskant, oppsto spørsmålet om vinkelstørrelse tidlig hos begge elevparene. Selv om dette er to grunnleggende geometriske figurer, viser elevene til manglende kjennskap til figurenes vinkelstørrelse. Under programmeringen av en likesidet trekant, observerte jeg at begge elevparene leste oppgaveteksten fort uten å fange opp at oppgaveteksten etterspurte en likesidet trekant. I utdraget under kommer dette frem når elevpar 2 skal programmere en likesidet trekant:

Tabell 12 Utdrag fra transkripsjon, elevpar 2, sitat 57- 70

- |    |          |   |
|----|----------|---|
| 57 | Josefine | Sånn ja. Da har vi funnet den, da må vi bare skrive ned koden, også er det neste trekant.. Hva var vinkelen i en trekant?                                       |
| 58 | Sara     | Eehm, jeg tror det er 45?   |
| 59 | Josefine | Jo, jaa det må være 45, vi sier at det er 45..Ok, da må vi gå, vi tar 100 for den er safe, også tar vi, vi bare beholder den samme koden, 45 grader, gå motsatt |
| 60 | Sara     | Alle må være 45, så det er<br>(snakker om hvilke knapper de skal trykke på)   |
| 69 | Sara     | Den og den da... Også bare trykk start  |
| 70 | Josefine | Hæ, da må vi ta andre grader, vi tok feil. Det må bli 75, blir det ikke?  |

Allerede ved lesing av oppgaveteksten observerer ikke jentene at det spesifikt etterspørres en likesidet trekant. Som et resultat av dette referer elevene til den likesidede trekanten som «trekant» gjennom resten av oppgaveløsningen. Dette viser at elevene ikke legger merke til detaljene i oppgaveteksten. Da Josefine spør Sara om vinkelstørrelsen i en trekanten, påpeker Sara at vinkelstørrelsen må være 45 grader. Josefine er enig i dette, og begge gir uttrykk for å være sikre på vinkelstørrelsen. Imidlertid oppdager elevene at dette er feil. På bakgrunn av dette fortsetter de å utforske flere muligheter, hvor de øker vinkelstørrelsen til trekanten litt og litt. Til slutt finner de ut at riktig svar er 120 grader.

På grunn av elevenes begrensede erfaring med Scratch, er de ikke klar over at de må anvende verdien til *nabovinkelen* i en likesidet trekant når de skal utforme koden. I motsetning til tegning av geometrisk figurer, hvor man vanligvis begynner med å tegne formen og bevege seg til vinklene, ønsker Scratch å begynne med vinklene og bevege seg mot formen. På bakgrunn av dette trenger elevene å kjenne til verdien til den ytre vinkelen i trekanten, altså

*supplementsvinkelen* for å programmere en likesidet trekant. Dette viser at elevene trenger ytterligere kunnskap om Scratch for å programmere de geometriske figurene korrekt.

Begge elevparene møtte også på utfordringer med å finne riktig vinkelstørrelse under programmeringen av en sekskant. Elevpar 1, som tidligere hadde programmert en åttekant under programmeringen av en likesidet trekant, ønsket å gjenbruke denne koden for sekskanten. Siden både en sekskanten og åttekanten er en mangekant, ønsket elevene å anvende erfaringer fra koden de fikk, under programmering av en likesidet trekant:

Tabell 13 Utdrag fra transkripsjon, elevpar 1, sitat 267- 276

267	Jens	X Den var 135, var den ikke?
268	Ola	Ehh, njaa, ehh, det var jo litt ned tror jeg kanskje, men bare prøv..
269	Jens	Den var hvertfall i hvertfall du som lagde en sekskant i stad
270	Ola	Det var ikke en sekskant
271	Jens	Jo, men, det er fordi vi ikke har tatt på.. gjenta
272	Ola	Der, åhhh..
273	Jens	Nå lagde lagde vi jo en stjerne, teknisk så er jo det en sekskant
274	Ola	Var det ikke noe med den? En, to, tre, fire, fem, seks sju
275	Jens	Jeg lagde en sju –
276	Ola	Åttekant

I utdraget over påpeker Ola at figuren de programmerte tidligere i programmeringsprosessen ikke var en sekskant. Selv om Ola påpeker dette, ønsker Jens at de gjenbraker den samme koden. Dette resulterer i at guttene denne gangen programmerer en stjerne, før de deretter får en åttekant. Videre diskuterer de frem og tilbake om de skal øke eller redusere vinkelstørrelsen. Etter gjentatte ganger med prøving og feiling, ser de at de at vinkelstørrelsen må være 60 grader:

Tabell 14 Utdrag fra transkripsjon, elevpar 1, sitat 290- 292

290	Ola	Mere grader
291	Jens	60 grader, siden den er 60. En, to, tre..
292	Ola	Vi klarte det!

Tidligere i prosessen anvender elevene en vinkelstørrelse på 50 grader. Dette resulterer i en syvkant. På bakgrunn av elevene både har programmert en åttekant og syvkant, ser de at de trenger å øke gradantallet for å programmer en sekskant. Gjennom gjentatte forsøk, utforsker



elevene hvor stor vinkelstørrelsen skal være, slik at de får en sekskant. Til slutt foreslår Jens en vinkelstørrelse på 60 grader. Etter mye fikling, og forslag som blir fremstilt uten en plan, har elevene en evne til å se et mønster mellom figurene.

#### 4.2.2 Ord, begreper og definisjoner i geometrien

Et funn fra observasjonen som i stor grad gjentok seg, var elevenes evne til å uttrykke seg gjennom matematikkspråket. I løpet av datainnsamling endret elevenes språkbruk seg gjentatte ganger. Under de første oppgavene hadde de et upresis språk hvor de blandet ord og begreper. Etterhvert ble elevene mer bevisst på en eget språk, og begynte å anvende de samme begrepene som Scratch og oppgavesettet tok i bruk. Et eksempel på dette, er når Josefine bruker både skritt og steg:

*Tabell 15 Utdrag fra transkripsjon, elevpar 2, sitat 51*

51 Josefine      Ok, også, når skritter vi, fordi at nå tar vi bare og sier at den går, 200 steg, der, også går den 100 som er oppover, også går den 200 skritt som er –

I dette sitatet bruker Josefine både skritt og steg, for å beskrive hvor langt Felix skal bevege seg for å tegne et rektangel. Variasjonen i språket hennes fører til et uklart språk, hvor hun blander begrepene. Selv om hun senere erstatter «skritt» med «steg», er språket hennes mindre konsekvent. Da hun går over til å kun bruke «steg», anvender hun det samme begrepet som blir brukt i Scratch og oppgavesettet. I overgangen mellom begrepene viser hun til gradvis tilpasning av språket, som etterhvert utvikler seg til å bli mer faglig.

Et annet funn fra observasjonen er at elevene har et uklart språk når de diskuterer vinkler. De benytter både begrepene kanter og hjørner. Da de blander begrepene skjer dette ubevist, og de gir uttrykk for at blandingen av begrepene er en naturlig del av deres vanlige- og matematiske språk.

u r  
b) Alle har over to vinkler, over 99 steg,  
og alle har en kode som sier at  
Felix skal gå 100 steg.  
Og forskjellen er at trekanten har  
tre kanter og firkantene har  
4 kanter.

Figur 4.3 Utdrag fra elevbesvarelse, elevpar 2

I utdraget ovenfor skal elevene svare på forskjellene og likhetene mellom de tre geometriske figurene som ble programmert i oppgave én. På samme måte som eksempelet ovenfor, bruker elevpar 2 både vinkler og kanter når de beskriver vinklene i de ulike figurene de har programmert. Dette fører til at eleven bruker et varierende språk og ulike begreper når de snakker om en ting. Et annet lignende eksempel er når Jens i elevpar 1 henviser til «loop» i stedet for «gjenta- blokken». I dette tilfellet blir det varierende språket til elevene tydelig, og noe som kan skape misforståelser innad i paret.

Som nevnt tidligere, hadde elevene en tendens til å raskt leste over oppgaveteksten. Da de skulle programmere en likesidet trekant, refererte de til figuren som en trekant. På bakgrunn av dette tydeliggjorde ikke elevene at alle sidene på figuren skulle være like lange, under samtalen om figurens egenskaper. Dette gjorde at elevene ikke var like nøyaktige når de snakket om figurens egenskaper.

Et annet lignende eksempel er når elevene skulle programmere et rektangel. I stedet for å anvende riktig begrep, omtalte elevpar 1, dette som en firkant:

Tabell 16 Utdrag fra transkripsjon, elevpar 2, sitat 103

103 Ola Vi får bare firkant!

I utdrag over bruker Ola firkant, når han skal referere til et rektangel. Ved å bruke ulike begreper for å omtale rektangelet skaper han usikkerhet i paret, noe som gjør at elevene blir usikre på

hvilken figur som skal programmeres. Dette er et resultat av deres varierende språk.

### 4.3 Oppsummering av resultater

Med dette er funnene fra observasjon supplert med skjerm- og lydopptak lagt frem. Ut ifra Bocconi et al. (2016 & 2018) og Gjøvik og Torkildsen (2019) seks kjerneelementer var fem av disse presentert under elevens arbeid. Blant disse kjerneelementene var feilsøking det mest fremtredende kjerneelementet blant elevene. Feilsøkingen bar preg av elevenes manglende erfaring med Scratch og usikkerhet rundt figurenes egenskaper. I tillegg til dette hadde elevene en tilnærming til oppgaveløsningen som bar preg av at de ikke hadde en klar plan for hva de skulle gjøre. Dette resulterte i at elevene testet alle forslag som de kom frem til. Et annet steg som var sentralt i elevenes oppgaveløsning var algoritmebehandling. Elevpar 1 praktiserte dette gjennom bruk av variabler hvor de lagret dataene i ulike sekvenser. Elevpar 2 praktiserte dette ved å bruke ufullstendige algoritmer, gjennom gjentakende bruk av «grønt flagg» hvor elevene trykket seg frem til ønsket resultat. Generaliseringen til elevene kom frem gjennom identifisering av mønster, likheter og logikk.

I tillegg til de tre nevnte kjerneelementene hadde elevpar 1 et tilfelle av dekomponering hvor de måtte gå fra å anvende en løkke til å dele opp koden i separate blokker, som var mer håndterlige å anvende. I tillegg til dette hadde de hyppig bruk av automatisering gjennom effektiv bruk av løkker og «gjenta- blokken». For begge elevparene ble Utdanningsdirektoratets (2019) fremstilling av den algoritmiske tenkeren synlig gjennom nøkkelbegrepene *logikk*, *algoritmer*, *dekomposisjon*, *mønstre* og *evaluering*. I tillegg til dette kom arbeidsmetodene *fikle*, *skape*, *feilsøke*, *holde ut* og *samarbeid* til syne i elevenes arbeid.

Elevene hadde mange gode refleksjonen og løsningsmetoder når de jobbet med oppgavesettet. Selv om de hadde mange gode forslag, var forslagene deres preg av usikkerhet rundt figurenes geometriske egenskaper, spesielt vinkelstørrelsen til en likesidet trekant og en sekskant var utfordrende. Da elevene skulle uttrykke seg gjennom matematikkspråket ble deres manglende kunnskaper om figurenes egenskaper og unøyaktig språk tydelig. I flere tilfeller hadde elevene et unøyaktig språk, hvor de blandet ord og begreper, samtidig som de bytte på å bruke begreper som steg og skritt, og hjørner og kanter. I tillegg til å bytte på begrepene omtale de figurene som trekant og firkant. Etterhvert i oppgaveløsningen ble matematikkspråket til elevene styrket, og de fikk et mer presis språk, uten like mye variasjon.

## 5 Diskusjon

I dette kapitlet vil jeg diskutere de kvalitative dataene fra observasjonen i lys av tidligere forskning og teori, som ble presentert i kapittel 2. Hensikten med denne oppgaven er å undersøke om geometrioppgaver i Scratch bidrar til utvikling av algoritmiske tenkning, og forståelse av geometriske figurers egenskaper hos barneskoleelever. Problemstillingen som ble presentert i kapittel 1 er:

*«På hvilke måter kan arbeid med geometrioppgaver i Scratch bidra til å utvikle barneskoleelevers algoritmiske tenkning, og forståelse av egenskapene til utvalgte geometriske figurer i planet?»*

I kapittel 5.1- 5.3 vil jeg, basert på mine funn, teori og tidligere forskning, besvare min problemstillingen ved å drøfte hvordan elevene har utviklet algoritmisk tenkning, geometrisk kompetanse og tenkning. I kapittel 5.4 vil jeg oppsummere diskusjonen. I det siste delkapitlet vil jeg trekke frem studiens begrensninger. Didaktiske implikasjoner vil bli presentert fortløpende i teksten, for å sikre en jevn flyt og unngå gjentakelser.

### 5.1 Utvikling av algoritmisk tenkning gjennom oppgaveløsning i Scratch

NOU (2020:2) fremhever betydningen av at elever kan anvende algoritmisk tenkning som en grunnleggende forutsetning for å utvikle og benytte den digitale kompetansen som er nødvendig i dagens samfunn. Oppnåelse av forståelse og ferdigheter innen algoritmisk tenkning krever at elevene har en grunnleggende forståelse av begrepet og hvordan de kan praktisere dette.

I denne studien hadde elevene begrenset med erfaring og kunnskap innen programmering. Tross begrenset erfaring og kunnskap viste begge elevparene algoritmisk tenkning gjennom *feilsøking, algoritmebehandling og generalisering*. I tillegg til dette var *dekomponering og automatisering* fremtredene hos elevpar 1.

## **Prøving og feiling som arbeidsmetode**

En viktig observasjon i denne studien er feilsøkingsprosessen til elevene. Blant mine funn er det tydelig at feilsøking ble aktivt brukt under arbeidet med geometrioppgaver i Scratch. Elevene erfarte at programmering ikke alltid fungerte som forventet, og at det å gjøre feil var en viktig del av problemløsningen. Ifølge Brennan & Resnick (2012) og Kunnskapsdepartementet (2019c) er evnen til å feilsøke og korrigere feil grunnleggende for å mestre programmering. Gjennom håndtering av feil i samspill med hverandre, kunne elevene se på problemene med et overordnet syn, og utforske hvilke strategier og tilnærminger, som kunne være med på å løse oppgaven deres.

I Brennan og Resnick (2012) sin dimensjon Computational thinking practice, er det et større fokus på læringsprosessen, fremfor hva man lærer (s. 6- 7). Denne dimensjonen understreker viktigheten av å se på elevenes tankeprosess og valg av strategier, for å forstå hvordan de lærer og håndterer utfordringer, slik at man kan støtte elevene utvikling og mestring (Brennan & Resnick, 2012, s. 6). Under programmering av en sekskant møtte elevene på flere utfordringer, særlig knyttet til å bestemmelsen av riktig vinkelstørrelse for figuren. Disse utfordringene førte til at elevene måtte utforske flere ulike løsninger og strategier. Gjennom denne utforskningsprosessen fikk elevene en grundig innføring i hvordan de programmerer en sekskant i Scratch. Samtidig som de fikk en innføring i programmeringen, måtte de belyse viktige prinsipper innen geometrien for å oppnå ønsket resultat (Battista, 2007, s. 843). Kombinasjonen av praktisk og teoretisk læring førte til en større forståelse blant elevene, som aktivt måtte håndtere og mestre utfordringer gjennom valg av hensiktsmessige strategier og tilnærminger, samt integrere og kombinere kunnskap om geometri med programmering (Bocconi et al, 2016, s. 15; The Royal Society, 2012, s. 29). Elevenes tankeprosess og strategivalg (Brennan & Resnick, 2012) var derfor sentrale i deres oppgaveløsning, og bidro med å belyse deres utvikling og mestring.

Under feilsøkingsprosessen i denne studien tok elevene i bruk ulike strategier for å identifisere og analysere problemer som oppstod (Wing, 2006). En interessant observasjon er at elevene ofte tok i bruk løsningsstrategier som bestod av eksperimentering og kjappe avgjørelser, hvor målet var å «effektivt» finne den riktige koden. Selv om denne tilnærmingen var fremtredende hos begge elevparene, var det særlig representert hos elevpar 1. Gjennom denne tilnærmingen fiklet (Utdanningsforbundet, 2019) elevene en del, samtidig som de eksperimenterte med flere ulike løsninger.

Selv om dette til tider så kaotisk ut, var det en viktig del av læringsprosessen til elevene, hvor de etterhvert kunne se et mønster, hvor de feilsøkte, oppdaget feilen og rettet den opp (Utdanningsforbundet, 2019). Feilsøkingen resulterte i at elevenes kunnskaper til Scratch utviklet seg positivt, og at de etterhvert navigerte seg frem på en mer systematisk måte (Sevik et al. 2016; Utdanningsdirektoratet, 2019). I tråd med Brennan og Resnicks (2012) fokus på læringsprosessen, ble det tydelig at elevenes evne til å først utforske og eksperimentere, og deretter rette opp i feil, var avgjørende for deres mestring av oppgavene.

I likhet med Brennan og Resnick (2012), påpeker Papert (1991) at å komme frem til en løsning er ikke det viktigste, men prosessen frem til å nå målet er det som bidrar til læring og utvikling hos den enkelte (Papert, 1991). For elevpar 1 var feilsøking det mest fremtredende og dominerende kjerneelementet i deres oppgaveløsning. Feilsøking ble uttrykket gjennom hyppige endringer av kodene, hvor de måtte se etter feil, rette dem opp, og søke etter en bedre løsning på problemet (Papert, 1980). Denne tilnærmingen gjorde at elevene aktivt måtte bidra til egen læring, slik at de kunne bruke Scratch som et verktøy ved problemløsning. Feilsøkingen skulle ikke bli sett på som et hinder, men som en del av læringsprosessen (Papert, 1980).

Da elevene eksperimenterte, måtte de være kreative og bruke idéene sine til å reflektere over ulike løsninger og en «learning- by making» tilnærming, hvor kunnskapen ble skapt under problemløsningen og i samspill med hverandre (Papert, 1991). På bakgrunn av at elevene skulle løse oppgavene på egenhånd, ble de tvunget til å være kreative og finne løsninger selv (Utdanningsdirektoratet, 2019). Hver gang elevene testet en ny kode og så at dette var feil, måtte de i samspill med hverandre justere koden slik at resultatet ble riktig.

Gjennom denne prosessen utviklet elevene en «learning- by- making» tilnærming der de sammen måtte finne løsninger på problemene. Læringsprosessen til elevene var preget av at de aktivt måtte ta del i egen læring, gjennom eksperimentering og utforskning i samspill med den andre eleven i paret. Vygotsky (1978) påpeker at vi ofte lærer i samspill med andre i en sosial kontekst. Ved å bygge videre på hverandres idéer og reflektere over løsninger i fellesskap, kunne elevene løse oppgaven og skape kunnskap sammen. Når elevene oppnådde mestring av oppgavene gjennom samarbeid, utviklet de erfaring og kunnskap i interaksjon med hverandre.

Et annet interessant funn som oppstod under feilsøkingprosessen, var elevenes evne til å validere egne svar og endelige resultater. I prosessen med å validere eget svar, feilsøke begge elevparene på en systematisk måte (Bocconi et al., 2016, s. 18; Csizmadia et al., 2015, s. 9). Elevpar 1 hadde også en tendens til å «skyldte på» Scratch, når programmeringsverktøyet ikke ga ønsket resultat. Etter at begge elevparene hadde fått feil resultat gjentatte ganger, i tillegg til å være kritiske til egen programmering, ønsket de å få en bekreftelse fra læreren på om de hadde løst oppgaven riktig. Selv om en av fordelene til Scratch, er at Scratch vil gi elevene en øyeblikkelig tilbakemelding på om de har fått ønsket resultat eller ikke (The Royal Society, 2012, s. 29), ønsket de fortsatt en bekreftelse på om de hadde løst oppgaven på en tilstrekkelig måte. På bakgrunn av at jeg som lærer måtte fatte meg i korthet, slik at jeg ikke påvirket løsningen eller elevene, måtte de stole på seg selv. Da elevene sto helt fast reflekterte vi over valgene deres i fellesskap, slik at elevene ble mer bevist på hvor feilen var. Videre rettet de opp i feilen på egenhånd og løste resten av oppgaven i parene.

Elevenes anvendelse av prøving og feiling som arbeidsmetode kan være et resultat av usikkerheten blant elevene. Fellestrekket bak elevenes feilsøking baserte seg på arbeidsmåter hvor de enten angrep oppgaven uten en klar plan, fikk et resultat som lignet på det riktige resultatet eller at de tidligere i prosessen hadde programmert noe de kunne assosiere med og bygge videre på (Utdanningsdirektoratet, 2019). Mangelen på kunnskap om Scratch kan skyldes at elevene har lite erfaring med programmeringsverktøyet. På den andre siden kan man si at feilsøking er en naturlig tilnærming for elevene i denne studien, på bakgrunn av de måtte bli kjent med programmet på egenhånd gjennom gjentatt fikling (Utdanningsdirektoratet, 2019)

### **Strategisk problemløsning**

En annen observasjon i denne studien var algoritmebehandlingen til elevene. Selv om algoritmebehandling ikke oppstod like ofte som feilsøking, var dette fremtredende blant elevenes oppgaveløsning gjennom bruk av løkker, «gjenta- blokken» og «grønt flagg». «Gjenta- blokken» ble anvendt av elevpar 1 under algoritmebehandlingen, for å gjennomføre gjentatte termer med samme sekvens (Brennan & Resnick, 2012, s. 3- 4). Elevene viste denne ferdigheten ved å forenkle kodene under programmering av ulike geometriske figurer. I stedet for å anvende separate blokker, utviklet elevpar 1 en algoritme som gjentok grupper av gitte instruksjoner (Csizmadia et al., 2015, s.7)

Algoritmebehandling innebærer å definere en klar og tydelig sekvens gjennom å utvikle instruksjoner for å løse et spesifikt problem (Csizmadia et al., 2015, s. 7). Når elevene skal

utvikle sine digitale ferdigheter for å løse komplekse problemer, har algoritmisk tenkning en sentral rolle. Når elevene skal lage en algoritme er det viktig at de kan identifisere hva koden skal bestå av, og tenke systematisk i termer av sekvenser for å forstå situasjonen. Ved å gjøre dette viser elevene til viktigheten av struktur, og hvordan algoritmebehandling kan vises til gjennom kommunikasjon med logikk (Gjøvik & Torkildsen, 2019, s. 34).

Da elevpar 1 skulle programmere en blomst, ble algoritmebehandlingen en sentral tilnærming for elevenes oppgaveløsning. Algoritmebehandlingen bestod av variabler og strukturerte sekvenser, som var med på å organisere koden til elevene. Ved å bruke variabler kunne de lagre koden til blomstens blader. Gjennom en organisert rekkefølge av variabler, kunne elevene systematisk bygge opp koden sin i ulike sekvenser av instruksjoner, steg for steg (Csizmadia et al., 2015, s. 7; Utdanningsforbundet, 2019), som til sammen skulle danne en blomst.

Elevenes bruk av algoritmebehandling ble fremtredene gjennom repetisjon og automatisering av oppgaven. For elevpar 1 var «gjenta- blokken» kjent. Elevpar 1 benyttet blokken til å lage fullstendige algoritmer, som automatiserte og effektiviserte oppgaveløsningen deres. Når elevene benyttet seg av «gjenta- blokken» viste de til bruk av aritmetiske og logiske operasjoner i Scratch, hvor de kombinerte grunnleggende matematikk med logikk (Gjøvik & Torkildsen, 2019, s. 34). Dette ble spesielt tydelig når elevene opprettet flere repetisjoner av bevegelser for å programmere bladene til blomsten av sekskanter. Denne tilnærmingen resulterte i en strukturert kode, som gjorde det mulig for elevene å analysere og forutse den praktisk anvendelsen av programmering (Utdanningsdirektoratet, 2019).

I motsetning til elevpar 1 var ikke elevpar 2 kjent med «gjenta- blokken». Elevpar 2 sin tilnærming til oppgavesettet involverte aktivt bruk av «grønt flagg», for å etablere algoritmer på en ufullstendig og mindre strukturert måte. Årsaken til at deres algoritmer var ufullstendig, er et resultatet av at de ikke brukte «gjenta- blokken» for å definere en klar og tydelig sekvens (Csizmadia et al., 2015, s. 7). Istedenfor å anvende «gjenta- blokken» til å repetere handlingen, trykket elevene på «grønt flagg» gjentatte ganger. Dette førte til at elevene etablert en ubestemt sekvens. Uten å oppgi en bestemt kode for programmeringssekvensen, trykket de seg frem til ønsket resultat gjennom «grønt flagg». Denne tilnærmingen til oppgaveløsningen førte til at elevene manglet den systematiske strukturen som definerer den tydelig sekvens hos algoritmebehandlingen.



En av grunnene til at det var ulikheter i parene, og deres bruk av algoritmebehandling, kan være et resultat av deres kjennskap til funksjoner i Scratch. Ettersom elevpar 1 hadde kjennskap til «gjenta- blokken», hadde de kunnskapen for å etablere fullstendige algoritmer. I motsetning brukte elevpar 2 brukte «grønt flagg» til å gjennomføre mye av de samme oppgavene som «gjenta- blokken». Dette resulterte i at de etablerte ufullstendige algoritmer.

### **Likheter og mønster ved oppgavene**

I likhet med algoritmebehandling var generalisering fremtredende blant begge elevparene, men ikke i like stor grad som feilsøking. Når elevene kan gjenkjenne mønster og sammenhenger, for å lage allmenne regler og metoder bruker de generalisering (Gjøvik & Torkildsen, 2019, s. 33). I mine funn ble generalisering observert ved at elevene uttrykte dette gjennom identifisering av likheter, mønster og logikk, som var med på å effektivisere løsningen deres (Csizmadia et al., 2015, s. 8; Utdanningsdirektoratet, 2019).

Under løsningen av oppgavesettet viste elevpar 1 og elevpar 2 ulike tilnærminger til generalisering. Dette ble reflektert gjennom variasjon i deres tilnærming til oppgavesettet. Elevpar 2 hadde en konkret tilnærming hvor de identifiserte mønster og likheter i oppgavene. Elevene observerte sammenhengen mellom et kvadrat og et rektangel, ved å finne og bruke likheter i figurenes geometriske former og egenskaper (Csizmadia et al., 2015; Utdanningsforbundet, 2019). Dette resulterte i at elevene generaliserte de geometriske prinsippene, noe som gjorde at oppgaveløsningen deres ble mer effektiv.

På den andre siden hadde elevpar 1 en mer abstrakt tilnærming til generalisering. I stedet for å fokusere på de geometriske egenskapene til figuren de skulle programmere, hadde elevene fokus på hvordan Scratch kunne programmere figuren og løse problemet. Ved å tenke på denne måten hadde elevene i elevpar 1 en evne til å tenke på hva datamaskinen kan gjøre slik at den kan bidra til problemløsning (Wing, 2008, s. 3717- 3718). Da elevene skulle programmere en sekskant, foreslo Ola at de skulle gjenta handlingen seks ganger. Dette viser at elevene tenker på hva datamaskinen og programmeringsverktøyet kan gjøre, for å oppnå ønsket resultat. Denne tilnærmingen viser at elevene kan tenke abstrakt og anvende programmeringen på en praktisk måte.

## **Dekomponering og automatisering**

I tillegg til de nevnte kjerneferdighetene viste elevpar 1 til eksempler av dekomponering og automatisering i sin oppgaveløsning. Automatisering var kun fremtredende hos elevpar 1, da de var det eneste elevparet som tok i bruk «gjenta- blokken». Dette gjorde at de automatiserte og effektiviserte oppgaveløsningen sin (Lee, 2011, s. 33). Under alle programmeringer, med unntak av rektangelet brukte de «gjenta- blokken» for å tegne sidene på figuren. I tillegg til automatisering, viste elevpar 1 til et tilfelle av dekomponering da de møtte på utfordringer ved programmeringen av et rektangel. I stedet for å fortsette prosessen med «gjenta- blokken», gikk elevene over til å programmere rektangelet med separate blokker.

Bakgrunnen for at kun elevpar 1 tok i bruk «gjenta- blokken», kan være et resultat av at elevpar 2 aldri hadde brukt Scratch tidligere. Elevene var derfor ikke kjent med alle funksjonene til programmeringsverktøyet. På bakgrunn av at de ikke tok i bruk «gjenta- blokken» og kun programmerte koder av separate blokker, hadde de ikke noe behov for å dekomponere og bryte ned problemene i mindre delproblemer (Bocconi et al., 2016).

## **5.2 Utvikling av geometrisk kompetanse gjennom programmering i Scratch**

Å skille mellom van Hiele sine fem nivåer av geometrisk tankegang og elevenes bruk av ord og begreper for å gjenspeile deres geometriske kunnskap, har vist seg å være noe utfordrende. Til tross for dette har jeg avdekket noen forskjeller blant resultatene, der elevenes formuleringer og kunnskapshull er fremtredene.

### **Egenskaper ved geometriske figurer**

*Forståelse av geometri* går ut på å utforske og analysere form og rom (Battista, 2007, s. 843). Forståelsen av geometri ble demonstrert av begge elevparene under den første programmeringsoppgaven [kapittel 4.2.1]. Gjennom anvendelse av riktig vinkelstørrelse og lengde på sidene mestret elevene denne oppgaven fort. Elevenes evne til å overføre eksisterende kunnskap om geometri til programmeringsspråket Scratch viste til geometri- og programmeringskompetanse.

Selv om elevene viser til å riktig bruk av kunnskap til programmering under den første programmeringsoppgaven, uttrykker de også usikkerhet rundt programmering og kjennskap til de geometriske figurene videre i oppgavesettet. Mye av arbeidet med geometri tar utgangspunkt i at elevene har utviklet evnen til å se, slik at de kan dele opp figuren i enklere deler. Å dele opp figuren i enklere deler vil kunne hjelpe elevene med å beskrive og forstå egenskapene til en geometrisk figur (Solem et al., 2017, s. 142). Når elevene skal programmere geometriske figurer i Scratch, er det avgjørende at de har kjennskap til figurens geometriske egenskaper for å oppnå ønsket resultat.

Et av kompetansemålene etter 6. trinn sier at elevene skal beskrive egenskaper ved to- og tredimensjonale figurer (Kunnskapsdepartementet, 2019a). På bakgrunn av at elevene i min studie har middel- høy måloppnåelse i faget, forventes det at elevene er kjent med de grunnleggende egenskapene hos geometriske figurer. Likevel observerte jeg at elevene blandet begreper og geometriske egenskaper, som for eksempel vinkelstørrelse og lengde på parallelle sider. Et eksempel på dette var når elevene skulle programmere en likesidet trekant. Ingen av elevparene gav uttrykk for å kjenne til vinkelstørrelsen som trengs ved programmeringen av en likesidet trekant. Dette resulterte i gjentatt eksperimentering, hvor elevene testet seg frem. I tillegg til gjentatt eksperimentering syntes elevene at det var utfordrende å overføre geometriske egenskaper til Scratch, noe som var med på å svekke matematikkforståelsen deres.

En av årsakene til dette, er at elevene la merke til forskjellen mellom å programmere i Scratch og tegning av en figur med penn og papir. Elevene i denne studien var vant med å jobbe med geometri gjennom en mer tradisjonell kontekst. Når elevene skal programmere ulike geometriske figurer i Scratch må de ha kjennskap til et bestemt sett med figurer, med en referanse til geometri som en del av deres umiddelbare miljø (Fletcher, 1919, s. 654). I tillegg til dette skal de sette sammen blokker av kommandoer som kjøres etter hverandre (Gjøvik og Høyland, 2021, s. 17). Blokkene og kommandoene vil representerer sider og vinkler i figuren.

En av årsakene til at elevene møtte på utfordringer med programmering, kan skyldes deres tradisjonelle tilnærmingen til geometri. Selv om elevene har erfart geometri gjennom tegning mer enn programmering, gav de uttrykk for å være usikre på egenskapene til figurene, gjennom feil bruk av ord og uttrykk, og gjentatte forslag uten en tydelig plan eller kjennskap til figurene.

For å hjelpe elevene med programmering og oppgaveløsningen fikk de utdelt et eget ark hvor de kunne skrive ned hva de gjorde, visualiser og anvende det som et «kladdeark». På den måten kunne elevene utforske geometrien på en mer konkret måte, ved å visualisere deres geometriske former og relasjoner (Samara & Clements, 2009). Til tross for at elevene kunne kladde på et eget ark, erfarte jeg at elevene var så ivrige etter å sette i gang, at de ikke ønsket å bruke «kladdearket». Dette resulterte i gjentatt prøving og feiling, og en tilnærming hvor elevene ikke hadde noe plan.

### **Bruk av ord, begreper og definisjoner**

Et resultat av elevenes begrensede kompetanse og kjennskap til geometriske figurers egenskaper, kan føre til begrensninger i deres evne til å uttrykke seg matematisk. Dette inkluderer evnen til å snakke matematisk på en presis og nøyaktig måte, hvor elevene behersker språket gjennom riktig bruk av ord, begreper og definisjoner (Jensen, 2020, s. 169- 170). Når elevene ikke har tilstrekkelig kunnskap om de geometriske figurene, vil dette være med på å hemme deres evne til å uttrykke seg (Jensen, 2020, s. 171). Dette understreker viktigheten av å gi elevene en grundig opplæring i forskjellen mellom ord, begreper og definisjon.

I flere tilfeller kommer det frem eksempler hvor elevene bruker feil begrep når de skal omtale et kvadrat eller rektangel. I sitat 103 sier Ola, «vi får bare firkant!». I stedet for å bruke rektangel som er riktig begrep, omtaler han figuren som en firkant. Bakgrunnen for dette kan være en manglende konsensus til definisjonen av en firkant, hvor både lærerne og lærebøkene bruker ulike definisjoner når de skal bygge opp elevenes geometriforståelse knyttet til firkanter (Hovtun, 2024, s. 45- 46). For å unngå at elevene anvender feil begrep i matematikken, er det viktig å tilpasse språket til elevene, og praktiserer et matematisk språk som er tydelig (Jensen, 2020, s. 169- 170).

Elevene anvender språket som et verktøy for å argumentere, reflektere og løse oppgaver når de samarbeider i par. Gjennom språket må de kommunisere idéer, foreslå løsningsforslag og stille spørsmål til hverandre. En tydelig sammenheng med dette finner vi i Vygotsky (1978) sin teori om *redskaper* hvor vi tilegner oss kultur og felles kunnskap gjennom *språket*. Dette blir tydelig når elevene kommer frem med en ny løsningsstrategi, som de ikke har testet tidligere. Elevene bruker da språket til å tilegne seg ferdigheter som trengs for å løse geometrioppgaven i Scratch. Når elevene arbeider i par, bruker de språket for å stimulere tankeprosessen og dele erfaringer og ferdigheter med hverandre (Vygotsky, 1978). Et eksempel på dette er når elevene diskuterer vinkelstørrelsen i en sekskant. I denne samtalen må elevene utveksle idéer,

løsningsstrategier og ulike tilnærminger. Ferdighetene elevene tilegner seg gjennom samtalen og bruk av språket som redskap, forbedrer elevenes evne til å kjenne igjen geometrien i hverdagen og andre kontekster.

Når språket anvendes som et redskap er det viktig at det bli tatt i bruk på en riktig måte, slik at elevene forstår hverandre. Når elevene anvender begrepet «firkant», kan det oppstå utfordringer i kommunikasjonen. Dette skyldes at elevene ofte knytter begrepet firkant til bestemte visuelle fremstillinger av prototyper som har to parallelle sider og fire vinkler (Solem et al., 2017, s. 138). Når Ola sier «vi får bare firkant», kan han og Jens møte prototypefenomenet. Dette kan forklares ved at elevene assosierer firkant med en bestemt prototype, uten å vurdere de ulike variasjonene (Solem m. fl., 2017, s. 137- 138). Når elevene hører «firkant» kan Jens og Ola tenke på ulike prototyper av en «firkant». Dette kan føre til misforståelser blant eleven om de har ulik oppfatning av hva en «firkant» er. For å unngå slike utfordringer, er det viktig med et presist og er nøyaktig språk, som er med på å utvide begrepsforståelsen hos elevene.

I dette tilfellet er det viktig at elevene tydeliggjør hvilken «firkant» de tenker på, slik at de er bevisst på hva de legger i «firkant». Bakgrunnen for dette er at det er både et ekskluderende og et inkluderende begrep. De Villers (1994) sier at en matematisk definisjon både kan være inkluderende og ekskluderende, på bakgrunn av om begrepet er en undergruppe eller et generelt begrep (s. 11). I dette tilfellet bruker Ola firkant som et inkluderende begrep fordi begrepet tar for grupper av firkanter i et hierarki. Eksempel på dette er rektangel, kvadrat og parallelogram, hvor firkant blir sett på i en større sammenheng som spesialtilfeller av firkanter.

### **5.3 Utvikling av geometrisk tenking gjennom van Hiele nivåer**

Som nevnt tidligere var det varierende bruk av nøyaktig definisjoner, begreper, og ferdigheter hos elevparene under programmeringen av de geometriske figurene. Variasjonen kan identifiseres som ulike nivåer av geometrisk forståelse. Geometrisk forståelsen kan reflekteres i van Hieles fem nivåer, som tar for seg elevenes geometriske tanker gjennom beskrivelse av tankenivåer.

Burger og Shaughnessy (1986) hevder i sin artikkel at det kan være utfordrende å plassere elevene innenfor nivå 0- 4 (s. 41). Selv om flere av elevene er innenfor det samme nivået, kan elevene vise til ulik kompetanse. Et eksempel på dette er Josefine og Ola. Josefine identifiserte og sorterte de geometriske figurene ut ifra deres karakteristiske egenskaper, og brukte riktig

begreper når hun omtalte kvadrat og rektangel. Ola sorterte figurene etter antall vinkler. Sortering av figurene etter vinkler, gjorde han enkelt ved å anvende «gjenta- blokken» til å gjennomføre handlingen like mange ganger som figuren har vinkler. Da Ola snakker om et kvadrat og rektangel, referer han ofte til firkant. I dette eksempelet kan begge elevene plasseres på nivå 1. Selv om Josefine viser til et høyre kompetansenivå gjennom bruk av riktig begreper, kvalifiseres dette til nivå 1, akkurat som Ola. Bakgrunnen for at begge elevene blir plassert i samme nivå, er fordi van Hieles fem nivåer ikke er et nøyaktig nok verktøy for elevenes tankenivåer (Burger & Shaughnessy, 1986, s. 41). Dette betyr at elever som oppfyller kriteriene til de respektive nivåene kan vise til ulik kompetanse.

Van Hieles nivåer er avhengig av hverandre og fremtrer som dynamiske. Nivåene er dynamiske fordi individet kan bevege seg mellom nivåene i takt med deres utvikling. Når elevene er imellom to ulike nivåer er det vanskeligere å plassere dem (Burger & Shaughnessy, 1986, s. 45). Dette er fordi elevene må gjennom en sti av utvikling før de kan nå det neste og høyre nivået av geometrisk tankegang (Van de Walle et al., 2020, s. 553).

Blant mine funn ble elevene i stor grad plassert mellom van Hieles nivå 0 og 1. Når elevene flytter seg fra nivå 0 til 1, fokuserer de på egenskapene til den enkelte figuren og sorterer disse i ulike klasser (Van de Walle et al., 2020, s. 554). Ola og Jens brukte i stor grad vinklene på figuren som sorteringskriterium. Elevene observerte antall vinkler på figuren, uten å fokusere på selve figuren som skulle programmeres. Når de skulle programmere neste figur startet de ofte på nytt, og brukte ikke sammenhengen mellom egenskapene til figurene for å effektivisere programmeringssekvensen (van Hiele, 1984). Unntaket er når elevene skulle programmere en sekskant. På bakgrunn av at de har programmert en åttekant tidligere i økten, ønsker de i dette tilfellet å gjenbruke vinkelstørrelsen, da en åttekant ifølge dem lignet mer på en sekskant, enn det et kvadrat gjør. Dette var det eneste tilfellet hvor elevene sorterte figuren annerledes. I dette tilfellet sorterer elevpar 1 figurene etter, figurer med vinkelstørrelse over 90 grader.

I likhet med elevpar 1, var oppgaveløsningen til elevpar 2 kvalifisert som nivå 1. I motsetning til elevpar 1 hadde elevpar 2 et mer presist språk hvor de tok i bruk riktige begreper når de omtalte figurene. Selv om flere av deres løsningsstrategier blir plassert i nivå 1, brukte de van Hieles nivå 2 når de skulle programmere et rektangel. I dette tilfellet programmerte de rektangelet basert på likheter ved de geometriske egenskapene til et kvadrat (Van de Walle et al., 2020, s. 554). Det ble gjort ved å identifisere sammenhengen mellom egenskapene til et

kvadrat og rektangel, slik at de kunne gjenbruke koden som ble laget for å programmere et kvadrat. De måtte kun endre lengde på to av de parallelle sidene. Da Josefine skulle argumentere for dette tok hun i bruk hvis og da argumentet (van Hiele, 1984) «da kan vi bruke den samme», for å resonnerer over likheten til figurene.

Et annet funn som er sentralt for van Hieles nivåer, er samarbeidet som oppstod mellom elevene. Van Hiele (1984) påpeker at elever som er på ulikt nivå, snakker forskjellig matematisk språk. Ut i fra utsagnene til elevpar to, ser elevene ut til å være på ulikt nivå. Dette gjenspeiles i resonnementet til Josefine, hvor hun gjentatte ganger viser til et høyre nivå enn Sara. Til tross for at elevene er på ulikt nivå, gir de uttrykk for å forstå hverandre under oppgaveløsningen.

I Sara og Josefine sitt tilfelle, ser det ut til at samarbeidet virker effektivt, selv om de ikke har samarbeidet tidligere. Dette kan forklares ved at elevene må diskutere og reflektere over løsninger i samspill med hverandre. Når Josefine er på et høyre nivå enn Sara, fungerer hun som en *stillasbygger*. Dette betyr at Josefine hjelper Sara med løsning av et problem, som Sara ser på som utfordrende å løse på egenhånd (Vygotsky, 1978). Et eksempel på dette er når Josefine ser sammenhengen mellom et kvadrat og et rektangel. Da er hun tydelig med Sara og forklarer likheten mellom figurenes egenskaper, før hun videre viser til verdien til koden som de trenger å bruke.

Handlingen som oppstod mellom elevene, befant seg i *den proksimale utviklingssonen*. Dette er fordi elevene fant trygghet i hverandre og kom frem til flere løsningsstrategier i samråd med hverandre. For Josefine kan det å fungere som stillasbygger virke positivt. Dette er fordi noen blir mer selvsikre av å kunne hjelpe andre. Å fungere som stillasbygger har ikke bare betydning for undervisningen og læringen som skjer i et klasserom, men også for andre perspektiver i livet (Edward & Jones, 1999, s. 282). Samarbeidet mellom Josefine og Sara er med på å illustrere hvordan elever kan løse utfordrende oppgaver ved hjelp av hverandres styrker og kunnskaper. Dette understreker Säljö (2001) sitt argument om at kunnskapen først oppstår i samspill mellom mennesker, før det overføres til det enkelte individet.

## 5.4 Oppsummering av diskusjon

Wing (2006) ønsker seg en fremtid der algoritmisk tenkning blir sett på som en grunnleggende ferdighet i likhet med skriving, regning og lesing. På bakgrunn av dagens digitale samfunn er dette en visjon som blir mer og mer relevant. Basert på de nevnte kjerneelementene, arbeidsmetodene og nøkkelbegrepene, kan jeg se at elevene aktivt anvender algoritmisk tenkning i sin oppgaveløsning.

Elevenes arbeidsprosessen ble i stor grad påvirket av at deres deltakelse i egen læring gjennom en «learning- by- making» tilnærming, hvor kunnskap ble skapt gjennom problemløsning og i samarbeid med hverandre. Dette oppstod gjennom felles refleksjon, utforskning, deling av kunnskap, muligheten til å identifisere likheter, effektivisering og dele opp større problemer i mindre og håndterlige deler. Denne tilnærmingen ble tydelig gjennom elevens arbeid med geometrioppgaver i Scratch, hvor de måtte anvende algoritmisk tenking for å løse oppgavene.

Innblikket i arbeidsprosessen til elevene ser jeg på som et bidrag til forskningsfeltet og som funn som er av betydning for matematikklærere. Når man jobber mer med algoritmisk tenking, vil man bli bedre på å håndtere problemer (Wing, 2006). Det er derfor avgjørende at lærere som ønsker å integrere algoritmisk tenkning gjennom programmering, har kjennskap til elevenes kunnskapsnivå og er oppmerksom på de utfordringene som kan oppstå.

For elevene i denne studien var forståelse av geometri tidvis utfordrende. Usikkerhet rundt programmering, kombinert med manglende kjennskap til geometriske egenskaper, gjorde at flere av oppgavene ble mer utfordrende enn nødvendig. Da elevene er vant med en tradisjonell tilnærming til geometrien, førte dette til utfordringer knyttet til figurenes egenskap, samt blanding av ord, begreper og definisjoner. Språket er et viktig redskap under oppgaveløsning og påvirker hvordan vi tilegner oss kultur og felles kunnskap. Feil språkbruk blant elevene førte til prototypefenomenet, hvor de hadde ulike tanker og assosiasjoner til de ulike geometriske figurene.

Dette kan være et resultat av at elevene var på ulike nivåer av van Hiele's teori. Blant mine funn var elevene i stor grad plassert mellom nivå 0-1. Selv om elevparene ofte var plassert innenfor det samme nivået hadde elevpar 2 flere tilfeller av et presist språkbruk. I de tilfellene hvor det var ulikt nivå på elevene i elevparet fungerte den sterke eleven som stillasbygger. Da løste



elevene oppgaven i interaksjon med hverandre hvor den sterke elevene fungerte som en støttende stillasbygger. Denne handlingen oppstod i den proksimale utviklingssonen.

Elevenes evne til å anvende algoritmisk tenkning for å løse geometrioppgaver i Scratch viser til viktigheten av å kombinere matematikk og programmering. Dette forbedrer ikke bare elevenes digitale ferdigheter, men også deres evne til å overføre kjent matematikk til å løse komplekse problemer. På den måten kan elevene utvikle en bedre forståelse innen flere felt i matematikken.

## 5.5 Studiens begrensninger

Når man skriver en masteroppgave er det vanlig at det oppstår begrensninger i forskningen, som kan være med på å påvirke funnene som presenteres oppgaven (Abbadia, 2022). Ifølge Abbadia (2022) vil begrensningen i forskningen være med på å begrense design, metoder og forskerens begrensninger som påvirker og påvirkning av forskerens endelige funn. Derfor er det viktig å sikre gyldighet og transparens i oppgaven (Anker, 2020, s. 109). På bakgrunn av dette ønsker jeg å legge frem begrensningene ved min oppgave.

Datainnsamlingen til min masteroppgave ble gjennomført med fire strategis utvalgte elever, fordelt i to elevpar. At studien tar for seg strategisk utvalgte elever kan være med på påvirke studiens funn. En annen faktor som kan være med på å påvirke studiens funn er at studien inneholder få informanter. På bakgrunn av dette kan ikke funnene fra denne studien generaliseres (Anker, 2020, s. 110) til å gjelde alle elever på mellomtrinnet, men som beskrevet tidligere er det heller ikke hensikten ved en kvalitativ studie. Til tross for dette skal studien likevel ha overføringsverdi (Anker, 2020, s. 110). Med min studie ønsker jeg gi en dypere forståelse av elevenes algoritmiske tenkning og deres forståelse av geometriske figurers egenskaper under arbeid med Scratch, for å bidra til forskningsfeltet.

Rekrutering av informanter ble gjennomført ved en skole med elever og lærer som jeg hadde kjennskap til. Selv om elevene ble strategisk utvalgt blant de som leverte samtykkeskjema, kan min personlige tilknytning til skolen, læreren og elevene potensielt være med på å påvirke resultatene. Elevene arbeidet godt og var positive til opplegget. De syntes at arbeid i Scratch var engasjerende og at de gav uttrykk for å være lojale til meg og min masteroppgave, samtidig som de var stolte over å være blant de fire som fikk delta som informanter.

Helt til slutt vil jeg si at min rolle som observatør, også kan ha vært med på å påvirke elevenes arbeid (Tjora, 2021, s. 83). Jeg ønsket å fatte meg i korthet og ikke påvirke elevenes beslutninger eller løsningsstrategier under oppgaveløsningen. Om de sto helt fast reflekterte vi i fellesskap over ulike løsninger og tilnærminger. Dette er ikke vanlig i den typiske lærerhverdagen, hvor man ofte må håndtere en hel klasse alene.

## 6 Avslutning

I denne studien har jeg undersøkt hvordan elever på mellomtrinnet, som arbeider i par, kan anvende programmeringsverktøyet Scratch for å utvikle algoritmisk tenkning og forståelse av egenskapene til utvalgte geometriske figurer i planet. For å undersøke dette har jeg analysert hvilke kjerneferdigheter innenfor algoritmisk tenkning som kom mest til uttrykk, samt hvordan elevene uttrykte forståelse for geometriske figurer under arbeidet i Scratch. Basert på disse funnene kan jeg utlede en konklusjon som svarer på studiens formål og problemstilling:

*«På hvilke måte kan arbeid med geometrioppgaver i Scratch bidra til å utvikle barneskoleelevers algoritmiske tenkning, og forståelse av egenskapene til utvalgte geometriske figurer i planet?»*

Det første forskningsspørsmålet i denne studien hadde som formål å identifisere hvilke kjerneferdigheter innen algoritmisk tenkning som var mest fremtredende i løpet av elevenes arbeid i Scratch. Funnene i denne studien viser til eksempler hvor elevene benyttet seg av alle kjerneferdighetene til Bocconi et al. (2016 & 2018) med unntak av abstraksjon. Forekomsten av de ulike kjerneferdighetene var varierende for de to elevparene. *Feilsøke*, *algoritmebehandling* og *generalisering* var de kjerneferdighetene som var mest fremtredende i oppgaveløsningen til begge elevparene. I tillegg til dette var *dekomponering* og *automatisering* fremtredende i løsningen til elevpar 1. Funnene viser til at elevenes tidligere erfaringer og kunnskap med geometri, var avgjørende for hvordan de algoritmiske kjerneferdighetene kom til uttrykk.

En av de største utfordringene som elevene møtte på under oppgaveløsningen, var at løsningsforslagene deres var preget av mye feilsøking, hvor elevene fiklet og eksperimenterte med ulike løsninger for å oppnå ønsket resultat. Dette resulterte i at elevene benytte seg av kunnskap som ble etablert gjennom aktiv problemløsning og samarbeid sammen med hverandre, i tråd med en «learning- by- making» tilnærming. «Learning- by- making» førte til at elevene måtte identifisere og løse problemer og utvikle sin evne til problemløsning og betraktet algoritmisk tenkning som en prosess. Prosessen utviklet seg fra å ha en mindre etablert tilnærming til oppgaven gjennom fikling og eksperimentering med kreative løsninger, til en mer reflekterende og systematisk oppgaveløsning. Elevene ble opptatt av å se mønster og sammenhenger hvor de utviklet bevisste og effektive tilnærminger til oppgaveløsningen.

Studiens andre og siste forskningsspørsmål tar sikte på å undersøke hvordan elevene uttrykker sin forståelse av geometriske figurer under arbeidet i Scratch. På lik linje med elevenes algoritmiske tenkning utviklet elevene sitt matematikkspråk og geometri kompetanse under oppgaveløsningen. Til å begynne med hadde elevene en tendens til å anvende et mindre presis språk, hvor de blandet begrepene med andre begreper med lignende betydning. I tillegg til å anvende et mindre presis språk unngikk elevene bruk av spesifikke fagbegreper relatert til de geometriske figurene som skulle programmeres. Dette førte til at formidlingen deres ble mindre presis og kunne skape konflikt i kommunikasjonen. Til tross det, utviklet språket til elevene seg gradvis gjennom oppgaveløsningen, og etterhvert tilegnet de seg det samme matematikkspråket som Scratch og oppgavesettet anvendte. Endringen av språket til elevene førte til at de utviklet sin geometriske kompetanse gjennom språket. Elevene fikk en mer presis og korrekt tilnærming til figurene og deres tilnærming til programmering.

Til tross for at språket ble mer presis, hadde elevene fremdeles utfordringer knyttet til geometriforståelsen. Utfordringene kom til syne, når elevene skulle programmere de ulike geometriske figurene, da de ofte slet med å finne riktig vinkelstørrelse og lengde på sidene. I tillegg til dette, var de ikke vant med å programmere i Scratch, noe som førte til en ytterligere utfordring. På bakgrunn av at Scratch programmer fra hjørne til side, noe som er annerledes fra tradisjonell konstruksjon, ble elevene forvirret over flere av verdiene som ble brukt i kodene for programmering.

Selv om elevene gikk inn i oppgaveløsningen med relativt liten erfaring til Scratch klarte elevene å programmere alle de utvalgte figurene i planet, med unntak av blomsten. Etterhvert som elevene løste oppgavene forstod det var nødvendig å tilegne seg en systematisk tilnærming til oppgavesettet. Samtidig som de forstod nødvendigheten av å tilegne seg en systematisk tilnærming til oppgavesettet, forstod de også viktigheten av et tydelig og presis språk.

På bakgrunn av følgende funn vil jeg si at det ikke er uproblematisk å starte opp med programmering i en klasse. Å kjenne til sentrale begreper og egenskaper er essensielt for at elevene skal klare å overføre kunnskap de har tilegnet gjennom konstruksjon og tegning til digital programmering. Når man innfører programmering i skolen handler det ikke bare om at elevene skal lære seg å forstå og bruke datamaskinen, eller utvikle sin digitale kompetanse. Det er vesentlig at elevene tilegner seg verktøy for å løse oppgavene og problemene de møter på underveis og tilegner seg ferdigheter og kompetanse som er nødvendig for den moderne

hverdagen og skolen. For at lærerne skal vite hvilke kompetanser og ferdigheter som skal være sentrale under opplæringen av programmering og algoritmisk tenkning er det vesentlig at de kjenner til begrepenes innhold og gjennomføring.

For elevene i denne studien ville det ha vært en fordel å ha erfaring med begreper innenfor programmering og geometri i planet. Erfaring med begreper kunne potensielt ha bidratt til å styrke elevenes digitale kompetanse, slik at overgangen fra å tegne geometriske figurer til å produsere geometriske figurer digitalt ville vært mindre. For å gjøre overgangen fra konstruksjon til digital tegning effektiv, er det viktig at læreren grundig introduserer programmering og inkluderer opplæring i analog programmering. Dette vil være verdifullt for elevene slik at de tilegner seg bedre forståelse for ulike prinsipper og metoder som blir anvendt i koding.

Jeg presenterer disse funnene, med hensikt å anerkjenne at arbeidet ikke har vært uten utfordringer. Spesielt har det vært utfordrende å starte opp programmering for elever uten erfaringer med Scratch. Likevel håper jeg at funnene i denne oppgaven vil veie opp mot de utfordringene som oppstod underveis. Jeg ønsker at matematikklærere tar oppfordringen om å være bevisste på sitt eget språk i undervisning, og gi elevene grundig opplæring i analog programmering som en introduksjon til digital programmering.

## **6.1 Forslag til videre forskning**

Basert på mine funn og avgrensninger ville det vært interessant å følge flere elever over en lenger periode for å se hvordan kompetansen deres hadde utviklet seg over tid. Dette ville gitt meg muligheten til å se om alle elever er like entusiastiske til programmering. Videre kunne studiet utvides til å inkludere tradisjonelle og digitale tilnærminger til matematikkundervisning om geometri. Dette ville gitt meg innsikt i likheter og forskjeller mellom de to ulike tilnærmingene, og hvordan dette påvirker elevenes geometriske forståelse. For å utvide dette enda mer kunne man ha intervjuet elevene for å høre deres refleksjoner om dette.

En annen moment som ville vært interessant å se nærmere på er sammenligningen av visuell blokkprogrammering som Scratch, med tekstbasert blokkprogrammering, som Python. Her kunne jeg ha undersøkt nærmere, hvilket programmeringsspråk som fremmer algoritmisk tenkning mest.

Jeg valgte å begrense mitt studie til elever på det siste året på barnetrinnet. Det kunne vært interessant å se hvordan dette utvikler seg til første året på ungdomstrinnet. Dette ville gitt meg et større bilde av hvordan programmering integreres i utdanningen på tvers av to påfølgende trinn, og hvordan dette påvirker elevenes algoritmiske tenking og motivasjon.

# Litteraturliste

- Abbadia, J. (2022). *Hva er begrensningene i forskningen, og hvordan skriver man dem?*  
<https://mindthegraph.com/blog/nb/begrensninger-i-forskningen/>
- Ackermann, E. (2001). *Piaget's Constructivism, Papert's Constructionism: What's the difference?*  
[https://learning.media.mit.edu/content/publications/EA.Piaget%20\\_%20Papert.pdf](https://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf)
- Andersen, S. S. (2013). *Casestudier: forskningsstrategi, generalisering og forklaring* (2. utg.)  
Fagbokforlaget
- Andersson- Bakken, E., Bjørnstad, E. & Dalland, C. P. (2021). Observasjon som metode i barnehage- og klasseromsforskning. I E. Andersson- Bakken & C. P. Dalland (Red.). *Metoder i klasseromsforskning*. (s. 125- 152). Universitetsforlaget.
- Anker, T. (2020). *Analyse i praksis: en håndbok for masterstudenter*. Cappelen Damm akademisk.
- Battista, M. T. (2007). The Development of Geometric and Spatial Thinking. I F. K. Lester (Ed.), *Second Handbook of Research on Mathematics Teaching and Learning* (s. 843-908). Charlotte, NC: Information Age.
- Blikstad- Balas, M. & Dalland C. P. (2021). Forskningsdesign- hva må du tenke på når du skal planlegge et forskningsprosjekt? I E. Andersson- Bakken & C. P. Dalland (Red.). *Metoder i klasseromsforskning*. (s. 21- 45). Universitetsforlaget.
- Braun, V & Clarke, V. (2006). Using Thematic Analysis in Psychology. I *Qualitative research in psychology*, 3(2), (s. 77- 101). <https://biotap.utk.edu/wp-content/uploads/2019/10/Using-thematic-analysis-in-psychology-1.pdf.pdf>
- Brinkmann, S., Hansen, W. & Tanggaard, L. (2012). *Kvalitative metoder: empiri og teoriutvikling*. Gyldendal akademisk.
- Brennan, K. & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. *American educational research association*.  
[https://web.media.mit.edu/~kbrennan/files/Brennan\\_Resnick\\_AERA2012\\_CT.pdf](https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf)
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P. & Punie, Y. (2016). Developing Computational Thinking in Compulsory Education: Implications for policy and practice. *EUR - Scientific and Technical Research Reports*.  
<https://doi.org/10.2791/792158>

- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). *The Nordic approach to introducing computational thinking and programming in compulsory education*. DOI:10.17471/54007
- Burger, W. F. & Shaughnessy, J. M. (1986). Characterizing the van Hiele levels of development in geometry. *National Council of Teachers of Mathematics*, 17(1), 31-48. <https://www.jstor.org/stable/749317>
- Clausen, E. (2017). *Undervisningsopplegg- Geometri og Scratch*. Espens klasserom. <https://espensklasserom.com/2017/02/24/geometri-og-scratch/>
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). *Computational Thinking: A Guide for Teachers*. *Computing At School*. [https://eprints.soton.ac.uk/424545/1/150818\\_Computational\\_Thinking\\_1\\_.pdf](https://eprints.soton.ac.uk/424545/1/150818_Computational_Thinking_1_.pdf)
- De Villiers, M. (1994). The Role and Function of a Hierarchical Classification of Quadrilaterals. *FLM Publishing Association*, 14(1), 11–18. [https://www.researchgate.net/publication/313730259\\_The\\_role\\_and\\_function\\_of\\_a\\_hierarchical\\_classification\\_of\\_quadrilaterals](https://www.researchgate.net/publication/313730259_The_role_and_function_of_a_hierarchical_classification_of_quadrilaterals)
- Edwards, J-A. & Jones, K. (1999). Students' views of learning mathematics in collaborative small groups. I O. Zaslavsky (Red.), *Proceedings of the 23rd Conference of the International Group for the Psychology of Mathematics Education* (s. 281- 288). [https://www.researchgate.net/publication/266497977\\_Students'\\_views\\_of\\_learning\\_mathematics\\_in\\_collaborative\\_small\\_groups](https://www.researchgate.net/publication/266497977_Students'_views_of_learning_mathematics_in_collaborative_small_groups)
- Fletcher, W. H. (1919). Concrete Geometry for the Seventh Grade. *Journal of Education*, 90(24), 654- 657. <https://doi.org/10.1177/002205741909002403>
- Forsström, S. E. & Kaufmann, O. T. (2018). A systematic literature review exploring the use of programming in mathematics education. *Østfold University Collage, Faculty of Education*, Norway, Østfold.
- Fuys, D., Geddes, D. og Tischler, R. (1988). The Van Hiele Model of Thinking in Geometry among Adolescents. *National Council of Teachers of Mathematics*. DOI:10.2307/749957
- Gjøvik, Ø. & Høyland, J. (2022). *Kloss for kloss: blokkprogrammering for lærere*. Universitetsforlaget.
- Gjøvik, Ø. & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten – tidsskrift for matematikundervisning*, 30(3), 31-37. <http://tangenten.no/wp-content/uploads/2021/12/Tangenten-3-2019-Gjovik-Torkildsen.pdf>



- Gold, R. L. (1958). Roles in Sociological Field Observations. *Social Forces*, 6(3), 217- 222.  
<https://doi.org/10.2307/2573808>
- Grover, S. & Pea, R. (2013). Computational Thinking i K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38- 43.  
<https://doi.org/10.3102/0013189X12463051>
- Hovtun, G., Dreyer, T. (2024). Definisjoner på firkanter i lærebøker. *Tangenten – tidsskrift for matematikk undervisning*, 35(1), 34– 47.
- Høgheim, S. (2020). *Masteroppgaven i GLU*. Fagbokforlaget.
- Imsen, G. (2020). *Elevens verden. Innføring i pedagogisk psykologi*. Universitetsforlaget.
- Jensen, R. (2020). Ord og begreper i matematikk. I Ulland, G. & Jensen, R. (Red). *Ord og begreper i klasserommet* (s. 169- 182). Fagbokforlaget.
- Johannessen, L. E. F., Rafoss, T. W., & Rasmussen, E. B. (2018). *Hvordan bruke teori?: nyttige verktøy i kvalitativ analyse*. Universitetsforlaget
- Kunnskapsdepartementet. (2017). Framtid, fornyelse og digitalisering: *Digitaliseringsstrategi for grunnpoplæringen 2017-2021 (F-4435 B)*.  
[https://www.regjeringen.no/contentassets/dc02a65c18a7464db394766247e5f5fc/kd\\_fr\\_amtid\\_fornyelse\\_digitalisering\\_net.pdf](https://www.regjeringen.no/contentassets/dc02a65c18a7464db394766247e5f5fc/kd_fr_amtid_fornyelse_digitalisering_net.pdf)
- Kunnskapsdepartementet. (2019a). *Læreplan i matematikk 1.-10. trinn (MAT01-05)*. Fastsatt som forskrift. Læreplanverket for kunnskapsløftet 2020.  
<https://www.udir.no/lk20/mat01-05? lang=nob>
- Kunnskapsdepartementet (2019b). *Hva er kjerneelementer?* <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hva-er-kjerneelementer/>
- Kunnskapsdepartementet. (2019c). *Kjerneelementer*. <https://www.udir.no/lk20/mat07-02/om-faget/kjerneelementer>
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J. & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.  
[https://www.researchgate.net/publication/234810765\\_Computational\\_thinking\\_for\\_youth\\_in\\_practice](https://www.researchgate.net/publication/234810765_Computational_thinking_for_youth_in_practice)
- Litherland, K. (2001). Programmeringsforskning i skolen - før og nå. *Kunnskapsfilm.no*.  
[https://kunnskapsfilm.no/wp-content/uploads/2021/11/Programmeringsforskning\\_i\\_skolen\\_foer\\_og\\_naa.pdf](https://kunnskapsfilm.no/wp-content/uploads/2021/11/Programmeringsforskning_i_skolen_foer_og_naa.pdf)

- Lund, M. (2023). *Slik påvirker algoritmene hverdagen din*. Advania. <https://www.advania.no/kunnskap/artikler/slik-pavirker-algoritmene-hverdagen-din>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
- Maxwell, J. A. (2009). Designing a Qualitative Study. I L. Bickman & D.J. Rog (Red.), *The 122 SAGE Handbook of Applied Social Research Methods* (s. 214-250). London: Sage
- NOU 2014:7. (2014). *Elevenes læring i fremtidens skole — Et kunnskapsgrunnlag*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/NOU-2014-7/id766593/>
- NOU 2020:2. (2020). *Fremtidige kompetansebehov III — Læring og kompetanse i alle ledd*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/nou-2020-2/id2689744/>
- Nyeng, F. (2012). *Nøkkelbegreper i forskningsmetode og vitenskapsteori*. Fagbokforlaget
- Nygaard, V. (2022). Transkribering med koding. *Din transkribent*. <https://www.dintranskribent.no/transkribering-med-koding/>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism*. Norwood, NJ: Ablex. [https://web.media.mit.edu/~calla/web\\_comunidad/Reading-En/situating\\_constructionism.pdf](https://web.media.mit.edu/~calla/web_comunidad/Reading-En/situating_constructionism.pdf)
- Papert, S. (1993). *Mindstorms: children, computers, and powerful ideas* (2. utg.). New York: Basic Books. [https://worrydream.com/refs/Papert\\_1980\\_-\\_Mindstorms,\\_1st\\_ed.pdf](https://worrydream.com/refs/Papert_1980_-_Mindstorms,_1st_ed.pdf)
- Postholm, M. B., Jacobsen, D. I., & Søbstad, R. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm akademisk.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). *Scratch: Programming For All*. *Communications of the ACM*, 52(11), 60- 67. DOI: 10.1145/1592761.1592779. <https://dl.acm.org/doi/10.1145/1592761.1592779>
- Sarama, J. & Clements, D. H. (2009) Learning Trajectories in Early Mathematics Education. *Researching and Using Progressions (Trajectories) in Mathematics Education*. DOI: [10.1163/9789004396449\\_002](https://doi.org/10.1163/9789004396449_002).

[https://www.researchgate.net/publication/335554829\\_Learning\\_Trajectories\\_in\\_Early\\_Mathematics\\_Education/stats](https://www.researchgate.net/publication/335554829_Learning_Trajectories_in_Early_Mathematics_Education/stats)

Sevik, K. (2016). *Programmering i skolen*. Notat fra Senter for IKT i utdanningen. [https://www.udir.no/globalassets/filer/programmering\\_i\\_skolen.pdf](https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf)

Sikt. (u.å.). *Meldeskjema for personopplysninger i forskning*. <https://sikt.no/fylle-ut-meldeskjema-personopplysninger>

Slotfeldt- Ellingsen, D. (2020). *Forskningsetikk: yrkesetikk ved forskningsvirksomhet*. Universitetsforlaget.

Smith, D. E. (1958). History of mathematics, 429 (30). *Courier Corporation*. <https://books.google.no/books?id=12qdOZ0gsWoC&lpg=PR13&lr&hl=no&pg=PA22#v=onepage&q&f=false>

Solem, I. H., Alseth, B., Eriksen, E., Smestad, B., Ødegaard, E., Vetlesen, E., & Paiam, V. (2017). *Tall og tanke 2: matematikkundervisning på 5. til 7. trinn*. Gyldendal akademisk.

Stengrundet, S. & Valbekmo, I. (2019). *Begrepslæring og begrepsforståelse i matematikk*. <https://realfagsloyper.no/sites/default/files/2019-03/T3.P1.M2A%20Begrepsl%C3%A6ring%20og%20begrepsforst%C3%A5else%20i%20matematikk.pdf>

Säljö, R. (2001). *Læring i praksis: et sosiokulturelt perspektiv*. Cappelen akademisk forlag.

Säljö, R. (2016). *Læring – introduksjon til perspektiver og metaforer*. Oslo: Cappelen Damm akademisk.

The Royal Society. (2012). Shut down or restart? The way forward of computing in UK schools. *The Royal Society*. <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>

Tjora, A. H. (2021). *Kvalitative forskningsmetoder i praksis* (4. utgave.). Gyldendal.

Universitetet i Oslo. (2017). *Nettskjema-diktafon mobilapp*. <https://www.uio.no/tjenester/it/adm-app/nettskjema/hjelp/diktafon.html>

Usiskin, Z. (1982). Van Hiele Levels and Achievement in secondary School Geometry. CDASSG Project. *The university of Chicago*. <https://eric.ed.gov/?id=ED220288>

Utdanningsdirektoratet. (2019). *Algoritmisk tenkning*. <https://www.udir.no/kvalitet-og-kompetanse/digitalisering/algoritmisk-tenkning/>

- Van de Walle, J. A., Karp, K. S., Bay-Williams, J. M., Wray, J., & Brown, E. T. (2020). *Elementary and middle school mathematics: teaching developmentally* (Tenth edition.; Global edition.). *Pearson*.
- van Hiele, P. M. (1984). *A Child's Thought and Geometry*. I D. Fuys (Red.), *English Translation of Selected Writings of Dina van Hiele-Geldof and Pierre M. van Hiele* (s. 247-255).
- Vygotsky, L. S. (1978). *Mind in Society: Development of Higher Psychological Processes*. *Harvard University Press*.
- Wing, J. M. (2006). *Computational thinking*. *Community. ACM* 49, 3 (March 2006), (s. 33–35). <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). *Computational thinking and thinking about computing*. *Philosophical Transactions Royal Society A.*, 366, 3717– 3725. [https://www.researchgate.net/publication/23142610\\_Computational\\_thinking\\_and\\_thinking\\_about\\_computing](https://www.researchgate.net/publication/23142610_Computational_thinking_and_thinking_about_computing)
- Wing, J. M. (2010). *Computational thinking: what and why?* <https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Wittek, L. & Brandmo, C. (2014). *Ulike tilnærminger til læring*. I J.H. Stray & L. Wittek (Red.), (s. 113- 132). *Pedagogikk: en grunnbok*. Cappelen Damm Akademisk.

# Oversikt over figurer og tabeller

Figur 2.1 Min fremstilling av Vygotskys (1978) proksimale utviklingssone .....	9
Figur 2.2 Utdanningsdirektoratets fremstilling av den algoritmiske tenkeren .....	13
Figur 2.3 Skjermdump fra CompuThink rapporten (Bocconi et al., 2016, s.17) .....	17
Figur 2.4 Min fremstilling av Bocconi et al. (2016 6 2018) og Gjøvik & Torkildsen (2019) rammeverk for algoritmisk tenkning .....	19
Figur 3.1 Utklipp fra Miro .....	37
Figur 3.2 Min fremstilling av koder lagt inn under kategorier i Miro .....	37
Figur 4.1 Oversikt over prosessen frem til en blomst av sekskanter .....	45
Figur 4.2 Kode tilhørende sitat 440 .....	46
Figur 4.3 Utdrag fra elevbesvarelse, elevpar 2 .....	58
Tabell 1 Utdrag fra koding .....	36
Tabell 2 Utdrag fra transkripsjon, feilsøking, elevpar 2, sitat 70-71 .....	44
Tabell 3 Utdrag fra transkripsjon, algoritmebehandling, elevpar 1, sitat 352- 360 .....	46
Tabell 4 Utdrag fra transkripsjon, algoritmebehandling, elevpar 2, sitat 27 .....	47
Tabell 5 Utdrag fra transkripsjon, algoritmebehandling, elevpar 2, sitat 160- 161 .....	48
Tabell 6 Utdrag fra transkripsjon, generalisering, elevpar 1, sitat 287 .....	49
Tabell 7 Utdrag fra transkripsjon, generalisering, elevpar 2, sitat 41 .....	50
Tabell 8 Utdrag fra transkripsjon, dekomponering, elevpar 1, sitat 105- 127 .....	51
Tabell 9 Utdrag fra transkripsjon, automatisering, elevpar 1, sitat 46- 51 .....	52
Tabell 10 Utdrag fra transkripsjon, elevpar 1, sitat 80- 103 .....	53
Tabell 11 Utdrag fra transkripsjon, elevpar 2, sitat 41- 53 .....	54
Tabell 12 Utdrag fra transkripsjon, elevpar 2, sitat 57- 70 .....	55
Tabell 13 Utdrag fra transkripsjon, elevpar 1, sitat 267- 276 .....	56
Tabell 14 Utdrag fra transkripsjon, elevpar 1, sitat 290- 292 .....	56
Tabell 15 Utdrag fra transkripsjon, elevpar 2, sitat 51 .....	57
Tabell 16 Utdrag fra transkripsjon, elevpar 2, sitat 103 .....	58

# Vedlegg

## Vedlegg I: Godkjenning fra Sikt



Norsk ▾ Cornelia Viken ▾

[Meldeskjema](#) / [En kvalitativ studie om fremstillingen av geometri gjennom programmer Scratch](#) / [Vurdering](#)

### Vurdering av behandling av personopplysninger

Skriv ut

28.09.2023 ▾

**Referansenummer**

997941

**Vurderingstype**

Standard

**Dato**

28.09.2023

**Tittel**

En kvalitativ studie om fremstillingen av geometri gjennom programmer Scratch

**Behandlingsansvarlig institusjon**

Universitetet i Sørøst-Norge / Fakultet for humaniora, idrett- og utdanningsvitenskap / Institutt for matematikk og naturfag

**Prosjektansvarlig**

Pål- Erik Eidsvig

**Student**

Cornelia Viken

**Prosjektperiode**

16.08.2023 - 03.06.2024

**Kategorier personopplysninger**

Alminnelige

**Lovlig grunnlag**

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

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 03.06.2024.

**Kommentar****OM VURDERINGEN**

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

**KOMMENTAR TIL INFORMASJONSSKRIVET**

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

- Hvilken institusjon som er behandlingsansvarlig

Ta gjerne en titt på våre nettsider for hjelp til formuleringer: <https://sikt.no/informasjon-til-deltakarane-i-forskingsprosjekt>

**FØLG DIN INSTITUSJONS RETNINGSLINJER**

Det er institusjonen du er ansatt/student ved som avgjør hvordan du må lagre og sikre data i ditt prosjekt og hvilke databehandlere du kan bruke. Husk å bruke leverandører som din institusjon har avtale med (f.eks. ved skylagring, nettspørreskjema, videosamtale el.).

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

**MELD VESENTLIGE ENDRINGER**

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


**OPPFØLGING AV PROSJEKTET**

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

Lykke til med prosjektet!

## Vedlegg II: Informasjonsskriv til foresatte med samtykkeerklæring

Vil ditt barn delta i mitt forskningsprosjekt  
«En kvalitativ studie om bruk av Scratch som verktøy i arbeid med geometri»

Til foresatte ved 6. trinn på 

Dette er et spørsmål til deg/dere om ditt/deres barn kan delta i et forskningsprosjekt hvor formålet er å se hvordan den algoritmiske tenkeren kommer til uttrykk gjennom arbeid med programmering i Scratch. Dette skrivet vil gi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for ditt barn.

### Formål

Dette er et masterprosjekt om programmering i matematikken, gjennomført av en lærerstudent. I den nye læreplanen (LK20) ble mye av innholdet endret, og programmering og den algoritmiske tenkeren fikk en fremtredende rolle. I den forbindelse ønsker jeg å se nærmere på hvordan Scratch kan brukes som et verktøy og for å se hvordan den algoritmiske tenkeren kommer til uttrykk, med søkelys på det matematiske temaet geometri.

I min oppgave vil jeg ta for meg

- *Hvordan kommer den matematiske samtalen til uttrykk gjennom arbeid med Scratch, og hvordan kan dette være med på å øke kompetansen om geometriske figurer?*

Dataene som samles inn vil kun brukes i dette masterprosjektet, og det er viktig å påpeke at alle deltakere vil bli anonymisert i oppgaven. Jeg ønsker å observere 1 matematikktime hvor ditt/ deres barn arbeider med programmeringsoppgaver i par. I tillegg ønsker jeg å ta skjermopptak av prosessen, og supplere med lydopptak, slik at jeg kan sikre at dataene er så detaljerte som mulig. Det vil verken skrives eller spørres om personopplysninger.

### Hvem er ansvarlig for forskningsprosjektet?

Cornelia Viken

### Hvorfor får du spørsmål om å delta?

Ditt/ deres barn får spørsmål om å delta i dette masterprosjektet fordi programmering er et sentralt aspekt i læreplanen LK20. I tillegg er din barneskole i kommunen som tilhører mitt universitet.

### Hva innebærer det for deg å delta?

Før timen starter vil det bli gitt informasjon til elevene på forhånd om hvem jeg er og hvorfor jeg er hos dem. Deretter vil jeg observere 1 matematikktime hos klassen til ditt/deres barn, hvor det bli gjort anonyme feltnotater under observasjonen, samt skjermopptak og lydopptak.

Opplysningene som samles inn, er navn på samtykkeskjema. I tillegg til dette er det en viss sannsynlighet for at elevene nevner hverandres navn i lydopptaket.

### Det er frivillig å delta

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

Det vil ikke påvirke ditt eller ditt barns forhold til verken skolen eller ansatte ved skolen. De som ikke deltar, vil få samme tilbud om undervisning uten å bli observert.

### **Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger**

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrevet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. Navnet og kontaktopplysningene dine vil vi erstatte med en kode som lagres på egen navneliste adskilt fra øvrige data.

Cornelia Viken, og veileder Pål- Erik Eidsvig vil ha tilgang til datamaterialet. Datamaterialet vil også kunne diskuteres med medstudenter, men vil da anonymiseres først.

### **Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?**

Prosjektet vil etter planen avsluttes 3. juni 2024. Alle opptak slettes etter prosjektslutt og observasjonsnotater vil bli makulert.

### **Hva gir oss rett til å behandle personopplysninger om deg?**

Jeg vil behandle opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Sørøst-Norge har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

### **Dine rettigheter**

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

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

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

- Universitetet i Sørøst- Norge ved Pål- Erik Eidsvig ([pal-erik.eidsvig@usn.no](mailto:pal-erik.eidsvig@usn.no)) og Cornelia Viken ([vcornelia@hotmail.no](mailto:vcornelia@hotmail.no))
- Vårt personvernombud: Paal Are Solberg ([personvernombud@usn.no](mailto:personvernombud@usn.no))

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

- Personverntjenester på epost ([personverntjenester@sikt.no](mailto:personverntjenester@sikt.no)) eller på telefon: 53 21 15 00

Med vennlig hilsen

Pål- Erik Eidsvig

(Veileder)

Cornelia Viken

(Student)



---

**Samtykkeerklæring**

Jeg har mottatt og forstått informasjon om prosjektet og har fått anledning til å stille spørsmål. Jeg samtykker til:

å delta i observasjon

å delta i lydopptak

Jeg samtykker til at mitt barns opplysninger behandles frem til prosjektet er avsluttet

---

(Ditt barns navn)

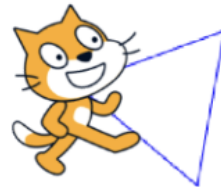
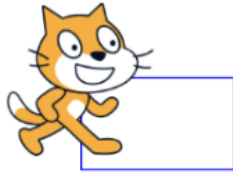
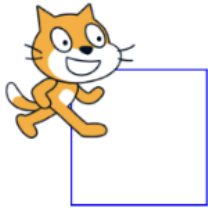
---

(Signert av foresatt, dato)

## Vedlegg III: Oppgavesett brukt ved datainnsamling

### Oppgave 1

- a) Hjelp Felix med å programmere de ulike geometriske figurene:  
kvadrat, rektangel og en likesidet trekant.



- b) Hvilke forskjeller og likheter finner du hos de ulike figurene og kodene?

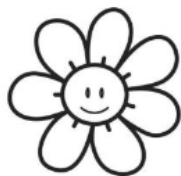
### Oppgave 2

- a) Tegn en sekskant i Scratch hvor alle sidene har lengde 100 steg.

### Oppgave 3

Felix ønsker å lage en blomst av sekskanter:

- a) Tegn en blomst i Scratch hvor du kun bruker sekskanter.  
b) Hvor mange sekskanter trenger du for at det skal bli en blomst?



## Vedlegg IV: Transkripsjonsnøkkel inspirert av Du Bois- system

X	En stavelse eller et ord som ikke er hørbart
..	Kortere pause med varighet på under 0,5 sekund
...	Middels lang pause som varer 0,5–1 sekund
...2.0	Langvarig pause
XordX	Usikker/ utydelig transkripsjon
@	Latter
<@ ord @>	Lattermild prating
@ sang @	Sang/ nynning
(HVISKING)	Hvisking
(GJESP)	Gjesp/ gipser
(HOST)	Hosting (eller kremting)
'ord	Ordet legges trykk på
[ord]	Personer snakker i munnen på hverandre
?	Spørrende / stigende intonasjonskontur
–	Ord avbrytes
---	Upassende ord
*skriver*	Skriver/ noterer på notatark