

# **MASTEROPPGAVE**

**MGMO5900**

**Mai 2023**

## **Tittel**

Hvilken rolle algoritmisk tenkning har i utforskende arbeid i et PRIMM-basert naturfagsopplegg

Computational Thinking's Role in Inquiry-Based Educational Schemes Using PRIMM

30 stp. oppgave

Mats Alexander Wennevold



**OsloMet – storbyuniversitetet**

**Fakultet for lærerutdanning og internasjonale studier**

**Institutt for grunnskole- og faglærerutdanning**

## Innhold

Forord .....	1
Sammendrag .....	3
Abstract .....	4
Hvilken rolle algoritmisk tenkning har i utforskende arbeid i et PRIMM-basert naturfagsopplegg .....	5
Forskningsspørsmål .....	6
MASCOT .....	7
Teori.....	7
Hva er algoritmisk tenkning?.....	7
Algoritmisk tenkning i skolen .....	12
Programmering i skolen .....	12
PRIMM .....	13
Begrepsbruk i naturfag .....	14
Dybdelæring i naturfag.....	15
Pedagogisk sammenhengsskaping .....	17
Stillasbygging .....	17
Studier av PRIMM i skolen.....	18
Bidrar PRIMM til algoritmisk tenkning .....	19
Tidligere empirisk forskning på algoritmisk tenkning i skolen .....	21
Utforskende arbeid i naturfag.....	21
Tidligere forskning.....	22
Metode .....	23
Kvalitativ metode .....	23
Utvalg.....	23
Datainnsamling.....	24
Undervisningsforløp .....	24
Transkribering.....	25
Analyse av datamaterialet.....	25
Etiske vurderinger .....	28
Metodens begrensninger .....	28
Kvalitet i studien .....	29
Resultater .....	29
Hvordan brukes algoritmisk tenkning i de ulike fasene av PRIMM.....	30

Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tekning .....	36
Diskusjon .....	43
Algoritmisk tenkning i et PRIMM-basert naturfagsopplegg .....	43
Stillasbygging .....	45
Dybdelæring .....	49
Studiens begrensninger .....	53
Videre studier .....	54
Konklusjon .....	54
Litteraturliste .....	58

## Forord

Sommeren 2018 fikk jeg beskjed om at jeg var kommet inn på OsloMet, lektorprogrammet i grunnskoleutdanning. Nå, snart fem år senere, synes det ufattelig hvor fort disse årene har gått. Men jeg ser hvor innholdsrike og meningsfulle de har vært, og hvor lærerikt studiet har vært for meg.

Da jeg denne våren skulle skrive masteroppgaven, var jeg innledningsvis overveldet. Det å ta fatt på en så stor oppgave som skulle være det siste jeg i mitt universitetsliv gjennomfører, syntes enormt – nesten altoppslukende. Likevel sitter jeg nå på slutten med en opplevelse av at det hele har vært en udelt positiv og lærerik prosess. Erfaringene som jeg har fått av å gå på OsloMet har vært uvurderlige. Kunnskapen og ferdighetene som jeg har opparbeidet meg, og alle opplevelsene som jeg har gjort både faglig og sosialt, har formet meg som menneske og vil ha stor betydning for livet som venter.

Jeg vil først og fremst takke min veileder Katarina Pajchel for all den gode veiledningen og støtten som jeg har fått gjennom hele dette prosjektet. Ditt engasjement har vært smittsomt. Du har inspirert meg hele veien med kunnskaper og positive innspill, og gitt uvurderlig motivasjon og bygget stå-på-vilje.

I tillegg til veilederen som jeg har hatt under dette prosjektet, vil jeg takke alle de gode lærerne som jeg har hatt i de ulike fagene gjennom hele mitt løp på OsloMet. Dere har alle bidratt positivt til at jeg nå står ved avslutningen av lektorprogrammet, og kan ta fatt på min lærergjerning.

Jeg vil også takke mine medstudenter, som jeg har delt mange opplevelser med. Det har vært forelesinger og undervisning, støttende kollokviegrupper og diskusjoner, sene studiekvelder. I tillegg har dere vært et sosialt ankerfeste. Hvem trodde vel at studentlivet ville utspille seg på Zoom da vi traff hverandre? Heldigvis fikk vi etter hvert tilbrakt mye tid sammen, og samholdet har skapt minner og ikke minst vennskap som jeg vil ta med meg videre i livet.

Avslutningsvis vil jeg takke mine venner for friluftsliv og fester, disc golf-runder, gamingkvelder og samtaler – dere er best! Og jeg vil takke familien som har støttet meg de siste årene – dere har vært en verdifull motvekt til universitetsstudiet, og heiet meg frem.

Helt til sist vil jeg takke min mor. Takk for at du har trodd på meg, oppmuntret meg, brygget kaffe og fylt termokopper, sørget for matpakker i praksisperioder, vist omsorg. Også takk for

alle samtaler om fag og læreryrket – det har vært til stor hjelp gjennom hele studiet. Og takk for at vi deler på Mille-pus, som er verdens beste koseklump og trøst når det stormer.

Mats Alexander

## Sammendrag

Bakgrunnen for denne oppgaven er å få innsikt i programmering og tenkemåtene som følger med denne ferdigheten i skolen. Jeg ønsket videre å se på det økende fokuset på utforskende arbeid og undring som blir fremhevet i LK20 sine kjerneelementer for naturfag. Denne oppgavens hensikt er å få et innblikk i hvilken rolle algoritmisk tenkning har i PRIMM-basert naturfagundervisning, og hvordan det tilrettelegger for utforskende arbeid.

For å svare på dette har jeg i åtte uker fulgt undervisningen i en 6.-klasse, som har gjennomgått et PRIMM-basert undervisningsopplegg knyttet til temaet elektrisitet. Elevene har i denne perioden arbeidet med programmering og elektriske kretser, hvor de til slutt skal designe og bygge et smart-hus ved hjelp av en microbit. Jeg har i denne oppgaven analysert elevenes samtaler fra undervisningen der de arbeidet med å lage elektriske kretser med hjelp av en Microbit som de har programmert.

I studien kom det frem at elevene anvendte (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning i et PRIMM-basert undervisningsopplegg. Når elevene anvendte disse praksisene, tok de også i bruk et utforskende språk, og brukte det nøkkelbegrepene som blir omtalt av (Haug, 2016). I denne studien kom det frem at elevene når de anvendte (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning og tok i bruk et utforskende språk, viste til stillasbygging og dybdelæring. ”

## Abstract

The background for this task is to gain insight into programming and the ways of thinking that come with this skill in school. I also wanted to look at the increasing focus on exploratory work and wonder that is highlighted in LK20's core elements for science. The purpose of this assignment is to gain an insight into the role of algorithmic thinking in PRIMM-based science education, and how it facilitates exploratory work.

To answer this, for eight weeks I have been teaching in a 6th grade, which has undergone a PRIMM-based teaching program related to the topic of electricity. During this period, the students have worked with programming and electrical circuits, where they will eventually design and build a smart house using a Microbit. In this assignment, I have analyzed the students' conversations from the teaching where they worked on creating electrical circuits with the help of a Microbit that they have programmed.

The study revealed that the students (Weintrop et al., 2016) applied their algorithmic thinking practices in a PRIMM-based teaching program. When the students applied these practices, they also used exploratory language, and used the key concepts discussed by it (Haug, 2016). In this study, it emerged that when the learners applied (Weintrop et al., 2016) their algorithmic thinking practices and used exploratory language, referred to scaffolding and deep learning.

## Hvilken rolle algoritmisk tenkning har i utforskende arbeid i et PRIMM-basert naturfagsopplegg

Ferdighetene som kreves i samfunnet er i konstant endring. Verden blir mer og mer digital og kunnskapen som en behøver for å fungere i verden er i stor forandring. Det er viktigere enn noen gang at elever i skolen får opplæring i ferdighetene som kreves for å kunne løse fremtidens problemer. Utviklingen av AI, som Chat-GPT, viser hvor raskt verden forandrer seg, og for å håndtere dette kreves en rekke ulike ferdigheter, kunnskaper og tenkemåter. Blant disse er lesing, skriving og regning, som er grunnleggende ferdigheter. Men i dag møter vi på nye problemer og utfordringer som krever at barn og unge utvikler nye ferdigheter for å møte verden. Ferdigheter som kritisk tenkning og kildekritikk er blant ferdighetene som har fått betegnelsen “21st Century Skills”. Disse behøver ifølge (Rotherham & Willingham, 2010) ikke å være nye ferdigheter, men de er nødvendige å beherske i samfunnet i dag og i fremtiden.

Algoritmisk tenkning omtales som en «21st Century Skill», og har fått et økt fokus de siste årene. Begrepet som stammer fra arbeid gjort av Seymour Papert, og ble gitt betegnelsen «computational thinking» av (Wing, 2006). I dag er programmering mer relevant enn noen gang før. Bruk av datamaskiner og teknologi stadig mer fremtredende del av livene våre. (Weintrop et al., 2016) har videre konkretisert algoritmisk tenkning til en rekke praksiser som inngår i algoritmisk tenkning både innenfor og utenfor programmering som en del av realfagene. Med det økte fokuset på programmering i verden, har begreper som algoritmisk tenkning kommet frem i skolene over hele verden. Videre har programmer som Scratch og produkter som Microbit gjort programmeringen mer tilgjengelig for elever i skolen. Blokkbasert koding har senket terskelen og gjort at elever kan arbeide med programmering, som åpner for at flere enn noen gang tidligere kan ta i bruk programmering i livene sine. PRIMM som står for Predic-Run-Invesigate-Modify-Make er laget for å hjelpe lærere å lage undervisning knyttet til programmering. Etersom programmeringen har blitt mer fremhevet i undervisning, er det viktig å kunne se på hva man kan overføre av kunnskaper fra programmering inn i nye situasjoner, også utenfor programmeringen som man møter i matematikken og naturfaget. LK20 utdyper det som blir kalt for Forskerspiren, ved å legge en større vekt på det som kalles for naturvitenskaplige praksiser og tenkemåter. Det blir påpekt i naturfagets kjerneelementer at elever skal oppleve naturfaget som et praktisk og utforskende fag. Naturfagets praksiser og tenkemåter, som er unike for faget, rommer mange av praksisene som vi ser i undervisningen. Kjerneelementene innenfor naturvitenskapliges praksiser og tenkemåter legger opp til stor



grad utforskning samt at elevene skal ta i bruk det egne språket og begrepene som eksisterer i faget. Den utforskende siden av naturfag, som har en sterk tilknytning til det naturvitenskaplige og forskningen som faget er basert på, blir derfor vektlagt i sterk grad.

Begreper som «forskerspore», som skal lære elever å arbeide som forskere for å erfare hvordan kunnskap dannes og utvikles.

Ifølge LK20 skal elever lære å programmere i matematikkundervisningen, men også ta i bruk disse ferdighetene i naturfaget. Elevene vil derfor ikke ha sitt første møte med programmering i naturfaget, men anvende det i naturfagundervisningen. Dette kommer også fram i skolen og kompetansemålene i naturfag, hvor det står at elever etter 7. trinn skal kunne «utforske, lage og programmere teknologiske systemer som består av deler som virker sammen» (Utdanningsdirektoratet, 2020). Man kan ut fra dette se at programmering går igjen gjennom hele skoleløpet, og elever vil være avhengige av å beherske dette.

Med fremveksten av de naturvitenskaplige praksisene og tenkemåtene og utforskning som metode i skolen, samt hvor mye elever i dag skal ta i bruk og anvende programmering og programmeringsferdigheter, ønsker jeg å se nærmere på nettopp dette.

Algoritmisk tenking har også blitt lagt inn i læreplanen, og er nå underlagt den digitale kompetansen som skoler skal gi opplæring i. *Hensikten med denne masteroppgaven er derfor å få et innblikk i hvilken rolle algoritmisk tenkning har i PRIMM-basert naturfagundervisning, og hvordan det tilrettelegger for utforskende arbeid.* På denne måten kan man få en større forståelse for hvordan elever arbeider med programmering i naturfagsundervisning.

#### Forskningsspørsmål

For å få innblikk i oppgavens hensikt vil jeg ta i bruk følgende forskningsspørsmål:

- *Hvordan brukes algoritmisk tenkning i de ulike fasene av PRIMM*
- *Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tenkning*

Ved å svare på disse forskningsspørsmålene vil jeg se på om elever tar i bruk algoritmisk tenkning når de deltar i et undervisningsopplegg basert på PRIMM, og hvordan samtalen deres hvor de praktiserer algoritmisk tenkning er utforskende. Gjennom å se på elevenes utforskende språk hvor de tar i bruk en av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Dette vil la meg få et innblikk i hvordan elevene benytter argumentasjon og begrunnelser, som er sentralt innenfor utforskende undervisning i naturfag.

## MASCOT

Denne oppgaven er skrevet i samarbeid med MASCOT-prosjektet, som står for Mathematics, Science and Computational Thinking og blir presentert av (OsloMet, u.å.) som et tverrfaglig forskningsprosjekt som sikter mot å utvikle undervisning, læring og vurderingsprosesser. MASCOT er et samarbeid mellom universiteter i Norden. Denne oppgaven er skrevet i et samarbeid mellom MASCOT og meg, hvor jeg gjennom de har fått tilgang på datamateriale og veiledning.

## Teori

Jeg vil i denne delen av besvarelsen presentere teorien og forskningen som jeg har brukt til å skrive denne oppgaven. Jeg vil definere hva jeg legger i begrepet algoritmisk tenkning og utforskende arbeid. I tillegg vil jeg presentere annen relevant teori. Teoridelen av oppgaven er strukturert slik at den først vil presentere teori knyttet til oppgaven og oppgavens studie. Jeg vil så trekke frem tidligere forskning som har blitt gjennomført knyttet til utforskning, algoritmisk tenkning og programmering i skolen.

### Hva er algoritmisk tenkning?

Algoritmisk tenkning er et begrep som kan være vanskelig å definere. Det finnes ingen klare definisjoner, og ulike tolkninger har kommet frem gjennom årene. Begrepet algoritmisk tenkning kommer opprinnelig fra (Wing, 2006), som bruker det for å beskrive måten å løse problemer, lage systemer og forstå menneskelig atferd ved å ta i bruk konsepter fra datavitenskapen. Algoritmisk tenkning vil benytte en mengde ulike mentale ferdigheter som bygger på å dele opp større og vanskeligere problemer i mindre og mer overkommelige deler, som en så kan løse. Metoden representerer ferdigheter og en tankegang som er nyttig for alle å kunne ta i bruk, og som ikke er bundet til å kun gjelde problemene som man møter på innenfor programmering. (Wing, 2006) eksemplifiserer bruk av algoritmisk tenkning ved at når man «mister» noe, vil sjekke de stedene man sist har vært. Metoden der man systematisk går bakover for å løse et problem, illustrert i form av å finne en mistet gjenstand, er noe som kan ses i programmeringen.

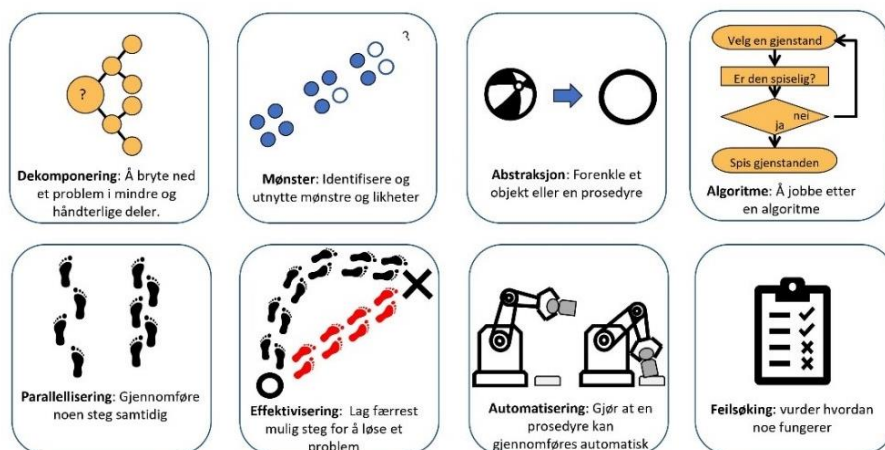
Algoritmisk tenkning blir av (Shute et al., 2017) beskrevet som en av de viktigste ferdighetene å besitte i dagens samfunn. De sammenligner algoritmisk tenkning med evnen til å lese og skrive. Tidligere var det kun enkelte i samfunnet som kunne lese og skrive, men når samfunnet endret seg, ble det nødvendig for hele befolkningen å beherske disse ferdighetene. På samme måte blir algoritmisk tenkning fremhevet som en ferdighet som med tiden vil være nødvendig for en større del av befolkningen. Dette støttes av (OsloMet, u.å.) sin definisjon av

algoritmisk tenkning, som omtaler det som en generisk ferdighet. Som en generisk ferdighet vil algoritmisk tenkning bidra til å styrke elevens evner til å tenke analytisk og utøve kritisk tenkning, kreativitet og problemløsning.

Algoritmisk tenkning er en egen tankegang som skiller seg fra andre tankeganger som systemtenkning, ved at den er basert på ferdigheter knyttet til programmering. Etersom dette er et relativt nytt begrep, fra 2006, har det blitt sett på hvordan det skiller seg fra andre tankeganger som har vært mer etablerte. (Shute et al., 2017) trekker frem at algoritmisk tenkning skiller seg fra andre tenkemåter ved at metodene som man tar i bruk og anvender, kan overføres og tilpasses andre oppgaver og problemer.

(Sengupta et al., 2013) påpeker at selv om algoritmisk tenkning tar utgangspunkt i de praksisene man lærer og tar i bruk i programmering, inkluderer det også andre ferdigheter i form av å kunne benytte problemfremstilling, abstraksjon, dekomposisjon, simuleringer og verifisering. Disse evnene er også essensielle innenfor vitenskap og matematikken, ved at de er grunnleggende for modellering, argumentering og problemløsning. De refererer til (Harel & Papert, 1991), som argumenterte for at å implementere programmering i andre fag vil være gunstig for både elevenes programmeringsferdigheter, men også for deres faglige utbytte.

Figur 1: MASCOT sine algoritmisk tenkning kort, inspirert av (Shute et al., 2017)



Ut fra (Shute et al., 2017) har MASCOT utviklet kort som i undervisning blir brukt for å illustrere algoritmisk tenkning for elever. Kortene fremstiller åtte prinsipper innenfor algoritmisk tenkning og hvordan de kan bli tatt i bruk i undervisningen. Dekomponering, mønstre, abstraksjon, algoritmer, parallellisering, effektivisering, automatisering og feilsøking er alle deler av algoritmisk tenkning som kan bli tatt i bruk i undervisning og i arbeid med og uten programmering.

(Weintrop et al., 2016) påpeker at det mangler en klar definisjon for algoritmisk tenking, spesielt innenfor fag som matematikk og naturfag. De poengterer videre at å bringe algoritmisk tenkning inn i matematikk og naturfaget ligger i de store forandringene som de fagene opplever. Dette ser man i at nesten alle delene av disse fagene har opparbeidet seg en digitalisert part som reflekterer den tidligere analoge delen av fagene. Her trekkes eksempler som bioinformatikk og datastatistikk frem.

De har laget et rammevært for algoritmisk tenkning praksiser i matematikk og naturfag. (Weintrop et al., 2016) fremhever fire kategorier av praksiser som sammen faller under algoritmisk tenking. Disse fire kategoriene inneholder ulike ferdigheter som kan brukes innenfor algoritmisk tenking i matematikk og naturfag. Rammeverket omhandler praksiser som blir tatt i bruk innenfor realfagene.

Disse kategoriene, og ferdighetene som ligger under dem, er flytende, og kan bevege seg inn i hverandre. De ulike kategoriene vil ikke bli brukt eksklusivt, og i praksis vil man bruke flere samtidig mens en arbeider for å løse et problem. (Weintrop et al., 2016) sin definisjon er mer rettet mot bruken av innenfor realfag. (Waters, 2020) har i sin masteroppgave oversatt (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Jeg vil i denne oppgaven ta i bruk disse oversatte praksisene.

Tabell 1: (Waters, 2020) sin oversettelse av (Weintrop et al., 2016) sine praksiser for  
 algoritmisk tenkning

<b>Datahåndtering</b>	<b>Modellering og simulering</b>	<b>Digital problemløsning</b>	<b>Systemtenkning</b>
Samle data	Bruke digitale modeller for å forstå et begrep	Forberede problemer til å løse dem digitalt	Undersøke et komplekst system som en helhet
Generere data	Bruke digitale modeller til å finne og teste løsninger	Programmere	Forstå sammenhenger innad i et system
Behandle data	Vurdere digitale modeller	Velge effektive beregningsverktøy	Tenke i nivåer
Analysere data	Utforme digitale modeller	Vurdere ulike tilnærminger/løsninger til et problem	Kommunisere informasjon om et system
Visualisere data	Implementere digitale modeller	Utvikle modulære digitale løsninger	Definere systemer og håndtere kompleksitet
		Lage digitale abstraksjoner	
		Feilsøking og feilretting	

(Utdanningsdirektoratet, 2019) definerer algoritmisk tenkning som en problemløsningsmetode, som handler om å gå systematisk fremover. Man vil vurdere hvilke trinn og handlinger som er nødvendige fremgangsmåten for å nå det resultatet en ønsker. Algoritmisk tenkning innebærer å bryte ned komplekse problemer til mindre delproblemer som man så kan løse. Dette krever å organisere og analysere informasjonen, for så å planlegge og gjennomføre en fremgangsmåte.

(Weintrop et al., 2016) sine praksiser er hentet fra hvordan algoritmisk tenkning brukes innenfor realfag. Dette gjør at disse praksisene er knyttet opp mot utforskningen som vi ser i naturfaget. Blant praksisene er det å samle data, analysere data og feilsøking og feilretting. Å anvende algoritmisk tenkning på en utforskende måte er noe som også (Palts & Pedaste, 2020) har fremhevet. Palts og Pedaste sine trinn innenfor algoritmisk tenkning viser likheter med praksisene man finner innenfor utforskende arbeid. De har undersøkt hvordan man kan tilrettelegge for algoritmisk tenkning i ulike fag. De påpeker at algoritmisk tenkning, som en tenkemåte, kan fremmes ved å anvende tenkemåten i tre trinn, når man møter på et problem som må løses.

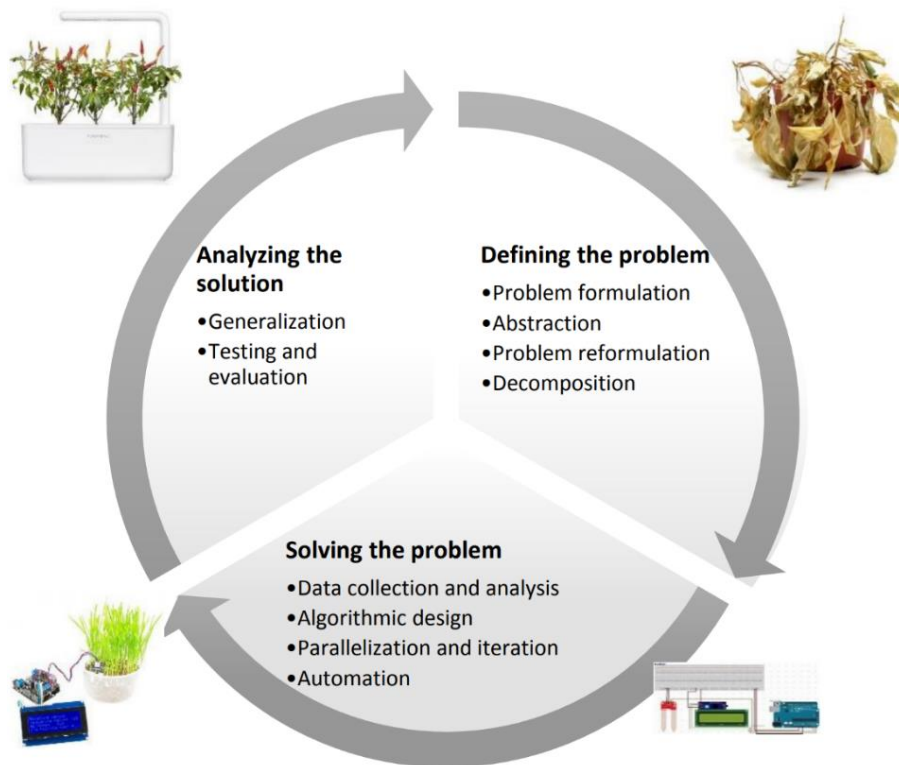
Det første trinnet er å formulere et spørsmål. Dette involverer alle algoritmiske tenkning-praksiser som blir tatt i bruk *før* man begynner å løse problemet. Innenfor det første trinnet blir også praksiser som abstraksjoner og dekomposisjon trukket frem. Det første trinnet omhandler det å avgrense hva som skal løses.

Det andre trinnet er å løse problemet. Stadiet involverer å ta i bruk alle praksisene som behøves, for å løse problemet som en møter på. I dette inngår i å samle og analysere data.

Dette trinnet er arbeidet som blir gjort underveis i problemløsningsprosessen.

Det siste trinnet er å analysere problemet. Dette involverer å bruke praksisen generalisering, hvor man tar med seg det man har lært fra prosessen, slik at man kan overføre kunnskapen til nye situasjoner.

Figur 2: (Palts & Pedaste, 2020) sine faser av algoritmisk tenkning



De tre stadiene av å anvende algoritmisk tenkning i problemløsning som blir presentert av (Palts & Pedaste, 2020), viser likheter med utforskningen og praksisene som man finner i naturfaget. Å arbeide utforskende bygger på å samle data og argumentere for sine synspunkter. Det siste trinnet som involverer generalisering, er også knyttet opp mot den siste fasen i PRIMM. Make-fasen, omhandler akkurat det å ta med seg det man har lært fra de tidligere fasene inn i nye situasjoner.

Ut fra disse definisjonene på algoritmisk tenkning, kan man konkretisere algoritmisk tenkning til en problemløsningsmetode som innebærer at man tar utgangspunkt i hovedtrekkene til programmering og koding for å løse ulike oppgaver. Algoritmisk tenkning er ikke bundet til kun å bli brukt innenfor arbeidsoppgaver som omhandler koding og programmering, men at man gjennom arbeidsmetoder som tidligere har blitt brukt i en programmeringssammenheng kan løse ulike oppgaver. Spesielt innenfor realfagene, som naturfaget ligger under, er algoritmisk tenkning en viktig egenskap hvor det vil bli et økende fokus på data, programmering og simulering i faget.

#### Algoritmisk tenkning i skolen

Ettersom algoritmisk tenkning har vokst frem, har det blitt et økt fokus på å inkludere det i skolen. Selv om algoritmisk tenkning kan brukes i flere fag, blir det hovedsakelig fokusert på i fagene innenfor matematikk og naturfaget, hvor programmeringen i skolen hovedsakelig foregår. Tidligere forskning har derfor sett på disse fagene når de undersøker hvorvidt algoritmisk tenkning kan brukes i skolen.

(Weintrop et al., 2016) argumenterer for at det er viktig å innarbeide algoritmisk tenkning i matematikk og naturfaget. De uttrykker at å inkludere algoritmisk tenkning i matematikk og naturfagene, vil gi tankegangen og ferdighetene den medbringer en autentisk følelse som ikke kan gjenskapes i et dedikert fag knyttet til algoritmisk tenkning. De påpeker også at ettersom fagene innenfor realfag er under så store endringer på grunn av de store fremskrittene innenfor data, vil det å beherske algoritmisk tenkning innenfor de ulike fagene være essensielt fremover.

#### Programmering i skolen

På 1980-tallet ble datamaskiner mer utbredt i samfunnet. Dette gjorde at en større andel av befolkningen hadde tilgang til datamaskiner. (Papert, 1980) begynte derfor å undersøke hvordan man kunne ta i bruk denne teknologien og programmering i skolen. Allerede da ble det begynt med bruk av programmering i skolen, hvor elevene skulle ta bruk programmet Turtle Talk, som lot de styre en mekanisk «skilpadde» for å lære de geometri. Turtle Talk lot barna skrive kommandoer som fikk den mekaniske «skilpadde», til å tegne en linje fra hvor den hadde gått basert på kommandoene den mottok. Målet med Turtle Talk var å lære barn om geometri ved å gi dem en konkret som de kunne styre. Undersøkelsens mål var å vise at programmering bidro til faglige ferdigheter som kunne bidra til læring på tvers av fag. (Papert, 1980) foreslo at bruken av programmeringsbaserte undervisningsformer som Turtle Talk, bidrar til økt heuristisk tenkning.

(Dolonen et al., 2019) legger frem forskjellen på tradisjonell og nyere måter å undervise programmering på. Tidligere har programmering i skolen krevd kjennskap til programmenes grammatikk, og har i stor grad vært styrt av lærere. Nyere arbeid som er inspirert av arbeid fra MIT (Massachusetts Institute of Technology) bygger på en undervisning som er mer ustrukturert og utforskende. Dette er noe som har ført til utviklingen av blokkbasert koding som Scratch. Forkjemperne for denne formen for koding begrunner dens relevans i at den bidrar til utvikling i generelle, ikke-faglige ferdigheter, som kreativitet og problemløsning. Blokkbasert koding er en av de nyeste formene for koding, og har blitt et populært i skolen. Scratch, som kan bli brukt på iPad-er og på PC-er, har en lav terskel, og tilbyr elever å lære programmering uten å måtte kunne programmeringsspråkets syntaks. (Brennan & Resnick, 2012) bruker Scratch som et utgangspunkt til å forklare hvordan barn gjennom arbeid med blokkbasert koding, møter en rekke med grunnleggende konsepter innenfor programmering. Elever kan gjennom programmer som Scratch, designe og lage sine egne produkter ved å anvende programmering til å arbeide kreativt. Det blir trukket frem eksempler på barn som helt ned i åtteårsalderen, designer spill, musikkvideoer og simuleringer ved hjelp av blokkbasert programmering. (Brennan & Resnick, 2012) argumenter for at Scratch bidrar til algoritmisk tenking ved at det gir en kontekst og aktiviteter hvor eleven kan anvende kunnskapene sine om programmering.

Gjennom å arbeide med å kode og programmere vil elevene over tid lære flere av de grunnleggende konseptene som finnes innenfor koding, slik som sekvenser, løkker, hendelser, paralleller, betingelser, operatører og data. (Brennan & Resnick, 2012) kommer med seks hovedpunkter som skal hjelpe med å vurdere algoritmisk tenking:

1. støtte videre læring
2. inkludere artefakter
3. vise prosesser
4. sjekke inn
5. verdsett flere metoder
6. inkludere flere syn

#### PRIMM

(Utdanningsdirektoratet, 2020) presenterer i LK20 at naturfaget og matematikken skal gi elever grunnleggende digitale ferdigheter, og påpeker at fagene skal inneholde bruk av programmering. Bruk av programmering og koding i undervisningen, og tenkemåtene som følger med dette, kan være nytt for både lærere og elever. (Lee et al., 2011) utarbeidet derfor Use-Modify-Create (UMC) som et rammeverk. (Lund, 2022) sier at UMC og PRIMM referer



til undervisningspraksiser hvor en tar en eksisterende kode og modifierer den til sin egen. Målet er å lære elever å lese kode, for så å modifisere den selv og lage et eget produkt. Gjennom UMC vil en som lærer å lese og skrive, kode. På lik linje som en som lærer å lese og skrive, vil elevene ta eksisterende tekst og bruke det til å lære å skrive selv. (Lund, 2022) bruker analogien at før man kan bygge en benk, vil man være nødt til å prøve ut og bruke en annen benk for å få en forståelse for hvordan en benk fungerer. (Futurelearn, u.å.) forklarer at PRIMM står for Predict-Run-Investigate-Modify-Make og er en fremgangsmåte som blir brukt i kode- og programmeringsundervisning. Den er en videreutvikling av den tidligere kjente UMC-modellen. Denne modellen ble utviklet for å bidra og hjelpe elever når de i opplæringen skulle arbeide med tekstbasert koding, men prinsippene i PRIMM kan også overføres til blokkbasert og visuell koding, som Scratch og Makecode.

(Futurelearn, u.å.) presenterer de fem fasene av PRIMM:

- Predict-fasen innebærer at elevene vil bli presentert med en kode som de skal tolke hva gjør. I denne fasen vil ikke koden bli kjørt, men de skal selv tolke og komme med sine teorier om hva de tror koden gjør når man kjører den.
- Run-fasen omhandler at elever vil kunne teste ut teoriene sine om koden ved å kjøre den og se hva den gjør.
- Investigate-fasen innebærer at elevene vil ta et «dypdykk» og se på hvordan koden fungerer.
- Modify-fasen av PRIMM er lik den som man finner i UMC, ved at de skal modifisere koden som de har blitt gitt. Elever vil i modiy-fasen av PRIMM endre på koden som de har blitt gitt og lage sin egen.
- Make-fasen i PRIMM innebærer at elever tar i bruk det som de har erfart og lært i de tidligere fasene, og bruker det til å løse et nytt problem. Make-fasen er derfor ikke en konkret slutt, men overgangen til en ny oppgave.

### Begrepsbruk i naturfag

Et av de mest unike kjennetegnene ved naturfag er dets mangfold av begreper. Elever vil gjennom undervisningen lære en rekke begreper og ord som har en egen betydning innad i faget. (Haug, 2016) fremhever det de kaller for nøkkelbegreper i naturfag. Nøkkelbegreper er begreper som er essensielle å beherske for å kunne praktisere naturfag. Nøkkelbegreper er delt inn i fagspesifikke begreper og forskerspirebegreper. Fagspesifikke begreper er nøkkelbegrepene som er knyttet til temaer innad i naturfaget. Disse begrepene er de som er nødvendig å beherske innenfor et tema, men som ikke overføres til andre temaer i faget.

Eksempler på fagspesifikke nøkkelbegreper innenfor temaet fenomener og stoffer er kjemiske reaksjoner, atomer og molekyler.

Forskerspirebegreper er begreper som er universelle og gjelder hele naturfaget.

Forskerspirebegrepene er lagt opp til å lære barn om hvordan forskere jobber når de skal finne svaret på det de undersøker. Begrepene som ligger under Forskerspiren, er lagt opp for å vise hvordan kunnskapen som vi kjenner til har blitt til. Eksempler på forskerspirebegreper er observasjon, bevis, data, påstand, forklaring, eksperiment og hypotese. Disse begrepene er tett knyttet opp mot arbeidsmåtene som man ser i forskning.

Ifølge (Haug og Mork, 2021) krever utviklingen av elevers begrepsforståelse tid.

Begrepsforståelsen er en kontinuerlig prosess, hvor elevenes utvikling gradvis bygger på å knytte flere assosiasjoner til begrepet. Ulike aktiviteter, både praktiske, teoretiske og utforskende, vil sammen kunne danne et grunnlag for elevenes begrepsvokabular i naturfaget. De refererer til (Osborne, 2014) som bruker analogien at på lik linje som å lære å spille et instrument eller utøve en sport, vil det å ta i bruk og praktisere bruken av begreper og praksisene knyttet til naturfaget være den beste måten å lære de på.

#### Dybdeløring i naturfag

Dybdeløring er, på lik linje med algoritmisk tenkning, et begrep som har blitt mer fremtredende de siste årene. Dette er den gradvise utviklingen og forståelsen av hvordan ulike begreper og fenomener henger sammen på tvers av fagområder. (Voll & Holt, 2019) påpeker at på samme måte som begrepslære, innebærer dybdeløring en organisering av kunnskap. De påpeker også at det foregår dybdeløring ved at elevene bygger mentale kunnskapsmodeller, sammensatt i hierarkiske strukturer. (Voll & Holt, 2019) påpeker at dybdeløring er en tidkrevende prosess, som krever en rekke varierte arbeidsformer. Det blir også spesifisert at dybdeløringen er en prosess og ikke et mål. En elev vil kontinuerlig oppleve dybdeløring gjennom oppbygningen av forståelse.

Løring er tett knyttet opp mot utforskende undervisning, hvor innhenting av data og argumentasjon er sentralt. (Øyehaug, 2019) fremhever at løring skjer i et sosialt fellesskap. Det å sette ord på og formulere sine synspunkter og forståelse lar en reflektere over sine egne kunnskaper på en helt annen måte. Dette kan bli knyttet opp mot hva (Mork, 2016) sier om utforskende arbeid. Hun fremmer at diskusjonen som foregår mellom elever er en del av utforskningen som kan bli opplevd i klasserommet.

(Waters, 2020) referer i sin oppgave til en undersøkelse av (Taub et al., 2015), hvor de har undersøkt hvordan programmering kan bidra til løring. De omtaler kunnskapsintegrasjon (KI) som referer til de ulike fasene som elever gjennomgår når de lærer. KI-prosessen viser

hvordan kunnskaper utvikles i fire faser. De ulike fasene har ingen kronologisk rekkefølge, men kommer naturlig frem i ulike deler av elevenes kunnskapsutvikling. Elevene vil derfor kunne ta i bruk de ulike prosessene i forskjellige sammenhenger for å kunne fremme sin egen forståelse.

De fire fasene er å elisitere ideer, å legge til nye ideer, å utvikle kriterier for å evaluere ideer og å sortere og reflektere. I likhet med hva (Voll & Holt, 2019) fremmer om dybdelæring, vil elever i det første trinnet i KI-prosessen bli gjort oppmerksom på sine forkunnskaper.

Forkunnskaper blir trukket frem, slik at elever kan få laget koblinger mellom den nye kunnskapen og sine tidligere kunnskaper. Det andre trinnet omhandler å legge til nye ideer. De nye ideene er den nye kunnskapen som elevene skal bearbeide og forstå, og kan komme fra lærer, medelev eller undervisningen. Det tredje trinnet som omhandler det å innhente og vurdere informasjon, ved å stille spørsmål og gjennomføre tester. Dette trinnet viser en likhet med utforskende undervisning i naturfag, særlig metodene som vi møter på i Forskerspiren, hvor elevene skal arbeide som forskere for å få svar på sine spørsmål. De vil i dette trinnet vurdere om den nye informasjonen som de har fått fra det andre trinnet er gyldig eller ugyldig. I det fjerde trinnet skal elevene sortere og reflektere kunnskapen som de har tilegnet seg og vurdert i de tidligere trinnene. Elever vil i dette trinnet lage hierarkiene av kunnskap som (Voll & Holt, 2019) omtaler som i dybdelæring.

Tabell 2: (Waters, 2020) sin oversettelse av (Taub et al., 2015) faser av kunnskapsintegrering

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

## Pedagogisk sammenhengsskaping

Knyttet til begrepsbygging og dybdel ring, er det (Scott et al., 2011) kaller for pedagogical link-making. De presenterer det som at en elev tar i bruk og anvender et begrep og knytter b nd til eksisterende kunnskaper for   f  en forst else for noe nytt. Dette fremmes av b de (Voll & Holt, 2019), og (Taub et al., 2015). Jeg har i denne oppgaven oversatt begrepet, og vil heretter referere til det som pedagogisk sammenhengsskaping. Jeg vil ogs  oversette underbegrepene som er knyttet til pedagogisk sammenhengsskaping. Jeg vil presentere to av formene for pedagogisk sammenhengsskaping som er knyttet opp mot dybdel ring.

Den f rste formen for pedagogisk sammenhengsskaping referer til   bygge kunnskap ved   trekke sammenhenger mellom elevenes hverdagslige forst else og den faglige forst elsen som de m ter i skolen. Dette kalles for support *knowledge building*. Denne formen har jeg oversatt til st tte-kunnskapsbygging. Denne formen for pedagogisk sammenhengsskaping omhandler sammenhengene man kan m te p  innenfor naturfaget og mellom naturfaget og den hverdagslige verden. Dette henger sammen med dybdel ringen, hvor det   trekke frem elevenes forkunnskaper er sentralt. Gjennom sammenhengsskaping hvor elevenes hverdagslige kunnskaper blir trukket frem slik at de kan se sammenhenger mellom ulike kunnskaper, vil det bli dannet grunnlag for dybdel ring i undervisning.

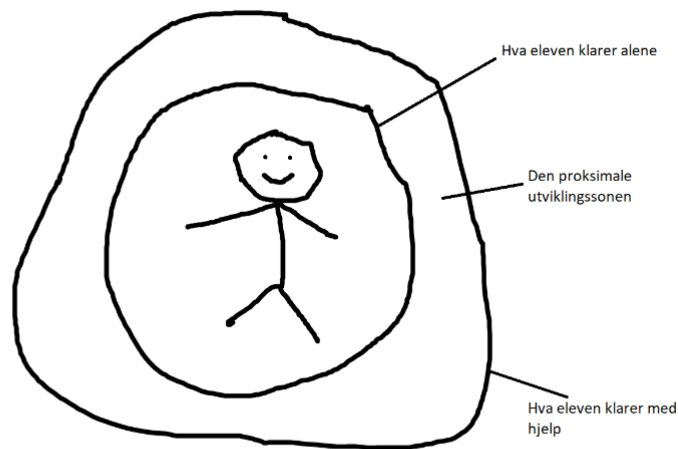
Den andre formen for pedagogisk sammenhengsskaping blir av (Scott et al., 2011) kalt *to promote continuity*. Jeg har i denne oppgaven oversatt dette til fremme kontinuitet. Denne formen for pedagogisk sammenhengsskaping omhandler det   referere tilbake til tidligere  kter, hvor en l rer eller elev kan ha kommet med et punkt i undervisningen. Dette vil igjen v re knyttet opp mot dybdel ringen som blir fremmet av (Voll & Holt, 2019), som p peker at dybdel ring er noe som tar tid og krever en rekke ulike tiln rmingsm ter.

## Stillasbygging

Begrepet stillasbygging (p  engelsk *scaffolding*) stammer fra Vygotsky, som kom med teorien i 1978 i boken «Mind and Society». (Maybin et al., 1992) p peker at stillasbygging er den st ttende hjelpen som eleven kan f  fra en annen person, og er ikke begrenset til   kun v re medelever. Elever som opplever stillasbygging vil kunne f  dette fra ulike personer, og det kan komme fra en medelev, l rer eller annen voksenperson. De p peker at stillasbygging kommer fra Bruner, som refererer til Vygotsky sin teori om den proksimale utviklingssonen. Stillasbygging tar i bruk det som kalles for den proksimale utviklingszone, ogs  hva eleven kan og ikke kan klare p  egen h nd.

(Gibbons, 2015) forklarer at den proksimale utviklingssonen er en grense for hva elever kan klare alene og hva som de kan klare med hjelp. Stillasbygging omhandler strukturene som blir bygget rundt en, for å kunne hjelpe en å nå gjennom sin proksimale utviklingszone. (Maybin et al., 1992) fremmer at stillasbygging ofte kommer opp rundt temaene muntlig språk. Ettersom stillasbygging i stor grad foregår mellom to personer hvor en er faglig sterkere enn den andre, enten det er en voksenperson eller medelev. Ettersom naturfaget er et muntlig fag, med et stort

*Figur 3: Den proksimale utviklingssonen*



Ovenfor illustreres det hvordan elever i den proksimale utviklingssonen har en grense for hva de klarer alene. Gjennom stillasbygging vil eleven med assistanse oppnå en høyere forståelse. Den proksimale utviklingssonen illustrerer skillene mellom hva en elev klarer alene, hva den samme eleven klarer med hjelp og hva den ikke klarer med hjelp. (Nyutstumo, 2021) skriver i sin masteroppgave om programmering og problemløsning i matematikk. I oppgaven påpeker han hvordan programmering, algoritmisk tenkning og PRIMM er knyttet opp mot stillasbygging.

#### Studier av PRIMM i skolen

Kodingen som utføres av uerfarne, bærer ofte en prøve-og-feil-mentalitet, som kan gjøre oppgaven slitsom og frustrerende for nybegynnere. PRIMM løser dette ved å la elever bruke en annen persons kode og modifisere den til sin egen. Ettersom PRIMM har blitt mer fremtredende i skolen, har det blitt gjort en rekke undersøkelser på hvordan det påvirker elevers læring. (Sentence & Waite, 2019) gjennomførte en studie med lærere med og uten erfaring med koding og programmering, på 6.–8. trinn. Studien ble gjennomført ved at 15

lærere med erfaring og 7 lærere med spesifikk erfaring innenfor programmering og koding gjennomførte et forhåndsplanlagt undervisningsopplegg knyttet til PRIMM. I etterkant ble deres egne oppfatninger målt. Resultatet av undersøkelsen var at 47% av lærerne som gjennomførte prosjektet ga tilbakemelding om at bruk av PRIMM var veldig nyttig, mens ytterligere 47% ga tilbakemelding om at det var nyttig. 7% av deltakerne lærere ga dog en tilbakemelding om at det var litt nyttig.

#### Bidrar PRIMM til algoritmisk tenkning

PRIMM tar også i bruk algoritmisk tenking ifølge (Lytle et al., 2019). De forklarer at algoritmisk tenking, som er en «21st Century Skill», er viktig for elevene å beherske. Det er derfor nødvendig å implementere det i fag som matematikken og naturfaget, og ikke kun i IT og data-relaterte fag. Å inkludere algoritmisk tenkning i matematikk og naturfagundervisningen byr dog på utfordringer, ettersom både lærere og elever kan ha manglende kunnskaper om algoritmisk tenking og bruk av programmering i skolen. (Lytle et al., 2019) gjennomførte en undersøkelse hvor de sammenlignet to ulike skoler i USA, hvor det ble gjennomført ett blokkbasert kodeopplegg og ett med UMC, for å få ett innblikk i hvordan naturfag og matematikk blir påvirket av å innarbeide algoritmisk tenking i undervisningen. Studien ble gjennomført ved at de fulgte de ulike skolenes 6.-klasser over fire dager.

Lærerne fra de ulike skolene kom med forskjellige tilbakemeldinger i etterkant av prosjektet. Lærerne fra skole 1, som gjennomgikk undervisningen med et UMC-opplegg, responderte med at de oppfattet at bruken av UMC bidro til å gi elevene stillasbygging i undervisningen. Lærerne til skole 2, som ikke brukte UMC, ga i tilbakemelding i etterkant av undersøkelsen at de oppfattet at det var et manglende element i undervisningsoppleggene. I tillegg ville elevene på skole 2 ha større utfordringer med å fullføre undervisningsoppleggene enn elevene ved skole 1. Dette ble begrunnet i at de måtte lage kodene sine fra bunnen av, og dermed brukte mer tid.

(Lytle et al., 2019) argumenterer for at det er positive sider ved å gjennomføre et opplegg hvor man tar i bruk UMC og hvor man programmerer fra bunnen av. Elever som lager sin egen kode fremfor å modifisere en tidligere en, vil kunne føle et større eierskap til det endelige produktet – selv om progresjonen kan oppfattes som vanskeligere for elevene som fra bunnen av lager sin egen kode. På den andre siden vil et undervisningsopplegg som er basert på UMC framfor å designe og lage egen kode, spare elevene tid på det grunnleggende i undervisningen, og heller kan bruke tid på delen å delta i lærerledede faglige diskusjoner.

I skolen brukes det for det meste blokkbasert koding. Programmerer slik som Scratch, er enkle å forstå og ta i bruk. Bruken av blokkbasert koding i undervisning som tilrettelegger for utvikling av algoritmisk tenking er noe som (Andersen, R., 2019) studerte. I sin case-studie, hvor hun undersøkt hvordan blokkbasert koding kan bli brukt i matematikkundervisningen og hvordan dette påvirker algoritmisk tenking. Hun ser inn i hvordan bruk av blokkbasert koding kan tilrettelegge for *Computer-supported collaborative learning* (CSCL). Andersen fant at å inkludere blokkbasert koding i matematikkundervisningen åpnet for samarbeid mellom elevene. I undervisningen vokste det frem en økt bruk av deling av kunnskap, konstruksjon av definisjoner.

Elevene ble gjennom bruk av blokkbasert programmering mer aktive i samarbeid med medelever, hvor de diskuterte faglig og hvor de samarbeidet om å finne løsninger på hvordan matematikken kunne brukes i programmene deres. Elevene fikk også tilegnet seg et mer programmeringsorientert språk. I samtalene mellom elevene, ble begreper som «loops» og «variables» framtrædende. Samtalene i gruppene flyter mellom matematikk og programmering i samsvar med (Harel & Papert, 1991) om at å inkludere programmering i skolefag vil løfte både elevens ferdigheter både innenfor fag og programmering.

Ettersom algoritmisk tenkning blir omtalt som en «21st Century Skill», har det blitt sett på hvordan det kan legges opp til å undervise og bygge kompetanse i den på skolen. (Yadav et al., 2011) refererer NRC (National Research Council), som omtaler algoritmisk tenkning som en ferdighet noen kan tilegne seg, og at dette er noe læreren og undervisningsopplegg kan tilrettelegge for at elever skal ta i bruk. For å undersøke nærmere på hvordan moderne opplegg kan tilrettelegge for elevene, kan vi se på en litteraturgjennomgang av (Dolonen et al., 2019) I oppgaven deres sammenligner de den tradisjonelle programmeringsundervisningen og den nyere blokkbaserte undervisningen. Endringene som har kommet siden 80-tallet, har gjort at det igjen blir nødvendig å vurdere hvordan programmering er relevant i skolen. (Dolonen et al., 2019) påpeker at å kalle tekstbasert for strukturert og blokkbasert for ustrukturert er en forenkling. De presenterer at gjennom tidligere forsøk gjort av TACCLE3 og senere av (Vivian, 2015) har det vært lagt stor vekt på programmerings-språkene, mens det programmeringsdidaktiske har vært mangelfullt. Videre refereres det til arbeid gjort av (Waite, 2018), som i en litteraturgjennomgang av programmeringspedagogikk ikke kunne komme frem til et klart svar på hva som var den mest effektive pedagogikken innenfor emnet programmering.

Tidligere empirisk forskning på algoritmisk tenkning i skolen

En studie av algoritmisk tenkning i skolen ble gjennomført av (Sengupta et al., 2013). I studiet undersøkte de hvordan algoritmisk tenkning henger sammen med realfagene. I undersøkelsen fant de en sammenheng mellom bruk av stillasbygging og læringsutbyttet til elevene. Elevgruppen som i undersøkelsen blir omtalt som gruppe S, hadde en økning i testresultatene i både fysikk og biologi i etterkant av å ha lært om algoritmisk tenkning. Gruppe C, som ikke hadde fått én-til-én stillasbygging, viste kun forbedring innenfor biologi.

Utforskende arbeid i naturfag

Det er ingen klar definisjon på hva utforskende undervisning i naturfag er. Gjennom tiden har det blitt fremmet flere ulike syn på hva som gjør undervisning utforskende. (Ødegaard, 2016) omtaler Forskerspiren, som ble introdusert i 2006. Forskerspiren var et svar på ønsker fra lærere om å få et rammeverk på hvordan man kunne legge opp undervisningen i naturfag, til å inkludere utforskningen som ble fremmet i LK06. Forskerspiren legger opp til utforskende aktiviteter i naturfag, hvor elevene får ta rollen som forskere i undervisningen. Den utforskende undervisningen som Forskerspiren la opp til, ble oppfattet som mer autentisk, og bidro til at elevene fikk en dypere forståelse for hvordan den naturfaglige kunnskapen utvikles.

Ettersom kjerneelementene i naturfag i LK20 også legger stor vekt på at elever skal arbeide utforskende og oppleve undring, er det like relevant som i 2006 å kunne tilrettelegge for dette i undervisning. (Minner et al., 2010) kommer med fem punkter som må oppfylles for at undervisning skal være utforskende undervisning innenfor naturfag.

- De lærende arbeider med et naturfaglig spørsmål.
- De lærende fokuserer på vitenskapelige forklaringer som lar de evaluere naturfaglige spørsmål.
- De lærende formulerer svar som begrunnes i vitenskapelige undersøkelser.
- De lærende må vurdere sine egne forklaringer ved å reflektere over andres forklaringer.
- De lærende kommuniserer og argumenterer for sine oppfatninger.

Utforskende arbeid er en av de mest fremtredende formene for undervisning som faller under definisjonen naturfaglige praksiser og tenkemåter. De naturfaglige praksiser og tenkemåtene rommer de ulike arbeidsmåtene som man møter på innenfor naturfaget. Utforskende undervisning ligger under de naturfaglige praksisene og tenkemåtene, og er et av de mest framtrede eksemplene i LK20. Under naturfagets kjerneelementer skal elevene i naturfag



oppleve det som et praktisk og utforskende fag, de skal gjennom praktisk arbeid få oppleve undring og erfare og forstå verden rundt seg. (Utdanningsdirektoratet, 2020). Allikevel blir begrepet naturfaglige praksiser og tenkemåter

Naturfaget har en helt spesiell og egen tenkemåte. Å snakke og tenke naturfag henger sammen. Når elever i grupper samarbeider om å løse et problem, vil de gjennom et naturfaglig språk kunne opparbeide formulere sine argumenter.

(Haug et al., 2021) definerer naturfaglige praksiser og tenkemåter innenfor

- Formulere spørsmål som kan undersøkes
- Samle og bearbeide data
- Lage forklaringer
- Bruke og lage modeller
- Utføre informasjonssøk og kildekritikk
- Argumentere
- Gjøre etiske vurderinger
- Formidle

Flere av punktene som (Haug et al., 2021) kommer med, overlapper med dem som blir trukket fram når man snakker om utforskende undervisning. De argumenterer for at naturfaglige praksiser og tenkemåter omhandler all naturfagundervisning. Selv om flere av punktene for utforskende undervisning og naturfaglige praksiser og tenkemåter overlapper, kommer (Haug et al., 2021) med punkter som å utføre informasjonssøk og kildekritikk, og gjøre etiske vurderinger. Disse punktene er blant dem som skiller utforskende undervisning fra de naturfaglige praksisene og tenkemåtene.

Ut fra disse definisjonene kan man betegne utforskende arbeidsmåter i naturfag som klasseroms-aktiviteter som tilrettelegger for naturfaglige praksiser ved å ta i bruk argumentasjon som begrunnes i observasjoner og undersøkelser. Elever vil i utforskende undervisning begrunne sine teorier i data, som de samler inn, og argumentere ut fra disse dataene. Elever vil i utforskende undervisning ha mye muntlig aktivitet som foregår mellom seg.

#### Tidligere forskning

(Abdi, 2014) gjennomførte en undersøkelse av hvilken effekt 5E-modellen og utforskende arbeid har for elever i skolen. Abdi gjennomført to runder med undervisningsopplegg på to 5.-klasser i Iran, med 20 elever i hver av klassene. Undersøkelsen gikk ut på at klassene ble gitt en for-prøve og en etter-prøve. Gruppe 1 ble i tillegg gitt et undervisningsopplegg med 5E-modellen mellom pre-testen og etter-testen. Undersøkelsen viste at gruppe 1, som

gjennomgikk et opplegg med 5E-utforskende undervisning, fikk bedre resultater på sluttprøven enn gruppe 2, som gjennomgikk et standard undervisningsløp mellom prøvene. Tidligere undersøkelser viser også lignende resultater. (Cardak et al., 2007), som undersøkte hvordan 5E-modellen påvirker elever i 6.-klasse. De gjennomførte to fire uker lange undervisningsopplegg med to elevgrupper, totalt 38 elever. Elevene i denne undersøkelsen undergikk også en prøve før og etter undervisningsoppleggene, hvor gruppe 1 gjennomgikk læring med 5E, mens gruppe 2 hadde tradisjonell undervisning. Resultatet av undersøkelsen viste til at det ble en signifikant økning i prøveresultatene til elevene som mottok 5E-læringsopplegg, i forhold til elevene som mottok tradisjonell undervisning.

## Metode

I denne delen av oppgaven vil jeg beskrive metoden jeg har tatt i bruk for å innhente data. Dette har blitt gjort i samarbeid med MASCOT-prosjektet, hvor jeg har deltatt i en datainnsamling på en skole. Jeg vil videre i denne delen av oppgaven beskrive hvordan jeg har analysert dataene som jeg tar i bruk for å få svar på oppgavens forskningsspørsmål.

- Hvordan kan PRIMM bidra til algoritmisk tenking
- Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tenkning

## Kvalitativ metode

For å besvare forskningsspørsmålene mine om hvordan PRIMM bidrar til algoritmisk tenkning og hvordan utforskende arbeid kommer til uttrykk i algoritmisk tenkning, vil jeg ta i bruk en kvalitativ metode. Jeg vil analysere elevenes samtaler ut fra videoopptak som jeg har fått tilgang til gjennom mitt samarbeid med MASCOT. (Krumsvik, 2015) sier at kvalitativ forskning omhandler å gå dypere inn i enkelttilfeller, fremfor å se på en bredere populasjon. Eksempler på kvalitative metoder er intervju og analyse av video. Han trekker frem Flyhavarikommisjonen som et eksempel. De vil når et fly krasjer, gå dypt inn i all dataen som de kan få for å finne ut akkurat hva som ledet frem til ulykken.

## Utvalg

Skolen og elevene som deltar i studien, er deltakere i MASCOT-prosjektet og er derfor ikke tilfeldig utvalgt. Skolen vil kunne bli beskrevet som en «gjennomsnittlig skole». Den ligger ikke i et sterkt sosioøkonomisk område, men heller ikke i et lavere sosioøkonomisk område. Klassen som har blitt fulgt gjennom undervisningsløpet kan bli beskrevet som «en normal norsk 6.-trinnsklasse».

## Datainnsamling

I mitt samarbeid med MASCOT-prosjektet har jeg deltatt i en datainnsamling på en skole utenfor Oslo. Datamaterialet som jeg har tatt i bruk, er videoer som er tatt fra elev- og klasseromsperspektiv, fra en klasse som har gjennomført et undervisningsopplegg i naturfag, knyttet til algoritmisk tenkning og programmering over en lengre periode. Elevene har i denne perioden jobbet med elektrisitet og energi, og har over åtte uker brukt ulike metoder og arbeidsmåter. Målet med opplegget er at elevene i slutten av perioden skal designe, bygge og programmere et smarthus. Ved å delta på datainnsamlingen har jeg fått en kjennskap til dataen og omstendighetene rundt utsagnene som læreren og elevene kommer med. Dette gir meg en unik mulighet til bedre å forstå elevenes utsagn og situasjonene som kommer frem seg ut i dataene.

Elevene har i undervisningen hatt på seg kameraer som fanger opp deres arbeid og samtaler mens de jobbet, slik at jeg får sett hvordan de har arbeidet med oppgavene sine, samt samtalene de har hatt med sine læringspartnere og andre rundt. Det har også blitt tatt opp video av hele klasserommet, slik at man får med de felles diskusjonene før og etter aktivitetene. Opptakene som jeg tar i bruk, er hentet fra tredje, fjerde og femte økt, hvor elevene jobber med å finne ut hva som leder elektrisitet, og hvordan man lager elektriske kretser.

## Undervisningsforløp

Undervisningsopplegget som jeg har fulgt er et åtte uker langt opplegg, knyttet til temaet elektrisitet. Elevene har i denne perioden lært om hvor strømmen som vi omgir oss med kommer fra, og hvordan vi bruker den. Klassen har hatt en lengre naturfagsøkt hver uke gjennom undervisningsperioden. I denne oppgaven fokuserer jeg på tredje, fjerde og femte økt, hvor det blir tatt i bruk PRIMM og der elevene skal lage sine egne elektriske kretser ved hjelp av en microbit.

*Tabell 3: Oversikt over øktene som har blitt transkribert i denne oppgaven*

Økt	Tema	Opplegg
3.	Elektrisk krets	<ul style="list-style-type: none"><li>• Elektriske ladninger som strømmer</li><li>• Hva er en elektrisk krets? Hva skjer hvis vi bryter kretsen?</li><li>• Aktivitet: Få en lyspære til å lyse. Bruke batteri, ledninger og små lyspærer.</li><li>• Aktivitet: Videreføre til dioder og microbit.</li></ul>

4.	Hvilke materialer leder strøm?	<ul style="list-style-type: none"> <li>• Utforske hvilke materialer som bruker strøm. Bruke «Jostein Volt» først, deretter skal elevene lage sin egen måler med microbit.</li> <li>• Lage hypoteser om hvilke stoffer som leder strøm. Teste ut hypotesene.</li> <li>• Koble dette til brytere. Hva vil en bryter si? Hvorfor bruker vi brytere?</li> <li>• Hvis tid: <a href="https://www.nysgjerrigper.no/eksperimenter/sitronbatteri/">https://www.nysgjerrigper.no/eksperimenter/sitronbatteri/</a> (Ønske fra elevene)</li> </ul>
5.	Sensorer microbit	<ul style="list-style-type: none"> <li>• Utforske sensorene på microbit</li> <li>• Begynne planlegging av smarthus. Gå gjennom kravspesifikasjoner, kriterier og vurdering</li> </ul>

### Transkribering

Datamaterialet som har blitt samlet inn, har blitt lagret i henhold til NSDs retningslinjer på en server på UiO. Denne serveren har jeg gjennom TSD tilgang til, via en virtuell desktop. All data, inkludert elevfilmer, har på forhånd blitt gitt anonyme navn og koder, slik at det ikke er mulig å finne personidentifikasjon. Elever som har blitt filmet, har blitt gitt navn etter rekkefølgen på alfabetet. Elevene som blir fulgt med et GoPro kamera, blir i denne oppgaven blitt gitt de fiktive navnene Anna, Benedicte, Ciri og Dina. Ytterligere har jeg også gitt navn til alle læringspartnere og medelever som kommer frem i elevenes og klassens samtaler og i videoen.

Transkripsjonen av datamaterialet foregikk ved at jeg først gjennomgikk en grovtranskribering, hvor jeg gjennomgikk datamaterialet i åtte-minutters segmenter. Innenfor disse segmentene ville jeg notere meg hva jeg så, og når. Dette lot meg få en oversikt over hva som kom opp i undervisningen. Jeg gikk deretter igjennom segmentene hvor elever hadde anvendt (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, og transkriberte deretter samtalen deres.

### Analyse av datamaterialet

#### *Hvordan kan PRIMM bidra til algoritmisk tenkning*

For å besvare forskningsspørsmålet *Hvordan kan PRIMM bidra til algoritmisk tenkning*, har jeg tatt i bruk (Weintrop et al., 2016) sine kategorier knyttet til algoritmisk tenkning i

matematikk og naturfag. Ved å gjøre dette får jeg et overblikk over ulike praksiser som inngår i de ulike fasene av PRIMM. Jeg har sett igjennom datamaterialet og funnet hvor elevene tar i bruk en eller flere av praksisene innenfor digital problemløsnings praksiser og systemtenkning. Jeg har valgt disse to kategoriene ettersom de er tett knyttet til å arbeide med og forstå kode og programmering.

*Tabell 4: (Weintrop et al., 2016) digital problemløsnings praksiser*

Forbrede problemer til å løse dem digitalt	Formulere et problem eller oppgave slik at den kan løses ved hjelp av et databasert hjelpemiddel
Programmere	Elevene tar i bruk programmering for å løse en oppgave. Dette inkluderer modifisering av en eksisterende kode som de blir delt ut.
Velge effektive beregningsverktøy	Vurdere ulike verktøys styrker og svakheter for å løse et problem
Vurdere ulike tilnærminger/løsninger til et problem	
Utvikle modulære digitale løsninger	Lage et verktøy som kan brukes til å løse et eller flere mindre problemer
Lage digitale abstraksjoner	Elevene visualiserer dataen som de besitter
Feilsøking og feilretting	Elevene søker etter

*Tabell 5: (Weintrop et al., 2016) systemtenkning*

Undersøke et komplekst system som en helhet	
Forstå sammenhenger innad i et system	Elevene viser forståelse for hvordan ulike deler henger sammen i et system som helhet, og hvilken rolle de ulike delene har.
Tenke i nivåer	Elevene vurderer de ulike nivåene som finnes innenfor et system. Dette går fra de større delene som finnes til de mindre delene.
Kommunisere informasjon om et system	Elevene kan kommunisere informasjon om systemets deler eller helet sin funksjon.
Definere systemer og håndtere kompleksitet	Elever definerer systemets rammer og begrensninger.

### Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tekning

For å besvare forskningsspørsmål 2, som er *Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tenking*, har jeg valgt å se på elevenes språk. Dette gjør jeg ved å ta i bruk Explora- rammeverket, som er designet for å analysere elevers og læreres samtaler i klasserom. (Kimen, 2011) forklarer at Explora er utviklet for å kunne analysere de unike samtalene som oppstår i naturfaget i skolen og som bærer preg av et sosialt og en kulturell virksomhet, og som befinner seg mellom hverdagskulturer og en naturvitenskaplig kultur. Explora er en kodebok/refleksjonsverktøy som er designet for å analysere videoer av naturfagundervisning. Kodeboken vil kunne bidra med å fange opp elementer av kommunikasjon, samspill og engasjement i utforskende arbeid.

For å besvare mitt andre forskningsspørsmål har jeg valgt å ta i bruk kodingen for elevkommunikasjon og innhold i elevsamtaler. Dette vil gi meg muligheten til å se på hvilke momenter som kommer opp i samtalene, og hvilke begrunnelser og tanker de gjør seg mens de jobber med oppgavene sine. Ved å fokusere på disse kategoriene innenfor kodeskjemaet, vil jeg få et innblikk i aspekter ved elevenes bruk av utforskende samtaler samt gi meg kunnskap om samtalenes faglige innhold og idé-generering og spørsmål til andre.

*Tabell 6: Innhold i elevsamtale*

Innhold i elevsamtale	Faglig innhold	Elevenes samtaler omhandler det faglige temaet som undervisningen
	Praktisk organisering	Et utsagn som legger opp til faglig diskusjon
	Sosial organisering	Elevene fordeler arbeidet som skal gjennomføres mellom gruppelemmene.
	Andre ting	Elevene har samtaler som ikke har et faglig innhold

*Tabell 7: Elevkommunikasjon i grupper av elever*

	Idé-generering	Eleven kommer med mulige løsninger på en oppgave eller et problem
	Spørsmål til andre	En elev stiller en eller flere andre elever spørsmål knyttet til det faglige innholdet

Elevkommunikasjon i grupper av elever	Elev-opptak	En elev svar referer eller bygger på det en annen elev tidligere har kommet med i samtalen
	Underkjennelse av andre elever	Eleven kommer med et utsagn som er rettet mot å
	Valg av lette løsninger og snarveier	Elevene velger å ikke fullføre oppgaven tilstrekkelig, men gir heller ikke opp.
	Hjelpeløs og/ eller gir opp	En elev kommer med påstander som signaliserer oppgittthet eller et ønske om å gi opp

Videre har jeg valgt å se på om elevene anvender nøkkelbegreper, som blir omtalt av (Haug, 2017). Dette vil la meg se om elevene klarer å vise en forståelse mellom begrepene og fenomenene som de beskriver, ved at de bruker de i samtale som de har i klasserommet. Jeg vil også her se om det kommer frem pedagogisk sammenhengsskaping, som bygger på å se sammenhenger mellom det hverdagslige og det teoretiske.

#### Etiske vurderinger

Innsamlingen av data har foregått ved observasjon og opptak av klasserom og elevperspektiv. Ettersom klassen er deltakere i MASCOT-prosjektet, har det på forhånd av datainnsamlingen blitt gitt skriftlig samtykke til å delta i prosjektet. De fire elevene som i tillegg har blitt filmet via GoPro, har frivillig meldt seg til dette.

#### Metodens begrensninger

I denne oppgaven har jeg sett på elevenes språk når de anvender (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, og om de har et utforskende språk når de tar i bruk disse praksisene. Videre har jeg tatt i bruk Explora-rammeverket. Jeg har også sett på om elever bruker nøkkelbegrepene som (Haug, 2016) omtaler, samt pedagogisk sammenhengsskaping-praksiser som blir omtalt av (Scott et al., 2011). Dette lot meg studere om elever har et utforskende språk når de tar i bruk algoritmisk tenkning-praksiser, men lar meg ikke se dypere enn samtale som de har med sine medelever. Dette fører til at resultatene som jeg får fra denne metoden, ikke vil gi innblikk i elevenes tanker og praksiser som ikke kommer til syne i elevenes samtaler.

## Kvalitet i studien

Oppgavens validitet vil jeg bedømme som god, da jeg har ved å bruke denne metoden har fått et klart innblikk i hvordan elever bruker algoritmisk tenkning i PRIMM, og hvordan utforskende arbeid kommer til uttrykk når de anvender (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. (Krumsvik, 2015) beskriver at kvalitative studier har validitet om de undersøker det som er studiens hensikt. Dette vil jeg påstå at denne oppgaven har gjort, og derfor er valid.

Jeg vil også si at oppgaven er reliabel. Relabilitet blir av (Krumsvik, 2015) beskrevet som hvorvidt oppgaven er etterprøvbart, noe denne oppgaven er. Ettersom jeg i denne oppgaven har fulgt det kan bli beskrevet som en vanlig klassens undervisningsopplegg, vil det også være mulig for andre i fremtiden å gjennomføre det samme.

## Resultater

I denne delen av oppgaven vil jeg presentere resultatene fra analysen av datamaterialet. Jeg vil presentere dem i rekkefølgen etter forskningsspørsmålene mine:

- Hvordan brukes algoritmisk tenkning i de ulike fasene av PRIMM
- Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tenkning

Jeg vil begynne med å presentere resultatene knyttet til forskningsspørsmål 1, for så å presentere resultatene knyttet til forskningsspørsmål 2. Dette lar meg først se på om og hvordan algoritmisk tenkning brukes i undervisningsopplegg basert på PRIMM, for så å se på hvordan utforskende arbeid kommer til uttrykk når elevene tar det i bruk. Jeg vil presentere hvilke algoritmisk tenkning-praksiser jeg finner hos elevene gjennom det PRIMM baserte undervisningsopplegget, for så å vise hvilke former for utforskende arbeid som kommer til uttrykk i elevenes samtaler når de tar i bruk disse praksisene. For å svare på forskningsspørsmål 1 vil jeg presentere de ulike fasene av PRIMM i kronologisk rekkefølge. Jeg vil fremme hvilke av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning som elevene tar i bruk i de ulike fasene.

Jeg vil så vise til elevenes samtaler når de har tatt i bruk én eller flere av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, og vise om elevene har brukt et utforskende språk gjennom å analysere samtaler deres med Explora-rammeverket, samt se om de benytter nøkkelbegreper eller pedagogisk sammenhengsskaping.



Hvordan brukes algoritmisk tekning i de ulike fasene av PRIMM

Innenfor de ulike fasene av PRIMM var det ulik bruk av algoritmisk tenkning-praksiser. I og med at det er en problemløsningsmetode, tok elevene dem i bruk når de løste oppgaver eller hjalp medelever. Når elevene i de ulike fasene av undervisningsopplegget tok i bruk praksisene, brukte de flere av praksisene samtidig for å løse oppgavene. I undervisningen hvor det var lagt opp til at elevene skulle forklare for hverandre, samarbeide og kommentere på hva koden gjør, ble det i stor grad brukt praksisen der de kommunisere informasjon om et system, men de tok også i bruk praksiser som programmere, undersøke et komplekst system som en helhet, og forstå sammenhengene innad i et system.

### *Predict*

I starten av undervisningen ble det gjennomført en felles predict-fase i undervisningsopplegget. Alle elevene fikk i fellesskap komme med innspill om hva de tenkte at koden som læreren viste på tavlen, ville gjøre. Den felles predict-fasen ble styrt av læreren, men la opp til at elevene skulle kunne komme med påstander om hva de tolket at koden ville gjøre når den ble aktivert. Dette ble gjort som del av alle de tre øktene jeg har transkribert. Predict-fasen i undervisningsopplegget foregikk under oppstarten av undervisningsoppleggene til hver av øktene.

Dette er et utdrag fra den første felles predict-fasen elevene møter på i naturfagundervisningen. Dette er hentet fra tredje økt, etter at de har jobbet med å lage elektriske kretser. Koden som læreren viser frem i Makecode er en bryter, som lar en skru av og på dioden ved å trykke på A-og B-knappen på microbiten.

- 1. Lærer: Dette her ser kanskje litt rart ut, men er det noen som kan tenke seg hva som skjer? Hvis jeg kjører den koden her med microbit.*
- 2. Hilde: Kanskje greia lyser, eller, kanskje skjer det noe med den greia.*
- 3. Lærer: Fredrik.*
- 4. Fredrik: Når man trykker på A og B knappen så lyser de forskjellige greiene.*
- 5. Lærer: Ja*
- 6. Fredrik: Akkurat det som skjer.*
- 7. Lærer: Guro.*
- 8. Guro: Når du trykker A så slås lyset på, og når du trykker B så slår du det av.*

I utdraget bruker elevene praksisene programmering og kommunisere informasjon om et system. Det vises til forståelse for koden til læreren, og elevene bruker praksisen kommunisere informasjon om et system for å dele dette med medelevene. Hilde forklarer kort at koden vil kunne la dioden lyse. Fredrik bygger på dette ved at A- og B-knappene er

bryterne, og til slutt presiserer Guro at A-knappen skrur på mens B-knappen skrur av lyset i dioden.

For å kommunisere dette måtte elevene først bruke praksisen undersøke et komplekst system som en helhet. Elevene har i utdraget sett på hva hele koden gjør. For å gjøre dette tar de i bruk praksisen forstå sammenhengene innad i et system for å se på hva de individuelle blokkene av kode som læreren viser fram gjør, og hvordan de samhandler med hverandre. Dette er fordi den blokkbaserte koden var delt inn i ulike blokker som samhandler.

Det ble også gjennomført en felles predict-fase i femte økt. Læreren viser frem en kode som gjør at dioden vil lyse når microbiten registrerer at det er mindre enn en gitt verdi med lys i rommet.

- 1. Lærer: Men her, i denne koden. Hva er det som gjør at dioden lyser, hva er det som gjør at vi får lys? Tenk tilbake til sist.*
- 2. Fredrik: At du sier til den røde commanden der sier at outputten på den (uhørlig ord) og da står det null eller noe og så skal den være verdi en som betyr at den skal sende ut strømsignaler som da.*
- 3. Lærer: Nettopp.*

Det andre utdragets felles predict-fase brukte også praksisene programmering og kommunisere informasjon om et system. Eleven bruker sin forståelse for kode og de individuelle blokkene til å lage seg et bilde av hva koden gjør som helhet, noe som eleven så formidler til resten av klassen.

I likhet med predict-fasen fra tredje økt, bruker elevene praksisene undersøke et komplekst system som en helhet og forstå sammenhenger innad i et system for å forstå hva koden gjør og hvordan den henger sammen. Predict-fasene brukte i stor grad kommunisere informasjon om et system, forstå sammenhengene innad i et system, og undersøke et komplekst system som en helhet. Dette kom fra at elevene skulle dele hva koden gjorde, og brukte da sin forståelse av de individuelle delene til å lage et helhetlig bilde som de kunne dele med medelevene sine. De skulle se på hva blokkene av kode, og tolke hvordan de henger sammen. Dette brukte de til å forstå hva koden gjorde.

#### *Etterkant av predict-fase*

I etterkant av en felles predict-fase, arbeidet elevene i læringspar. Elevene tok i sitt arbeid i bruk koden som de tidligere hadde gjennomgått. Her jobber elevene i læringspar og har ulike oppgaver hvor de skal anvende koden som de har fått utdelt. Ettersom de resterende fasene av PRIMM foregår i læringspar og ikke i plenum, vil jeg ta i bruk dataen som ble tatt opp fra

perspektivet til elevene som jeg har kalt for «Anna», «Benedicte», «Ciri» og «Dina», samt læringingspartnerene deres.

Selv om undervisningsoppleggene var basert på PRIMM, fulgte de ikke rekkefølgen til enhver tid. De startet alltid med en felles predict-fase, fikk så i oppgave å gjenskape koden og kretsen som læreren hadde vist de. Dette gjør at det ikke er like klare skiller mellom de ulike delene av PRIMM i undervisningsoppleggene, men ulike deler av oppleggene havner mer innenfor de ulike fasene.

#### *Run og Investigate*

Run og investigate-fasene i det PRIMM baserte undervisningsopplegget smeltet inn i hverandre. Elevene gikk i undervisningsopplegget direkte fra den felles predict-fasen til run-fasen, hvor de samtidig gikk inn i koden og så på den. Dette førte til at de to fasene ikke har et tydelig skille i undervisningen.

I tredje økt ble det etter predict-fasen lagt opp til at elevene selv skulle bygge kretsene før de skulle kjøre den. Elevene måtte derfor lage en krets og laste ned koden på microbiten. Elevene jobber her i læringspar, selv om de hadde egne kretser som de bygget og jobbet med. De samarbeidet likevel ved å hjelpe hverandre. I læringsparene vil den ene eleven hjelpe den andre ved å ta i bruk praksisen kommunisere informasjon om et system, som går igjen når elevene skal hjelpe hverandre med å forklare for medeleven hvordan de skulle løse problemene de hadde med kretsen eller koden sin.

I run-fasen av undervisningsopplegget så man at elevene tok i bruk praksisene knyttet til algoritmisk tenkning når de trengte eller fikk hjelp av læringspartnerene sine. Dette kom frem når de stilte spørsmål om hvordan de skulle sette opp kretsen og læringspartneren tok i bruk praksisen forstå sammenhenger innad i et system, ettersom kretsene består av flere mindre deler som må være koblet riktig sammen for å fungere.

Utdraget under er hentet fra den tredje økten, hvor elevene skal koble en krets og få en diode til å lyse. Elevene skulle bygge en krets som kunne bruke koden som de så på i predict-fasen. Elevene hadde her fått en gjennomgang av læreren om hvordan de skulle bygge kretsen. Elevene følger i dette utdraget derfor en algoritme i form av en brukanvisning.

1. *Erik: Trykka du her?*
2. *Benedicte: Trykka på A.*
3. *Erik: Ja, men. A.*
4. *Benedicte: Har du kobla til dioden din da?*
5. *Erik: Ja.*

6. *Benedicte: Åja, nei du har kobla den til feil vei. Den korteste skal på den siden, for det er minus.*

7. *Erik: Jeg visste det, da.*

8. *Benedicte: Også trykk på A... Nei? Hm... Har du lasta den ned da?*

9. *Erik: Ja*

10. *Benedicte: Har du gjort sånn her? (Mumrer til seg selv mens hun sjekker at microbit er tilkoblet) Sånn, og så kan vi laste ned. Når det er ferdig lastet ned kan du trykke på A, og så skal det fungere.*

Benedicte har her klart å koble sin krets slik at dioden lyser, og blir derfor spurt av Erik om hva hun har gjort. Benedicte hjelper så Erik med å feilsøke problemet. Her tar Benedicte i bruk feilsøking og feilretting. Benedicte kommer med en rekke løsninger for å hjelpe Erik med å få dioden hans til å lyse. Benedicte forklarer først hvilke inputs som trigger kretsen. Når det ikke fungerer, går hun dypere og prøver andre løsninger for å få kretsen til å fungere. Gjennom samtalen kommer hun med tre løsninger, som beveger seg fra å forklare hvordan koden virker til å se på den fysiske kretsen som de jobber med og så over til programmet som de bruker til å programmere.

For å kunne utføre feilsøking og feilretting bruker Benedicte sine egne kunnskaper om hvordan kretsen henger sammen som helhet. På linje to forklarer hun at hun trykket på A-knappen på microbiten for å få dioden til å lyse. Når dette ikke hjelper går hun videre og kommer med ytterligere forslag om hva som kan være feilen i kretsen til Erik. Til slutt kommer de frem til at microbiten mangler koden som gjør at den vet hva den skal gjøre med kretsen, når de ser at delene i kretsen er koblet riktig.

Benedicte bruker praksisen forstå sammenhengene innad i et system, for å kunne se på hvilke av de ulike delene som kan være årsaken til at Erik sin krets ikke fungerer. Hun viser at hun forstår hvordan de ulike delene samhandler, ved at man må trykke på bryteren for å få dioden til å lyse og at dioden er avhengig av at strømmen går i en spesifikk retning.

### *Modify*

Under fjerde økt fikk elevene i oppgave å endre på koden som de tidligere arbeidet med i tredje økt. I denne økten fikk de utdelt en ekstra diode og ledninger, og sto fritt til å gjøre endringer på koden slik at den kunne gjøre det de selv ønsket. Denne delen av økten går derfor under modify-fasen, fordi elevene får endre på koden og lage sitt eget produkt av koden som de opprinnelig gjennomgikk i predict-fasen. Ettersom elevene i denne delen av fasen kunne endre på koden som de skulle bruke, fikk de tatt i bruk praksisen programmering. Her fikk de laget sitt eget produkt og kode gjennom å modifisere en annen kode. Selv om ikke

elevene lagde sin egen kode fra grunnen av, har de gjennom å modifisere koden tatt i bruk kunnskapene sine om programmering og laget et produkt som henger sammen og som til slutt blir unikt for dem.

Utdraget under er hentet fra Erik og Benedicte's samtaler mens de jobber med hver sine kretser. Benedicte jobber her med å modifisere koden sin slik at den vil la diodene i kretsen hennes blinke når den blir aktivert, mens Erik forsøker å lage en krets hvor to dioder ved aktivering lyser samtidig. Benedicte tar i bruk praksisen for å forstå sammenhenger innad i et system til å hjelpe Erik med å få kretsen hans til å fungere.

1. *Erik: Skal man ha den lys på, på den lange greia?*
2. *Benedicte: Hm? Ja, den korte er til GND (ground på microbiten).*
3. *Erik: Den som er GND?*
4. *Benedicte: Ja.*
5. *Erik: Den her? ...*
6. *Benedicte: Nei, men det er bare sånn strømmen har et sted å gå ut.*
7. *Erik: Åja.*

Utdraget ovenfor viser hvordan en elev i en modify-fase av undervisningsopplegget får hjelp av en annen. Utdraget illustrerer hvordan en hjelper den andre og forklarer hvordan de ulike delene fungerer sammen. Benedicte, som hjelper Erik, ved å forklare hvilken side på dioden som er positiv og skal være koblet til P0 på microbiten. Benedicte bruker praksisen for å forstå sammenhenger innad i et system ved beskriver hva "GND" som er jordingen på microbiten gjør, for å klargjøre hvordan den henger sammen med resten. Dette gjør hun ved å forklare at det er dit strømmen går. Dette henger sammen med den positive og negative enden på dioden, hvor disse må være koblet til riktig side for å kunne lyse. Hun viser at hun ser sammenhengen med hvordan elektrisiteten i ledningene beveger seg i en spesifikk retning og at utstyret som de bruker i kretsen må passe med den bestemte retningen som elektronene beveger seg. Make-fasen i PRIMM tilsvarer make-fasen i UMC, hvor elevene får lage sitt eget produkt. Elevene har her gjort endringer og tilpasset koden til det de ønsker, og har laget sin egen kode. På slutten av modify-fasen ble det igjen brukt mest kommunisere informasjon om et system, ettersom elevene nå har ferdigstilt koden og kretsen sin og her forklarer hva den gjør og hvordan den fungerer. Elevene skulle da de var ferdige laste opp koden sin med kommentarer på hva de ulike blokkene av kode gjør, slik at læreren kunne se hvordan de har tenkt og løst oppgavene sine. Som i tidligere faser bruker elever sin forståelse av helheten av koden som de har gjort til sin egen.

### *Make*

Make-fasen involverer at elevene tar i bruk det de har lært i de tidligere fasene av PRIMM, og overfører det til en ny oppgave. Dette kom frem i femte økt, hvor elevene begynte med en ny kode som de skulle arbeide med. Dette er en kode som ville skru av og på dioden når lysnivået i rommet gikk under eller over et gitt nivå. Elevene fikk i denne økten de ulike blokkene til hele koden delt ut, men måtte selv sette de sammen i riktige grupper for at koden skulle fungere. Denne koden skulle de ta i bruk senere i temaet, hvor de skal arbeide med å lage smarthus med automatiske lysbrytere.

Elevene i klassen tok i denne situasjonen i bruk praksisene forstå sammenhengen innad i et system, samt programmering. Dette kommer av at de vet hva den ferdige koden skal gjøre, men må kunne se på de ulike blokkene av kode og forstå deres funksjon i det hele bildets for å kunne sette dem sammen riktig. Her måtte elevene selv tolke hvordan de ulike blokkene med kode fungerer sammen ved å se sammenhengene innad i et system.

Elevene tok også i bruk praksisen programmering, som omhandler det å programmere fra bunnen av eller å modifisere en eksisterende kode. Elevene viser at de har ferdigheter innenfor programmering ved at de forstår hva de ulike blokkene av kode gjør. Denne kunnskapen, sammen med å bruke praksisen å forstå sammenhenger innad i et system, lot dem løse utfordringen og lage en funksjonell kode, som gjorde det oppgaven ba om.

### *Hvilke praksiser kom til syne i det PRIMM-baserte undervisningsopplegget*

Gjennom det PRIMM baserte undervisningsopplegget kom det frem at elevene i stor grad brukte praksisene undersøke et komplekst system som en helhet, forstå sammenhengene innad i et system og kommunisere informasjon om et system. Dette kom frem når elevene skulle forklare hvordan hva kode gjorde og hvordan den fungerte. Dette skjedde enten når det ble gjennomført en felles predict-fase eller når en elev hjalp en annen med å forklare for læringspartnerene sine hvordan de måtte sette sammen kretser og kode for at de skulle fungere.

Elevene som satt i læringspar, hjalp hverandre når den ene ikke klarte å løse oppgaven de hadde fått, eller hadde misforstått hvordan noe skulle gjøres. Dette kom sterkt til syne mellom Erik og Benedicte, hvor Benedicte i flere tilfeller hjalp Erik med å løse ulike problemer. Det kom også fram innslag av andre former for algoritmisk tenkning-praksiser som feilsøking og feilretting, men disse var mindre fremtredende da de kom når begge elevene i læringsparet prøvde å løse oppgaven, slik som Benedicte og Erik i tredje økt.

Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tenkning

Jeg vil her presentere resultatene som er knyttet til forskningsspørsmål 2: Hvordan kommer utforskende arbeid til uttrykk i algoritmisk tenkning. Dette gjør jeg ved å se på elevenes språk i situasjoner hvor de har tatt i bruk én eller flere av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Jeg vil se på om elevenes språk viser utforskning, samt om de tar i bruk begreper innenfor Forskerspiren.

#### *Programmering*

Den mest brukte praksisen i det PRIMM-baserte undervisningsopplegget var programmering. Denne praksisen omhandler all bruk av programmering, også den hvor en elev modifiserer en allerede eksisterende kode, slik som i PRIMM og UMC. Dette gjorde at denne praksisen i stor grad ble brukt igjennom hele undervisningsopplegget. Programmering, som en praksis, ble i stor grad brukt med flere av de andre praksisene samtidig når elevene skulle løse et problem. Elevene som i undervisningen skulle programmere sine egne brytere, møtte ofte på problemer som de måtte løse, og tok i bruk andre praksiser mens de brukte praksisen programmering. Selv om elevene ikke tok i bruk et utforskende språk knyttet til praksisen programmering, ble det i stor grad brukt et utforskende språk når elevene anvendte andre praksiser mens de samtidig brukte programmering, som kommunisere informasjon om et system og feilsøking og feilretting.

#### *Kommunisere informasjon om et system*

Den mest framtrødende praksisen hvor elevene i stor grad brukte et utforskende språk, var kommunisere informasjon om et system. Dette gjaldt spesielt for den felles predict-fasen som ble gjennomført i de ulike undervisningstimene. Innenfor denne algoritmisk tenkning-praksisen ser vi at elevene tar i bruk sin evne til å se helheten i koden, for så å dele dette med resten. Etersom det i undervisningsopplegget blir lagt opp til en felles predict-fase, er det naturlig at denne algoritmisk tenkning-praksisen kommer fram der. Denne praksisen ble brukt for å dele løsningsforslag med medelever, enten det var hele klassen under predict-fasen, eller mellom læringspartnere underveis i øktene. Praksisen kommunisere informasjon om et system ble brukt, for å dele andre praksiser som direkte løste oppgaven, slik som forstå sammenhenger innad i et system.

Utdraget under er tatt fra den første timen hvor elevene arbeider med kode i undervisningsopplegget. De har tidligere i økten hatt om elektrisitet og hvor den kommer fra, men har ikke møtt på kode før i dette undervisningsopplegget. Utdraget under er hentet fra da elevene tok i bruk følgende tre algoritmisk tenkning-praksiser: kommunisere informasjon om

et system, undersøke et komplekst system som en hel og forstå sammenhenger innad i et system.

Tabell 8: Kommunisere informasjon om et system, tredje økt

Algoritmisk tenkning praksis		Transkripsjon	Innhold i elevsamtaler	Elev kommunikasjon i grupper av elever	Form for begrep/ bruk av pedagogisk sammenhengsskaping
<p>Kommunisere informasjon om et system</p> <p>Undersøke et komplekst system som en helhet</p> <p>Forstå sammenhenger innad i et system</p>	1.	Lærer: Dette her ser kanskje litt rart ut, men er det noen som kan tenke seg hva som skjer? Hvis jeg kjører den koden her med microbit.	Praktisk Organisering		Pedagogisk sammenhengsskaping
	2.	Guro: Kanskje greia lyser, eller, kanskje skjer det noe med den greia	Faglig Innhold	Idé-generering	
	3.	Lærer: Fredrik.			
	4.	Fredrik: Når man trykker på A- og B-knappen så lyser de forskjellige greiene.	Faglig innhold	Idé-generering	
	5.	Lærer: Ja.			
	6.	Fredrik: Akkurat det som skjer.			



	7.	Lærer: Hilde.			
	8.	Hilde: Når du trykker A så slås lyset på, og når du trykker B så slår du det av.	Faglig innhold	Idé-generering	

I tabellen ovenfor viser jeg utdrag fra den felles predict-fasen. I denne fasen har elevene og læreren i fellesskap sett på en kode og forutsett hva den vil gjøre når man kjører den. I begge tilfellene går dette under praktisk organisering, hvor læreren legger opp til en faglig samtale hvor elevene får komme med sine tanker og ideer. Videre i samtalen bruker elevene i stor grad idé-generering. De ulike elevene kommer alle med lignende påstander om hva koden gjør, men anerkjenner ikke de tidligere påstandene, men kommer kun med sine egne. Selv om dette er den tredje økten i undervisningsopplegget, blir det ikke brukt noen nøkkelbegreper av elevene. Elevenes idé-genererende påstander om koden går direkte på hva de ser på tavlen, og knytter ikke inn faglige eller utforskende begreper i forklaringene sine. I den påfølgende økten ble det også gjort en felles predict-fase, hvor elevene skulle tolke hva en kode ville gjøre når den ble kjørt. Koden de ser på, inneholder en variabel, som er satt til å få dioden i kretsen til å lyse om det blir tilstrekkelig mørkt i klasserommet.

*Tabell 9: Kommunisere informasjon om et system, femte økt*

Algoritmisk tenkning praksis		Transkripsjon	Innhold i elevsamtaler	Elev kommunikasjon i grupper av elever	Form for begrep / Bruk av pedagogisk sammenhengs skaping
Programmering  Kommunisere informasjon om et system	1.	Lærer: Men her, i denne koden. Hva er det som gjør at dioden lyser, hva er det som gjør at vi får lys? Tenk tilbake til sist.	Praktisk organisering		Pedagogisk sammenhengs skaping

Forstå sammenhengene innad i et system	2.	Fredrik: At du sier til den røde commanden der sier at outputten på den (uhørlig ord) og da står det null eller noe og så skal den være verdi en som betyr at den skal sende ut strømsignaler som da.		Idé-generering	Nøkkelbegrep: Fagspesifikke begrep
	3.	Lærer: Nettopp.			

Utdraget ovenfor er hentet fra den femte økten i undervisningsopplegget. I dette utdraget kan vi se at eleven tar i bruk et nøkkelbegrep, og at språket til eleven har fått et større innslag av naturfaglige begreper.

Vi ser at læreren igjen kommer med praktisk organisering, ved at hen stiller et spørsmål til klassen som de skal svare på. Her legger hun opp til at elevene skal tenke og vurdere, for så å kunne komme med sine tanker. Dette blir besvart med idé-generering fra Fredrik, som besvarer spørsmålet til læreren. I utdraget tar Fredrik i bruk et nøkkelbegrep når han kommer med sin påstand om hva koden gjør. Ettersom nøkkelbegrepet her er knyttet til temaet de arbeider med, faller det under fagspesifikke begreper. I predict-fasene ble det ikke brukt noen form for Forskerspire-begreper, hvor læreren etterspurte om elevene hadde hypoteser eller teorier om hva koden gjør. Likevel ser vi at kommunikasjonen i samtalen er utforskende. I utdraget ser vi også et innslag av pedagogisk sammenhengsskaping. Læreren tar i bruk fremme kontinuitet og refererer tilbake til en av de tidligere øktene i undervisningsopplegget. Utdraget som er hentet fra femte økt, lar læreren referere til tidligere undervisningsøkter før de begynte med programmeringen og har arbeidet med elektriske kretser. Dette går også under begrepsbygging, hvor det gjennom ulike situasjoner og gjentatt bruk av begrepene skal tilrettelegge for at elevene skal lære nøkkelbegreper.

### Feilsøking og feilretting

Elevene tok også i bruk feilsøking og feilretting. Dette ble gjort da elevene skulle lage kretser og bruke microbitene som brytere. Elevene hadde i oppgave å lage en krets hvor dioden lyser når de trykker på A, og slås av når de trykker på B.

Tabell 10: Feilsøking og feilretting

Algoritmisk tenkning praksis		Transkripsjon	Innhold i elevsamtaler	Elev kommunikasjon i grupper av elever	Form for begrep/ bruk av pedagogisk sammenhengsskaping
Kommunisere informasjon om et system  Feilsøking og feilretting	1.	Erik: Trykka du her?		Spørsmål til andre	
	2.	Benedicte: Trykka på A.		Elev-opptak	
	3.	Erik: Ja, men A.			
	4.	Benedicte: Har du kobla til dioden din da?	Faglig innhold	Idé-generering	Nøkkelbegrep: fagspesifikke begrep
	5.	Erik: Ja.			
	6.	Benedicte: Åja, nei du har kobla den til feil vei. Den korteste skal på den siden, for det er minus.		Idé-generering	
	7.	Erik: Jeg visste det da.			
	8.	Benedicte: Også trykk på A... Nei? Hm... Har du lasta den ned da?		Idé-generering	
	9.	Erik: Ja.			

	10.	Benedicte: Har du gjort sånn her? (Mumrer til seg selv mens hun sjekker at microbit er tilkoblet.) ... Sånn, og så kan vi laste ned. Når det er ferdig lastet ned kan du trykke på A, og så skal det fungere.		Idé-generering	
--	-----	--	--	----------------	--

I tabellen over, hvor Benedicte hjelper Erik å feilsøke hvorfor kretsen hans ikke lyser, blir det gjennomført en feilsøking. Dette gjør at Benedicte i stor grad med idé-generering hvor hun lister opp potensielle grunner til hvorfor kretsen til Erik ikke fungerer. Benedicte kommer med idé-generering på hvordan Erik kan få kretsen sin til å fungere på linje 4, 6 og 8.

I samtalen tar Benedicte i bruk et nøkkelbegrep. Diode, som er lyspæren som de ønsket skulle lyse, ligger under fagspesifikke begreper og er knyttet til temaet som de arbeider med. Begrepet diode er noe som har blitt brukt av læreren tidligere i undervisningen, slik at begrepet har blitt knyttet til objektet før de skulle lage kretser.

#### *Forstå sammenhengene innad i et system*

Når elevene tok i bruk praksisen forstå sammenhengene innad i et system, kom det frem når de forklarte hvordan de ulike delene i kretsene hang sammen. Dette var en praksis som ble tatt i bruk både i predict-fasen, men også når elevene i læringspar hjalp hverandre. Her brukte den ene eleven sin forståelse for de individuelle delene i kretsen og koden til å assistere medeleven sin.

Utdraget er fra da Benedicte hjalp læringspartnerene sin Erik ved forklare hvordan ledningene må kobles på microbiten for at dioden skulle lyse. Benedicte vil gjennom bruk av denne praksisen vise forståelse for hvordan de individuelle delene henger sammen for å lage en komplett krets som fungerer.

Tabell 11: Forstå sammenhenger innad i et system

Algoritmisk tenkning praksis		Transkripsjon	Innhold i elevsamtaler	Elev kommunikasjon i grupper av elever	Form for begrep/ bruk av pedagogisk sammenhengsskaping
Forstå sammenhengene innad i et system	1.	<i>Erik:</i> Skal man ha den lys på, på den lange greia?	Faglig-innhold	Spørsmål til andre	
	2.	<i>Benedicte:</i> Hm? Ja, den korte er til GND (ground på microbiten)	Faglig-innhold	Idé-generering	Nøkkelbegrep: fagspesifikke begrep
	3.	<i>Erik:</i> Den som er GND?	Faglig-innhold		
	4.	<i>Benedicte:</i> Ja			
	5.	<i>Erik:</i> Den her? ...			
	6.	<i>Benedicte:</i> Nei, men det er bare sånn strømmen har et sted å gå ut.	Faglig-innhold		
	7.	<i>Erik:</i> Åja.			

Elevene hjelper hverandre her i læringspar, hvor den ene forklarer sammenhengen mellom de ulike delene i kretsen slik, og dermed tar i bruk praksisen forstå sammenhenger innad i et system.

Selv om samtalene har innslag av utforskning ved at Erik stiller spørsmål til andre og at Benedicte kommer med idé-generering for å hjelpe han, kommer det ikke frem noen bruk av Forskerspire-begreper i samtalene deres. I denne økten hjalp Benedicte Erik med å forklare

hvordan de ulike delene i kretsen fungerte sammen. Her tok hun i bruk praksisen forstå sammenhenger innad i et system. Dette ser vi ved at hun kommer med uttalelser som forklarer sammenhengen mellom «GND» på microbiten og hvordan elektronene beveger seg inne i ledningene. Dette er også et nøkkelbegrep i form av fagspesifikke begreper, hvor «GND» (forkortelsen for ground) spesifikt beskriver et fenomen knyttet til temaet de har om i undervisningen.

#### *Andre ting*

Gjennom undervisningen var det varierende grad av faglig orientering i elevenes samtaler. I situasjoner hvor elevene i større grad selvstendig klarte å gjennomføre oppgaven som de fikk, ville samtalen med læringspartnerene deres være mindre faglige og havne i Explorakategorien andre ting under innhold i elevsamtaler. De ulike fasene i undervisningsopplegget hadde ulik grad av faglighet og utforskning i samtalen. Dette kom frem særlig mellom elevene Anna og Dina, som ofte arbeidet mer selvstendig og hvor samtalen ikke omhandlet temaet. Dette førte til at samtalen deres hadde liten grad av utforskning. Elever hadde derfor en større bruk av utforskende samtaler og begreper i fasene hvor de ble nødt til å løse problemer. Dette kom frem da de tok i bruk algoritmisk tenkning, hvor de måtte benytte en eller flere praksiser for å løse et problem eller hjelpe læringspartneren.

#### *Diskusjon*

Jeg vil i denne delen av oppgaven drøfte resultatene fra analysen som jeg har gjennomført, og trekke inn og sammenligne det med tidligere teori. Målet er å kunne se på oppgavens hensikt *å få et innblikk i hvilken rolle algoritmisk tenkning har i PRIMM-basert naturfagundervisning, og hvordan det tilrettelegger for utforskende arbeid.* Jeg vil i diskusjonen først drøfte hvordan algoritmisk tenkning kommer frem i elevs utforskning i et PRIMM-basert undervisningsopplegg. Jeg vil videre omtale stillasbyggingen som jeg så i undervisningen og da elevene anvendte algoritmisk tenkning og utforskning. Jeg vil også drøfte hvorvidt dybdelæring er fremtredende i disse situasjonene som jeg har observert. Til slutt vil jeg trekke frem studiens begrensninger og videre studier.

#### *Algoritmisk tenkning i et PRIMM-basert naturfagsopplegg*

Gjennom undervisningsopplegget i naturfag så vi at elevene gjennom PRIMM tok i bruk flere av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Elevene benyttet stor grad praksisen programmering, ettersom de i undervisningsopplegget skulle anvende denne ferdigheten, men de brukte også flere andre av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning underveis i undervisningen. Ved siden av programmering, var det

praksisen kommunisere informasjon om et system som var mest framtreddende. Dette kommer av at elevene i denne klassen satt i læringspar og arbeidet sammen, ved å hjelpe hverandre for å løse oppgavene som de fikk i undervisnings-opplegget. Mens de anvendte praksisen kommunisere informasjon om et system, ble det også tatt i bruk ytterligere praksiser samtidig for å hjelpe til med å løse medelevenes problemer.

Algoritmisk tenkning, som er en tenkemåte rettet mot å løse problemer, viste seg derfor oftest når en elev skulle hjelpe en av medelevene sine. Dette kommer av at det i undervisningen fra lærerens side var lagt opp til at elevene som satt i læringspar, skulle hjelpe hverandre.

Undervisning hvor det ikke er lagt opp til at elevene skal samarbeide, vil sannsynligvis ha en lavere grad av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning i form av å assistere medelever. (Haug et al., 2021) vektlegger bruken av dialog i sin definisjon på utforskende arbeid. Bruken av argumentasjon og formulering av spørsmål er punkter som gjenkjenner utforskende arbeid. Videre fremmer (Mork, 2016) at diskusjon er en viktig del av utforskende undervisning i naturfag. Elevene som i undervisningen tok i bruk en stor andel utforskende samtaler hvor de stilte spørsmål og besvarte dem, viser til utforskning i arbeidet sitt. Dette kom frem da de i det PRIMM-baserte undervisnings-opplegget anvendte algoritmisk tenkning-praksiser idet de skulle besvare medelevenes spørsmål.

Jeg så også stor bruk av et utforskende språk i undervisningen, da elever tok i bruk algoritmisk tenkning-praksisene. Dette kan komme av at (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning i matematikk og naturfag er knyttet opp mot utforskning. Elever vil bruke praksiser som å samle data, analysere data, forstå sammenhenger innad i et system og kommunisere informasjon om et system. Utforskende undervisning i naturfag bygger ifølge (Haug et al., 2019) på akkurat det å samle data og argumentere for sine synspunkter og teorier. Dette inngår i de naturfaglige praksisene og tenkemåtene som omfatter all naturfagundervisning. Naturfagundervisning som tar utgangspunkt i PRIMM, vil også ha innslag av naturfagets kjerneelementer, og det blir derfor naturlig å forvente man ser en grad av utforskning i undervisningen.

Videre så jeg på elevenes språk, hvor jeg undersøkte samtalene og begrepsbruken deres. Der fant jeg at når de tok i bruk (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, var samtalene deres utforskende. Elevene ville komme med spørsmål til andre elever og svare med idé-genering for å hjelpe dem med å løse problemene sine. Selv om de ikke brukte Forskerspire-begreper, som (Haug, 2016) fremmer, viste de begrepsbruk ved at de anvendte fagspesifikke begreper som var knyttet til temaet de arbeidet med.

I undervisningsopplegget så vi at elevene brukte praksiser fra algoritmisk tenkning. Elevene tok både i bruk praksisene til (Weintrop et al., 2016), som jeg så på i analysen av elevsamtalene, men også andre praksiser knyttet opp mot algoritmisk tenkning. Blant MASCOTs algoritmisk tenkning-kort, som er hentet fra (Shute et al., 2017) ser vi ferdigheter som automatisering. I den femte økten ble elevene i oppgave å lage en bryter som automatisk ville sku på dioden om lyset i rommet gikk under et visst lysnivå. For å gjøre dette kopierte de lærerens kode, men måtte selv legge inn verdiene på hvor svakt og sterkt lyset i rommet måtte være for å kunne aktivere bryteren. (Weintrop et al., 2016) sin algoritmisk tenkning-praksis programmering, inkluderer også å modifisere eksisterende kode. Å endre på en kode, viser kunnskaper om koden som et system, og at en elev behersker og klarer å se sammenhengene innad i koden.

Dette går inn under PRIMM, som legger opp til at elevene nettopp får en kode som de skal modifisere og lage til sin egen. Selv om de ikke nødvendigvis har ferdighetene til å lage sin egen kode fra bunnen, anvender de sin forståelse for hvordan koden henger sammen til å lage noe eget. For å gjøre dette anvender de praksisene som forstå sammenhenger innad i et system, hvor de må se på akkurat hvordan de ulike blokkene av kode har en innvirkning på hverandre, slik som inndata som aktiverer koden. Dette er noe elevene gjør i både tredje, fjerde og femte økt, hvor de arbeider med koden som de har fått fra læreren, og tilpasser den til å gjøre det de selv ønsker at den skal. Elevene får gjennom å anvende programmering, lært hvordan ulike blokker av kode henger sammen, og vil kunne opparbeide seg en dypere forståelse for hvordan man tar det i bruk.

#### Stillasbygging

Når jeg ser på det PRIMM-baserte undervisningsopplegget, ser jeg at det blir lagt opp til stillasbygging fra selve rammeverket. Gjennom de ulike fasene vil elever oppleve stillasbygging via undervisningen, ved at de ulike fasene legger opp til aktiviteter hvor elevene får samarbeide. Stillasbygging som blir definert av (Maybin et al., 1992) som hjelpen en elev får for å nå et høyere nivå på sin proksimale utviklingssone, vil også kunne bli overførbart til de andre hjelpemidlene som en elev støttes av. PRIMM fremstår som en hjelper, som gjennom de ulike fasene løfter en elev gjennom den proksimale utviklingssonen som blir omtalt av (Gibbons, P., 2015), som viser forskjellen på hva en elev klarer alene og hva hen klarer med hjelp. Eleven vil til slutt gjennom PRIMM tilegnet seg en større forståelse for programmering. PRIMM, som en fremgangsmåte å lære seg programmeringsferdigheter med eller uten blokkbaserte programmeringsverktøy, vil bidra til læring.



Ettersom hovedprinsippet innad i stillasbygging ifølge (Maybin et al., 1992) omhandler å løfte en elev gjennom sin proksimale utviklingssone fra det den klarer alene til det den klarer med hjelp, kan det argumenteres for at rammeverk og hjelpemidler også kan bidra til stillasbygging. PRIMM som et rammeverk for hvordan lærere kan legge opp undervisning i programmering, vil kunne fungere som et hjelpemiddel som bidrar til stillasbygging for elever. Elevene som uten oppsettet til PRIMM ikke ville ha klart å kunne lage sine egne koder får gjennom PRIMM-stillasbygging, til å nå det punktet for hva de klarer med hjelp i den proksimale utviklingssone.

Man kan også fremme at elevene i undervisning opplevde stillasbygging fra bruken av et blokkbasert kodespråk i naturfagundervisningen. (Brennan & Resnick, 2012) fremmer at blokkbasert kode gir et godt grunnlag for elever når de skal lære seg kode. Elever vil med blokkbasert kode fokusere på hvordan koden og blokkene fungerer, uten å ha kunnskaper om kodespråkets syntaks. Innenfor PRIMM blir dermed også det blokkbaserte programmeringsverktøyet som elevene bruker i det PRIMM-baserte naturfagsopplegget, også en del av elevenes stillasbygging.

I elevenes utforskende samtaler ser vi også innslag av stillasbygging. (Maybin et al., 1992) fremmer at stillasbygging kommer frem via språk mellom en elev og en annen person. Ettersom stillasbygging baserer seg på at en elev får hjelp av en annen person, for så å nå et høyere nivå enn det den klarer alene, kommer det frem stillasbygging i elevenes utforskende samtaler. Samtalene til elevene mens de arbeidet baserte seg i stor grad på spørsmål til andre og idé-generering når de arbeidet i det PRIMM-baserte undervisningsopplegget. Når elevene tok i bruk (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning og anvendte et utforskende språk, ser vi at spørsmålene og svarene som de gir til hverandre bidrar med å løfte hverandre opp. Her kommer Benedicte og Erik frem som eksempler hvor den ene opplever stillasbygging gjennom de utforskende samtalene som han har med læringspartneren sin. (Haug et al., 2021) fremmer at utforskning er tett knyttet opp mot diskusjon og formulering av spørsmål, som vi så mellom Benedicte og Erik.

Innenfor de ulike fasene av PRIMM ligger det til rette for at elever får oppleve stillasbygging i undervisningsopplegget som jeg har fulgt. Predict-fasen, hvor det blir lagt opp til at elevene skal snakke om hva koden gjør, vil la elever som forstår sammenhengene i koden, forklare dette for sine medelever. Dette lar elevene som i utgangspunktet ikke ville klart å forstå hvordan koden fungerer, hjelp til å se hva koden som helhet gjør. Dette så jeg i selve

undervisningen, hvor elever som var sterkere, som Fredrik, kunne komme med sine innspill. Fredrik får gjennom predict-fasen opptre som den sterkere eleven i stillasbygging, hvor han kan bidra til å skape en grunnforståelse for hva koden som blir vist frem til hele klassen, gjør. Predict-fasen tillater dermed at alle elevene i klassen får den samme grunnforståelsen for hvordan koden fungerer.

Run-fasen lar elevene få «hands on» og prøve å se hvordan koden fungerer. De kan se hvilke begrensninger den har. Videre kan elevene i investigate-fasen gå dypere inn og se hvordan den henger sammen, ved å kjøre koden og observere hva de ulike blokkene av kode gjør. Avslutningsvis lar make-fasen elevene ta i bruk det de har lært gjennom predict, run, investigate og modify-fasene, for så å anvende det i en oppgave. Dette går i likhet med tredje trinn av algoritmisk tenkning, hvor man gjennom generalisering vil kunne ta i bruk PRIMM ble i undervisningen kombinert med naturfagets kjerneelementer om utforskning, opplevelser og bruk av et naturfaglig språk.

Både i undervisning og i teorien ser vi at PRIMM, spesifikt det å modifisere en kode som har blitt gitt til en elev, er en av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Denne praksisen omhandler også det å modifisere en eksisterende kode, ettersom en elev vil ta i bruk sine kunnskaper om hvordan koden henger sammen. Elever som alene ikke ville klare å skape et eget produkt fra bunnen av, vil gjennom PRIMM få muligheten til å lage sitt eget produkt. Elevene kan i ulik grad mestre ulike nivåer av programmering, men får gjennom støtten av å kunne modifisere en egen kode til sitt eget produkt, tilegnet seg kunnskaper om hvordan kode henger sammen.

PRIMM som konsept, opptrer derfor som stillasbygging for elever, hvor de gjennom å få se en fungerende kode og bli leid gjennom prosessen å lage noe eget, får økt sin egen forståelse for programmering. Målet er at de gjennom prosessen får tilegnet seg ferdigheter som de kan ta med videre inn i den neste gangen de skal arbeide med programmering. Gjennom hele prosessen av de ulike fasene av PRIMM, får elevene da gjennom rammeverket og samarbeidet som foregår i de ulike fasene, oppleve stillasbygging.

Det kommer også frem at PRIMM bidrar til stillasbygging mellom elevene. Når elever i det PRIMM-baserte undervisningsopplegget samarbeidet gjennom å ta i bruk praksisen kommunisere informasjon om et system sammen med andre praksiser, viste de at de tok i bruk stillasbygging. Stillasbygging, som ifølge (Maybin et al., 1992) bygger på at en elev ved hjelp av en annen elev, klarer mer enn hva hen hadde mestret alene. I samarbeidet til elevene

hadde de utforskende samtaler hvor de hjalp hverandre. Her tok de i bruk praksisen kommunisere informasjon om et system, som sammen med andre av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Elevene tok i bruk algoritmisk tenking-praksiser under den felles predict-fasen eller når de hjalp medelever. Innenfor praksisene tok benyttet elevene et utforskende språk med spørsmål til andre og idé-generering, samt at læreren tilrettela med praktisk organisering. Dette gjorde at de gjennom praksiser som undersøke et komplekst system som en helhet og forstå sammenhenger innad i et system kunne anvende praksisen kommunisere informasjon om et system for å bidra med stillasbygging hos sine medelever.

Under predict-fasene så vi at Fredrik, Guro og Hilde tok i bruk praksisen kommunisere informasjon om et system da de kom med sine forslag til hva koden som læreren viste frem, innebar. I denne fasen av undervisningsopplegget kom elevene med idé-genererende utsagn om hva de antok at koden gjør når den ble kjørt. Elevene som forklarer hva koden gjør, hjelper i dette tilfellet elevene som ikke klarer å bruke praksisen forstå sammenhenger innad i et system, med å kunne forklare hva koden som helhet gjør når den kjøres.

I det PRIMM-baserte undervisningsopplegget ser vi at elevene tar i bruk (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning når de bruker stillasbygging. Dette kommer frem når de viser språklig utforskning, ved at de stiller spørsmål og kommer med idé-generering i samtalene sine. I læringsparene ville den ene eleven få hjelp av den andre med å løse eller forstå oppgaven. Benedicte hjelper Erik med å svare på hans faglige spørsmål. Benedicte, som i tredje økt hjalp Erik ved å ta i bruk praksisen feilsøking og feilretting, bidro med idé-generering hvor hun tok i bruk praksisen forstå sammenhenger innad i et system. Erik har i denne situasjonen nådd sin grense for hva han klarer alene i sin proksimale utviklingszone. Benedicte som anvender (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, bruker stillasbygging til å hjelpe han å nå et høyere faglig nivå og opparbeide seg en større forståelse.

Dette gjaldt ikke kun Benedicte og Erik, men også de andre elevene. De svarte på spørsmål på kryss og tvers, og elever som ikke var læringspartnere kunne fort spørre andre elever rundt seg når de ikke klarte å løse en av oppgavene. Den åpne strukturen i klassen hvor elever kunne svare på hverandres spørsmål når de hadde behov for assistanse, bidro derfor til stillasbygging. Dette ser vi også ved at medelevene kunne benytte seg av de ulike andre elevene rundt seg for å løse oppgavene.

Utforskende undervisning, som ifølge (Haug et al., 2021) i stor grad bygger på dialog, legger opp til at elever skal diskutere og argumentere for sine synspunkter. Elevenes utforskende samtaler og bruk av stillasbygging overlapper dermed i undervisningen, hvor utforskningen kommer til syne via stillas-byggingen mellom elevene selv.

Det er viktig å påpeke at elevenes samarbeid i læringspar i undervisningsopplegget ikke er noe som blir lagt til rette for i PRIMM, men er en praksis som blir fasilitert på ulike skoler. Dette fører til at selv om PRIMM i seg selv ikke direkte tilrettelegger for stillasbygging mellom elever i alle sine faser alene, vil den i undervisning hvor elevene får arbeide i læringspar bidra til stillasbygging gjennom progresjonen i undervisningsopplegget. Dette gjør at resultatene fra en skole hvor elevene ikke sitter i læringspar og snakker like fritt med hverandre, vil kunne ha en mindre grad av stillasbygging i sine PRIMM-baserte naturfagsopplegg. Samtidig kan man også anta at skoler som praktiserer gruppe-arbeid på tilsvarende måte, vil kunne ha en lignende mengde med stillasbygging, hvor elever med ulike styrker i større grad kan være paret for å utfylle hverandre og bidra til å hjelpe hverandre.

Algoritmisk tenkning som en tenkemåte som ifølge (Wing, 2006) tar utgangspunkt i ferdighetene som man lærer i programmeringen, synes også utover (Weintrop et al., 2016) sine praksiser. Elevene anvender praksiser som er knyttet opp mot bruk av programmering, også i oppgaver hvor programmering ikke er hovedfokuset. Elevene som i undervisningen har fått tildelt seg en kode fra læreren, anvender (Weintrop et al., 2016) sin praksis programmering, men bruker også stillasbygging og utforskende samtaler for å løse oppgavene som de har fått.

#### Dybdelæring

Gjennom det PRIMM-baserte undervisningsopplegget viste elevene tegn på dybdelæring i sine samtaler da de tok i bruk (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Når de i de algoritmisk tenkning-praksisene hadde samtaler som var utforskende, viste elevene at de kunne se sammenhenger og sortere kunnskap. (Voll & Holt, 2019) fremhever at dybdelæring foregår over lengre tid og bygger på at elevene skal kunne se sammenhengene mellom ulike faglige elementer og trekke inn sine tidligere erfaringer i nye situasjoner. Dette vises i deres bruk av pedagogisk sammenhengsskaping, i formen av støtte-kunnskapsbygging, som omhandler å se sammenhenger mellom begreper, fenomener og teorier, hvor man trekker det hverdagslige inn i det teoretiske som man møter i skolen. Dette kommer frem når læreren ber elevene tenke tilbake på tidligere økter i undervisningsopplegget, hvor elevene har arbeidet med temaet elektrisitet.

Det blir også sett bruk av formen for pedagogisk sammenhengsskaping som kalles for fremme kontinuitet, som handler om å referere til tidligere økter. Dette er noe læreren gjør i predict-fasene når hun ber elevene tenke tilbake på tidligere undervisning for å aktivisere forutforståelse. Dette bygger kontinuitet og bidrar til å la elevene se sammenhengene mellom temaene som kommer opp i de ulike øktene. Ettersom elevene i starten av det åtte uker lange undervisningsopplegget startet med å ta i bruk støtte-kunnskapsbygging, hvor deres hverdagslige forståelser ble innhentet og knyttet til det faglige, ser vi at de gjennom fremme kontinuitet får aktivert disse kunnskapene og overført dem til de nye øktene hvor de møter på nye fenomener.

Elevene har i undervisningsoppleggene arbeidet med temaet elektrisitet. I disse undervisningsoppleggene har de trukket inn sine erfaringer av hva som bruker strøm, og hvor strøm kommer fra. Ulike elever har fått kommet med innspill i hvordan dette henger sammen, og de har sett på brytere og snakket om hvordan de fungerer i en krets. I dybdelæring er elevenes forkunnskaper en sentral del av det å bygge forståelse, ettersom det i dybdelæring er viktig å bygge robuste nettverk hvor man ser sammenhenger. I undervisningen hvor vi så pedagogisk sammenhengsskaping bli brukt, kommer det frem at elevene i større grad får oppleve dybdelæring gjennom å kunne anerkjenne sammenhengene som finnes.

Dette kommer også frem i via læreplanen, som uttrykker at programmering er noe som skal læres i matematikken, men anvendes i naturfaget. Elevene skal ikke i naturfaget lære de grunnleggende kunnskapene om programmering, men bruke det som et verktøy til å lære naturfag. Dette gjør at programmeringen i naturfaget handler mer om å innlemme denne ferdigheten inn i det faglige. I undervisningen som jeg har fulgt, ser vi at dette blir gjort gjennom å innlemme programmeringen i temaet elektrisitet hvor elevene får lage sine egne elektriske kretser med ulike brytere.

Dette passer med (Taub et al., 2015) sine kjennetegn på dybdelæring. I KI-prosessen, elisitere ideer, blir elever gjort oppmerksom på sine forkunnskaper ved å trekke fram fakta og kunnskaper som de allerede besitter. Elever har i matematikken lært å programmere, men selve anvendelsen skjer i naturfaget. Disse forkunnskapene er noe som læreren henter inn i tredje økt når de felles har en predict-fase før de begynner å arbeide med kretsene og koden sin. Elevene får her gjennom idé-generering trukke frem forkunnskapene sine. Elever som Fredrik, som viser større kjennskap til kodingen, får også trukket opp andre elever med å forklare hva koden gjør.

Det neste trinnet i KI-prosessene er legge til nye ideer. Den siste fasen av PRIMM, som er modify, går ut på at elevene skal anvende det de tidligere har lært i møte med en ny kode. Dette kommer frem i femte økt, hvor de skal programmere en automatisk bryter som slår seg på når lysnivået i rommet går under en viss grense. Elevene møter på dette punktet i undervisningen en mer avansert kode enn de tidligere har arbeidet med, og må bruke kunnskapene som de har tilegnet seg gjennom de tidligere PRIMM-fasene til å løse oppgaven.

Det tredje trinnet i (Taub et al., 2015) sin KI-prosess er å utvikle kriterier for å evaluere ideer. Elever vil innhente data for å bedømme om resultatene som de kommer fram til er gyldige eller ikke. Dette er sterkt knyttet opp mot utforskning hvor elevene skal samle inn data for å svare på spørsmålene sine. Dette går også inn i (Weintrop e al., 2016) sine praksiser for algoritmisk tenkning, hvor det å innhente data er én av praksisene. Å samle data er felles for alle tre, og vi kan se at elevene i deres samtaler bærer preg av dette. Utforskende arbeid foregår i et sosialt fellesskap, hvor elevene sammen bygger kunnskaper ved å trekke inn sine tidligere erfaringer, samler data og argumenterer for sine synspunkter. Dybdelæring, i likhet med annen læring, foregår i sosiale sammenhenger. Elever utvikler kunnskap gjennom samhandling og bruk av språk.

Elevene tar i samtalene sine i bruk nøkkelbegreper i form av fagspesifikke begreper. Disse begrepene, som er knyttet opp mot naturfaget og fagets egne språk, er en viktig del av utviklingen av dybde-læring i naturfag. (Øyehaug, 2019) fremmet at læring er noe som foregår i et sosialt fellesskap, hvor elevene kan anvende et språk og sette ord på ting. Elevene har i det PRIMM-baserte undervisningsopplegget tatt i bruk flere nøkkelbegreper, hvor de viser at de ser sammenhengen mellom det naturfaglige og begrepene. Benedicte, som omtaler «GND», altså ground på microbiten, forklarer hvilken rolle den har i kretsen. Elevene som gjennom undervisningen tar i bruk fagspesifikke begreper i sine utforskende samtaler, viser at de har klart å se koblingene som oppstår i dybdelæring.

Elevene som har hatt utforskende samtaler i undervisningen og der de har tatt i bruk algoritmisk tenkning-praksiser, viser dybdelæring gjennom å benytte et utforskende språk med idé-generering. Dybdelæringen kommer frem gjennom utforskende samtaler og ved at de anvender kunnskapen som de har tilegnet seg til å forklare og komme med ideene sine. Elevenes idé-generering kommer fra deres evne til å se ulike sammenhenger. Ettersom læring foregår i et sosialt fellesskap hvor elever kan anvende kunnskaper og formulere forklaringer, ser vi at de i sine utforskende samtaler opplever dybdelæring.

Målet med dybdelæring, som er å tilegne seg ferdigheter som kan overføres til andre situasjoner gjennom å bygge opp hierarkiske kunnskapsstrukturer, kommer frem i både PRIMM og algoritmisk tenkning. I det PRIMM-baserte undervisningsopplegget så vi at elevene fikk tatt i bruk sine for-kunnskaper i både programmering og i naturfaget, for så å kunne implementere dem i hverandre og se sammenhengene som eksisterer. Ifølge (Palts & Pedaste, 2020) sine faser for algoritmisk tenking, vil elever i det siste trinnet ta i bruk generalisering, som vil la de overføre kunnskapen som de har lært i prosessen til nye situasjoner.

Dette utviklet elevene gjennom all undervisningen, ved at de viste progresjon. Elevene, som tidligere har hatt opplæring i programmering i matematikken, tok dette med inn i naturfagundervisningen. Gjennom det PRIMM-baserte undervisningsopplegget hvor de gjennom de ulike øktene arbeider med mer og mer kompliserte koder, fikk de vist økt kompetanse. Koden som de møtte i tredje økt, som kun består av en enkel bryter, gir elevene muligheten til å forstå hvordan de ulike blokkene fungerer. Elevene tar så dette videre til femte økt hvor de skal sette sammen blokkene med kode slik at de kan lage en automatisk lysbryter som blir skrudd av og på basert på lysnivået.

Elevene viser gjennom de ulike øktene hva de lærer, og klarer å bygge kunnskaper og se sammen-henger. Øktene bygger på hverandre, og oppgavene som elevene får blir stadig mer komplekse. De går fra å bygge en enkel bryter som kan blir skrudd av og på, til å lage en bryter som automatisk skrur seg selv av og på basert på gitte kriterier. Mellom hver av øktene, og de ulike syklusene av PRIMM som elevene får deltatt i, opplever de dermed generaliseringen som (Palts & Pedaste, 2020) omtaler.

(Voll & Holt, 2019) påpeker at dybdelæring er noe som tar tid, og en variasjon av tilnærminger. Dette er noe som undervisningsopplegget la opp til. Undervisningsopplegget foregikk over åtte uker. Innenfor de ulike undervisningsøktene, ble det tatt i bruk et variert undervisningsopplegg. Elevene som i undervisningen tok i bruk (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, som bygger på erfaringer fra programmeringen som de har hatt i matematikken, og som vi ser i matematikk og naturfaget. Disse kunnskapene blir overført over til et helt nytt fagområde, hvor de må se sammenhengene og likhetene mellom de ulike fagene.

LK20 vektlegger i større grad tverrfaglighet. Fagene matematikk og naturfag kan ofte oppleves som separate, men ved å anvende dybdelæring og la elevene se sammenhengene

mellom fagene med å anvende ferdighetene fra matematikken i naturfaget, får elevene oppleve tverrfaglige progresjonen. Ved å også ta i bruk utforskningen som elevene viste i undervisningen, vil det kunne bli brukt til å fremme de varierte tilnærmingene som kreves for å oppnå dybdelæringen som (Voll & Holt, 2019) omtaler.

#### Studiens begrensninger

Gjennom det PRIMM-baserte undervisningsopplegget kom det tydelig frem at læreren som hadde planlagt og gjennomført selve undervisningsopplegget, hadde god kjennskap til både programmering, naturfaget og algoritmisk tenkning. Undervisningsoppleggene var spesielt lagt opp til å utvikle bruk av algoritmisk tenkning i naturfag, som kan komme fra samarbeidet læreren har hatt med MASCOT-prosjektet. Læreren har gjennom sin deltakelse i dette hatt tilgang på kurs, som har gitt læreren gode forutsetninger for å kunne planlegge og gjennomføre undervisningsopplegg som tar i bruk programmering og algoritmisk tenkning.

Undervisningsopplegget som har blitt fulgt i denne oppgaven, som er knyttet til naturfaget, vil også ha en større vekt på utforskning og undring ettersom dette er et kjerneelement i naturfaget. Dette fører til at det vil være større innslag av utforskende aktiviteter og undervisningsopplegg. Dette vil påvirke hvordan (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning blir brukt undervisning, og vil da kun gjelde for naturfaget eller tverrfaglige opplegg som tar utgangspunkt i de naturfaglige kjerneelementene.

Det er også viktig å bemerke at læreren som har laget og gjennomført undervisningsopplegget som

har blitt fulgt i denne oppgaven, har hatt ansvaret for klassen i lengre tid. Det er verdt å merke seg at læreren som gjennomførte den første programmeringsundervisningen også hadde elevene i matematikken. Dette ga læreren en kjennskap til elevenes kunnskaper om programmering og algoritmisk-tenkning ferdigheter. Læreren kunne ut fra dette planlegge og tilrettelegge undervisningen bedre, og vil naturlig kunne bygge på det som elevenes forkunnskaper i matematikk.

Dette er ikke forutsetninger som vil kunne overføres til alle andre undervisningssituasjoner, hvor lærere ikke sitter på den samme kompetansen innenfor fag, metode og algoritmisk tenkning, ettersom ikke alle lærere vil ha gjennomgått kurs knyttet til algoritmisk tenkning og bruk av programmering i klasserommet. Undervisningen som ble fulgt i denne oppgaven, vil derfor ikke kunne overføres til andre PRIMM-baserte naturfagundervisning. Det vil heller ikke kunne være garantert at læreren som har gjennomført programmeringsopplæringen i



matematikken, senere vil være læreren som skal hjelpe elevene å anvende programmering i naturfaget.

Det vi likevel har fått, er et innblikk i et optimalt scenario, hvor læreren har kunnet følge elevene og ha gode kjennskaper til sine elevers ferdigheter innenfor algoritmisk tenkning og programmering. Vi har fått sett hvordan en lærer som besitter gode grunnkunnskaper kan tilrettelegge for og bidra til god undervisning som lar elevene oppleve naturfaget som et praktisk og utforskende fag, hvor de anvender programmering og tar i bruk en algoritmisk tenkemåte når de skal løse ulike problemer.

I denne oppgaven har jeg også kun sett på elevenes samtaler i klasserommet. Dette påvirker hvilke av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning som man ser blir brukt. I elev-opptakene som jeg har transkribert, får jeg ikke sett på elevenes tanker eller hvordan de går frem for å løse et problem, hvor de ikke snakker om det med en av medelevene sine. Dette har gjort at samtale som jeg har sett på i denne oppgaven, hovedsakelig blir rettet mot samtale som foregikk når elevene tok i bruk stillasbygging.

#### Videre studier

Ettersom denne studien er bygget på et kvalitativt datagrunnlag, vil den ikke kunne bli brukt til å generalisere. I denne studien har jeg kun sett på én klasse over en periode, så vil flere innblikk i hvordan elever tar i bruk algoritmisk tenkning bidra til større forståelse for hvordan det blir brukt og hvordan det kan bli lagt opp til å bygge algoritmisk tenkning. Med fremveksten av programmering og algoritmisk tenkning i skolen, kan det være gunstig å gjennomføre ytterligere studier knyttet til temaet i Norge. Videre kan det være interessant å undersøke elevers tankeprosesser og bruk av algoritmisk tenkning når de skal løse oppgaver. Det vil være gunstig å gjennomføre en studie hvor man får en dypere innsikt i hva elever tenker når de løser oppgaver ved hjelp av algoritmisk tenkning.

#### Konklusjon

I denne oppgaven har jeg fått et innblikk i hvilken rolle algoritmisk tenkning har i et PRIMM-basert undervisningsopplegg, og hvordan det tilrettelegger for utforskende arbeid. For å besvare dette har jeg fulgt en klasse over åtte uker, mens de gjennomgikk et PRIMM-basert undervisningsopplegg i naturfag knyttet til temaet elektrisitet. Jeg har sett på elevenes samtaler mens de anvender (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. I samtale som jeg har sett på har elevene tar i bruk et utforskende språk og anvender fagspesifikke begreper i samtale sine.

Gjennom det PRIMM-baserte undervisningsopplegget kommer det tydelig frem at elevene tar i bruk (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning. Disse praksisene bidrar til innlæring og utvikling av ferdigheter, og ble hyppig brukt av elevene i undervisningen. Programmering og kommunisere informasjon om et system var de mest fremtredende praksisene, men andre praksiser som feilsøking og feilretting og forstå sammenhenger innad i et system var også brukt. Ettersom disse praksisene er spesifikt orientert mot STEM-fagene, som matematikk og naturfag, kommer de naturlig frem i naturfagloppet som jeg har fulgt i denne oppgaven.

(Weintrop et al., 2016) sine praksiser for algoritmisk tenkning, som er rettet mot naturfaget, er tett knyttet opp mot utforskning. Elevene viste derfor ofte et utforskende språk da de tok i bruk praksisene i undervisningen. Dette passer med (Haug et al., 2021) sin definisjon på utforskende arbeid, som påpeker at diskusjon, datainnsamling og å formulere spørsmål er kjennetegn på utforskning i naturfagundervisning. Elevenes samtaler hadde inneholdt i stor grad idé-generering i de ulike situasjonene hvor de skulle assistere hverandre. Ettersom det i utforskende arbeid er vektlagt det å argumentere og formulere svar og spørsmål, kan vi se utforskning i undervisningen. Elevene som har samtaler hvor de hjelper hverandre, ved å svare på hverandres spørsmål.

Vi kan også se (Palts & Pedaste, 2020) sine faser i algoritmisk tenkning komme frem. Spesielt deres tredje fase, som involverer bruk av generalisering for å kunne overføre kunnskapene som en har lært gjennom prosessen inn i nye, samsvarer med PRIMM sin make-fase. Det å overføre kunnskapen som de har opparbeidet seg er et moment i både PRIMM og i algoritmisk tenkning. Elevene skal gjennom PRIMM opparbeide seg kunnskaper om programmering som kan generaliseres og overføres videre, slik en vil gjennom algoritmisk tenkning.

Elevene tok i bruk et utforskende språk og (Weintrop & et al., 2016) sine praksiser for algoritmisk tenkning når de hjalp hverandre. Dette kom ofte fram når de tok i bruk stillasbygging, som blir frem-met av (Maybin et al., 1992) som hjelpen en elev får fra en annen person. Eleven vil gjennom denne hjelpen oppnå en høyere forståelse enn hva hen ville klart alene. Elevene som satt i læringspar, stilte spørsmål til hverandre og hjalp dem som ikke forsto hvordan de skulle løse oppgavene. Dette blir det også lagt opp til, ved at predict-fasen i PRIMM skal legger en felles grunn for koden. Her fikk de sterkere elevene fremmet sine

tanker om hva koden som de brukte i undervisningen gjør, slik at alle elevene får et felles grunnlag for å forstå hva koden.

Videre får elevene via samarbeidet som de gjennomfører i de resterende fasene, gjennomgått stillas-bygging ved at de sterkere elevene i undervisningsopplegget hjelper de andre medelevene sine gjennom de utforskende samtale som de har. (Øyehaug, 2019) sier at læring skjer i et sosialt fellesskap. Dette var noe som kom til syne da elevene hadde samtale sine. Her fikk de brukt både det naturfaglige språket, og begrepene som er knyttet til det. Elevene tok i bruk nøkkelbegreper som ble omtalt av (Haug, 2016), men tok ikke i bruk Forskerspire-begreper som er knyttet til utforskning. De anvende allikevel fagspesifikke begreper i undervisningen.

Dette går også over til dybdelæringen som jeg har sett i undervisningen. Elevene får gjennom PRIMM, overført tidligere kunnskaper inn i undervisningen gjennom å ha felles predict-faser, hvor deres tidligere kunnskaper om programmering fra matematikken bidrar til å se sammenhenger også mellom ulike fag. Elevene får gjennom overføringen av programmeringsferdighetene fra matematikken til naturfaget sett sammenhengene mellom de to ulike fagene.

Elevene fikk også gjennom bruken av pedagogisk sammenhengsskaping, benyttet sine egne kunnskaper og erfaringer fra utenfor skolen. Her ble elevenes hverdagslige forkunnskaper og forklaringer trukket inn i skolens faglige forklaringer slik at elevene kan knytte kunnskapene sammen i de hierarkiske strukturene som (Voll & Holt, 2019) fremhever som en del av dybdelæringen.

Ettersom utforskning er så tett knyttet opp mot samtaler og argumentasjon, kommer det naturlig frem en grad av utforskning blant elevene når de hjelper hverandre i undervisningen. I elevenes utforskende samtaler kan man se at de viser at de begrunner svarene til hverandre, når de forklarer hvordan de ulike delene i kretsen henger sammen.

I undervisningsopplegget jeg observerte, var det tydelig at det sosiale fellesskapet var essensielt for læringen, da elevene hjalp hverandre og snakket sammen. Elever som Benedicte, svarte på Erik sine spørsmål. Gjennom stillasbygging, fikk Benedicte tatt i bruk sitt naturfaglige språk og brukt begreper i samtale deres. Hun fikk festet begrep til praksis, og forklart hvordan de ulike delene i kretsen virker sammen, for å fungere.

I undervisningen ser man at flere av (Taub et al., 2015) sine KI-prosesser, som (Waters, 2020) fremhevet. Det kommer både frem at elevene får tatt i bruk elisistere ideer, legge til nye ideer, utvikle kriterier for å evaluere ideer og sortere og reflektere. Elevene blir tidlig bedt om å se tilbake på tidligere økter, der de har sett på temaet elektrisitet. Dette har de bygget på gjennom å legge til nye ideer i undervisningen og koble disse til forkunnskapene som elevene har hatt.

Når elevene i det PRIMM-baserte undervisningsopplegget tar i bruk algoritmisk tenkning-praksiser, ser vi også at de tar i bruk et utforskende språk, hvor de i stor grad kommer med idé-generering. Dette vil kunne forekomme ved at flere av (Weintrop et al., 2016) sine praksiser for algoritmisk tenkning er tett knyttet opp mot utforskning. Ettersom tankemåten er knyttet opp mot det å løse problemer, vil elevene naturlig anvende disse praksisene når de skal løse en arbeidsoppgave i undervisningen. Dette kom frem når de i tok i bruk et utforskende språk for å bidra med stillasbygging hos medelevene deres.

## Litteraturliste

- Abdi, A. (2014). The Effect of Inquiry-based Learning Method on Students' Academic Achievement in Science Course. *Universal Journal of Educational Research*, 2(1), 37-41.
- Anderse, R. (2022). Blokkbasert programmering og algoritmisk tenkning i en samarbeidslæringskontekst: En case-studie av programmering integrert i et matematikkfag. *Acta Didactica Norden*, 16(4). 1-22.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada, 1*, 1-24.
- Cardak, O., Onder, K., & Dikmenli, M. (2007). Effect of the usage of laboratory method in primary school education for the achievement of the students' learning. *In Asia-Pacific forum on science learning and teaching*, 8(2), 1-11.
- Dolonen, J. A., Kluge, A., Litherland, K., & Mørch, A. I. (2019). Litteraturgjennomgang av programmering i skolen.
- Futurelearn. (u.å.). *What is PRIMM?* Hentet 12. Mai 2023 fra <https://www.futurelearn.com/info/courses/secondary-programming-pedagogy/0/steps/68416>
- Gibbons, P. (2002). *Scaffolding language, scaffolding learning*. Portsmouth, NH: Heinemann.
- Papert, S., & Harel, I. (1991). Situating constructionism. *constructionism*, 36(2), 1-11.
- Haug, S. B., & Mork, S. M. (2021). *Nøkkeltbegreper i utforskende arbeid*. Universitetsforlaget.
- Haug, B. S., Sørborg, Ø., Mork, S. M. & Frøyland, M. (2021). Naturvitenskaplige praksiser og tenkemåter – på vei mot et tolkningsfellesskap. *NorDiNa*. 17(3), 293-310.
- Voll, L. O. & Holt, A. (2019). Dybdeløring i naturfag. | Holt, A. (Red.), Dybdeløring i naturfag. (s. 17-36). Universitetsforlaget.
- Ødegaard, M., Möller Andersen, H., Schoultz, J., Hultman, G., Lund Nielsen, B., Löfgren, R., & Mork, S. M. (2011). *Explora: Koding av elevers og læreres samtaler ved praktisk arbeid i skandinaviske klasserom*. Naturfagsenteret, Oslo universitet.
- Krumsvik, R. B. (2015). *Forskningsdesign og kvalitativ metode – ei innføring* (2. utg.). Fagbokforlaget.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L. & Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, 2(1), 32-37.

Lund, M. (2022) Use-Modify-Create framework. Ellipsis education.

<https://ellipsiseducation.com/blog/use-modify-create-framework>

Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A. & Barnes, T. (2019). Use, modify, create: Comparing computational thinking lesson progressions for stem classes. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* 395-401.

Maybin, J., Mercer, N., & Stierer, B. (1992). Scaffolding learning in the classroom. *Thinking voices: The work of the national oracy project*, 186, 195.

Minner, D. D., Levy, A. J., & Century, J. (2010). Inquiry-based science instruction—what is it and does it matter? Results from a research synthesis years 1984 to 2002. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 47(4), 474-496.

Nyutstumoen, S. (2021) Programmering og problemløsning i matematikk [Masteroppgave, Norges teknisk-naturvitenskapelige universitet].

OsloMet. (u.å.) Matematikk, naturfag og algoritmisk tenkning (MASCOT). Hentet 12. Mai 2023 fra <https://www.OsloMet.no/forskning/forskningsprosjekter/matematikk-naturfag-algoritmisk-tenkning>

Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, 19(1), 113-128.

Papert, S. (1980). *Children, computers, and powerful ideas*. Harvester.

Rotherham, A. J., & Willingham, D. T. (2010). 21st-century” skills. *American Educator*, 17(1), ArtikkelEJ889143 <https://eric.ed.gov/?id=EJ889143>

Scott, Mortimer, E., & Ametller, J. (2011). Pedagogical link-making: a fundamental aspect of teaching and learning scientific conceptual knowledge. *Studies in science education*, 47(1), 3–36.

- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18, 351-380.
- Sentance, S., Waite, J. & Kallia, M. (2019) Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2-3), 136-176.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22, 142-158.
- Utdanningsdirektoratet. (2019, 27. mars). *Algoritmisk tenkning*. Udir. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2020). *Læreplan i matematikk (MAT01-05)*. Fastsatt som forskrift. Læreplan for kunnskapsløftet 2020. <https://www.udir.no/lk20/mat01-05?lang=nob>
- Utdanningsdirektoratet. (2020). *Læreplan i naturfag (NAT01-04)*. Fastsatt som forskrift. Læreplanverket for kunnskapsløftet 2020. <https://www.udir.no/lk20/nat01-04>
- Waters, J. B. (2020). Programmering og dybdeløring i fysikk - en kvalitativ studie av elevers arbeid med programmering i fysikk 1.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology* 25, 127-147.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 60(4), 55-62.
- Ødegaard, M., Haug, B. S., Mork, S. M. & Sørvik, G. O. (2016) *På forskerfötter i naturfag*. Universitetsforlaget.
- Øyehaug, A. B. (2019). Kjennetegn på undervisning som gir dyp forståelse. | Holt, A. (Red.), *Dybdeløring i naturfag*. (s. 38-57). Universitetsforlaget.