

## **Programmering som verktøy i matematikkundervisning**

En flercasestudie av tre læreres kunnskap om- og adaptasjon av programmering, på bakgrunn av TPACK-rammeverket

SOROUSH AGHAKOUCHAKI  
HENRIK SNILSBERG

VEILEDER

Said Hadjerrouit

**Universitetet i Agder, 2024**  
Fakultet for teknologi og realfag  
Institutt for matematiske fag

Master



# Forord

Denne masteroppgaven markerer enden på vår tid på grunnskolelærerutdanningen, men også starten på et liv med læring i skolen. Vi startet utdanningen vår på Universitetet i Agder i 2019, der begge bestemte seg tidlig for å ta en master i matematikdidaktikk. Vi har begge hatt en fascinasjon for dataspråk, og hvordan den digitale hverdagen har forandret seg fra da vi vokste opp. Da vi oppdaget TPACK rammeverket, visste vi tidlig at vi ønsket å skrive en oppgave om nettopp dette. Programmering viste seg å være det mest interessante temaet innen digital teknologi, spesielt med tanke på det nye læreplanverket.

Det er mange vi må takke for i forbindelse med denne oppgaven. Først må vi takke hverandre for godt samarbeid og holdning til å skrive denne oppgaven. Vi er veldig fornøyd med å ikke måtte stå alene i valgene om metodebruk, analyse og lignende. Videre må vi takke våre medstudenter, venner og familie som har vært god støtte gjennom den lange skriveprosessen. Takk for gode råd, vennlighet og optimisme.

Vi er veldig takknemlig for de tre lærerne som ønsket å stille til intervju og observasjon i forbindelse med oppgaven. Det var en vanskelig prosess å finne informanter til oppgaven, så vi kunne aldri laget denne oppgaven uten hjelpen vi fikk derfra.

Sist men ikke minst må vi rette en stor takk til vår veileder, Said Hadjerrouit. Takk for at du trodde på oss hele veien gjennom. Det har vært en ubeskrivelig hjelp med rådene vi har fått, artiklene du har vist oss og de grundige og konsekvente tilbakemeldingene vi har fått gjennom arbeidet med oppgaven. Uten forventningene du satt til oss, ville den ikke blitt av det kaliberet den er nå.

Kristiansand, mai 2024

Soroush Aghakouchaki & Henrik Snilsberg



# Oppsummering

I denne masteroppgaven presenteres et kvalitativ studie av integreringen av programmering i matematikkundervisningen hos matematikklærere på ungdomstrinnet. Studiens hensikt er å undersøke læreres vektlegging av kunnskaper der programmering ble benyttet som hjelpemiddel i matematikkundervisning. Oppgaven baseres på teori om lærernes teknologiske pedagogiske faginnholdskunnskap (TPACK-rammeverket) og en 5-steps modell om adaptoring av teknologi i klasserommet. Problemstillingen går ut på å undersøke lærernes grunnleggende kunnskaper og integrasjon av programmering ved bruk av TPACK-rammeverket. Vi har utviklet tre konkrete forskningsspørsmål for å besvare denne problemstillingen:

- (1) Hvordan vektlegger lærerne teknologisk-, pedagogisk- og faginnholdskunnskap i arbeidet med programmering i matematikkundervisning på ungdomsskolen?
- (2) Hvilke elementer i TPACK rammeverket identifiseres i den påståtte planleggingen og utføringen av undervisning i matematikk med programmering som verktøy i ungdomsskolen?
- (3) I hvilken grad utfyller lærerne stegene i Niess et al. (2009) sin 5-steps modell om integrering av teknologi, spesifikt programmering, i matematikkundervisning på ungdomsskolen?

For å kunne svare på disse har vi gjennomført et studie hvor vi intervjuet og observerte tre lærere for å høre om deres opplevelser med og tanker om å bruke programmering som hjelpemiddel i matematikkundervisning. Ved bruk av TPACK-rammeverket har vi analysert svar og observasjoner for å finne hvilke elementer av kunnskap lærerne omtaler og bruker i arbeidet.

Resultatene viser til at kunnskap om teknologi, faginnhold og pedagogikk er alle tre viktige faktorer i bruk av programmering i undervisning, der ingen av dem skiller seg spesielt ut. Lærerne vektlegger vanskeligheten av skillet mellom programmering som hjelpemiddel og kompetansemål, samt hvordan dette passer til de forskjellige temaene. Likevel viser alle lærerne at de har adaptert programmering godt inn i undervisningen, og evaluerer stadig virkningen den har på elevene.

**Nøkkelord:** TPACK, programmering, algoritmisk tenking, computational thinking ungdomsskole, matematikkundervisning



# Abstract

In this master thesis a quantitative study of the integration of programming in mathematics education among secondary school teachers is presented. The purpose of this study is identifying the teachers emphasis of different knowledge in teaching of mathematics education with programming as a learning tool. The thesis is based on theory of teachers technological pedagogical content knowledge (TPACK-framework) and a 5-step model of adapting technology into the classroom. The main research question involves identifying teachers basic knowledge and integration of programming using the TPACK-framework. We have developed 3 specified accompanying research questions:

- (1) How do the teachers emphasise technological, pedagogical and content knowledge in practicing programming in mathematics education in secondary school?
- (2) What elements in the TPACK-framework is identified in the alleged planning and execution of education of mathematics with programming as aid in lower secondary school?
- (3) To which extent do the teachers fulfill the 5-step model of integrating technology from Niess et al. (2009), programming specifically, in mathematics education in lower secondary school?

In order to answer these questions we have conducted a study where we interviewed and observed three teachers to determine what their experiences and thoughts are with using programming as a learning tool in mathematics education. With usage of the TPACK-framework, we have analyzed answers and observations to find which elements of knowledge the teachers mentions and uses in their work.

The results shows that knowledge of technology, pedagogy and content are all equally important in usage of programming in mathematics education, where none of the knowledge differs much. The teachers implies that the difficulty in differentiating programming as a tool and as a competence goal, as well as how they fit the different subjects. Still all the teachers show that they have adapted programming well in their education, and continually evaluates the impact it has on the pupils.

**Keywords:** TPACK, programming, computational thinking, teachers in lower secondary schools , mathematics education





# Innholdsfortegnelse

<b>Forord</b> .....	<b>I</b>
<b>Oppsummering</b> .....	<b>III</b>
<b>Abstract</b> .....	<b>V</b>
<b>Innholdsfortegnelse</b> .....	<b>1</b>
<b>1 Introduksjon</b> .....	<b>3</b>
1.1 Bakgrunn.....	3
1.2 Problemstilling og forskningsspørsmål.....	4
1.3 Litteraturgjennomgang.....	4
<b>2 Teori</b> .....	<b>6</b>
2.1 TPACK.....	6
2.1.1 Faginnholdskunnskap (CK).....	7
2.1.2 Pedagogisk kunnskap (PK).....	8
2.1.3 Teknologisk kunnskap (TK).....	8
2.1.4 Pedagogisk faginnholdskunnskap (PCK).....	8
2.1.5 Teknologisk pedagogisk kunnskap (TPK).....	9
2.1.6 Teknologisk faginnholdskunnskap (TCK).....	9
2.2 TPACK i matematikk.....	9
2.3 Programmering i skolen.....	11
2.4 Computational thinking og algoritmisk tenking.....	12
<b>3 Metode</b> .....	<b>14</b>
3.1 Forskningsparadigme.....	14
3.2 Forskningsstrategi: Kvalitativt studie.....	15
3.3 Forskningsdesign: Case-studie.....	15
3.3.1 Multiple case-studie – Flercasestudie (komparative casestudier).....	16
3.3.2 Utvalg av deltagere.....	16
3.4 Datainnsamling.....	17
3.4.1 Intervju.....	17
3.4.1.1 Intervjuguide.....	18
3.4.2 Observasjon.....	18
3.4.2.1 Observasjonsskjema.....	19
3.5 Metode for dataanalyse.....	19
3.5.1 Induktiv- Deduktiv tilnærming.....	20
3.5.2 TPACK analyseverktøy.....	20
3.5.3 Tematisk analyse.....	23
3.5.3.1 Utvikling av analyseverktøy.....	23
3.6 Etske betraktninger.....	25
3.7 Forskningskvalitet.....	26
3.7.1 Trustworthiness (Troverdighet).....	26
3.7.2 Validitet og reliabilitet.....	28
3.7.2.1 Reliabilitet.....	28

3.7.2.2 Validitet.....	29
<b>4 Analyse og resultat.....</b>	<b>30</b>
4.1 Case 1: Lærer 1.....	31
4.1.1 Intervju.....	31
4.1.2 Observasjon.....	34
4.1.3 Oppsummering – triangulering.....	35
4.2 Case 2: Lærer 2.....	36
4.2.1 Intervju.....	36
4.2.2 Observasjon.....	39
4.2.3 Oppsummering - triangulering.....	40
4.3 Case 3: Lærer 3.....	41
4.3.1 Intervju.....	41
4.3.2 Observasjon.....	45
4.3.4 Oppsummering - triangulering.....	46
4.4 Cross case – multiple case design.....	47
4.4.1 Teknologisk, pedagogisk og faginnholdskunnskap.....	47
4.4.2 TPACK.....	48
4.4.3 Adaptering av teknologi i undervisning.....	50
<b>5 Diskusjon.....</b>	<b>52</b>
5.1 Svar på forskningsspørsmål.....	52
5.1.1 Læreres vektlegging av hovedkunnskapene i TPACK.....	52
5.1.2 Identifisering av kategorier i TPACK rammeverket.....	54
5.1.3 Integrering av programmering.....	55
5.2 Pedagogiske og praktiske implikasjoner.....	58
5.3 Kritiske refleksjoner om metode og teori.....	60
<b>6 Avslutning.....</b>	<b>65</b>
6.1 Konklusjon.....	65
6.2 Videre forskning.....	67
6.3 Egne refleksjoner.....	68
6.3.1 Student 1.....	68
6.3.2 Student 2.....	68
<b>7 Referanseliste.....</b>	<b>70</b>
Litteratur.....	70
Tabeller og figurer.....	74
<b>8 Vedlegg.....</b>	<b>75</b>
Vedlegg 1 - sikt godkjenning.....	75
Vedlegg 2 - Intervjuguide.....	77
Vedlegg 3 - observasjonsguide.....	79
Vedlegg 4 - Samtykkeskjema.....	81
Vedlegg 5 - Infoskriv til elever og foresatte.....	85
Vedlegg 6 - Data av intervju og observasjon.....	86
Vedlegg 7 - Transkripsjoner.....	87
Vedlegg 8 - Observasjoner.....	106

# 1 Introduksjon

## 1.1 Bakgrunn

Digital teknologi har i løpet av de siste ti årene fått en stadig større plass i samfunnet og skolen. Teknologiens rolle har utfordret arbeidsforutsetningen til skolen og lærerne som jobber i den, i tillegg til kompetansemålene til elevene. Mulighetene lærere står overfor er store, derimot kreves det visse forutsetninger til grunn for at læreren kan utnytte mulighetene på best mulig vis (Sanne et al., 2016).

Det digitale hjelpemiddelet kan ikke bare benyttes uten en gjennomtenkt plan på hvordan det skal hjelpe. Fokuset på forutsetninger og endringsprosesser hos elever og lærere er nødvendig for utnyttelsen av de nye digitale mulighetene. Samtidig kan ikke det digitale verktøyet overta eller erstatte det kognitive arbeidet som kreves for at elever skal lære (Ludvigsen, 2012, sitert i Sanne et al., 2016). Mishra & Koehler (Mishra & Koehler, 2006, sitert i Koehler & Mishra 2009) beskriver et problem ved dette: At teknologien selv står i fokus, og ikke hvordan den kan brukes til å forbedre læringen hos elevene. Ut i fra dette har de utarbeidet et rammeverk kalt TPACK, der lærerens evne til å lære bort ved hjelp av teknologi først er komplett når vedkommende klarer å kombinere kunnskapene sine innen pedagogikk, teknologi og faginnhold.

I den nye læreplanen, LK20, har en andel av programmeringen gått inn under matematikk. I løpet av ungdomsskolen skal blant annet elevene kunne forstå programmering av løkker, variabler og hvis-formler. I tillegg skal de kunne utforske matematiske egenskaper og sammenhenger (Kunnskapsdepartementet, 2019). Nødvendigheten av kompetanse i digital teknologi faller på faglærere, da det ikke finnes et eget fag i teknologi og programmering. Den digitale kompetansen som kreves inneholder hvordan datamaskinen fungerer og styres, samt hvordan datamaskinen brukes til å programmere (Sanne et al., 2016).

Hensikten med studien er å undersøke hvilken erfaring matematikklærere på ungdomsskolen har med programmering som verktøy i matematikkundervisning. Samt å kunne si noe om hvilke kunnskaper som er nødvendig for å kunne lære bort målene i læreplanen. Vi har valgt å begrense oss til lærere på ungdomsskolen da vi opplever at læreplanmålet som innebærer programmering etter 10. trinn er mindre konkret, og da stiller større krav til læreren i form av

planlegging, vurdering og undervisning. I løpet av snart fem år på lærerstudiet har vi selv fått flere inntrykk av ulike skoler, samt diskutert med andre studenter, at programmering er et tabubelagt tema. Noe vi bemerket oss som en gjenganger var læreres lettelse over å få studenter som kunne bidra med programmering. Dette har lærere begrunnet med vår erfaring med den “nye” læreplanen. Vi ble nysgjerrige på hvordan lærere på praksis- og lærerutdanningskoler ikke var trygge på dette verktøyet som læreplanen eksplisitt etterspør.

## 1.2 Problemstilling og forskningsspørsmål

Vår masteroppgave handler om lærere og deres kunnskaper, samt hvordan programmering ble benyttet i matematikkundervisningen. For å tilfredsstille våre kriterier på hva vi så etter, benyttet vi TPACK rammeverket for å se nærmere på lærernes grunnleggende kunnskaper og integreringen av teknologi i disse. Vi tar utgangspunkt i rammeverket for å besvare våre 3 forskningsspørsmål:

- (1) Hvordan vektlegger lærerne teknologisk-, pedagogisk- og faginnholdskunnskap i arbeidet med programmering i matematikkundervisning på ungdomsskolen?
- (2) Hvilke kategorier i TPACK rammeverket identifiseres i den påståtte planleggingen og utføringen av undervisning i matematikk med programmering som verktøy i ungdomsskolen?
- (3) I hvilken grad utfyller lærerne stegene i Niess et al. (2009) sin 5-steps modell om integrering av teknologi, spesifikt programmering, i matematikkundervisning på ungdomsskolen?

## 1.3 Litteraturgjennomgang

I dette delkapitlet legger vi frem prosessen vi brukte for å finne relevant forskningslitteratur som omhandler bruken av programmering i matematikkundervisning, og hvordan man kobler dette til matematisk TPACK. Vi skal også beskrive noen utfordringer og muligheter som kommer frem i den aktuelle faglitteraturen som stod som grunnlag for valg av forskningsområde, problemstilling og forskningsspørsmål

For å danne et litteraturgrunnlag gjorde vi søk på nettsidene oria.no og scholar.google.com med nøkkelord som TPACK, matematikk, (integrering av) programmering, algoritmisk tenking og “21st century skills”. Her leste vi abstract i mange artikler for å finne de som var mest relevant for vårt prosjekt. Vi valgte å begrense oss til artikler skrevet mot læreplanen

LK20, samt å fokusere på den mest oppdaterte forskningen. Vi brukte også referansene i disse artiklene til å videre utvide litteraturgrunnlaget, da noen av artiklene bygget på andre begreper, eller progresjonen i skolen.

Noen mulige bidrag programmering kan ha til matematikkundervisningen er økt engasjement og utvikling av problemløsningsevner. Gjennom at elever får arbeide på en interaktiv og praktisk måte kan de være mer deltagende og utforskende. (Kaufman & Stenseth, 2021)

I Kaufmann og Stenseth (2021) kommer også funnene til Kafaj og Burke (2013) frem, de så på hvilke utfordringer som oppstod på 80- og 90-tallet da programmering først ble introdusert i undervisningen. Disse utfordringene var utilstrekkelig kompetanse i programmering hos læreren, begrenset tid til å arbeide med programmering på skolen, mangel på maskinvare og programvare og til slutt motstand til endring av undervisningsmetoder

For å avdekke den digitale tilstanden i norske skoler og barnehager ble det gjennomført en kartlegging, Monitor 2019. I denne kartleggingen kommer det fram at dekningsgraden maskin- og programvare har forbedret seg siden 2016, da den forrige kartleggingen ble gjort. Tilgangen på maskinvare var altså på et tilfredsstillende nivå. Likevel vil kvaliteten på utstyret og lærerens kunnskap om det, være avgjørende for om de digitale hjelpemidlene bidrar til læring (Fjørtoft, Thun & Buvik, 2019).

Å integrere programmering i undervisning på en hensiktsmessig måte er en av utfordringene som har oppstått etter implementeringen av LK20. Programmering skal ikke være et nytt verktøy for å utføre de samme algoritmene som tidligere. Det må brukes til å løse nye typer problemer og finne en generell løsning for flere typer problemer. Arbeidet mot disse generelle løsningene kalles algoritmisk tenking eller CT, og er det viktigste i arbeid med programmering. Dette arbeidet trenger ikke nødvendigvis gå gjennom digitale verktøy (Kaufman & Stenseth, 2023).

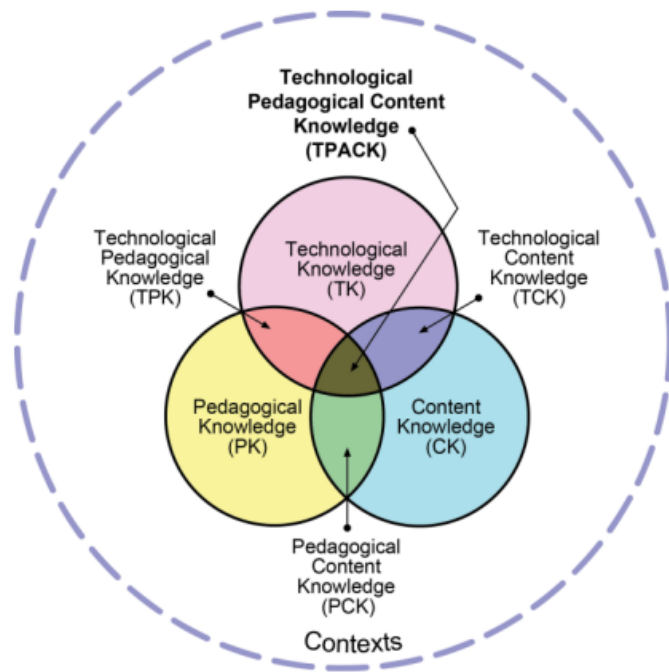
## 2 Teori

Dette kapittelet kommer til å inneholde relevant teori for vår studie. Vi starter med å beskrive TPACK rammeverket og hvordan det kan integreres i matematikkundervisning. Deretter kommer vi til å se på digital kompetanse i skolen og begrepene computational thinking og algoritmisk tenking.

### 2.1 TPACK

Technological pedagogical content knowledge, eller forkortet TPACK, er et rammeverk som skal hjelpe lærere å definere kunnskapen som er nødvendig for å bruke teknologi i undervisningen (Baran, Chuang & Thompson, 2011). TPACK består av tre komponenter som skaper grunnpilaren i en teknologisk undervisning for læreren: Content knowledge, pedagogical knowledge og technological knowledge (Koehler, Mishra & Cain, 2013). Dette velger vi å oversette til faginnholdskunnskap, pedagogisk kunnskap og teknologisk kunnskap. Herfra kommer vi til å referere til kunnskapene på den norske måten. Samlet skaper de tre komponentene sammen teorien om TPACK. Teorien baserer seg på flere års forskning der målet var å kartlegge nødvendige ferdigheter en lærer må ha for å undervise i teknologisk undervisning. Konklusjonen av forskningen tilsier at først når TPACK oppnås, vil den digitale undervisningen kunne hensiktsmessig fremme elevers læring (Mishra & Koehler, 2006, sitert i Koehler & Mishra, 2009).

Rammeverket har tatt utgangspunkt i Shulman (1986, 1987) sin teori om PCK, og bygger videre på denne modellen når teknologi skal implementeres (Koehler & Mishra, 2009). PCK, eller pedagogisk faginnholdskunnskap vises i figur 1, og forklares som møtepunktet mellom pedagogisk og faginnholdskunnskap. Ved hjelp av forskningen utført av Mishra og Koehler(2006) har det nye rammeverket som inkluderer teknologisk kunnskap inspirert lærere til å analysere deres bruk av teknologi i klasserommet, og hvordan teknologien kan skape læring (Cox & Graham, 2009). Videre skal vi utdype de forskjellige elementene, vist i figur 1, som utgjør TPACK.



**Figur 1:** TPACK rammeverket (Koehler & Mishra, 2009)

### 2.1.1 Faginnholdskunnskap (CK)

Faginnholdskunnskap er lærerens egen kunnskap om faget vedkommende underviser i eller lærer om (Koehler & Mishra, 2009). En lærer må kunne mestre sentrale begreper, fagets egenart og relevant teori innen faget for å kunne tilfredsstille faginnholdskunnskapen tiltrengt (Mishra & Koehler, 2006). Faginnholdet er derimot ikke den samme for en lærer på barneskolen som den er på høyskole, noe som gjør faginnholdskunnskapen avhengig av både fag og klasse (Koehler & Mishra, 2009). Kunnskapen er uavhengig av lærerens pedagogiske økter eller representasjoner for læring (Cox & Graham, 2009).

### 2.1.2 Pedagogisk kunnskap (PK)

Pedagogisk kunnskap viser til lærerens evne og kunnskap til å lære bort. Samlingen av praksis- og metodekunnskaper sammen med erfaring om undervisning skaper et pedagogisk repertoar hos læreren som beskrives som pedagogisk kunnskap. Dette gjelder også kunnskap om overordnet del og læreplanmål (Koehler, Mishra & Cain, 2013). Magnusson, Krajcik og Borko (1999) deler den pedagogiske kunnskapen inn i generell pedagogisk kunnskap og fagspesifikk pedagogisk kunnskap (Cox & Graham, 2009, s.62). I denne sammenhengen blir den fagspesifikke kunnskapen beskrevet under PCK senere. Den generelle kunnskapen faller

utenfor faget og beskriver lærerens aktiviteter og strategier for å optimalisere læringsplattformen. Dette inkluderer motivasjon av studenter, kommunikasjon med elever, lærere og foreldre, formidlingsevne, klasseromsledelse og mye mer (Cox & Graham, 2009).

### 2.1.3 Teknologisk kunnskap (TK)

Teknologisk kunnskap er ifølge Cox og Graham (2009, s.64) kunnskapen om å bruke teknologiske hjelpemidler som dukker opp på en hensiktsmessig måte. Siden teknologien stadig utvikles og er i en fluktuerende form, er det ufattelig vanskelig å definere kunnskapen på en konkret og god måte. Definisjonen i TPACK rameverket beskrives dermed med en lengre definisjon. I Kunnskapen av teknologi inngår forståelsen av teknologi bredt nok til å implementere det produktivt i jobb/undervisning, bruke teknologien hensiktsmessig til å forenkle oppgaver og å kontinuerlig adaptere nye metoder (Koehler & Mishra 2009, s. 64). Kunnskapen kan bli utdatert, og må stadig videreutvikles for å opprettholdes. Likevel kan visse tankesett og teknologiske metoder bli videreført til nye teknologiske verktøy (Koehler, Mishra & Cain, 2013).

### 2.1.4 Pedagogisk faginnholdskunnskap (PCK)

Pedagogisk faginnholdskunnskap ble introdusert av Shulman (1986, 1987) der ideen om pedagogisk kunnskap utnyttes for å dele den faginnholdsspesifikke kunnskapen. Læreren bruker sin pedagogiske kunnskap til å skreddersy presentasjonsmetoder, materiale og alternative konsepter tilpasset elevenes læringsvilje og tidligere kunnskaper (Koehler, Mishra & Cain, 2013). Kunnskapen bygger med andre ord på lærerens forståelse av fagets representasjoner, og hvordan kunnskapene kan brukes for å fremme elevens læring (Cox & Graham, 2009). Magnusson et al. (1999) deler de fagspesifikke pedagogiske kunnskapene nevnt under pedagogisk kunnskap (PK), inn i to ulike kategorier: "Subject-specific" og "topic-specific" strategier. De fagspesifikke, oversatt fra "subject", aktivitetene er arsenalet av aktiviteter læreren har som kan benyttes i forskjellige temaer innen faget. Temaspesifikke, oversatt fra "topic-specific", aktiviteter derimot er spesifikke aktiviteter som knyttes til et spesifikt tema. Dette inkluderer forståelsen av at representasjoner kan brukes til hjelp for elevene i temaet, for eksempel at en kurve kan hjelpe elever å se konseptet av flere søyler i diagram (Cox & Graham, 2009).



### 2.1.5 Teknologisk pedagogisk kunnskap (TPK)

Teknologisk pedagogisk kunnskap er lærerens forståelse til å bruke riktig teknologi til riktig tid. Det å mestre teknologisk pedagogisk kunnskap vil si å kunne utnytte når et teknologisk verktøy kan forandre læringen og undervisningen til det bedre, samt se begrensningene og fordelene ved metodene eller aktivitetene (Koehler & Mishra, 2009). TPK inneholder også bruken av teknologi for å fremme motivasjon hos elevene. (Cox & Graham, 2009).

### 2.1.6 Teknologisk faginnholdskunnskap (TCK)

Teknologisk faginnholdskunnskap er en forståelse av begrensningen og mulighetene teknologi og faginnhold har for hverandre. I tillegg til å forstå faget i seg selv, må læreren også forstå hvordan faget kan endres ved innføringen av diverse teknologiske verktøy. Eksempelvis har flere felt som medisin, historie og fysikk blitt fremskredet ved hjelp av nye teknologier som representerer eller manipulerer data på en ny og fruktbar måte (Koehler & Mishra, 2009). Teknologisk faginnholdskunnskap for lærere består oftest og hovedsakelig av temaspesifikke representasjoner, og selv om representasjoner ikke fullt beskriver toveisforholdet mellom faginnhold og teknologi understreker det viktigheten av kunnskapen om hvilken teknologi som representerer konsepter korrekt og hensiktsmessig (Cox & Graham, 2009).

## 2.2 TPACK i matematikk

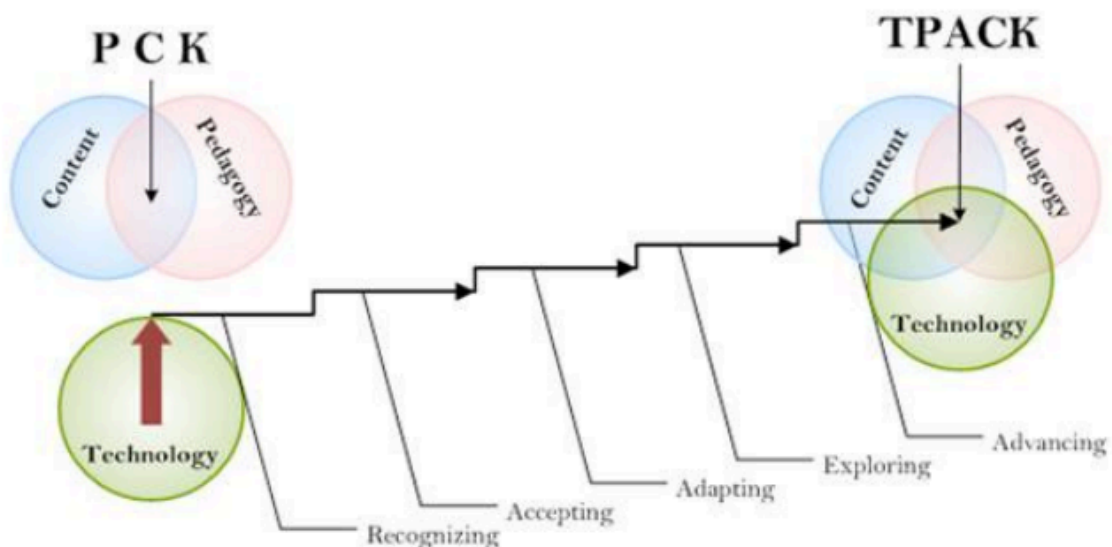
Etter hvert som teknologi og samfunnet utviklet seg, og teknologien ble mer tilgjengelig, ble lærere utfordret til å inkorporere teknologiske verktøy i matematikkundervisningen. I 2000 kom *International Society for Technology* ut med den amerikanske boken *National education technology standards for students*, med forslag og mål for effektiv bruk av teknologi i skolen (Niess et al., 2009). Regjeringen fulgte etter og har i nyere tid også inkludert programmering som en del av matematikkfaget i læreplanmålene. Dette blir bedre beskrevet i 2.3.

Programmering i skolen.

Niess, Sandri og Lee (2007) dannet en modell for å gå fra pedagogisk faginnholdskunnskap til TPACK. Modellen er basert på en 5-steps prosess som først ble introdusert av Everett Rogers (1995) om å adaptere en ny invensjon. Stegene i modellen ble så omdannet til å passe teknologisk integrering i klasserommet (Niess et al., 2009).

Niess et al. (2009) gjengir de fem stegene som: *recognizing*, *accepting*, *adapting*, *exploring* og *advancing* (se figur 2). Beskrivelsene vi presenterer er hentet fra samme artikkel, med egen oversettelse av ordene.

1. *Gjenkjennelse (Kunnskap)*: Lærerne har kunnskap nok til å bruke teknologien. De forstår sammenhengen mellom teknologien og matematikk, men bruker ikke teknologien til et verktøy i undervisning eller læring av matematikk.
2. *Akseptere (Overbevisning)*: I denne fasen skaper lærere en selvstendig mening om teknologien er gunstig eller ugunstig i undervisning av matematikk med teknologi som verktøy.
3. *Tilpasse (Bestemmelse)*: Det tredje steget består av å engasjere seg i aktiviteter som leder til en bestemmelse om å tilpasse matematikkundervisningen til den gitte teknologien.
4. *Utforske (Implementering)*: Her bruker lærerne teknologi i undervisningen eller læringen, i håp om at elevene finner undervisningen meningsfull.
5. *Fremskritt*: Lærerne tar tak i matematikkundervisningen og evaluerer resultatene. Evalueringen avgjør om den teknologiske metoden var tilfredsstillende, eller om det finnes en mer passende teknologi eller metode.



**Figur 2:** Integrering av teknologi i PCK (Niess et al. 2009)

## 2.3 Programmering i skolen

I dag henter elevene mye av sin informasjon og lærdom fra digitale virkemidler. På skolen skal elever utvikle egne ferdigheter til å referere til korrekte kilder, være kildekritiske, samhandle med egne og andres etiske holdninger og meninger og reflektere over dette. Som lærer har man en stor innflytelse innen disse ferdighetene, og må selv kunne utvikle egen digital kompetanse for å korrekt lære elevene. (Kelentrić et al., 2017).

Programmeringsbegrepet inkluderer forskjellige arbeidsmåter, konsepter og innhold på tvers av undervisningsfag. Bruken av begrepet varierer i forhold til fagfelt og fag, og kan inkludere ulike meninger (Forsström & Kaufmann, 2019). Videre beskrives programmering som “prosessen relatert til utvikling og implementeringen av instruksjoner for dataprogrammer så datamaskinen kan utføre spesifikke oppgaver, løse problemer og støtte menneskelige interaksjoner”. For at elevene skal kunne programmere må de derfor ha kunnskap om vurdering, utforming, implementering og analysering av algoritmer i programmet (Forsström & Kaufmann, 2019, s. 19).

I utbedringen av fagfornyelsen betegnet Sanne-utvalget (2016, s. 25) programmering som “en tilegnelse av algoritmisk tenkning som har likhetspunkter med matematisk logikk” og “en måte å forstå de grunnleggende mekanismene i digital teknologi”. Med andre ord utvider de programmering fra utforming av program, eller koder, til å inkludere forståelse og algoritmisk tenkning.

Sanne-utvalget ble ansatt til å konsultere Utdanningsdirektoratet i planleggingen av teknologisk undervisning i skolen. Teknologi i skolen, og dermed programmering, ble foreslått av Sanne-utvalget (2016) som et eget fag i deres rapport. Dette var for å gi elevene et relevant grunnlag innen teknologi som gjenspeiler det 21. århundres samfunn. Det står også at programmering kan innføres som en del av de allerede etablerte fagene, med forutsetningen at elevene allerede har fått en praktisk innføring i programmering. I rapporten ble det beskrevet at innføringen av programmering krevde lærerkompetanse, dedikerte ressurser og engasjement fra skoleledere (Sanne et al., 2016). Utdanningsdirektoratet valgte å ilegge programmering blant annet under matematikk, og ikke et eget fag, noe som betyr at matematikklærere må oppfylle kravene om programmeringskompetanse. I læreplanen ble det inkludert et kompetansemål per trinn fra 5-10 trinn som inkluderte programmering. I 8. trinn er det generelle programmeringsmål som kunnskap om vilkår, løkker og variabler. I de andre

trinnene er programmeringen knyttet opp til matematiske temaer som simulere tilfeldige forsøk og beregne sannsynlighet (9. trinn) og utforske matematiske egenskaper og sammenhenger (10. trinn) (Kunnskapsdepartementet, 2019). I denne studien vil programmering være et samlebegrep tilnærmet det Sanne et al. (2016) beskriver.

## 2.4 Computational thinking og algoritmisk tenking

Nouri et al. (2019) presenterer Computational thinking, eller CT, som et begrep først tatt i bruk på 80-tallet av Seymour Papert og omhandlet samspillet mellom programmering og thinking skills. I nyere tid var det Jeanette Wing som gjeninnførte begrepet i 2006 og ga det en ny definisjon. Denne definisjonen handlet om å løse problemer og konseptualisere systemer basert på grunnleggende datakunnskap. Definisjonen fikk en ny oppfriskning av Cuny, Snyder og Wing (2010). CT ble da sett på som prosessen ved å angripe et problem hvor man formulerer konkrete løsninger på mindre deloppgaver som løser det større problemet. Disse deloppgavene skal enkelt kunne gjennomføres av maskiner (Nouri et al., 2019).

Det er gjort flere forsøk på å oversette begrepet CT til norsk. Noen sentrale begreper for å forklare CT er ifølge Wing (2011) abstraksjon, generalisering og algoritmisk tenkning. Likevel brukes ofte algoritmisk thinking som en oversettelse av CT begrepet i mangel på et likeverdig norsk begrep (Kaufman & Stenseth, 2023). Gjennom fagfornyelsen har både algoritmisk tenking og programmering blitt sentrale begreper i matematikkundervisningen. Da vil det være nyttig med en avklaring om begrepene koding og programmering, samt hvordan disse knyttes til CT og algoritmisk tenkning. Kaufmann og Stenseth (2023) sier at prosessen hvor man vurderer et problem og dets løsninger, for å realisere denne løsningen, er programmering. Når man skal implementere løsningen i et programmeringsspråk, enten med tekst, blokker eller annet, heter det koding.

### **Integrering av programmering og algoritmisk tenking i matematikkundervisning**

Tidligere har vi sett på noen av utfordringene som har oppstått ved å integrere programmering i undervisningen. Vi kommer derfor til å se på måter integrering av programmering og CT eller algoritmisk tenkning kan berike matematikkundervisningen. Sammenhengen mellom CT og matematikk har vært argumentet for å tilføre programmering som pensum i matematikkfaget i de nordiske landene (Bocconi et al., 2018 sitert i Kaufmann

& Stenseth, 2021) For at CT skal kunne anvendes i matematisk problemløsning kreves en kombinasjon av flere forkunnskaper. Blant dem matematisk tenking, kjennskap til programmering og validering av programmet og dets utfall. For å kunne dele opp matematikkoppgaver i mindre deler og utarbeide algoritmer før disse løsningene implementeres gjennom programmering, trengs også tilstrekkelig matematisk kunnskap (Hadjerrouit & Hansen, 2022). Videre poengterer de at den pedagogiske situasjonen er avgjørende. Oppgavene som blir gitt bør oppleves motiverende og på mottakerens nivå. God integrering av CT bør fremme autonomi og eierskap til arbeidet. Hadjerrouit og Hansen (2022) påpeker at dette er vanskelig å oppnå hos uerfarne universitetsstudenter uten tilstrekkelige forkunnskaper.

I Kaufmann og Stenseth (2023) reflekterer de over erfaringer de har gjort seg om ulike måter å integrere programmering og problemløsning i matematikkfaget. De ser også på hvilken rolle læreren bør innta for å gjøre dette best mulig. Kaufmann og Stenseth (2023) konkluderer med at lærerrollen bør være tydelig, og at læreren burde hjelpe elevene med å definere problemer, delmål og vurdere hvorvidt de gradvis har blitt oppfylt. De påpeker også at læreren må unngå å gå i fellen med å bli usynlig når elevene jobber i grupper. I tillegg at læreren planlegger hvilke oppgaver elevene skal løse, hvem som skal jobbe sammen og motivere bør læreren også hjelpe elever med å bruke terminologi som er relevant for det aktuelle temaet, trekke de lange linjene mellom oppgaver og temaer, overse at arbeidsmetodene er hensiktsmessige, og vurdere progresjonen og produksjonen av koden til elevene (Kaufmann & Stenseth, 2023). Dette samsvarer med Hadjerrouit og Hansen (2022) sin beskrivelse av lærerrollen i situasjoner hvor CT og programmering blir brukt. Læreren skal fasilitere læring og være med å trekke linjene mellom matematikk, CT og programmering.

Renate Andersen gjennomførte et forskningsprosjekt i samarbeid med ProSkap (PROgrammering og SKAPerverksted i skolen), hvor hun hentet data fra elever i 12-16 års alderen. Elevene løste matematikkoppgaver de fikk av læreren sin med hjelp av blokkbasert programmering, tre timer i uken over 16 uker fordelt på to intervensjoner i semesteret. Konklusjonene på studien var at det å integrere blokkbasert programmering med matematikk gjorde at elevene tilegnet seg kunnskap aktivt og gjennom fellesskap. Det å slå sammen verktøyet og faget fikk elevene utviklet sine CT ferdigheter. Og til slutt at det å arbeide med blokkbasert programmering fremmet læring i matematikk (Andersen, 2022).

## 3 Metode

I dette kapitlet vil vi gi en beskrivelse av vår studie, Det vil innebære å beskrive vår tilnærming til forskningen og designet av studien. Vi kommer også til å legge frem vårt datagrunnlag og hvilke metoder vi valgte å benytte for å samle inn data, samt hva disse metodene innebærer. Videre kommer vi til å beskrive hvordan vi behandlet dataene og hvilke prosesser som har inngått. Til slutt vil vi gi en oversikt over etiske betraktninger gjort og grep tatt for å sikre forskningskvalitet.

I denne studien ønsker vi å se hvordan lærere påstår at de bruker programmering som verktøy i matematikkundervisningen. Vi har undersøkt dette gjennom metoden multiple-case studie der vi har fulgt tre lærere plassert på to forskjellige ungdomsskoler. Undersøkelsen baseres på Mishra og Koehler (2006) sitt TPACK rammeverk, som vi også har brukt som analysemetode. I dette kapitlet redegjør vi de metodiske tilnærmingene vi har brukt for å svare på egen problemstilling og forskningsspørsmål. Vi beskriver først forskningsparadigme og forskningsstrategi brukt i undersøkelsen. Siden våre spørsmål baserer seg på læreres erfaringer og tanker, falt det naturlig å bruke et kvalitativt design og intervju og observere lærere. Vi kommer da med en beskrivelse av forskningsdesignet. Videre beskriver vi bruken av intervju og observasjon, valg og begrunnelse for innhenting av informasjon, analyse og behandling av dataene og hvordan metodene passer sammen med problemstillingene. Til slutt reflekterer og vurderer vi de etiske konsekvensene av vår studie, samt dens reliabilitet og validitet.

### 3.1 Forskningsparadigme

I oppgaven er det viktig å være transparent med hvordan dataen tolkes, og hvor pålitelig dataene er. Vår tolkning baserer seg på det Johannessen, Tufte og Christoffersen (2016) kaller et fenomenologisk paradigme. Fenomenologi er et studie av et fenomen, der forskere skal undersøke hvordan andre oppfatter fenomenet. Målet er å beskrive objekters egne perspektiver, opplevelse og forståelse presist (Johannessen et al., 2016).

Som forskere er det viktig å artikulere korrekt hvordan vi oppnår kunnskapen. Et valg av forskningsparadigme bør ikke være et påtvunget til å følge et spesifikt filosofisk syn av personlig interesse, men heller tilpasses til kravet og nødvendigheten til forskningsstudiet.

Her kommer vi til å beskrive vår tilnærming til fortolkning av dataene med hensyn til ontologi og epistemologi, og anerkjennelsen av kompleksiteten av fenomenets natur (Rehman & Alharthi, 2016).

Ontologien beskriver hvordan vi som forskere anser realiteten. I konstruksjonismen er det enighet om at det finnes sosialt konstruerte realiteter, der flere virkeligheter kan påvirke hverandre og synet på virkeligheten kan forandres gjennom lærdom. I vår oppgave mener vi det ikke finnes en entydig realitet, og vi forsøker å finne flere sosialt lagde meninger og kunnskaper (Mertens, D.M., 2019).

I epistemologien har oppgaven vår en interpretivistisk tilnærming. Med andre ord påvirker intervjuer og intervjuobjekter hverandre. Dataen, tolkningen og resultatet baserer seg på kontekst, og det teoretiske rammeverket TPACK. Det er nødvendig at vi holder oss så nøytrale som mulig så verken data eller analyse blir påvirket av egen mening eller tro (Mertens, D.M., 2019).

Mesteparten av dataen som er innhentet er tvetydig, og krever en grad av tolkning. Vår vurdering av meningen bak hva som blir sagt i intervjuene spiller da en rolle i hvordan dataen blir analysert og utfallet av resultatet, og det er ikke mulig å konstruere en entydig mening. Med det sagt har vi en forpliktelse til å holde oss så nøytral som mulig.

### 3.2 Forskningsstrategi: Kvalitativt studie

Forskningstilnærmingen vi har valgt baserer seg på en kvalitativ studie. Tilnærmingen er basert på en todelt innhenting av informasjon, der vi har intervjuet lærere, og observert en undervisningsøkt. Dette blir spesifisert i delkapittelet om datainnsamling. Den kvalitative studien bidrar til en muligens mer detaljert og utfyllende informasjon fra deltakerne. I tillegg skaper det en mer flytende og spontan data, og deltar til å forstå hvordan skolen og lærere vil operere med digital teknologi i undervisningstimene (Christoffersen & Johannessen, 2012). Ettersom vi ønsker å lære om lærernes tanker og ideer rundt programmering som verktøy, har vi konkludert at kvalitativ forskning er det mest hensiktsmessige i denne oppgaven. Dette begrunner vi også med Bryman (2016) som tilsier at kvalitativ forskning kan gi en mer konkret forklaring på hva, hvorfor og hvordan lærere tenker. Dette kan igjen føre til at vi får en helhetlig og dyp forståelse av informantenes tolkning av erfaringer.

### 3.3 Forskningsdesign: Case-studie

Case-studier har som fellesnevner at de undersøker en “case”, en enkel enhet som skole, gruppe mennesker, person eller samfunn, som er kontekst spesifikk og avgrenset både i tid og sted. Selve casen kan variere i alt fra en elev, til det norske skolesystemet og varierer etter hvilket nivå man ønsker å analysere på (Postholm & Jacobsen, 2018). Målet med studien er å få en dypere oppfatning av enheten, der man undersøker bestemte deler av enhetens egenskaper og karakter (Bryman, 2016).

Denne oppgaven baserer seg på tre forskjellige caser, som også omtales som multiple case-studier. Målet med undersøkelsen er å undersøke læreres tilnærming til programmering som verktøy i matematikkundervisning. I tillegg går forskningen ut på å forstå lærernes egne erfaringer innen kunnskapene. For at en studie skal kvalifiseres som case-studie må konteksten være av spesifikk betydning, mer generelle undersøkelser kvalifiserer derfor ikke som case-studier. (Postholm & Jacobsen, 2018).

### 3.3.1 Multiple case-studie – Flercasestudie (komparative casestudier)

I tilfeller hvor man ønsker å undersøke flere caser vil man benytte en multiple case-studie. Det vil da være naturlig å se på kontekstene i hvert enkelt case for så å sammenligne dem. Multiple case-studie er en samleparaply på en forskningsmetoder som består av to eller flere case-studier, hvor både den enkelte enheten behandles som sin egen case og casene sammenlignes for å finne fellesnevner og ulikheter (Stake, 2006). Hensikten med å ha med flere caser må drøftes av forskeren for å gi et overbevisende svar på hvorfor det er av hensikt. Dette er grunnlagt av at alle forskningsprosjekter er tids- eller ressursbegrenset. I casestudier gjelder det å gå i dybden for å undersøke den spesifikke konteksten, dermed betraktes det å se på flere caser som en nedprioritering av dybde for å ha med større bredde. Komparative case-studier bør dermed ha et smalere fokus, for å beholde dybden samtidig som man har noe bredde (Postholm & Jacobsen, 2018).

### 3.3.2 Utvalg av deltagere

Første runde med å skaffe informanter til masterprosjektet startet med å søke gjennom LU. I søknaden ønsket vi oss lærere som underviste i matematikk på ungdomsskolen, og som brukte programmering i undervisningen. Vi skrev også at vi helst ville ha lærere som jobbet på 9. trinn. Dette mest for å spisse søket mest mulig, og for å skåne 10. klasse som skal ha eksamen. Valget med å prioritere 9. klasse framfor 8. kom som resultat av en hypotese hvor



programmering var i større grad integrert i matematikkundervisningen på 9. trinn enn 8. da de har et ekstra år på skolen. Fra LU fikk vi som svar først at de ikke hadde funnet noen informanter til oss. Neste steg i letingen etter informanter var å komme i dialog med vår veileder og andre på fakultetet. Der fikk vi noen ledetråder om noen vi kunne spørre om å delta som informanter. Herfra fikk vi to informanter (Lærer 1 og Lærer 2), der begge lærerne jobbet på samme skole (Skole 1). Dette er noe vi tar betraktning når vi sammenligner praksis og oppfatninger. Spesielt med tanke på at disse lærerne jobbet tett på 9. trinn på skole 1, hvor lærer 1 også hadde deler av sin stilling som inspektør på skolen. Søket stoppet opp på dette stadiet en periode, frem til vi ble enige om at vi var nødt til å utvide søket vårt for å få et bredere grunnlag. Derfra utvidet vi søket til kontakter vi har i skoler rundt om for å høre om det fantes matematikklærere som jobbet på ungdomsskoler som hadde mulighet til å delta. Derav fikk vi svar fra lærer 3, skole 2, som jobbet på 8. trinn.

Etter vi hadde innhentet dataen fra de tre enhetene, gikk vi gjennom mengden informasjon vi fikk. Med et ekstra fokus på mangel av tid og frivillige informanter kom vi frem til at den innsamlede dataen var tilstrekkelig nok til å avslutte innhenting av informasjon. Videre konsulterte vi med veileder, og gjennomgikk den innsamlede dataen, og bekreftet at grunnlaget var godt nok til å starte på analyse av innhold.

## 3.4 Datainnsamling

### 3.4.1 Intervju

Intervju er en datainnsamlingsmetode som kan gjøres på ulike måter. I motsetning til hvordan en samtale flyter i det daglige, beskriver Postholm og Jacobsen (2018) at intervjuet bør ta utgangspunkt i problemstilling og forskningsspørsmål. Dermed kan forskeren holde seg til tema og spørsmålene er med på å gi konkretiserte svar til studien. Vi har gjennomført individuelle, semistrukturerte intervjuer med lærere. Ved å gjennomføre et semistrukturert intervju har vi et utgangspunkt med spørsmål som er forankret i TPACK vi ønsker å stille, samtidig som det er rom for å stille oppfølgingsspørsmål for å grave videre eller tilstrekkelig forstå konteksten læreren presenterer (Postholm & Jacobsen, 2018). I gjennomførelsen av et multiple case-studie er det en stor fordel og anbefaling å ha en grad av struktur i spørsmålene for å kunne sammenligne svarene som blir oppgitt (Bryman, 2016).

For å sikre at dataene ble innhentet etter retningslinjene for personvern, brukte vi en kryptert diktafonapplikasjon med backup, som senere ble transkribert så konkret som mulig. Applikasjonen lagret opptakene rett inn på nettskjema hvor kun vi studenter og veileder hadde tilgang, uten å legge igjen spor lokalt på enhetene.

### **3.4.1.1 Intervjuguide**

I alle intervjuene vi hadde fulgte vi den samme intervjuguiden (se vedlegg 2). Intervjuguiden ble laget på bakgrunn av hvordan vi kunne innhente informasjon om lærernes tanker rundt planlegging og gjennomføring av matematikkundervisning med programmering som hjelpemiddel. Vi fokuserte spørsmålene inn mot tilnærmingen til TPACK hovedbegrepene, samt utfordringene og mulighetene de opplevde ved de forskjellige kunnskapene. Ved utbedringen av intervjuguiden fokuserte vi på spørsmål som omfattet intervjuobjektets personlige mening. Likevel inkluderte vi noen spørsmål som omhandler hvordan intervjuobjektets erfaringer er i forhold til andre læreres tanker.

Under forskningskvalitet påstår Postholm og Jacobsen (2018) at man bør unngå formuleringer i spørsmål som kan oppleves ledende eller som kan resultere i tvetydighet i svarene. Derfor forsøkte vi, så langt det lot seg gjøre, å unngå dette i utbedringen av intervjuguiden. For å sikre oss et datagrunnlag som gjorde oss i stand til å besvare forskningsspørsmålene våre, tok vi hensyn til de tre spørsmålstypene som blir nevnt om semistrukturerte intervjuer i Postholm og Jacobsen (2018) . Oppfølgingsspørsmål for å gi forklaring for nye tematikker som kunne oppstå i intervjuet. Inngående spørsmål for å holde informanten på rett spor, slik at vi fikk data som omhandlet tematikken vår studie gikk ut på. Til slutt også oppklarings spørsmål til der det eventuelt oppstod situasjoner hvor informantene hadde svart tvetydig.

### **3.4.2 Observasjon**

For å muliggjøre en triangulering av dataen vi innhenter, valgte vi å gjennomføre en observasjon av alle informantene. Observasjonene i klasserommet ga oss direkte tilgang til informasjonen vi ønsket å innhente: Hvordan det tekniske aspektet utspilte seg og ble organisert sammen med faget og pedagogikken (Christoffersen & Johannessen, 2012). Observasjon handler om å bruke sansene til å oppfatte omgivelsene. I forskningssammenheng vil det være nyttig å ha et spisset fokus på hva man vil observere for å innhente data som vil

kunne svare på forskningsspørsmålene (Postholm & Jacobsen, 2018). For å oppnå denne spissingen tok vi utgangspunkt i et observasjonsskjema vi forberedte i forkant, som gjorde det enklere å klassifisere de ulike observasjonene i kategoriene i TPACK rammeverket.

Postholm og Jacobsen (2018) skriver om de fire ulike observatørrollene, originalt beskrevet av Gold (1958). Rollene plasseres etter to kriterier, forskerens deltakelse og avstand. Deltakelsen beskriver graden av interaksjon observatøren har med aktiviteten som foregår. Avstanden gir uttrykk for grad av tilknytning til observasjonssituasjonen. I tabellen (Tabell 1) vises de fire forskjellige rollene man kan innta: fullstendig observatør, observatør som deltaker, deltaker som observatør og fullstendig deltaker. I vår studie bestemte vi i forkant at begge skulle gå inn i rollen som fullstendig observatør. Dette betyr at vi ikke hadde noen tilknytning til situasjonen som ble observert, og heller ikke noen samhandling. (Postholm & Jacobsen, 2018).

		Forskerens deltakelse	
		Liten	Stor
Forskerens avstand	Liten	Deltaker-som-observatør	Fullstendig deltaker
	Stor	Fullstendig observatør	Observatør-som-deltaker

**Tabell 1:** Ulike observatørroller (Postholm & Jacobsen, 2018, s.115)

### 3.4.2.1 Observasjonsskjema

I forkant av observasjonene utarbeidet vi et observasjonsskjema (se vedlegg 3). Dette skjemaet hadde som hensikt å være en hjelp for oss som forskere til å sortere våre observasjoner underveis i prosessen. Skjemaet bestod av felter til å notere generelle observasjoner, observasjoner knyttet til hver av kategoriene i TPACK rammeverket, og spørsmål som kunne bli tatt opp i intervjuet.

## 3.5 Metode for dataanalyse

Når man analyserer en case-studie, vil man gjøre mening av dataene og sette casen i en kontekst. Dermed kan leseren gi mening til casen ved å sette den i et større perspektiv blant

andre lignende caser som er studert (Stake, 1995 sitert i Postholm & Jacobsen, 2018). I vår analyse har vi først gjort en analyse av lærerne hver for seg. Som nevnt er multiple-case basert på flere enkeltcaser, og vi representerer dermed også lærerne i hver sin case. Analysen inneholder en tredelt analyse, hvor vi først tar for oss intervjuet. For å best mulig skille informasjonen er intervjuanalysen strukturert med: utdanning & erfaring, TPACK analyse, forklart i 3.4.1, og påstått kunnskap om teknologi, pedagogikk og faginnhold, og anvendelse av kunnskapen. Videre analyserer vi observasjonene gjort i undervisningsøkten. Og til slutt oppsummerer vi funnene, og triangulerer funnene i intervju og resultat. I neste delkapittel presenterer vi tilnærminger vi har valgt til analysering av observasjoner og intervjuer.

### 3.5.1 Induktiv- Deduktiv tilnærming

Vi ønsker å forstå hver lærers oppfattelse av det å integrere programmering i matematikkundervisningen. Derav er konteksten hvor hver enkelt lærer befinner seg av betydning. Denne tilnærmingen å jobbe på beskrives av noen som induktiv - deduktiv. Vi starter med et utgangspunkt, TPACK rammeverket, forankret i vår teori. Deretter utformer vi intervjuguide og observasjonsguide, og tar disse i bruk under datainnsamling. I analysene kan det da oppstå nye empiriske data. Herfra prøver vi å knytte disse nye dataene induktivt til TPACK (Bryman, 2016)

### 3.5.2 TPACK analyseverktøy

Som nevnt i introduksjonen av kapittel 3, baserer vi analysen vår på Mishra og Koehler (2006) sitt TPACK rammeverk. Dette bruker vi for å få oversikt over lærernes påståtte kunnskaper og bruk innen pedagogikk, fagkunnskap og teknologi. Vi identifiserer alle deler av TPACK, forklart i kapittel 2.1 TPACK. Vi har laget et analyseskjema på bakgrunn av TPACK rammeverket, hvor vi i tillegg har plassert to forskjellige underkategorier innen hver TPACK kategori:

- Kan: Underkategorien bruker vi der en lærer omtaler egen eller andres forståelse eller kunnskap av en kategori, eksempelvis teoretisk, pedagogisk faginnhold eller lignende. Dersom intervjuobjektet omtaler bruk av en læringsmetode med programmering, koder vi dette som teknologisk pedagogisk kunnskap i "kan"-kategorien. Koden vil også falle innenfor "kan" underkategorien dersom læreren beskriver kurs innen en kategori, eller positiv omtale av egen kunnskap.

- Kan ikke: Underkategorien brukes der læreren omtaler manglende forståelse eller vanskeligheter rundt en kategori. Eksempelvis hvis læreren beskriver vanskeligheter med å finne undervisningsmetoder til statistikk med programmering som hjelpemiddel. Det vil kodes innenfor kategorien teknologisk pedagogisk kunnskap og underkategorien “kan ikke”. Uttaler om manglende kunnskap, ønsket om mer kunnskap eller lignende vil også kodes som “kan ikke” underkategorien.

Når kodingen brukes som “kan” eller “kan ikke”, inkluderer dette også når lærer reflekterer over interaksjon med andre læreres uttalte kunnskap, tilfredshet ved læreplanmål og erfaringer med medarbeidere og skoleorganisasjon. Vi begrunner dette med at lærerens kunnskaper innen teknologi, pedagogikk og faginnhold i stor grad avhenger av medarbeidere, skolekultur og retningslinjer fra Kunnskapsdepartementet. Dersom lærer har uttrykt misnøye rundt et av disse punktene, anser vi det som vanskeligheter rundt temaet. Det er også viktig å bemerke at underkategorien “kan” inkluderer alt fra “kan noe” til “kan mye eller alt” og “kan ikke” inkluderer alt fra “kan lite” til “kan ingenting”.

Selv om vi har en kvalitativ undersøkelse, valgte vi også å presentere dataene vi fant i en empirisk tabell (Se vedlegg 6). Dataene beskriver kun hvor mange ganger de forskjellige kategoriene ble nevnt i løpet av intervjuet. Dette er for å få en større oversikt over fokusområdet til intervjuobjektet gjennom intervjuet. Dataen vil ikke være nok i seg selv til å generalisere fullstendig, men gir en indikasjon på fokusområder og gjør det lettere å sammenligne de forskjellige lærernes svar. Det er likevel viktig å nevne at tabellen ikke differensierer på graden av benevnelsene.

<b>Komponent</b>	<b>Kjennetegn</b>	<b>Kan ikke</b>	<b>Kan</b>
<b>1. Teknologisk kunnskap (TK)</b>	Omtale om forståelsen av programmer eller utnyttelse av digitale hjelpemidler for å fremme effektivitet eller skape nye representasjoner.		
<b>2. Pedagogisk kunnskap (PK)</b>	Omtale om eller utnyttelse av metoder eller aktiviteter for å fremme elevers læring.		
<b>3.Faginnholds-kunnskap (CK)</b>	Omtale om eller utnyttelse av matematiske formler, betegnelser eller læreplanmål.		
<b>4. Teknologisk pedagogisk kunnskap (TPK)</b>	Omtale eller utnyttelse der teknologisk kunnskap knyttes til metode eller aktivitet for å fremme elevers læring.		
<b>5. Teknologisk faginnholds-kunnskap (TCK)</b>	Omtale eller utnyttelse der teknologisk brukes for å forbedre eller forenkle matematiske formler, strategier, betegnelser eller læreplanmål.  Det kan også være omtale av nye representasjoner teknologi kan gi.		

<b>6. Pedagogisk faginnholds-kunnskap (PCK)</b>	Omtale om eller utnyttelse av metoder eller aktiviteter som fremmer elevers forståelse av matematiske formler, strategier, betegnelser eller læreplanmål		
<b>7. TPACK</b>	Omtaler teknologiske metoder eller aktiviteter som fremmer elevers forståelse av matematiske formler, strategier, betegnelser eller læreplanmål		

**Tabell 2** TPACK analyseverktøy

For å kode de forskjellige kunnskapene i TPACK brukte vi tallene 1-7 som presentert i tabellen over, samt bokstaven “a” for “kan ikke” underkategorien, og “c” for underkategorien “kan”. Etter at kodingen var ferdig, oppførte vi de aktuelle utsagnene i tabellen.

### 3.5.3 Tematisk analyse

Når vi tok for oss datamaterialet ønsket vi at dette skulle gjøres på en systematisk og oversiktlig måte slik at det i etterkant vil være tydelig hva som er gjort og det vil være mulig å etterprøve analysene. Vi tok utgangspunkt i Braun og Clarke (2006) sin fremgangsmåte for tematisk analyse når vi videreutviklet et analyseverktøy som passer til vårt formål. Videre vil vi presentere noen av de ulike stegene i Braun og Clarke (2006), samt komme med en mer nøyaktig beskrivelse av prosessen vi gjennomgikk for å utvikle et rammeverk som passer for vår studie. Etersom vi allerede har et utarbeidet analyseverktøy fra Mishra og Koehler (2006), bruker vi ikke stegene 4 og 5, og kommer derfor ikke til å forklare stegene.

#### 3.5.3.1 Utvikling av analyseverktøy

1. Bli kjent med datamaterialet

De verbale delene av datamaterialet må transkriberes for å kunne utføre tematisk analyse. Arbeidet med transkripsjonen er ingen bortkastet tid likevel, arbeidet man legger ned er med på å forberede enn på analysen, samt at man får en god oversikt over materialet man senere

skal analysere (Braun & Clarke, 2006). Transkripsjonen ble gjennomført ut fra lydopptakene vi har gjort. Transkripsjonen ble gjort så ordrett som mulig, der alle ord med unntak av “hmm” og “umm” er inkludert. Tidsbruken på svar er notert ned, men ettersom vi ikke hadde fokus på hvor lang tid enhetene brukte på å svare, er dette ikke relevant for oppgaven.

For vår del har begge studentene vært til stede under all observasjon, begge har tilgang på lydopptak og begge har transkribert, og hatt tilgang til disse transkripsjonene. For å vite hva man skal analysere må man sette seg inn i datamaterialet. Det innebærer at vi leste gjennom gjentatte ganger, for å se etter mønstre og gjentatte fenomener. Dette for å sikre seg en god forståelse av dybden og bredden i datamaterialet (Braun & Clarke, 2006).

## 2. Utvikle initielle koder

Etter at datamaterialet er gjennomgått og gjennomlest, har man fått en viss oversikt over hvilke elementer som er ønskelig å se videre etter. Man kan da begynne å skape seg noen koder å fordele korte tekstutklipp i. Dette vil være første runde med koding. Hvilke koder du bestemmer deg for kan variere ut ifra om de baserer seg på data eller teori (Braun & Clarke, 2006). I vår studie ønsker vi å se på datamaterialet gjennom TPACK rammeverket, og det er derfor naturlig å basere disse kodene på de ulike kunnskapene i rammeverket. Likevel påpeker Braun og Clarke (2006) at man heller bør ha for mange koder enn for få. Ved å ha med flere koder får man plukket opp sidespor og detaljer som ellers kan glippe. I tillegg har man ikke et totalt omfang om hva som kan være interessant å kikke på senere. Vi utviklet her tre midlertidige underkategorier vi omtalte som “negativ omtale”, “nøytral omtale” og “positiv omtale”. Disse ble lagt til i hver av de 7 kategoriene av TPACK. Vi kodet etter underkategoriene “negativ omtale”, “nøytral omtale” og “positiv omtale” gjennom et intervju, før vi oppdaget en misrepresentasjon av dataen vi hadde innsamlet. Vi begynte da arbeidet med å finne mer tilpassede kategorier til vår undersøkelse. Etter veiledning og samarbeid med medstudenter kom vi frem til underkategoriene “kan” og “kan ikke”. Disse passet godt inn med hva vi ønsket å kartlegge, og dermed ble disse definert og brukt i analysen.

Kodingen ble gjort på nytt, der begge studenter kodet hver for seg. Vi sammenlignet funnene våre, og kom frem til en felles avgjørelse på kodingen. Derfra gjorde vi hvert vårt intervju, ettersom vi vurderte at kodingen vår var lik nok til å sammenlignes.

## 3. Se etter kategorier



Som nevnt begynte vi å se etter kan og kan ikke i kodingen. Dette valgte vi å kode i Excel, der vi startet med en tallanalyse av frekvensen av benevninger som ble gjort innen de forskjellige kategoriene i tpack. I analysen tok vi videre utdrag av de forskjellige settingene vi anså som viktige for beskrivelsen. Vi satt opp to kategorier innen temaet programmering som verktøy, med kategoriene “kan” og “kan ikke”. Dette ble gjort i Word.

Braun og Clarke (2006) beskriver at i denne delen av arbeidet kan kategoriene fint ha underkategorier, og det er helt i orden å ha noen koder som ikke faller inn under en konkret kategori på dette tidspunktet som blir plassert under diverse. Disse kodene vil finne sin plass senere i form av at temaene justeres eller at noe sløyfes.

## 6. Produsere rapporten

Den siste fasen starter med ferdigstilte kategorier som analysen utføres gjennom, samt at prosessen med å utarbeide en rapport om funn. Denne rapporten skal presentere hva dataene forteller på en presis måte. I tillegg til å presentere dataene skal det trekkes en tråd opp mot hvordan disse dataene er med på å svare på de forskningsspørsmålene studien ønsker å besvare. Dette står beskrevet tidligere i 3.5.2 TPACK analyseverktøy.

## 3.6 Etske betraktninger

Før vi begynte innhenting av data fikk vi godkjenning av SIKT (Kunnskapssektorens tjenesteleverandør). Dette var nødvendig ettersom informasjonen innhentet til oppgaven gjennom intervjuer og observasjoner skal oppbevares og behandles på riktig måte. I tillegg har vi holdt oss etter Universitetet i Agder sine retningslinjer for innhenting og behandling av informasjon. Ingen personopplysninger ble lagret på personlige enheter, og lydopptakene ble tatt rett opp i nettskjema gjennom diktafon applikasjonen, slik at det ikke skulle lagres fysisk hos oss. Informantene ble totalt anonymisert, derfor har vi brukt “han” som personlig pronomen uansett kjønn på informantene.

Først etter godkjenning fra SIKT (Se vedlegg 1) kontaktet vi informantene. Søknaden var en nødvendighet ettersom vi ønsket å ta lydopptak og observere deltakerne i undersøkelsen. Ved hjelp av SIKT sin mal, laget vi et informasjonsskriv som ble utdelt og signert av deltakerne (Se vedlegg 4). For å informere elever og foresatte i klassene som skulle observeres, leverte vi også et informasjonsskriv til e-posten til deltakerne så de kunne dele ut dette (se vedlegg 5). Her beskrev vi blant annet målet med observasjonen, og understreket at ingen elever vil

bli nevnt eller beskrevet i oppgaven. Ut ifra disse kriteriene konkluderte vi med at det ikke var nødvendig med underskrift fra elever/foresatte, ettersom disse ikke er deltagere i vår forskning.

## 3.7 Forskningskvalitet

### 3.7.1 Trustworthiness (Troverdighet)

Bryman (2016) presenterer ideer fra Lincoln og Guba(1985) og Guba og Lincoln(1994) hvor de tar for seg spørsmålet om hvorvidt kvalitativ forskning trenger å bli vurdert etter andre kriterier enn kvantitativ. Derav kommer det behovet for andre begreper som fortsatt speiler reliabilitet og validitet. Begrepene credibility, transferability, dependability og confirmability skal passe bedre for å beskrive troverdigheten (trustworthiness) av kvalitativ forskning, gitt dens natur. Vi har valgt å bruke både troverdighet og validitet & reliabilitet for å vurdere vår forskningskvalitet. Begge teoriene blir presentert av Bryman (2016), og selv om noen punkter går over hverandre gir det leseren et bedre innsyn på kvaliteten i oppgaven. Vi har også tatt utgangspunkt i flere artikler i validitet og reliabilitet.

**Credibility** eller troverdighet går ut på å spille med kortene på bordet. Forskingen skal utføres etter de prinsipper som god forskning baserer seg på. For å etablere god credibility vil man være åpen om arbeidet man gjør. Vi ga derfor alle enheter i oppgaven et skriv som inneholdt målet med undersøkelsen og informasjonen vi ønsket å forhøre oss om. Alle informanter har også blitt informert om at de kan trekke seg på ethvert tidspunkt. Informanter bør ha mulighet til å bekrefte at innsamlet data stemmer, og ofte vil man sikre bredde i innsamlingene for å hindre at tilfeldigheter spiller en større faktor enn nødvendig (Bryman, 2016).

Når transkripsjonene ble ferdigstilt totalt anonymisert, ble de sendt tilbake til informanten slik at vedkommende kunne enten bekrefte eller avkrefte at vi har forstått det de ønsker å få frem på rett måte. Dette beskriver Bryman (2016) som “respondent validation”. En av grunnene til å ha flere datainnsamlingsmetoder er å kunne triangulere, da sjekker man dataene på tvers av innsamlingsmetodene. Gitt at funnene fra begge metodene gir samsvarende resultat, vil dette bidra til å styrke Credibility (Bryman, 2016).

Neste grunnpilar i troverdigheten er **transferability**, eller på norsk overførbarhet. Kvalitative studier, og multiple case-studier spesielt, får røttene sine av dybden man kan gå i da bredden er begrenset. Derfor bærer studiene også preg av at konteksten spiller en betydelig rolle, og dette er vi også bevisste på når det kommer til å presentere funn gjort i vår studie. Ved å gi innsyn i metode og data kan leseren gjøre sine egne tanker om hvordan funnene kan settes i en bredere kontekst. Ettersom vi kun har datamateriale fra tre forskjellige lærere fordelt på to skoler, er det viktig å notere seg at det ikke kan gjøres en god generalisering av funnene gjort i oppgaven. Det kan være vanskelig å gi et godt svar på om det kan gjøres noen form for generalisering basert kun på et smalt utvalg, men det å arbeide transparent gjør at leseren kan sammenligne med lignende studier og selv trekke linjene om de måtte eksistere (Bryman, 2016).

Det er også en bemerkning at alle enhetene i studien selv valgte å bli med i studien. Ettersom det også spesifikt ble etterspurt programmering i matematikkundervisning før intervjuet, kan det ha påvirket utvalget av lærere som ønsket å bli intervjuet.

Gjennom å beskrive analyseprosessen og vise hvilke data som resulterer i de ulike funnene vil man oppnå en bedre **dependability**, også kjent som pålitelighet. I dette prosjektet er vi to studenter, det byr naturligvis på noen utfordringer, men også noen muligheter. Bryman (2016) nevner under dependability å ha tilsyn til sine kollegaer underveis i prosessen, samt å dokumentere stegene man gjør fra start til slutt for å ha en oversikt over hvilke steg som blir utført, av hvem og når. Det å være to om masteroppgaven gjør at vi kan passe på at vi begge opptrer i god ånd og holder oss på rett side, både etisk, juridisk og i forskningspraksis. Som nevnt er det vanskelig å avgjøre om en tilsvarende forskning vil komme frem til samme svar, da vi ikke har stort nok grunnlag til å generalisere funnene.

**Confirmability**, oversatt som bekreftelse, handler om å sette til side ens personlige verdier og teoretisk ståsted og la forskningen være upåvirket. Likevel vil total objektivitet være uoppnåelig. Vi anerkjenner at vi som mennesker har tidligere erfaringer, og at dette vil i noen grad påvirke hvordan man tolker ulike inntrykk. Når vi skal overse hverandres arbeid, mener Lincoln og Guba(1985, sitert i Bryman 2016) at et av målene våre burde være å sørge for god confirmability gjennom blant annet at vi opptrer sannferdig overfor forskningen.

### 3.7.2 Validitet og reliabilitet

Bryman (2016) skriver at god validitet og reliabilitet i en oppgave er kriterier som etablerer og vurderer forskningens kvalitet. Den kvalitative forskningen kan gjøre det vanskelig å generalisere funnene, da våre enheter ikke nødvendigvis representerer alle matematikklærere.

Vi som forskere har prøvd å ha mest mulig objektiv tilnærming på valg og tolkninger gjort, noe som vil ha en innvirkning på resultatene funnet. Gjennom en nøye beskrivelse av metode og analyse gir dette åpenhet som gjør at leseren kan få innblikk i prosessen. Før arbeidet startet brukte vi mye tid på å sette oss inn i teorien skrevet av andre for å begrense subjektiviteten. Desto bedre vi beskriver metode- og analysebruken, samt underliggende teori, desto bedre tolkningsprobabilitet har oppgaven (Postholm & Jacobsen, 2018)

#### 3.7.2.1 Reliabilitet

Reliabilitet omhandler påliteligheten og troverdigheten til forskningsresultatet. Siden det er omtrent umulig å reprodusere den samme studien ved senere anledning, er reliabiliteten avhengig av forskerens egen refleksjon over påvirkningen forskeren selv kan ha i forskningsprosessen (Postholm & Jacobsen, 2018).

Som nevnt har vi prøvd å finne intervjuobjekter vi selv ikke har noe tidligere kjennskap til. Wadel (2014) beskriver at nær tilknytning til steder eller informanter man forsker på kan føre til at viktig data går tapt, grunnet forskerens kjennskap til praksisen. Det er derfor viktig å nevne at den ene studenten har en indirekte kjennskap til Skole 2, gjennom jobb på barnetrinnet som er tilknyttet ungdomsskolen. Ingen av studentene kjente til noen av lærerne som ble intervjuet, og heller ikke skolekulturen på skolene.

I intervjuguiden er det essensielt at spørsmålene ikke virker ledende eller er for vanskelige å forstå. Dette kan føre til at informanten svarer ut fra det de tror forskeren ønsker å høre (Kvale & Brinkmann, 2015, sitert i Postholm & Jacobsen, 2018) Intervjuguiden er derfor gjennomgått flere ganger, og er basert på åpne spørsmål der intervjuobjektet selv må bestemme hvordan vedkommende ønsker å svare. Intervjuguiden er også lagt ved som vedlegg (se vedlegg 2) så leseren selv kan gjennomgå spørsmålene.

I observasjon av enheten var det viktig å ikke observere det vi ønsket å se, men heller skrive ned alle observasjoner nøytralt. Dette er for å unngå “observasjons-biås” (Fangen, 2010).

Først var det viktig å ha et observasjonsskjema som understreket hvilke observasjoner vi skulle skrive ned. Skjemaet ble laget veldig åpent, da vi ikke ønsket for mange retningslinjer på hva vi skulle se etter. En stor fordel i forskningen er at vi under alle 3 observasjonene hadde to individer som gjorde hver sine observasjoner. Det er da lettere å sammenligne hvilke observasjoner som er objektive og nødvendige for oppgaven. Likevel er begge forskerne i oppgaven interesserte i programmering, og det er ikke mulig å ha en fullstendig nøytral observasjon. Vi vil likevel argumentere for at bruken av observasjon og intervju sammen kvalitetssikrer studien mot egen påvirkning av resultatene.

### **3.7.2.2 Validitet**

Relevansen av den innsamlede dataen, og hvordan denne dataen kan generaliseres, kalles validitet (Christoffersen & Johannessen, 2012). Med tanke på at vi har tre individer som forskningsgrunnlag, kan vi ikke generalisere oppgaven til alle skoler i Norge. Vi kan heller ikke fastslå med sikkerhet at lærere i området passer inn i vår undersøkelse. En mulig viktig faktor i innhenting av data er at vi spesifiserte at vi trengte lærere som har brukt programmering i matematikkundervisning på ungdomsskolen. Vi vil da ikke ha et utvalg av lærere som representerer matematikklærere på landsbasis, ettersom flere lærere trolig ikke har brukt programmering som hjelpemiddel. Det er også en mulighet at lærere med lite eller ingen kunnskap om programmering følte seg utilpass med å delta i forskningen vår. Dette er en faktor vi har tatt høyde for når vi prøver å generalisere funnene våre i oppgaven.

## 4 Analyse og resultat

Dette kapittelet har som hensikt å gi innblikk i analyseprosessen. Vi kommer til å presentere hver case individuelt. Vi legger da frem prosessen og resultatene fra intervju og observasjon, etterfulgt av en liten oppsummering. Når alle casene er presentert kommer vi til å legge frem en bredere oversikt over resultatene i en “cross case” analyse for å gi en oversikt over helheten av resultatene.

For å kunne svare på forskningsspørsmålene:

- (1) Hvordan vektlegger lærerne teknologisk-, pedagogisk- og faginnholdskunnskap i arbeidet med programmering i matematikkundervisning på ungdomsskolen?
- (2) Hvilke kategorier i TPACK rammeverket identifiseres i den påståtte planleggingen og utføringen av undervisning i matematikk med programmering som verktøy i ungdomsskolen?
- (3) I hvilken grad utfyller lærerne stegene i Niess et al. (2009) sin 5-steps modell om integrering av teknologi, spesifikt programmering, i matematikkundervisning på ungdomsskolen?

Analyserte vi intervjuene og observasjonene ved hjelp av TPACK rammeverket. Det vår analyse skiller seg fra en tematisk analyse slik Braun og Clarke (2006) beskriver det er i hovedsak fasene hvor man redigerer og fjerner kategoriene man koder i slik at man har mest mulig klarhet og minst mulig overlapp. I vår analyse har vi en deduktiv tilnærming hvor vi starter med forutbestemte kategorier vi koder i. Vi har også innslag av induktiv analyse. Dermed endrer vi ikke kategoriene gjennom de ulike fasene, men vi spesifiserer mer hvilke kjennetegn vi er ute etter for at datamaterialet oppfyller kriteriene for hver kategori. Analysen er basert på en enkelt case-studie av enhetene vi har intervjuet og observert. Vi presenterer alle casene individuelt, for så å sammenligne casene til en multiple-case til slutt.

Casene vil bestå av en intervju-, observasjons- og oppsummeringsdel. I intervjudelen begynner vi med enhetens utdanning og erfaringer relevant for yrket. Videre presenterer vi analysen av intervjuet basert på TPACK rammeverket vi har utarbeidet. Vi presenterer også lærerens påståtte kunnskaper innen de forskjellige delene av TPACK, og vektleggingen av disse under arbeidet med matematikk. Til slutt presenterer vi deres mening om hensiktsmessigheten ved programmering som verktøy.

I observasjonen kommer vi først med frekvensen av observasjoner vi har gjort innen de forskjellige kategoriene. Videre beskriver vi observasjonene gjort.

Til slutt triangulerer vi observasjonene med intervjuet. Her kommer en undersøkelse av hvordan lærere har påstått deres kunnskap er, sammenlignet med våre observasjoner av undervisningsøkten. Ettersom observasjonen er basert på antall observasjoner, vil sammenligningen bli forholdet mellom benevnelsene i intervjuet og observasjonene.

## 4.1 Case 1: Lærer 1

### 4.1.1 Intervju

#### **Utdanning og erfaring:**

Lærer 1 beskriver at han tok individuelle fag innen matematikk og kristendom, for så å utdanne seg innen PPU til lærer. Der hadde vedkommende hovedfag innen matematikdidaktikk. Dette er rundt 30 år siden. Læreren begynte så å jobbe som realfaglærer, som spesialist innen matematikk. Når læreren ble spurt om han ønsket mer fagkunnskaper svarte vedkommende: «I matte til å jobbe på ungdomstrinnet så føler jeg at jeg kan mer enn nok. Det har jeg ikke noe behov for.»

I programmering beskriver læreren at han fikk litt programmeringsinnføring under studier, i et fag som kalles informatikk. I senere tid hadde personen et kurs innen programmering. Etter det har personen ingen erfaringer med programmering før de senere årene. Da valgte vedkommende å kontakte universitetet i nærheten for et oppfriskningskurs i programmering. Nå mener han at programmeringskunnskapene er gode nok til å programmere med elevene.

#### **TPACK- analyse**

Tabellen (Tabell 2) viser resultatene våre fra TPACK analysen. I fra tabellen kan vi se at det ble kodet totalt 69 frekvenser av komponenter i intervjuet med lærer 1, 20 på «kan ikke» og 49 på «kan». Under alle teknologiske kunnskaper er det 46 benevninger av teknologisk kunnskap. Dette inkluderer *Teknologisk kunnskap (TK)*, *Teknologisk pedagogisk kunnskap (TPK)*, *Teknologisk faginnholdskunnskap (TCK)* og *TPACK*. Av disse er 18 ren teknologisk kunnskap. Videre er det 34 benevninger av komponentene som inneholder pedagogiske kunnskaper, hvor kun 4 er rent basert på pedagogiske kunnskaper. Av faginnholdskunnskaper

er det 42 benevnelser av komponentene som inneholder CK, hvor 9 er rent basert på CK. I benevningene som treffer alle komponentene (TPACK) er det 13 av 15 mulige som havner innenfor «kan» underkategorien.

Kunnskapskomponent	Lærer 1	
	Kan ikke	Kan
Frekvensen av komponenter	20	49
TK	9	9
PK	1	3
CK	1	8
Alle TK (TK, TCK, TPK, TPACK)	16	30
Alle PK (PK, PCK, TPK, TPACK)	6	28
Alle CK (CK, PCK, TCK, TPACK)	9	33
TPK	1	4
TPACK	2	13

**Tabell 2:** Lærer 1 sine svar

En interessant bemerkelse er at læreren har jevnt pratet og svart mer om dårlig kompetanse eller kunnskaper rundt alle TK enn CK og PK. Dette støttes opp av blant annet lærerens uttalelse:

«Det er ganske langt ifra det å bare lese et program og se “ja, men dette programmet skjønner jeg kan kjøre”, til å kunne lage det programmet selv. Og det tenker jeg er veldig sånn i forhold til når vi gjør det her på skolen, at de lærere som ikke har vært borte i det hele tatt, har en utrolig lang vei å gå».



Videre beskriver læreren følgende: «Når noen bestemmer at alle mattelærere i Norge skal programmere, så er det noen som aldri har sett det før, og det er ikke bare å gjøre det altså. Det er et helt eget fag, det er ikke matematikk»

I disse tilfellene snakker læreren generelt om lærere, og ikke seg selv. I utdanning og erfaring kategoriserer han seg til å ha god kunnskap i teknologien.

På den motsatte siden har læreren pratet veldig positivt rundt kombinasjonen teknologi, pedagogikk og faginnhold (TPACK). Programmering som hjelpemiddel blir flere ganger nevnt som en viktig faktor for elevenes læring. Læreren sier i intervjuet følgende:

«Så ser vi selvfølgelig at det (programmering) har noen veldig positive sider. Det er noen ting som vi mener at programmering gjør at elevene lærer bedre. Men det er nok sånn at vi er ganske selektive med hvilket tema vi tenker at vi skal bruke programmering på ... Jeg mener jo at noen ting i matematikken tror jeg de kan lære bedre ved hjelp av programmering»

Læreren nevner ved flere tilfeller misnøye ved retningslinjene som er oppført i læreplanen i matematikk. Det er få retningslinjer på hva som er rett og galt program å bruke, samtidig som eksamensoppgavene ikke konkretiserer noe spesifikt. Når læreren beskriver eksamensoppgavene sier han følgende: «Jeg føler litt de oppgavene som har kommet der er hverken fugl eller fisk. Det er litt sånn noen blokker og noe "hvis" ditt.»

### **Påstått kunnskap om teknologi, pedagogikk og faginnhold, og anvendelse av kunnskapen**

Som nevnt tidligere beskriver enheten at matematikk-kompetansen er god nok til å undervise matematikk på ungdomstrinnet. han sier også at programmeringskunnskapene (TK) er gode nok til å undervise. han nevner også at han kommer inn på elevene og føler de lærer det de skal, noe vi anser som god kunnskap innen PK.

Læreren nevner at det er vanskelig å innføre programmering i faget, og at elevene mister verdifull tid og kunnskap på å kode for mye. Likevel understreker læreren at det skaper en stor motivasjon hos flere av elevene, noe som gjør han motivert til å bruke programmeringen der det kan passe inn. Våre funn indikerer at det er mer kritiske svar til egen og andres teknologiske kunnskap, enn de andre kunnskapene.

Læreren fikk spørsmålet om hvilken av de tre kunnskapene (faginnhold, pedagogikk og teknologi) han brukte fokuset på i planlegging og utføring av undervisning. Der fikk vi svaret: «Jeg fokuserer mest på de to første, minst på den teknologiske. Det er helt åpenbart.» Dette underbygger våre funn om at den teknologiske kunnskapen blir nedprioritert i undervisningen. han nevner også: «Vi trenger jo ikke de programmene i forhold til vårt pensum»

Læreren beskriver at han har brukt programmering i flere temaer som sannsynlighet, variabel (algebra) og algoritmisk tenkning. Ut ifra dette blir programmeringen beskrevet som et godt hjelpemiddel hvor den totale TPAC- kunnskapen til læreren virker kompatibel til hva som er forventet av en lærer på ungdomsskolen.

#### 4.1.2 Observasjon

I observasjonstimen av læreren noterte student 1 ned 1 observasjon av teknologisk kunnskap, 2 bemerkelser av pedagogikk og 1 av faginnholdskunnskap. Hvis vi summerer opp alle kategoriene som inneholdt de tre grunnkunnskapene (TK, PK og CK) var det 10 bemerkelser av TK, 12 av PK og 12 av CK. Med andre ord er det en veldig jevn distribusjon av observasjoner innen de forskjellige utførte kunnskapene.

Student 2 hadde flere observasjoner totalt sett, men sammenhengen er like påfallende med 17 observasjoner av TK, 18 av PK og 19 av CK. Et eksempel på en teknologisk-faginnholdskunnskaps observasjon skrevet ned er «Lærer viser en metode for utregning (i programmeringsverktøyet scratch), men han er åpen og oppfordrer til å prøve alternative løsningsstrategier». Samtidig viser lærer gode representasjoner innen pedagogisk-faginnholdskunnskap der læreren viser visuelt en gruppe elever som hadde problemer løsning ved å brette et ark som hjelp til å forklare formel for omkrets og areal av den 3-dimensjonale figuren de skulle regne ut i programmet.

Læreren observerte viser også flere TPACK koblinger i løpet av undervisningsøktene, der student 1 har notert ned 3 observasjoner og student 2 har notert 4. Et eksempel på observasjon er en interaksjon med 2 elever der lærer sier «du har laget en variabel uten å gi den en verdi, da regner den (scratch programmet) som null. Derfor blir svaret feil. Hva bør dere gjøre her da?». Lærer effektivt bruker kunnskap om faginnhold til å se at svaret blir feil,

teknologisk kunnskap til å finne feilen raskt og pedagogisk kunnskap til å hjelpe elevene videre.

I løpet av observasjonsøkten var det ingen observasjoner der elever ikke fikk tilstrekkelig hjelp til å løse de gitte oppgavene, eller spørsmål angående temaet. Læreren ga ikke ut noen direkte svar i spørsmålet om fremgangsmåte, men hintet elevene i riktig rekkefølge. Læreren hadde tilstrekkelig med tid til å sjekke flere elevers arbeid, og vi observerte kort ventetid hos elevene som trengte hjelp.

#### 4.1.3 Oppsummering – triangulering

I intervjuet beskrev læreren godt hvordan han følte seg komfortabel med sin kunnskap innen pedagogikk, teknologi og matematikk. Dette støttes opp av undervisningsøkten der vi observerte flere tilfeller av alle tre kategoriene. Vi så også flere tilfeller der lærer brukte kunnskapene sammen til å undervise elevene. Dette passer godt inn med lærerens generelt positive omtale rundt TPACK sammen, selv om vedkommende nevnte at det kunne være vanskelig å kombinere kunnskapene sammen.

Læreren beskriver at han helst ville klare seg uten den teknologiske kunnskapen, dersom han måtte velge en av de tre grunnkategoriene teknologisk, pedagogisk og faginnholdskunnskap. Dette passer best med å mangle det første punktet i Niess et al. (2009) sin modell, der gjenkjennelse omhandler kunnskap til å bruke teknologien. Hvis det er det læreren tenker på, har han fortsatt de pedagogiske kunnskapene til å velge riktig teknologisk hjelpemiddel til de forskjellige temaene, bare ikke kunnskapen til å hjelpe elevene med selve programmeringen.

Vi observerte også i timen at læreren virker veldig trygg på scratch som et verktøy, og forventer det samme av elevene. Ut i fra dette, og intervjuet der læreren forteller oss det, tar vi utgangspunkt i at scratch er et programmeringsverktøy som fungerer bra i klassen. Læreren har da vært igjennom 2. stedet i Niess et al. (2009) sin modell som valget av teknologi og selvstendig mening om gunstigheten av verktøyet. De neste stegene omhandler å finne, tilpasse og utforske aktivitetene eller metodene som programmeringen kan brukes til. I intervjuet beskriver læreren at han har funnet flere temaer hvor programmering som hjelpemiddel er godt egnet, og at han er fornøyd med utfallet. Dette tyder på at læreren har vært igjennom alle stegene av integrasjon av teknologien. I observasjonen fant vi flere tilfeller som bygger opp under påstanden, der elevene så engasjerte ut i undervisningsøkten,

lærer fikk vist faginnholdskunnskap gjennom programmeringen og at læreren viste pedagogisk kunnskap til å forklare og vise elevene faginnholdet og teknologien.

## 4.2 Case 2: Lærer 2

### 4.2.1 Intervju

#### **Utdanning og erfaring:**

Læreren tok en 4. års adjunktutdannelse innen grunnskoleutdanning 1-10. trinn. Denne utdannelsen kalles ALU eller adjunkt lærerutdanning. Etter utdanningen tok læreren videreutdanning i 2 år, så et totalt antall av 6 år med utdanning. I disse årene er 75 studiepoeng matematikk. Etter utdannelsen begynte læreren i jobb på ungdomsskole og har jobbet med det i ca 15 år. Selv beskriver læreren kunnskapen som god nok, og er fornøyd med utdannelses- og erfaringskunnskapen innen både matematikk og pedagogikk.

I programmering beskriver læreren at kunnskapen hovedsakelig har kommet igjennom selvlæring av programmene. Programmeringskunnskapene beskrives av læreren som relativt gode, og tilfredsstillende nok til å lære bort til elever. Dette kommer av lærerens evne til å jobbe godt selvstendig, og trivselen med å utforske nye ting på egenhånd. Likevel beskriver han at det var tungt å jobbe så mye med et nytt tema selv, og uttrykker mangel på utdanningsmuligheter innen programmering. Etter oppfølgingsspørsmål om læreren hadde tatt kurs dersom mulig får vi dette svaret:

«Ja, 100 prosent. Det vil jeg si. Det er ikke noe tvil om at hvis det hadde vært et kurs som kunne ha begynt å forklare litt hvordan du skulle programmere og hva tanken bak det var, så hadde du forstått mer tidligere, og det tror jeg hadde dratt nytte av»

#### **TPACK analyse**

Tabellen (Tabell 3) viser lærerens svar i intervjuet etter analyse gjennom vår analysemetode beskrevet i kapittel 3 metode. Av antall kodede utsagn finner vi et total av 15 i underkategorien «kan ikke» og 51 i underkategorien «kan». Læreren beskriver her kunnskapene teknologi, pedagogikk og faginnhold over 3 ganger så mye innenfor «kan» kategorien i forhold til «kan ikke» kategorien. Læreren har ganske nært like mange utsagn som inneholder teknologisk kunnskap, pedagogisk kunnskap og faginnholdskunnskap med

32 utsagn innen teknologisk kunnskap, 31 innen pedagogisk kunnskap og 39 innen faginnholdskunnskap. Totalt har læreren 7 uttalelser som inkluderer kombinasjonen av alle tre (TPACK) der alle har vært innenfor «kan» underkategorien.

Kunnskapskomponent	Lærer 2	
	Kan ikke	Kan
Frekvensen av komponenter	15	51
TK	4	10
PK	0	9
CK	5	9
Alle TK (TK, TCK, TPK, TPACK)	8	24
Alle PK (PK, PCK, TPK, TPACK)	2	29
Alle CK (CK, PCK, TCK, TPACK)	11	28
TCK	4	3
TPK	0	4
TPACK	0	7

**Tabell 3:** Lærer 2 sine svar

Læreren har kun to ganger omtalt pedagogisk kunnskap innenfor vår «kan ikke» underkategori. På motsatt side har læreren hele 29 uttalelser som faller innenfor «kan» kategorien. Læreren har altså et veldig positivt syn på den pedagogiske tilnærmingen sin, og den pedagogiske tilnærmingen ved teknologi. Dette kommer blant annet frem i svaret der læreren blir spurt om programmering som verktøy: «jeg tror mange opplever at det er, altså at du er litt mer kreativ, at du har muligheten til å flytte og prøve ut og feile, gøy å se.»

I kategorien faginnholdskunnskap har også læreren generelt mange uttalelser om at han kan og forstår læreplan og faget. Likevel er det en stor andel (11) som peker på manglende kunnskaper i faget. Flere av uttalelsene omhandler manglende veiledelse fra utdanningsdirektoratet. Når læreren uttaler seg om læreplanmålene i fra utdanningsdirektoratet sier han følgende: «Hvordan man skal bryte ned disse målene, hva det er de (Utdanningsdirektoratet) egentlig mener. Gjør vi dette for å lære programmering, eller gjør vi dette for å lære matematikk? Og så det var jo mye av det vi lurte på.»

På den andre siden beskriver læreren at han får dekt alle kunnskapsmålene. han sier: «Jeg tror jo at vi er innom det vi skal, og at vi får jobba gjennom disse kompetansemålene på en god og riktig måte. Det gjør jeg egentlig, ut fra hvordan vi tolker det.»

Læreren har også 7 omtaler som inneholder alle kunnskapene i TPACK. I disse uttalelsene beskriver læreren ofte spesifikke eksempler på metoder de har brukt, eller hvordan elevene best kan lære. Et eksempel på dette er uttalelsen:

«Men jeg håper og tror at flere av elevene opplevde at dette med volum er mer visuelt, at man forstår det bedre (etter en observasjonstime med programmering av volum). At du nå faktisk har et program som regner ut volum av alle verdens rektangulære prismer, for eksempel. Og det er jo litt kult for mange.»

I flere av disse eksemplene nevner læreren at han bruker programmeringen mer som et verktøy i matematikkundervisningen, i stedet for å fokusere på læringen av programmet selv. Læreren bekrefter igjennom intervjuet forsøk på å optimalisere programmeringen til de spesifikke temaene, og beskriver også «pedagogikken er jo veldig vid, men det med å prøve å formidle dette ut til elevene og vise dem hva programmering går ut på, og samtidig få inn matematikkfaget.»

### **Påstått kunnskap om teknologi, pedagogikk og faginnhold, og anvendelse av kunnskapen**

Læreren utdyper i intervjuet at kunnskapen nå er god nok til å presentere et godt resultat innen både teknologi, pedagogikk og matematikk for elevene. Den største utfordringen var å få inn programmering i undervisningen og han beskrev:

«Jeg var jo småskeptisk i starten, men det var jo basert på egen kompetanse. Men når jeg da kastet meg ut i det og forsto at dette kunne brukes som et godt verktøy inn mot matematikkfaget, og la vekk den stoltheten om at jeg skulle kunne alt best.»

Det blir gjentatt flere ganger med vanskeligheten av adaptering av programmering i starten, at programmeringen ble et eget tema uten særlig bruk i andre temaer. I teorien om adaptering av ny teknologi, blir dette beskrevet som utforske-fasen der lærer må teste ut programmeringen i håp om at metoden er hensiktsmessig. Lærer 2 beskriver utilsiktet om flere av stegene i løpet av sitt intervju, noe som tyder på en tydelig adaptasjon av teknologien med årene. Han beskriver også at det ble lettere å undervise i matematikk med programmering som hjelpemiddel når vedkommende ble mer komfortabel med sin egen kunnskap rundt teknologien.

Likevel beskriver læreren at det er teknologien han helst vil fokusere minst på av de tre grunnkunnskapene (TK, PK og CK). Læreren begrunner påstanden med følgende: «Pedagogikken må jeg ha i bunnen, uansett hvilken form av pedagogikk. Samtidig som programmering er en del av matematikkfaget. Og da ville jeg heller kunne de 95 prosent andre». Dette samsvarer også med funnene vi har gjort i analysen, der læreren viser seg mer kritisk til den teknologiske kunnskapen, spesifikt hva som kreves i læreplanmålene.

#### 4.2.2 Observasjon

I observasjonen av læreren ble det gjort flere observasjoner av de forskjellige kunnskapene teknologi, pedagogikk og faginnholdskunnskaper. Observasjonene som ble gjort var veldig jevnt fordelt over de tre kunnskapene, der det ble gjort 16 observasjoner vist teknologisk kunnskap, 15 av pedagogisk kunnskap og 15 av faginnholdskunnskap hos student 1. Student 2 har observert henholdsvis 16 observasjoner av vist teknologisk kunnskap, 18 av pedagogisk kunnskap og 20 av faginnholdskunnskap. Læreren har da vist forskjellige kunnskaper både hver for seg og samlet for å undervise elevene i løpet av økten.

Læreren brukte scratch (programmeringsverktøy) som representasjon, der matematikktemaet var geometri. I observasjonen så vi at læreren først gjennomgikk et eksempel av en geometrisk figur, rektangel, på whiteboard før han viste elevene fremgangsmåte på scratch. I denne tiden brukte læreren elever ved flere tilfeller til å komme med neste steg på hvilke blokker i scratch som skulle brukes, og formler til den geometriske figuren.

Det er flere tilfeller av en kombinasjon av alle de tre grunnkategoriene eller TPACK hos læreren. Blant annet ble det notert ned «Lærer viser rask kopiering av klosser på tavle (i scratch) i stedet for unødvendig tid, viser forståelse av effektivitet og teknologi». Med dette viser lærer teknologisk kunnskap, lærer det bort på en teknologisk metode og har matematisk kompetanse til å se at blokkene brukt i oppgaven kan brukes videre i neste ledd.

Gjennom timen oppfordret lærer til å kladde ned formler på ark før elevene skulle programmere. Læreren viser en teknologisk pedagogisk faginnholdskunnskap kunnskap her hvor han understreker viktigheten av riktige regnerekkefølger, samtidig som han viser at representasjonen fra ark til programmering er en mulig strategi for forståelsen av programmet.

Lærer bruker som nevnt kunnskap innen de tre forskjellige kategoriene. I faginnholdet viser lærer kunnskaper som korrekte formler til regning av geometriske figurer, og beskriver  $\pi$  som et fast tall og ikke en variabel. Videre bruker lærer forskjellige pedagogiske kunnskaper som fremvisning på tavle, klasseledelse i form av irettesettelse av elever som prater under fremvisning og gir hjelpende hint til elever i stedet for å gi direkte svar. Lærer viser også kunnskap innen teknologi ved rask og konkret fremvisning av blokkene i scratch, bruker snarveier for tidsbesparing og gjenkjenner feil raskt og retter feilene opp.

#### 4.2.3 Oppsummering - triangulering

I analysen av intervjuet med læreren kommer det frem at alle de tre kunnskapene brukes i planleggingen og gjennomføringen av undervisning med programmering som verktøy. Dette samsvarer bra med observasjonen der læreren utnytter alle tre kunnskapene jevnt over timen. Læreren beskriver at han føler de tre kunnskapene er gode nok til å undervise, noe som verifiseres i observasjonen. Det ble ikke observert noen manglende kunnskaper, og lærer brukte ofte flere av kunnskapene sammen (PCK, TPK, TCK, TPACK) for å lære elevene.

Læreren var positiv til kombinasjonen i TPACK under intervjuet, og nevnte blant annet at han strebet etter å implementere pedagogikk, matematikk og teknologi i undervisningsøkten. Dette samsvarer med våre observasjoner der vi så ulike kombinasjoner av kunnskapene, og flere tilfeller av fullverdig TPACK.



Det var ingen av kunnskapene som ble adskillig mer vektlagt enn andre i observasjonsøkten. Det var et veldig jevnt antall observasjoner av bruken av de forskjellige kunnskapene. Læreren beskrev også dette i intervjuet selv, der viktigheten av alle kunnskapene ble nevnt.

I intervjuet beskrev læreren at teknologien var den minst viktige av de tre kunnskapene, der teknologien kun er nødvendig for en liten andel av matematikken. Likevel presiserer han at det ikke er mer tidskrevende å bruke teknologien i undervisningen, og at representasjonene ved programmering er en stor fordel for mange elever. han beskrev at det var krevende og lite effektivt med programmering i begynnelsen, men at det ikke lenger er noe stor utfordring. Dette samsvarer godt med Niess et al. (2009) sine adaptasjonssteg der det først er når læreren oppnår erfaring og bruker tilfredsstillende teknologiske metoder. Observasjonene gjort i timen underbygger dette, da kunnskapene til læreren ble brukt i samsvar til temaet.

## 4.3 Case 3: Lærer 3

### 4.3.1 Intervju

#### **Utdanning og erfaring:**

Lærer 3 har en grunnskoleutdanning innen allmennfag, et 4 åring studie. Som tilleggstudning har læreren tatt 30 studiepoeng innen matematikk, noe som resulterer i 60 studiepoeng med matematikk. han har jobbet som lærer i ca 15 år, og beskriver at det ikke føles som nødvendig med mer kunnskap innen matematikk og pedagogikk på dette tidspunktet.

I programmering fikk læreren tilbud om å ta et 30-timers kurs, fra tilbud i kommunen han jobbet i tidligere. Læreren beskriver at hen, og alle medkollegaer valgte å takke ja til dette tilbudet. Læreren sier kurset ga et godt grunnlag for kunnskap, men at det ikke følte nok i det læreren skulle bruke programmeringskunnskapene i undervisning. Videre beskriver læreren at han har brukt tid på å utvikle kunnskaper innen å lære bort programmeringen til elevene, og føler nå at kunnskapen er god nok til undervisning.

#### **TPACK analyse**

I tabellen (Tabell 4) vises frekvensen av benevninger lærer 3 har gjort i intervjuet vi hadde. Her kan vi lese at han har i større grad avgitt svar om teknologisk (37) og

faginnholdskunnskap (35) enn han har gjort om pedagogikk (21). Likevel har læreren beskrevet alle de tre grunnkunnskapene pedagogikk, teknologi og faginnhold flere ganger.

Kunnskapskomponent	Lærer 2	
	Kan ikke	Kan
Frekvensen av komponenter	19	34
TK	3	7
PK	2	3
CK	2	5
Alle TK (TK, TCK, TPK, TPACK)	14	23
Alle PK (PK, PCK, TPK, TPACK)	7	14
Alle CK (CK, PCK, TCK, TPACK)	13	22
TCK	7	8
TPK	1	2
TPACK	3	6

**Tabell 4:** Lærer 3 sine svar

Flere av lærerens uttalelser om mangel på kunnskap handler om frustrasjon ved læreplanmålene, og en manglende felles forståelse av hva teknologi og programmering er i skolen. I spørsmål om læreren føler det har vært en felles fremgangsmåte på programmering får vi følgende svar: «Nei, det er ikke min opplevelse. Fremdeles opplever jeg at hver lærer står liksom på sin tue, og de lærerne hos oss gjetter.» han fortsetter å forklare at han opplever flere lærere som usikre på programmering, og at det ikke er landet noen felles plan for hvordan fremgangsmåte de skal ha innen programmering.

Likevel beskriver læreren at han gjerne ønsker å ha programmering inne i mange av de forskjellige temaene matematikk inneholder. Han forklarer at det brukes programmering hvor de bruker en «funksjonsmaskin». Dette er en metode der elevene leverer et tall inn i en ukjent ligning med én variabel skjult av en boks. Tallet de leverer inn skiftes ut med variabelen. Ut får de et nytt tall som er svaret på ligningen etter utregning. Derfra må elevene teste flere tall for å finne ut algoritmen eller ligningen som brukes på de innputtede tallene. Videre beskriver han følgende:

«Kanskje også i forbindelse med figurtall. Figurtall er også litt den samme, opplever jeg i hvert fall litt samme strategi for hvordan man går frem for å nøste opp hvilke ulike deler av denne figuren er satt sammen, knytter det til trekantall, eller kvadrattall, eller whatever man gjør for noe.»

Læreren har totalt i intervjuet beskrevet 9 tilfeller av TPACK, der 3 av de går under «kan ikke»-kategorien. Uavhengig av dette, beskriver læreren at han er veldig glad i å bruke programmering som hjelpemiddel. Sitert fra læreren ved spørsmål om synet på programmering som verktøy:

«Ja, det synes jeg er veldig kult. Jeg elsker det. For det er veldig ofte jeg ser, hvis du knytter det sammen med at utforskning har fått en mye større plass i matematikk læreplanen enn det hadde tidligere. Når jeg jobber nå med utforskningsoppgaver, altså matematisk problemløsning, så tenker jeg veldig ofte at «Faen heller, kunne jeg programmert løsninger på dette spørsmålet?»

“Vi hadde jo faktisk her om dagen en problemløsningsoppgave som vi gjorde i hele personalet knyttet til matematikk, hvor jeg bare var irritert for at jeg ikke hadde en PC. For jeg hadde lyst til å prøve å se om jeg kunne programmere meg fram til... For det var jo en generalisering vi skulle prøve å gjøre da. Vi ser at hvis du legger sammen sånn og sånn og sånn, så får du alltid ni. Uansett, du får alltid ni. Da ble jeg sånn «Faen heller, hvorfor det?» «Jo, dette her bør jeg kunne ha programmert.» Så jeg synes innføring av programmeringen hører veldig bra sammen med det utvidet fokus på utforskning og problemløsning i matematikken”

Samtidig understreker læreren at det er flere utfordringer ved programmering i klasserommet. Blant annet sier han at problemløsningsoppgaver ofte blir for vanskelig for mange elever, og at det er for tidkrevende å sette av tiden til å forstå det. I tillegg understreker han at

programmeringen kan bli veldig abstrakt for elever, spesielt de som har mindre grunnlag i de matematiske formlene.

For å hjelpe med disse utfordringene, anbefaler læreren å bruke blokkprogrammering. Læreren beskriver at han forandret læreplanen for elever som synes det var vanskelig med tekst-programmering til blokkprogrammering. han beskriver følgende: «Kikora hadde en blokkinnlæringsløp, så det ble kjempemotiverende for dem (elevene) å redde programmeringen deres.»

Læreren understreker at det ønskes et mer stabilt programmeringsløp og integrering for elever på barneskolen. Når læreren snakker om fremtidens ungdomsskoleelever sier han følgende: «Det som jeg håper, det er at hele skoleløpet gjør jobben sin(...) At de har såpass godt grunnlaget at jeg i større grad kan bruke programmering som verktøy for å løse matematiske problemer. Enn å bare drive opplæring i programmering. For det har det vært så langt.»

### **Påstått kunnskap om teknologi, pedagogikk og faginnhold, og anvendelse av kunnskapen**

Læreren beskriver at individuelt har han god kunnskap i både teknologi, pedagogikk og faginnhold. han sier at det vanskeligste med undervisningen av programmering er å innføre en god pedagogisk tilnærming til teknologien. I dette tilfellet beskriver læreren utilsiktet vanskeligheten med utforske-fasen av Niess et al. (2009) sin integrering av teknologi i klasserommet. Dette beskrives godt i utsagnet:

«Så jeg tror den utfordringen min er den pedagogikk- og undervisningsmetode-biten. For nå har jeg gjort det en gang, og da var jeg helt ... Det var første gang. Så jeg tenker at det er det som er dårligst integrert. Og så er jeg litt spent på hvis det nå gjør det sånn at vi tar fire uker med programmering, og så går vi bare videre. Da er jeg veldig spent på om jeg i det hele tatt får integrert matematikk noe særlig».

Læreren har hatt mange år med erfaring innen læring av faget, men beskriver her vanskeligheten av å kunne de pedagogiske tilnærmingene til et helt nytt tema.

Videre forklarer læreren om utfordringen med å få inn nok matematikk i programmeringstimene. han sier at fokuset ofte forsvinner fra det matematiske og over på det teknologiske der elevene får kun utbytte av de teknologiske kunnskapene til læreren. Dette

kan knyttes tett opp til fremskritt-fasen av Niess et al. (2009) sin modell som omhandler å videreutvikle metodene tilpasset til hva elevene lærer.

Likevel beskriver læreren at han er fornøyd med å lære bort programmering, og at metodene han bruker har utviklet seg. Som nevnt tidligere beskriver han flere temaer som viser seg å være passende å bruke programmering som hjelpemiddel i.

### 4.3.2 Observasjon

I observasjonsøkten av lærer 3, var det hovedsakelig et introduksjonskurs innen programmering i matematikk. Det var derfor ingen spesifikke temaer utenom programmering som ble introdusert eller jobbet med. Det var likevel fokus på matematiske funksjoner innen multiplikasjon, divisjon, subtraksjon og addisjon ved bruk av variabler.

Spesielt i denne observasjonsøkten blir det et veldig smalt skille på hva som er faginnholdskunnskaper og hva som er teknologisk kunnskap ettersom læreplanmålene beskriver programmering som en faginnholdskunnskap lærer og elevene skal kunne.

Undervisningstimen var pedagogisk satt opp der lærer først viste hele algoritmen på whiteboard mens elevene satt og fulgte med. Dersom det var nye formler eller koder elevene måtte kunne, ble de bedt skrive disse ned i et word-dokument de kunne bruke. Etter læreren var ferdig, skulle elevene prøve å gjenskape koden selv. Her bruker læreren en pedagogisk metode hvor han «tvinger» elevene til å memorere fremgangsmåten, for så å utnytte denne kunnskapen selv. Lærer viser kunnskap om teknologien ved å forklare elevene hvordan variablene de betegner i koden blir ført videre til neste gang de bruker samme variabel. Eksempelvis sier læreren til en elev: «Det er stor A der \*peker\* og liten a der \*peker\*, da blir det forskjellige variabler og programmet skjønner det ikke.» Dette kodet vi en interaksjon som teknologisk-pedagogisk kunnskap der læreren viser kunnskap om hvordan eleven kan omkode programmet for å få riktig resultat.

I observasjonen har vi mange tilfeller av faginnholdskunnskap hos læreren. Mange av disse observasjonene er av kunnskapen om selve programmeringen, og hvordan lærer bruker kunnskapen til å lære elevene. I tillegg til programmeringen observerte vi nøye bruk av korrekte faginnholdsord som «variabel», «subtraksjon», «addisjon» og lignende. Læreren viser først hvordan man koder de fire regneartene i python, for så å be elevene bruke regneartene til å lage egne regnestykker. han beskriver for elevene sammenhengen mellom

koding og matematikk. I oppsummering av timen beskriver læreren at elevene der har laget en kalkulator – og dermed sammenligner med en representasjon elevene er kjent med.

Av det totale antallet observasjoner gjort i timen, var det flest tilfeller av pedagogisk kunnskap læreren utøvde. Læreren brukte forskjellige pedagogiske metoder som fremvisning på tavle, selvstendig arbeid, arbeid med hjelp av lærer og «kode-ark» med hjelpemidler til å løse oppgavene gitt.

Læreren viser i flere tilfeller utnyttelse av alle kunnskapene (TPACK) samtidig. Eksempelvis gir læreren en aktivitet til elevene der de skal bruke programmering til å først lage 4 ukjente variabler, for så å bruke variablene til å lage multiplikasjons-, divisjons-, addisjons- og subtraksjonsregnestykker. Lærer går så rundt i klasserommet og hjelper elever med nettopp dette. Her kommer det fram både forståelse av koblingen mellom de forskjellige kodene i programmeringen, sammenhengen med matematikk og aktivitet der elevene kan lære.

#### 4.3.4 Oppsummering - triangulering

I intervjudelen av analysen bemerket vi at lærer flere ganger beskriver pedagogikk som den minst viktige av de tre delene kategoriene i TPACK, og fokuserte mye på kunnskapen av teknologi og faginnhold. Læreren beskrev fortsatt at pedagogikk var viktig, bare benevnte ikke dette så ofte som et fokusområde når han pratet om undervisning av programmering i matematikk. Likevel bemerket vi i observasjonene at læreren har stort fokus på både en til en pedagogikk, teknologiske metoder og aktiviteter og klasseledelse. Når vi teller opp antall observasjoner, er det faktisk flest av de pedagogiske kunnskapene som blir vist i undervisningsøkten.

I intervjuet nevnte læreren alle tre kunnskapene, med litt ekstra fokus på teknologisk og faginnholdskunnskap. Videre beskriver læreren i intervjuet at han selv føler han har god kunnskap i alle de tre grunnkategoriene teknologi, pedagogikk og faginnhold. Læreren sier det er vanskelig å implementere teknologien på en god måte. Som nevnt passer dette godt med de siste punktene i teorien om implementering av ny teknologi av Niess et al. (2009), utforskning og fremskritt. Læreren må få implementert teknologien på en god måte, og så revaluere om den teknologiske metoden og temaet passer sammen. I observasjonen vi hadde er det vanskelig å se om læreren har klart dette, da undervisningsøkten hovedsakelig omhandlet koding og de fire regneartene.

## 4.4 Cross case – multiple case design

I denne delen av analysen sammenligner vi de tre casene eller lærerne for å se hvilke likheter og ulikheter som finnes. Målet med sammenligningen er å se om det er mulig å generalisere funnene våre. Analysedelen er delt inn i 3 deler der vi først ser på de tre kunnskapene lærerne har og utnytter i planlegging og utførelse av undervisning. Videre studerer vi lærernes bruk av alle kunnskapene sammen, og om de har klart å oppnå TPACK. Til slutt ser vi på lærernes adaptivering av teknologi i forhold til Niess et al. (2009) sin modell.

### 4.4.1 Teknologisk, pedagogisk og faginnholdskunnskap

Det ble beskrevet i intervjuene at alle tre lærerne hadde gode kunnskaper innen faginnhold, teknologi og pedagogikk. Ingen av de tre lærerne beskrev at de følte noen mangler innen kunnskapene. Likevel beskriver alle tre lærerne enten at de har hatt tilleggskurs i programmering, eller ville takket ja dersom de fikk tilbudet. Dette kan tyde på at informantene er både villige og interesserte i å lære den nye teknologien. Ingen av lærerne følte behov for mer utdanning innen matematikk eller pedagogikk, selv om lærer 3 understreket vanskeligheten av å innføre teknologisk pedagogisk kunnskap i undervisningen.

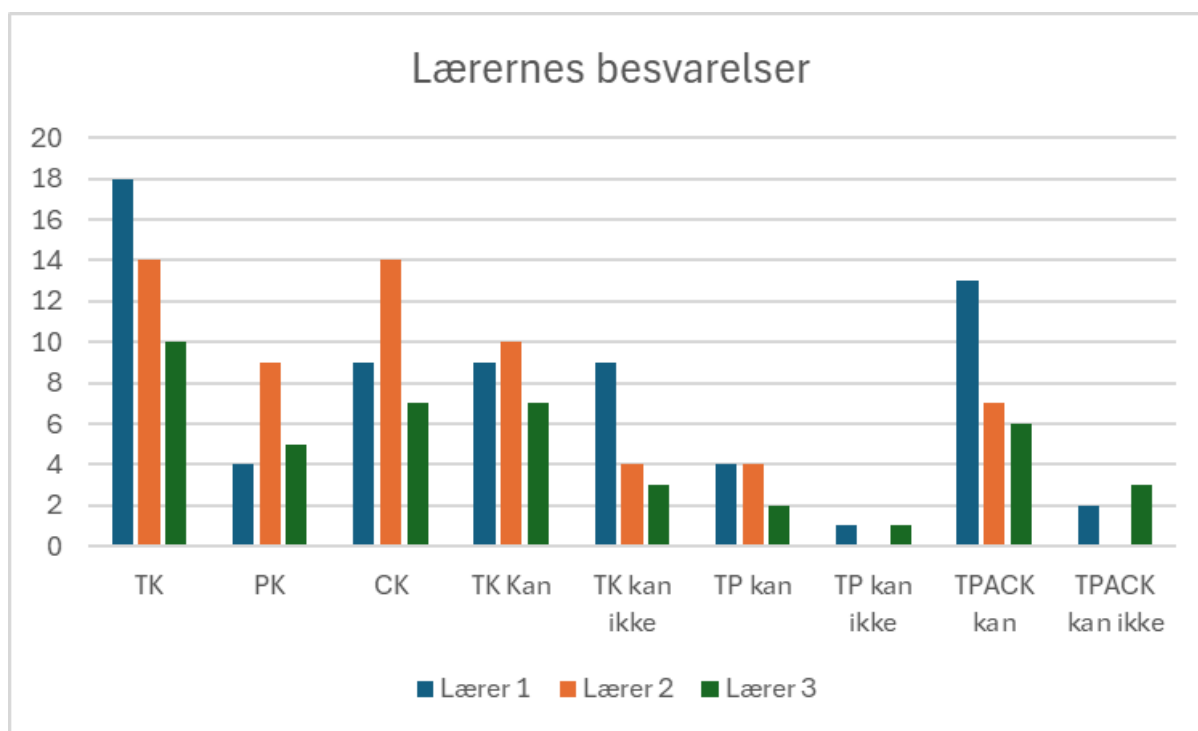
Også i observasjonene bemerket vi flere observasjoner i alle de tre forskjellige kunnskapene, både hver for seg og felles. Vi observerte at alle lærerne brukte tavleundervisning, selvstendig jobbing, og gikk rundt i klasserommet og hjalp enkeltelever med vanskeligheter dersom de trengte det. De to første lærerne fikk vist frem mange tilfeller av faginnholdskunnskap, og hadde større fokus på andre matematiske temaer ved siden av enn programmering enn lærer 3. Lærer 3 derimot hadde et mer grunnleggende kurs for elevene om koding, noe som gikk utover muligheten til å vise faginnholdskunnskap eller aktiviteter som fremmer faginnholdskunnskapene til elevene.

I Figur 4 fremkommer det at alle lærerne omtalte mange tilfeller av teknologisk, pedagogisk og faginnholdskunnskaper. Figuren inkluderer kombinasjonskategorier som TCK, PCK og TPK. Lærer 1 og 2 hadde et veldig jevnt fokus på alle tre kunnskapene, mens lærer 3 hadde litt mindre fokus på den pedagogiske kunnskapen. Dette kom også frem da lærer 3 ville valgt bort pedagogikk, mens lærer 1 og 2 ville klart seg uten teknologisk kunnskap dersom de måtte velge.

## 4.4.2 TPACK

Når lærerne skal integrere den teknologiske kunnskapen sin inn i undervisningen må læreren vurdere flere faktorer som nytte, tidsbruk, stemmer representasjoner og lignende.

Hvordan læreren har vurdert disse punktene avgjør hvor godt utfall det har for elevene. Figur 3 er en representasjon av antallet benevninger lærerne hadde innen de forskjellige kunnskapskomponentene vi så etter. I intervjuene med lærerne identifiserte vi totalt 31 tilfeller av TPACK der 26 av de falt innenfor «kan» kategorien. Selv om tallene i seg selv ikke sier veldig mye, gir det en indikasjon på at lærerne viser kunnskaper innen fullverdig TPACK. Dette har vi triangulert med observasjonene, der vi også fant flere tilfeller av TPACK når læreren underviste.

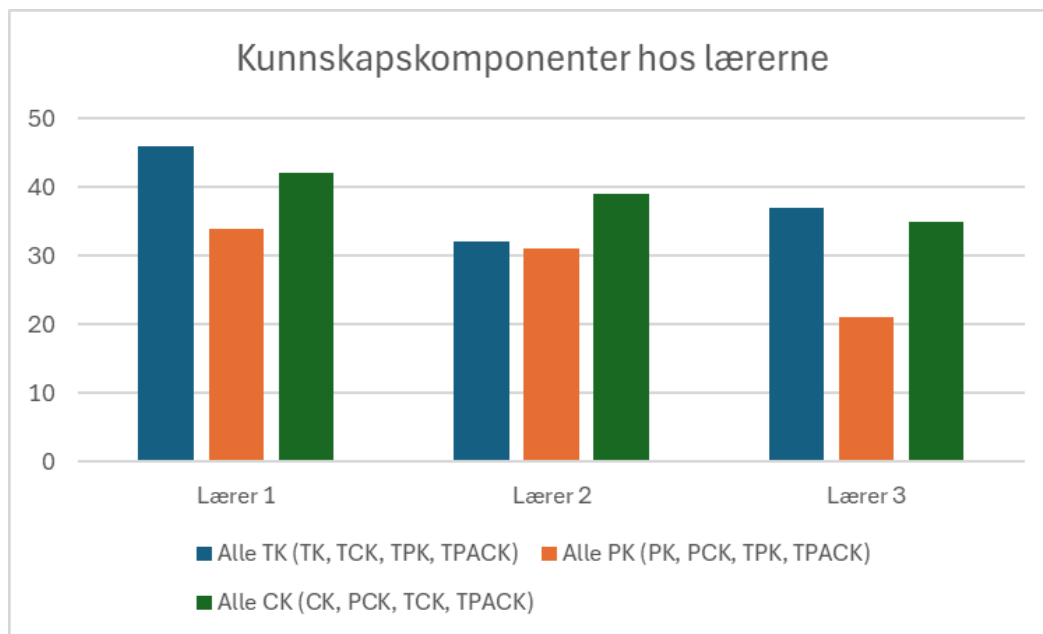


**Figur 3:** Sammenligning av lærernes svar

Alle tre lærerne pratet varmt om mulighetene som tilbød seg når programmering ble inkludert i undervisningen som et hjelpemiddel. Likevel er det 16 benevnelser blant lærerne som treffer innen «kan-ikke» underkategorien av teknologisk kunnskap. Mange av disse benevnelser, som nevnt i de enkelte casene, handler om manglende kunnskaper eller motivasjon hos kollegaer. Dette er noe som kan spille inn på lærerens egen motivasjon og syn om programmering. En annen ting å bemerke er at ingen av de tre lærerne beskrev tidsbruken som en utfordring når de ble spurt om vanskelighetene med å bruke programmering som



hjelpemiddel. Flere av de negative omtalene rundt programmering omhandlet manglende retningslinjer fra utdanningsdirektoratets læreplanmål.



**Figur 4:** Kunnskapskomponenter hos lærerne

Når vi ser på lærernes omtaler om teknologisk-pedagogisk kunnskap og TPACK ser vi en prosentvis økning fra «kan» til «kan-ikke» underkategorien i forhold til benevningene i teknologisk kunnskap kategorien. Vi kan også se gjennom trianguleringen i de forskjellige casene at alle lærerne har oppnådd flere tilfeller av TPACK. Lærerne har som nevnt beskrevet alle de ulike kunnskapene om faginnhold, pedagogikk og teknologi relativt likt. Dette peker på at lærerne må fokusere på alle de forskjellige kunnskapene individuelt, for så å kombinere kunnskapene for elevenes fortjeneste.

I undervisningene som ble observert brukte som sagt alle lærerne tavleundervisning i starten av undervisningen. Det kan være at alle lærerne individet har kommet frem til at de får utnyttet sin TPACK best ved tavlegjennomgang med elevene først. Videre brukte alle tre lærerne selvstendig arbeid, der elevene fikk utdelt en oppgave eller et problem de måtte løse. Om dette viser en komplett TPACK hos læreren har vi for lite grunnlag til å konkludere med, men ut fra våre observasjoner ser dette ut som en god metode for lærerne til å utnytte kunnskapene i undervisning.

Innen pedagogisk-faginnholdskunnskap fant vi hovedsakelig omtale innenfor “kan” kategorien. Lærerne hadde flere metoder og aktiviteter de kunne bruke for å fremme elevenes læring, og ut ifra vår data hadde lærerne få problemer med å utnytte denne kunnskapen.

I planleggingen som fører opp til undervisningsøkten pekte 2 av 3 lærere på pedagogikk og faginnhold som fokusområdene, mens den siste læreren pekte på faginnhold og teknologi. Alle lærerne bekreftet i intervjuet at de kun bruker programmering dersom de anser det som hensiktsmessig for elevene i forhold til temaet. Eksempler på tilpassede temaer var funksjoner, geometri og algebra. Ut ifra våre funn er det dermed ikke alle temaene det er like enkelt å tilpasse til programmering. Likevel beskriver lærerne at de hele tiden er på utkikk etter metoder som kan passe inn i temaene de enda ikke har brukt programmering i.

#### 4.4.3 Adaptering av teknologi i undervisning

##### **Gjenkjennelse**

Alle de tre lærerne sier selv de har kunnskapen til å programmere selv. De har vært på kurs for å oppdatere kunnskapen sin innen programmering. Dette støttes av observasjonene, der alle lærerne viste kunnskap om programmet de underviste i. Vi observerte ingen tilfeller der læreren ikke kunne hjelpe elevene med programmeringen.

##### **Akseptere**

To av lærerne brukte scratch som hjelpemiddel, og sa de hadde bestemt seg for det flere år siden. De mente dette var den beste formen for programmering for elevene, da blokkprogrammering gjorde det lettere å kode. Den siste læreren brukte både Python og blokkprogrammering (ikke spesifisert hvilket) til å lære elevene. Han understreker at Python kunne være for vanskelig for noen elever, og i de tilfellene byttet de over til blokkprogrammering. Alle lærerne har her gjort en vurdering av teknologien.

##### **Tilpasse**

I denne fasen har lærerne beskrevet samarbeid med ledelse eller team, der de har kommet til en felles avgjørelse om hvilke program som skal brukes og hvordan programmet skal brukes til å hjelpe elevene i faget. Alle tre lærerne har nevnt vanskeligheter ved dette punktet, da kollegaer og skoleledelse ikke har hatt kunnskapen til å komme med gode innspill. Lærer 1 og lærer 2 jobbet på samme trinn, og beskriver at de to har selv funnet ut hvordan de ønsket å

implementere programmeringen i matematikkundervisningen. Lærer 3 nevnte at han ble stående litt alene, og trolig måtte bestemme programmeringsverktøy og undervisning for hele trinnet alene. I observasjonen så vi at valget av program falt på Python, noe læreren også spekulerte i under intervjuet.

### **Utforske og fremskritt**

Av lærerne vi snakket med, hadde alle tre testet ut forskjellige undervisningsopplegg innen matematikk med programmering som hjelpemiddel. Som nevnt var det kommentarer om vanskeligheten med å velge program og undervisningsmetode som var den pågående utfordringen til lærerne. Likevel beskriver alle tre lærere programmering som hjelpemiddel som en positiv ting, og alle kom med flere eksempler på oppgaver og undervisningsmetoder de kunne bruke til programmering. De vurderte generelt programmeringen som en positiv del av matematikken, noe som plasserer alle de tre lærerne innenfor “fremskritt” steget i modellen.

## 5 Diskusjon

Her kommer vi til å besvare våre forskningsspørsmål gjennom å diskutere de funnene som er gjort i analysen. Vi kommer også til å sammenligne resultatene med funnene i lignende studier. Vi kommer også til å diskutere hvilke implikasjoner funnene vil ha for hver lærer i klasserommet, og hva som må til for at programmering skal integreres optimalt i matematikkundervisning. Etter det kommer vi til å gjøre et tilbakeblikk på de valgene vi har gjort når vi har etablert det teoretiske grunnlaget og utarbeidet metoden. Dette innebærer også kritiske refleksjoner rundt valgene som ble gjort.

### 5.1 Svar på forskningsspørsmål

#### 5.1.1 Læreres vektlegging av hovedkunnskapene i TPACK

Vi skal her ta for oss forskningsspørsmålet: “Hvordan vektlegger lærerne teknologisk-, pedagogisk- og faginnholdskunnskap i arbeidet med programmering i matematikkundervisning på ungdomsskolen?” Først er det viktig å nevne at målet vårt ikke er å måle kunnskapene i teknologi, pedagogikk og faginnhold hos læreren, men å undersøke hvordan læreren vektlegger kunnskapene i sin planlegging og undervisning. Gjennom observasjonene så vi at alle lærerne hadde et jevnt fordelt notasjon i hovedkunnskapene i TPACK, noe som stort sett treffer godt med hva enhetene sa i intervjuet.

En mulig utfordring med undersøkelsen er forskjellen på programmering som et verktøy eller hjelpemiddel, og programmering som en egen kunnskap eller tema. Å avklare når programmering er et verktøy og når det er fagkunnskap kan være vanskelig. Her kunne vi stilt inngående og oppklarende spørsmål for å gi svarene en mer omfattende kontekst og større tydelighet (Postholm & Jacobsen, 2018 s. 125). Vi kunne også hatt en begrepsavklaring med informantene underveis eller i forkant av intervjuene, da hadde det vært tydeligere for dem at vi kategoriserte teknologi som når programmering ble brukt til å berike matematikk. Og når programmering ble behandlet som et kunnskaps- eller læreplanmål ble det kategorisert som faginnhold kunnskap.

Dette ble en utfordring i observasjonen med lærer 3, da vi observerte en undervisningstime i programmering hvor elevene kun hadde hatt en undervisningsøkt med python fra før. 8. trinns kompetansemål fra Utdanningsdirektoratet er å “utforske hvordan algoritmer kan skapes,

testes og forbedres ved hjelp av programmering” (Kunnskapsdepartementet, 2019). Her vil programmeringen hovedsakelig være en faginnholdskunnskap og ikke en teknologisk kunnskap. Likevel observerte vi teknologiske kunnskaper der læreren brukte programmeringen til å skape nye representasjoner hos elevene, som for eksempel tilknytningen mellom programmering av regnestykker og kalkulator.

I arbeidet med programmering beskrev alle tre lærerne at de behøvde kurs og egentrening i programmering før de følte seg komfortable med å bringe programmering inn i undervisningen, og at programmeringen var stort vektlagt i starten. I Monitor 2019 (Fjørtoft et al., 2019) besvarte lærere at deres utvikling av digital kompetanse ble til takket være prøving og feiling (83,1%), selvstudium (64,7 %), kollegaveiledning (58,3%) og felles opplæring med kollegaer (42%). Dette bygger opp under hypotesen om at læreren trenger å utvikle programmeringskunnskapene gjennom egentrening.

Gjennom spørsmål om hvilken kunnskap lærerne helst måtte klare seg uten, valgte 2 stykker teknologisk kunnskap som den minst viktige, og ville heller ha tydeligere fokus på pedagogikk og matematikk. Lærer 2 beskriver at pedagogikken må i bunn uansett fag, og at matematikken står for 95% av kompetansemålene, så programmeringen kommer sist i rekken. Likevel beskriver begge de to lærerne at programmeringen har begynt å spille en stor rolle i undervisningen der de ikke lengre ser utfordringer med programmering som undervisningsverktøy.

Samsvaret mellom de to lærerne kommer nok av at de har arbeidet sammen i flere år på samme skole. Lærer 3 uttalte derimot at man ikke klarer seg uten teknologien. Læreren ville heller klart seg uten den pedagogiske kunnskapen, og begrunner det med at man ikke kan hjelpe elevene med programmering dersom man ikke har den teknologiske kunnskapen til å forstå det selv. Ulikhetene i svarene kan være en tolkningsforskjell i spørsmålet. Lærer 3 besvarer tydelig ut fra en tanke om undervisning i programmering som hjelpemiddel. De to andre lærerne kan ha forstått spørsmålet som generell kunnskap, uavhengig om programmering er i fokus eller ikke. Det er derfor viktig å ikke lese alt for nøye inn i sammenligningen mellom de tre lærerne, da de kan ha svart på forskjellig grunnlag.

Imidlertid påpekte alle lærerne at alle tre hovedkunnskapene var viktige. Det er vesentlig å bemerke seg potensialet for at ulike oppfatninger av hva teknologi innebærer kan påvirke hva lærerne svarer her. Alle lærerne beskriver i løpet av intervjuet forskjellige metoder eller pedagogisk kunnskap om hvordan de skal bruke teknologien som hjelpemiddel. Lærer 1 og 2

sier begge at det var vanskelig og tidkrevende i begynnelsen, mens kunnskapen de har tilegnet seg etterhvert har gjort det lettere å bruke programmering som representasjon eller hjelpemiddel. Lærer 3 beskriver også at programmeringen oppleves som et godt hjelpemiddel, men nevner også utfordringen med den pedagogiske kunnskapen i forhold til programmering (Teknologisk-pedagogisk kunnskap). Dette kan være en følgefeil av kollegaer, ettersom læreren mente vedkommende selv måtte undervise hele trinnet i programmering. Fjørtoft et al. (2019) peker på at et fåtall av skoleledere sjeldent opplever at lærere observerer og gir tilbakemeldinger om bruken av digital teknologi i klasserommet.

I planleggingen av undervisning sier alle tre lærerne at de begynner med å ta utgangspunkt i et tema, eller i vårt tilfelle faginnholds-kunnskap, før de vurderer de pedagogiske mulighetene de har. Først når læreplanmålene og den faglige kompetansen er på plass ser de etter muligheten til å inkludere programmering. Lærerne kommer her med få eksempler på metoder eller aktiviteter de ønsker å bruke i undervisningen, men ettersom vi ikke spesifikt spurte etter undervisningsmetoder kan det være de ikke tenkte over å ta det med i svarene. Det er også en mulighet at lærerne anser programmering som selve aktiviteten og ikke ser nødvendigheten med å beskrive at elevene skal skrive selvstendige koder eller tavleundervisning.

### 5.1.2 Identifisering av kategorier i TPACK rammeverket

I dette delkapittelet prøver vi å besvare forskningsspørsmålet: “Hvilke kategorier i TPACK rammeverket identifiseres i den påståtte planleggingen og utføringen av undervisning i matematikk med programmering som verktøy i ungdomsskolen?” Vi skal her drøfte hvilke kategorier læreren har benevnt og hvordan beskriver læreren viktigheten av de nevnte kategoriene.

Pedagogisk kunnskap beskrives av to av tre lærere som grunnsteinen i arbeidet med undervisning i matematikk, også med programmering som verktøy. Dette støttes også opp i Niess et al. der utforsknings- og fremskrittstegene omhandler å velge en aktivitet der elevene får positiv utnyttelse av teknologien. Den siste læreren beskriver pedagogikk som en viktig del, men mener at pedagogikken er den siste kunnskapen av de tre grunnkunnskapene som trengs i programmeringsundervisning. I observasjonene noterte vi ned mange tilfeller av pedagogisk inngripen, og en klar pedagogisk plan på hvordan elevene skulle jobbe og lære.

Som nevnt i 5.1.1. beskriver alle tre lærerne at i planleggingsfasen begynner de med temaet de ønsker å undervise i. De snakker også generelt innenfor “kan” kategorien om Pedagogisk faginnholds-kunnskapen der det er få tilfeller av vanskeligheter rundt aktiviteter eller metoder rettet mot faginnholdet.

I teknologi -kunnskapskategorien var alle tre lærerne positive til programmering som hjelpemiddel. Lærerne beskrev at de har fått et større fokus på programmering de siste årene, og ser nå etter hvilke temaer de kan bruke programmeringen i. Når de skal inkludere teknologien, beskriver lærer 3 at det har vært vanskelig å vite hvilke metoder som fungerer på elevene. Lærer 1 og 2 beskriver at de føler de nå har bedre kontroll på dette, og at de har opparbeidet seg erfaring som er viktig for gjennomførelsen av undervisningstimene.

Alle tre lærerne har beskrevet en utfordring om kompetansemålene utdanningsdirektoratet har gitt angående programmering. Derav ser vi også mindre av “kan” TCK i intervjuene da lærerne beskriver at de bare har valgt seg et programmeringsverktøy, og håper dette dekker kravene til kompetansemålene. En av grunnene til dette kan også ha sammenheng med kursingen som er gitt til lærerne. De beskriver at de alle har vært gjennom kursing i programmering, men at kursene ikke har vært i en offisiell utdanning.

Totalt sett havner veldig mange av utsagnene til enhetene innen en kombinasjonskategori som teknologisk pedagogisk kunnskap eller teknologisk faginnholdskunnskap. Det at vi ser så mange overlappende kunnskaper er et tegn på at det lærerne klarer å knytte de ulike hovedkunnskapene sammen for å gi elevene en meningsfull undervisning. Gjennom empirien så vi at alle informantene har flere situasjoner hvor TPACK er komplett. Vi bemerker oss også at begge datainnsamlingsmetodene har gitt flere merknader i overlappskategoriene i TPACK. Totalt sett har vi funnet flere tilfeller av alle grunnkategoriene, og alle overlappskategoriene. Det virker dermed som at lærerne klarer å kombinere kunnskapene til å lage et undervisningsopplegg. Dette beskrives som nødvendig for å integrere teknologi i undervisningen (Mishra og Koehler, 2006, sitert i Koehler & Mishra 2009)

### 5.1.3 Integrering av programmering

I delkapittelet her svarer vi på det siste forskningsspørsmålet: “I hvilken grad utfyller lærerne stegene i Niess et al. (2009) sin 5-steps modell om integrering av teknologi, spesifikt programmering, i matematikkundervisning på ungdomsskolen? “

Vi ser på informantens besvarelser opp mot Niess et al. (2009) sine 5 steg for teknologisk integrering i klasseromssituasjoner. Da ser vi at alle har oppfylt første steg, gjenkjennelse. Alle har tatt kurs eller fag om programmering for å selv ha en bedre forståelse. Da vi intervjuet lærerne hadde vi med et spørsmål om de så noen muligheter og utfordringer med programmering som verktøy i matematikkundervisningen. Det at informantene kunne gi omfattende svar på dette spørsmålet forteller oss at de også har oppfylt andre steg i modellen, akseptere.

Noe som var felles for alle informantene var oppgittheten over utydelighetene om hvordan man bør gå frem for å bruke programmering i undervisningen. Fra utdanningsdirektoratet er det svært få føringer om hvilket programmeringsverktøy som skal brukes, og hvilke krav som skal settes til elevene. Som nevnt i 2.3 finnes det kompetansemål elevene skal igjennom, men våre informanter beskrev vanskeligheter med å tyde kompetansemålene godt nok. I grunnleggende ferdigheter hos Utdanningsdirektoratet står det at elevene skal utvikle digitale ferdigheter gjennom bruk av digitale verktøy. Og gjennom læreplanmålet etter 10. trinn at elevene skal utforske matematiske konsepter. (Kunnskapsdepartementet, 2017) (Kunnskapsdepartementet, 2019). Dermed står lærerne fritt til å se hvilken måte som passer best å benytte verktøyet. Hvis vi ser tilbake til 5-stegs modellen vi diskuterte over, kan det sammenfalle med trinn 2. Trinnet går ut på at man aksepterer programmering som et verktøy, benytter dette og ser de ulike mulighetene og utfordringene som medfølger. For våre informanter var ikke det et problem da alle hadde det første steget ivaretatt gjennom ekstra kursing. For den gjennomsnittlige lærer er det ikke nødvendigvis slik at man har denne typen kurs eller interesse til å sette seg inn i det på egen hånd. Informantene uttrykte også at kollegaer og andre lærere ikke hadde denne kunnskapen. Derfor kan det å vurdere hvilken retning man skal ta i matematikkundervisningen være et stort hinder, dersom man ikke har oppfylt første steg. Da tror vi at en type situasjon kan oppstå, hvor lærerne sammen med elevene tilegner seg den nødvendige kunnskapen. Da blir ikke programmeringen en berikelse til matematikken, og det blir heller en utfordring å få nok matematikk inn i undervisningen av “programmering” eller koding. Dette samsvarer med Sanne-utvalget sin konklusjon om at programmering bør gå inn under et eget fag, og ikke matematikk (Sanne et al., 2016).

Det tredje steget kan være vanskelig å identifisere da det går ut på at lærerne endrer deres pedagogiske praksis og samarbeider om hvordan teknologien, eller programmeringen, skal inkluderes i undervisning. Våre caser gir ikke innblikk i hvordan lærerne har arbeidet over



tid, men ut fra hva vi har observert og hva de har uttrykt tyder det på at informantene har oppfylt tilpasning også.

Steg 4 går ut på at læreren utforsker hvilke temaer teknologien kan berike og hvor det gir mening å ta det i bruk. Hadjerrouit og Hansen (2022) understreker at arbeidets algoritmiske natur skal gi berikelsen man ikke oppnår gjennom tradisjonell undervisning. Gjennom intervjuene kom det frem at informantene hadde gjort seg opp en mening om hvilke temaer det egnet seg å bruke programmering som verktøy. De har brukt programmeringen som verktøy i undervisningstimer, og testet hvordan elevenes utbytte av undervisningen var. Det kan være verdt å bemerke seg at observasjonen av lærer 3 ikke ga bekreftelse av dette, siden timen i stor grad gikk til opplæring av koding i Python. Læreren hadde derimot gode refleksjoner rundt dette i intervjuet. Derfor mener vi at også dette steget er oppfylt.

I det femte og siste steget skal læreren inkludere programmering som et av de grunnleggende verktøyene for å løse problemer i undervisningen, og det skal brukes for at elevene skal få en dypere forståelse av matematikken. Videre skal læreren gjøre opp egen evaluering av undervisningsmetoden, og om metoden er tilstrekkelig. Her er læreren avhengig av at en selv har gått gjennom de fire foregående stegene, samt at elevene må ha god nok kjennskap til verktøyet og faget. Alle tre lærerne har hatt refleksjoner rundt utbytte av programmering som verktøy i intervjuet. I intervjuene ble det beskrevet temaer som var tilpasset til programmering, men ikke mye om hvordan de spesifikt brukte programmeringen som representasjon. Ut ifra observasjonene kan det antas at introduksjon på tavle eller smartboard, og så oppgavejobbing med programmering, er den vanligste fremgangsmåten.

Fremskritt-steget er vanskelig å oppfylle, ettersom kriteriene timen vurderes ut ifra forandres gjennom tid. Koehler et al. (2013) beskriver at teknologisk kunnskap stadig må forbedres og utvikles for å gjelde. Litt det samme gjelder med Niess et al. (2009) sitt 5. steg. Elevene varierer i hvor mye de kan om programmering, faglig kompetanse, kompetansemål forandres og selve teknologien forandres. Lærerne beskriver i flere tilfeller at de har vurdert programmeringsmetoden de bruker. Blant annet beskriver lærer 3 at elever som synes det er vanskelig med Python kan gå over til blokkprogrammering. Vi mener informantene til en viss grad oppfyller dette steget, men kanskje er det ikke mulig å si at man er ferdig utlært, da hver klasse er unik og krever en annen tilnærming. Lærer 3 påpekte også at det var ytterst få elever som hadde en god programmeringsforståelse i løpet av ungdomsskolen, og at det var noe han så for seg var mer naturlig på videregående- eller universitetsnivå.

En forskjell vi så mellom informantene på 9. trinn og den på 8. var at de førstnevnte fikk i større grad brukt programmering som verktøy. Dette kan det være flere årsaker til. En grunn kan være at elevene var eldre eller mer kjent med å bruke programmering som verktøy for problemløsning i matematikk. En annen kan være de rammene som ble gitt av læreren. I observasjonen av læreren på 8. trinn bemerket vi oss at kodingen ble gjort i Python, i motsetning til de andre lærerne som valgte blokkprogrammering i Scratch. Andersen (2022) fant ut at gjennom å arbeide aktivt i fellesskap med blokkprogrammering som verktøy, tilegnet elever seg CT ferdigheter og matematiske egenskaper. Vi ser at CT og programmering opptrer sammen, dette trolig av tankeprosessen man gjennomgår når man skal løse et problem ved hjelp av programmering. Integrering av programmering medfører trolig også tankeprosessen CT innebærer.

Blokkprogrammering har sine begrensninger, men lærerne var kjent med disse og lot elevene finne ut hvordan man kunne løse problemene gitt disse ekstra beregningene. Det at 8. klassen ikke var like komfortabel med å bruke Python, kan ha vært med på å gjøre at vi observerte i mindre grad at dette ble brukt som verktøy for berikelse av matematikkundervisningen.

Gitt rammene for vår studie er det vanskelig å si noe generelt om lærere. Vi fikk et innblikk av hvordan informantene opptrer i én undervisningstime, samt et intervju hvor informantene skulle fortelle om deres praksis. Det er ikke tilstrekkelig med data til å generalisere den enkelte læreres praksis ut fra disse dataene, men vi kan si noe om graden av integrering fra det inntrykket vi har fått.

## 5.2 Pedagogiske og praktiske implikasjoner

I dette delkapitlet ønsker vi å se på hvilke pedagogiske og praktiske implikasjoner det har for læreren i klasserommet dersom TPACK er ivaretatt i undervisningen. For å gjøre det må vi avklare hva det vil si at TPACK er ivaretatt. Med det mener vi at læreren har adaptert en ny teknologi i sin undervisningspraksis i matematikk, og utnytter teknologien på en hensiktsmessig måte for å skape nye representasjoner for elevene. Da har læreren fått til å benytte verktøyet på en meningsfull måte, hvor det beriker undervisningen. For å kunne si om teknologien beriker undervisningen behøver læreren å gjøre en evaluering om den teknologiske metoden er passende, eller om det eventuelt er noen forbedringer som kan gjøres. Som Cox og Graham (2009) påpeker er teknologien fluktuerende, derfor er det nødvendig at læreren gjør denne typen vurdering ofte. Selv om læreren har kunnskap om

teknologien, er det derimot ikke gitt at den blir integrert på en berikende måte i undervisningen. Niess et al. (2009) sine siste steg om integrering er sentral i denne vurderingen. Å utforske muligheter og evaluere utbytte og respons hos elevene.

Vi så at alle lærerne var positive til programmering som verktøy i matematikkundervisningen. Likevel uttrykte de at fokuset ofte kan bli i overkant mye på selve programmeringsstegene, og bort fra programmeringsprosessen og matematikken. Det at informantene våre føler mye av tiden går til å lære bort syntaks og hvordan man koder, tyder på at elevene ikke innehar den kunnskapen de behøver for å benytte verktøyet på en måte som beriker matematikken. Dermed må lærere være bevisst på at elever stiller med svært varierende kunnskaper om programmering. Lærer 1 påpekte om lærere i Norge: ”Noen synes det er gøy å gjøre det fordi det er gøy og de kan det og alt. Men noen har ikke peiling. Og det er klart, elevene blir jo da, liksom prisgitt etter hvilken lærer de har.” Disse faktorene kan utløse en situasjon hvor den eller de lærerne på skolen som har kompetanse innen programmering, blir nødt til å ha undervisning for de ulike klassene. Lærer 1 påstod at denne situasjonen oppstod i fjor, og at de måtte gå inn i hver 10. klasse og kurse elevene i det de måtte for å kunne besvare den type oppgaver som hadde kommet på eksamen tidligere. I observasjon av lærer 3 ble vi gjort oppmerksom på at vedkommende også skulle holde en tilsvarende undervisningsøkt i de tre gjenstående parallellklassene på 8. trinn. Når slike tilfeller oppstår tror vi kan det være utfordrende for læreren som må gjennomføre disse timene. Ikke nødvendigvis bare på hensyn av mengden undervisning, men også med hensyn til de tilpasningene man gjør for hver enkelt klasse når man kjenner dem godt.

Enda et fellestrekk ved våre informanter var at de alle hadde ekstra kursing på programmering, og var positive til dette. For at alle elever skal få tilstrekkelig undervisning om og gjennom programmering i løpet av hele skolegangen, kan det være et forslag å etterutdanne lærere i programmering. På denne måten kan lærere bli tryggere på egen praksis, og programmering blir ikke nedprioritert i løpet av året. Dette vil kunne skape muligheter der elevene får undervisningsøkter med programmering fordelt over hele læringsløpet.

Læreren må være bevisst på sin rolle i arbeid med programmering og CT blir diskutert av Kaufmann og Stenseth (2023) samt Hadjerrouit og Hansen (2022). Læreren inntar en rolle som veileder og ikke kunnskapsformidler. Dette kan oppleves som uvant for enkelte, men likevel viktig for at elevene skal oppleve å ta eierskap i tankeprosessen og utviklingen av løsningene på deloppgavene de har skapt seg.

## 5.3 Kritiske refleksjoner om metode og teori

Under kommer vi til å gå gjennom noen kritiske refleksjoner knyttet til masteroppgavens teori og metode. Vi kommer til å diskutere utfordringer, svakheter og presentere alternative strategier som kunne blitt benyttet.

### **Teori**

I arbeidet med å velge ut relevant forskningslitteratur forsøkte vi å holde oss så objektive som mulig og se på kvaliteten av studiene og relevansen for vår egen oppgave. Artikler som ble grunnlaget for masteroppgaven er basert på en nøyere gjennomgang der vi har prøvd å finne utgivelser som er korrekt dokumentert, og enten er eller refererer til originalkilden av teorien. Likevel er det alltid en risiko for at teorien funnet ikke er tilstrekkelig. Vi har derfor prøvd å finne flere teorier som bekrefter eller avkrefter teorien funnet. Vi har valgt et forskningsområde som det finnes mindre konkret litteratur på. Kombinasjonene matematikk & programmering og matematikk & TPACK finnes det noe litteratur. Når vi derimot så etter studier gjort hvor man undersøkte på hensyn av alle tre, gikk mengden treff betraktelig ned. I oppgaven har vi tydelig fremstilt resultatene, og det stemmer overens med de teoretiske formuleringene vi har formulert. Imidlertid åpner dette muligheten til å være kritisk til selve rammeverket. TPACK-rammeverket er en generell modell som tilsier at pedagogisk faginnholds- og teknologisk kunnskap er avgjørende, ikke noe om graden av viktigheten blant de tre komponentene. Undersøkelsen vår baserer seg på empiriske data der vi har intervjuet og observert informantene beskrevet. Dette er ikke et konkret teoretisk spørsmål, som kan være en ulempe, spesielt i form av generalisering.

Utfordringen gjør at det er et spennende forskningshull å utforske. Siden forskningshullet er av betydelig størrelse, har vi ressurser eller tid til å dekke problemstillingens omfang. Det er derfor et spørsmål det er mulig å forske videre på. I tillegg til forskningen har vi også sett oss nødt til å sammenligne teori på tvers av funn i ulike studier for å sammenligne om våre resultater samsvarer med noe av det vi kunne forventet å finne. Denne prosessen krever naturligvis en form for tolking av oss for å sette resultatene i en kontekst. For å sikre god kvalitet i litteraturen vår har vi tatt utgangspunkt i fegelleverderte artikler publisert i anerkjente journaler, nasjonalt og internasjonalt. Noen ganger har jakten på primærkilder

sendt oss i retning av bøker og artikler som ikke er peer reviewed etter samme strenge kriterier. I disse tilfeller har vi vært oppmerksom på å vurdere hver enkelt kilde for å sikre at kvaliteten ikke blir svekket. Videre har vi sett etter vitenskapelig oppbygning og metode, samt å sjekke om referanselisten viser til gode og trygge kilder.

For å ankre den teoretiske bakgrunnen baserte vi oss på artikler av Kaufmann og Stenseth for å knytte programmering og algoritmisk tenkning opp mot matematikkundervisning i den norske skolen. Vi har brukt Kunnskapsdepartementet (2019) sine læreplanmål for å beskrive hva læreren må kunne lære bort. I tillegg har vi undersøkt Sanne-utvalget sin rapport om programmering for å definere og få en oversikt over hva en ekstern kilde mener om programmering i skolen. Som teori til TPACK har vi tatt utgangspunkt i originalkildene Koehler og Mishra ( & Cain), samt artikkel fra Cox og Graham for å definere TPACK. Vi har prøvd å lese oss opp på forskjellige artikler og masteroppgaver som omhandler TPACK som analyseverktøy, for å sammenligne fremgangsmåte og selve bruken av TPACK. Masteroppgavene er ikke referert til i oppgaven, da vi ikke har brukt data eller teori i vår forskning.

TPACK rammeverket er et av mange verktøy som kan brukes til å forske på programmering. En annen mulighet er Vygotsky sin sosiokulturelle teori. Den sosiokulturelle teorien omhandler den sosiale konteksten der relasjon mellom lærer og elev gir bedre innsikt i hvordan undervisningen bør legges opp for elevenes utbytte (Moen, 2013). I dette tilfellet ville fokuset blitt større på hvordan samholdet mellom lærer og elev er, og hvordan dette kan brukes til å implementere programmering i undervisningen.

## **Metode**

Vår metode er i stor grad påvirket av annen kvalitativ forskning på samme fagfelt. Intervju og observasjon var kombinasjonen vi landet på, da vi ønsket å ha flere innsamlingsmetoder som kunne underbygge hverandre.

Vi hadde i utgangspunktet planer om å ha informanter som underviste på 9. trinn. Dette for å kunne spisse forskningsområdet mest mulig. Videre ønsket vi oss fire til fem informanter som kom fra ulike skoler for å ha god bredde i datagrunnlaget slik at det ville være mindre sjans for at tilfeldigheter spilte inn på funnene våre. Etter å ha startet opp prosjektet og startet søket etter informanter innså vi at det ikke lot seg gjøre å ha disse strenge kriteriene til informanter, siden vi da hadde blitt stående uten datagrunnlag. Etter å ha reflektert over situasjonen, kom

vi frem til at de tre informantene vi hadde ville passe studiens omfang godt. Hadde vi hatt flere informanter kunne bredden av data vært større, men da på bekostning av dybde. Dette ville ikke vært gunstig med tanke på studiens forskningsdesign, multippel case-studie, som baserer seg på konteksten i hver enkelt case. For å utforske denne konteksten vil dybde være en viktigere faktor (Postholm & Jacobsen, 2018).

Det at vi havnet i en situasjon hvor vi ikke visste om vi måtte endre fokuset i masteroppgaven eller endre søket vårt etter informanter, gjorde at vi ikke fikk gjort en pilotundersøkelse for å teste observasjonsskjema og intervjuguide i forkant av datainnsamlingene. I observasjonstimene gjorde vi ingen opptak av lyd eller video. Vi la merke til at vi ikke hørte like tydelig hva læreren snakket om i dialog med elevene som satt lengst bort fra oss. En alternativ strategi ville vært å enten ha en mikrofon på informanten gjennom timen. Eventuelt kunne vi endret observatørrolle, og beveget oss litt mer rundt i klasserommet. I Postholm og Jacobsen (2018) foreslår de også lignende tiltak, men nevner også at slike hjelpemidler kan være forstyrrende og at forskeren bør vurdere hva som er nødvendig.

Vi sendte informasjon om oppgaven og medfølgende samtykkerklæring (se vedlegg 4) til informantene, samt et informasjonsskriv til elever og foresatte. Annet forarbeid som kunne vært med på å påvirke svarene i intervjuene vi gjennomførte hadde vært å sende intervjuguiden på forhånd, så lærerne kunne sette seg inn i hva vi skulle spørre om. Likevel er det ikke sikkert at vi hadde fått deres genuine og opprinnelige tanker om temaene. En risiko vi forespeilet ved å sende intervjuguiden på forhånd var at lærerne forbereder svar som de mener bedre tilfredsstillende det som forventes av en lærer i klasserommet. Likevel kunne vi fått svar som var mer gjennomtenkt eller at lærerne hadde hatt tid til å sette seg inn i TPACK og da kunne gitt mer utfyllende svar om dette i intervjuet. Da dette var det vi i hovedsak så etter. Det ble derfor opp til oss i analysen å tolke det lærerne sa og plassere det inn i rameverket, i stedet for at de selv kunne plassere deres praksis der de følte den passet.

En annen potensiell svakhet i datainnsamlingen var tidspunktet for observasjon av lærer 3. Som nevnt over, hadde vi utfordringer med å skaffe informanter, og ble helt nødt til å gjennomføre datainnsamlingene i stor grad på deres premisser. Det innebar å utsette observasjonen av lærer 3 til en senere dato etter intervjuet. Når det gjelder skolen som lærer 3 jobbet på, hadde en av studentene tidligere jobbet på den tilknyttede barneskolen. Likevel var det ingen relasjon til ungdomsskolen eller de ansatte der.

En av fordelene med å være flere om dette prosjektet er at vi kan passe på hverandres praksis. Vi har opplevd det som nyttig å ha en i ryggen som dobbeltsjekker arbeid og uttalelser, slik at man opptrer etter våre retningslinjer og at vi gjør oss forstått på rett måte. Det at begge studentene har hatt tilgang til transaksjonsdokumentene og lydfilene gjør at vi hele veien kan bekrefte at det arbeidet som blir gjort er av god kvalitet og i henhold til de retningslinjene vi har sagt vi skal føye oss etter. For å bli enda tryggere på at vi har forstått informanten korrekt i intervjuet og derav i transkripsjonen, har vi sendt hver informant sitt intervju transkribert, totalt anonymisert og uten personopplysninger. Dermed kan informantene vurdere om vi har forstått dem rett, og om det er noe som eventuelt må oppklares i et oppfølgingsintervju eller om dette resonnerer så lite med deres meninger at de ønsker å trekke seg fra studien. Dette er også med på å styrke troverdigheten av funnene i denne typen kvalitativ forskning (Bryman, 2016) En mulig feilkilde her vil være at våre informanter er lærere som allerede har mye å gjøre. Å be dem lese gjennom 20 sider transkripsjon kan ende opp langt nede på prioriteringslisten av arbeid som må gjøres. Derfor kan vi ende opp med å få bekreftelse på noe de ikke mener, siden de ikke har lest nøye nok.

Når intervju dataene er validert var det opp til oss som forskere å kategorisere svarene vi fikk. Slik tematisk analyse er beskrevet i Braun og Clarke (2006) sine seks faser vil man vanligvis luke bort eller justere kategoriene så det er minst mulig overlapp og stor tydelighet i hver kategori. Dermed blir det også lettere å besvare forskningsspørsmål ved hjelp av resultatene. I vår TPACK analyse er kategoriene forutbestemt, dette sammen med at TPACK modellen baserer seg på et venn diagram med de tre hovedkunnskapene teknologi, pedagogikk og faginnhold, gjør at det er mye overlapp i kategoriene våre. noe som har gjort det utfordrende for oss som forskere å skille læreres påstander i de ulike kategoriene. For å minske variasjoner i kodingen tok vi utgangspunkt i to etterfølgende, tilfeldige sider fra ett av intervjuene. Disse ble brukt i en pilotanalyse vi gjennomførte hver for oss, hvor vi brukte tabellen for TPACK analyse. I etterkant satt vi oss ned sammen for å sammenligne hvordan vi hadde kodet i denne pilotanalysen. Vi ble enige om at kodene sammenfalt til et tilstrekkelig nivå hvor det ikke var nødvendig med ny pilotanalyse. Likevel ble vi enige om noen konkretiseringer i kjennetegn for de ulike kategoriene vi hadde. Etterfulgt av at vi gikk gjennom hele første transkripsjon sammen. Etter dette så vi det som en tidsbesparelse å kode hvert vårt av de gjenstående transkripsjonene, da vi hadde stor enighet og likhet i hvordan analysen ble utført på dette tidspunktet. Sammenlignet med tematisk analyse har da noen av

de avsluttende fasene blitt sløyfet til fordel for konkretisering i beskrivelsene, da vi skulle bevare de opprinnelige kodene våre.

Selv om vi så på vår enighet som tilstrekkelig for å sikre at analysen ble utført konsekvent på samme måte betyr ikke det at det er uten påvirkning. Vi vil ha et noe farget syn basert på de oppfatninger og forkunnskaper vi har med oss. Derfor vil det hende at vi har valgt å kode noe som en kategori, hvor andre forskere ville kodet ulikt.



## 6 Avslutning

I dette kapitlet kommer vi til å starte med konklusjon av problemstillingene. Konklusjonen er en kort oppsummering av resultatene og drøftingen, samt selve konklusjonen i oppgaven. I neste punkt tar vi for oss noen retninger man kan gå videre i forskningen som ikke falt under denne masteroppgavens omfang. Til slutt avsluttes oppgaven med individuelle refleksjoner fra oss forskere om oppgaven.

### 6.1 Konklusjon

*(1) Hvordan vektlegger lærerne teknologisk-, pedagogisk- og faginnholdskunnskap i arbeidet med programmering i matematikkundervisning på ungdomsskolen?*

I både intervjuer og observasjoner har vi oppdaget flere tilfeller av de tre komponentene pedagogisk, teknologisk og faginnholdskunnskap. Koehler og Mishra (2009) beskriver de tre kunnskapene som nødvendige både alene og i samhold for å kunne hensiktsmessig undervise med teknologi. De tre komponentene ble identifisert tilnærmet like mange ganger i løpet av undersøkelsen, med unntak av en nedgang av benevninger av pedagogisk kunnskap hos lærer 3. De tre forskningsobjektene uttrykker også viktigheten av balansen mellom disse. Lærerne beskrev et fokus på temaet, eller faginnholdskunnskapen, i begynnelsen av planleggingen og skiftet fokus over til elevaktivitet eller undervisningsmetode. Her kom også teknologisk kunnskap inn, der lærerne vurderte om programmering passet til temaet. Den største forskjellen vi fant i forskningen er hvilken kunnskap lærerne minst vektlegger, der to mente de kunne klart seg uten teknologisk kunnskap, mens lærer 3 anså pedagogikken som den minst viktige kunnskapen. Likevel har alle tre lærerne tatt et etterutdanningskurs i programmering, noe som peker på viktigheten av den teknologiske kunnskapen.

*(2) Hvilke kategorier i TPACK rammeverket identifiseres i den påståtte planleggingen og utføringen av undervisning i matematikk med programmering som verktøy i ungdomsskolen?*

Våre funn av data samsvarer med Mishra og Koehler (2006) sin påstand i Koehler & Mishra (2009) om at alle kunnskapene er nødvendige og må brukes sammen for å undervise hensiktsmessig. De tre grunnkunnskapene teknologi, pedagogikk og faginnhold ble funnet i alle stadier av undervisningen. I tillegg til dette fant vi mange tilfeller av kombinasjoner av

kunnskapene. Lærerne nevnte ved flere tilfeller av at teknologien var en god representasjon av faginnhold, og en positiv aktivitet for elevene. Også i observasjonene fant vi tilfeller av alle de forskjellige kategoriene, der det var en relativt jevn fordeling av kunnskapene brukt i undervisning. Dette tyder på at lærerne er flinke til å variere mellom de forskjellige kunnskapene, og kombinere dem når det er hensiktsmessig.

Vi identifiserte flere tilfeller av TPACK hos alle informantene i både intervju- og observasjonssammenheng. Lærerne har vist kunnskaper innen alle de tre kategoriene teknologi, pedagogikk og faginnhold, samt kombinasjonen av disse.

*(3) I hvilken grad utfyller lærerne stegene i Niess et al. (2009) sin 5-steps modell om integrering av teknologi, spesifikt programmering, i matematikkundervisning på ungdomsskolen?*

Resultatene våre tilsier at alle 3 lærerne hadde oppfylt de 4 første stegene i Niess et al. (2009) sin modell for teknologisk integrering i klasserommet. Gjenkjennelses- og aksepterings-steget ble oppfylt der alle lærere hadde etterutdannet seg innen programmering. I tillegg fortalte de at de hadde jobbet alene med å utvikle ferdighetene, og teste forskjellige programmeringsverktøy. 2 av 3 lærere beskrev utfordringer i læringen av programmering, da det krevde mye selvstendig arbeid for forståelse. Ettersom to av lærerne jobbet i fellesskap fikk vi innsyn i planleggingsfasen av innføring av programmering for elevene. I deres tilfelle havnet avgjørelsen på Scratch, da blokkprogrammering ble ansett som det riktige valget for de. Lærer 3 beskrev at han ble arbeidende litt alene med planleggingen, der informanten i utgangspunktet ville jobbe med Python. Dette viser til det tredje steget i Niess et al. (2009) sin modell der lærerne skal søke aktiviteter for å tilpasse matematikkundervisningen til programmering. Det siste steget vi med sikkerhet kan si lærerne har oppfylt er utforskningssteget. Her har de prøvd programmering i klasserommet, i håp om at elevene fant undervisningen meningsfull.

I det siste steget, fremskritt, skal læreren aktivt vurdere effekten av programmering som hjelpemiddel for elevene. Det ble beskrevet utfordringer for lærerne, der de undret på hvordan programmeringen skulle brukes som et riktig hjelpemiddel. Likevel så vi et potensial for at dette steget også skulle bli oppfylt. Den teknologiske pedagogiske kunnskapen hos lærerne virker god, samt de har gode refleksjoner rundt programmeringen. Utbyttet av programmeringen virker likevel ikke perfekt, da det er rom for forbedring. Vi har også stilt

spørsmålet om det er mulig å utfylle dette steget fullstendig, da omgivelsene og vilkårene til lærerne stadig er i utvikling.

## 6.2 Videre forskning

I arbeidet med masteroppgaven avdekket vi noen interessante spørsmål som ikke falt under masterens omfang, og som vi derfor ikke får undersøkt. Dermed kommer vi her til å presentere noen spørsmål som kan forskes videre på, av oss på et senere tidspunkt eller av den heldige leser av dette.

1. Hvor mye kunnskap trenger lærere om programmering for å effektivt benytte dette som verktøy i matematikkundervisning?

Vi så at alle våre informanter hadde rene programmeringsfag eller videreutdanning/ kursing i programmering, dermed stilte vi oss naturligvis spørsmålet om hvor mye en lærer må kunne. For å ha et rent a-didaktisk opplegg har få til ingen krav til læreren, da tilpasser læringsløpet oppgavene og vanskelighetsgraden på dem. I tillegg har den didaktiske refleksjonen allerede blitt gjort av utvikleren. Men hva hvis læreren skal gjennomføre et egenutviklet didaktisk matematikkopplegg hvor man benytter programmering som verktøy i større eller mindre grad. Holder det å bare få til det samme som elevene skal lære? Vi fikk høre fra informantene og har hørt fra andre lærere at noen elever overgår deres nivå når det kommer til akkurat dette med programmering. Det blir ikke direkte relevant å sammenligne kunnskap om programmering med kunnskap innen de ulike matematiske temaene, men hvis vi gjør det stiller lærerutdanningen krav om at studentene ligger på et betraktelig høyere nivå enn det de selv skal undervise i. Man kan jo da stille spørsmålet om man skulle hatt samme kriteriet til lærere når det kommer til programmering. På den andre siden kan man også argumentere for at alle har vært gjennom matematikk i løpet av hele utdanningsløpet. Man kan jo tenke at majoriteten av lærere i skolen i dag begynte å jobbe etter at LK20 trådte i kraft. Disse vil være avhengig av at skolen og kommunen har ressurser til å etterutdanne og kurse for å tilegne seg de nødvendige kunnskapene om ikke de skal gjøre det på eget initiativ.

2. Hvordan foregår den interrelasjonelle læringen hos elevene når læreren ikke innehar tilstrekkelige ferdigheter i programmering?

Kaufmann og Stenseth (2023) problematiserer deres påstand fra Kaufmann og Stenseth (2021) om at elever ofte blir selvstendiggjort i arbeid med programmering. De diskuterer

også at en mulig årsak er lærerens manglende kompetanse. Vi lurer da på om, og isåfall hvordan læring foregår mellom disse elevene og hvordan dette kan knyttes til Lev Vygotskys sosiokulturelle læringsteori. I motsetning til vår studie vil dette forskningsområdet vært rettet mot elevene og deres utbytte, gitt situasjonen hvor læreren har manglende kunnskap. Vi avdekket ikke disse manglene hos våre informanter, men de uttrykte det som et utbredt problem.

### 3. Hvor lang tid før læreplanen LK20s konsepter er innført i alle ledd i skolen?

I samtale med informantene våre kom det frem at ungdomsskoleelever ikke har de nødvendige ferdighetene til å programmere på et nivå hvor det beriker matematikundervisningen. Hvor lang tid vil det ta før lærere gjennom hele skoleløpet er komfortable med å bruke denne arbeidsmetoden slik at elever som kommer på ungdomsskolen er kjent med hele prosessen som programmering er. Slik at de kan utforske matematiske konsepter gjennom programmering slik det er beskrevet under læreplanmålene i matematikk etter 10. trinn, men også trekke “de lange linjene” mellom CT/algoritmisk tenkning, programmering og matematisk tenking.

## 6.3 Egne refleksjoner

### 6.3.1 Student 1

Jeg har satt pris på arbeidet med masteroppgaven, samtidig har jeg fått kjenne på den belastningen det har vært. Likevel sitter jeg igjen med en positiv ettersmak på det hele, da jeg tolker det som at masterprosjektet har hatt en viss betydning for meg. Målet har hele tiden vært å bli lærer, og på en måte har dette prosjektet noen ganger virket som et hinder. Tross alle negative tanker ser jeg tilbake på det som en læringsprosess, både om TPACK, programmering og akademisk arbeid, men også om meg selv. Når lærerutdanningen krever en avslutning i form av en masteroppgave, har jeg opplevd det som nyttig å skrive om noe både jeg og kanskje andre kan dra nytte av i lærerprofesjonen.

### 6.3.2 Student 2

Masteroppgavens innhold har vært både utfordrende og interessant å skrive om. Jeg har lært mye i løpet av prosessen, der jeg ikke hadde stor kunnskap til TPACK rammeverket eller dets omfang før arbeidet med oppgaven startet. Jeg er fornøyd med å ha valgt et tema som har

passet så bra til mitt interessefelt, der jeg har kunnet utforske hvordan kunnskapene kommer frem i planlegging og undervisning av programmering. Jeg vil påstå dette kommer til å være en nyttig erfaring for meg som fremtidig lærer generelt, og ikke minst innen programmering.

Utfordringen med valg av metode, teori og analyse ble veldig mye lettere takket være god hjelp fra medforfatter (Student 1) og veileder. Dermed gikk det relativt raskt å finne både intervjuguide og observasjonsguide til innhenting av data. Den største uforutsette utfordringen var vanskeligheten med å finne lærere som var villige til å bli med i undersøkelsen. Det er en fin erfaring å ha, ettersom det setter forskningen litt i et nytt syn.

Analyse og diskusjon var nok den vanskeligste biten av oppgaven, spesielt da vi ikke fant forskning som hadde lik fremgangsmåte som oss. Det ble derfor en lang prosess med gjennomgang av de forskjellige avsnittene, og mye redigering. Heldigvis er jeg fornøyd med utfallet til slutt.

Jeg har nå fått et godt innblikk i hva som kreves av en god programmeringslærer, og har fått gode eksempler på oppgaver og temaer jeg kan bruke programmering til. I tillegg har jeg fått en større forståelse av hvordan jeg kan forske eller undersøke forskjellige fenomener i skolen, noe som vil passe godt inn under egenvurderinger og videreutvikling av skolene.

## 7 Referanseliste

### Litteratur

- Aalbergsjø, S. G. (2022). Learning to make and use computer simulations in science education. *Acta Didactica Norden*, 16(4) <https://doi.org/10.5617/adno.9174>
- Andersen, R. (2022). Block-based programming and computational thinking in a collaborative setting: A case study of integrating programming into a maths subject. *Acta Didactica Norden*, 16(4). <https://doi.org/10.5617/adno.9169>
- Baran, E., Chuang, H.H. & Thompson, A.(2011) TPACK: An Emerging Research and Development Tool for Teacher Educators. *Turkish Online Journal of Educational Technology - TOJET*, 10(4), 370-377  
[https://www.researchgate.net/publication/265215357\\_TPACK\\_An\\_emerging\\_research\\_and\\_development\\_tool\\_for\\_teacher\\_educators](https://www.researchgate.net/publication/265215357_TPACK_An_emerging_research_and_development_tool_for_teacher_educators)
- Braun, V., & Clarke, V. (2006). Using *thematic analysis in psychology*. *Qualitative research in psychology*, 3(2), 77-101. <https://doi.org/10.1191/1478088706qp063oa>
- Bryman, A. (2016). *Social research methods*(5. utg.). Oxford: Oxford University Press.
- Christoffersen, L. & Johannessen, A. (2012). *Forskningsmetode for lærerutdanningene*. Abstrakt forlag
- Cox, S. & Graham, C. R. (2009) Diagramming TPACK in Practice: Using an Elaborated Model of the TPACK Framework to Analyze and Depict Teacher Knowledge. *TECHTRENDS TECH TRENDS* 53(5), 60–69  
<https://doi.org/10.1007/s11528-009-0327-1>
- Fangen, K. (2010). *Deltagende observasjon* (2. utg.). Fagbokforlaget.
- Fjørtoft, S.O., Thun, S. & Buvik, M.P. (2019). *Monitor 2019. En deskriptiv kartlegging av digital tilstand i norske skoler og barnehager*. Utdanningsdirektoratet.  
[https://www.udir.no/contentassets/92b2822fa64e4759b4372d67bcc8bc61/monitor-2019-sluttrapport\\_sintef.pdf](https://www.udir.no/contentassets/92b2822fa64e4759b4372d67bcc8bc61/monitor-2019-sluttrapport_sintef.pdf)

- Forsström, S. E. & Kaufmann, O. T. (2019). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18-32. <http://hdl.handle.net/11250/2599710>
- Hadjerrouit, S. & Hansen, N. (2022). Undergraduate Mathematics Students Engaging in Problem-Solving Through Computational Thinking and Programming: A Case Study. I D. Ifenthaler, P. Isaïs & D. G. Sampson (Red.), *Orchestration of Learning Environments in the Digital World* (s. 197–214). Springer International Publishing. [https://doi.org/10.1007/978-3-030-90944-4\\_11](https://doi.org/10.1007/978-3-030-90944-4_11)
- Johannessen, A., Tufte, P. A., & Christoffersen, L. (2016). *Introduksjon til samfunnsvitenskapelig metode*(5. utg.). Abstrakt Forlag.
- Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 52(7), 1029–1048. <https://doi.org/10.1080/0020739X.2020.1736349>
- Kaufmann, O. T., & Stenseth, B. (2023). Programmering som del av matematikkfaget. *Norsk pedagogisk tidsskrift*, 107(2), 109-123. <https://doi.org/10.18261/npt.107.2.2>
- Kelentrić, M., Helland, K. & Arstorp, A.-T. (2017). *Rammeverk for lærerens profesjonsfaglige digitale kompetanse*. Senter for IKT i utdanningen. [https://www.researchgate.net/publication/321796206\\_Rammeverk\\_for\\_laererens\\_profesjonsfaglige\\_digitale\\_kompetanse](https://www.researchgate.net/publication/321796206_Rammeverk_for_laererens_profesjonsfaglige_digitale_kompetanse)
- Koehler, M. & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)? *Contemporary issues in technology and teacher education*, 9(1), 60-70. [https://www.researchgate.net/publication/241616400\\_What\\_Is\\_Technological\\_Pedagogical\\_Content\\_Knowledge](https://www.researchgate.net/publication/241616400_What_Is_Technological_Pedagogical_Content_Knowledge)
- Koehler, M. J., Mishra, P. & Cain W. (2013) What is Technological Pedagogical Content Knowledge (TPACK)?. *Journal of Education*, 193(3), 13-19. <https://doi.org/10.1177/002205741319300303>
- Kunnskapsdepartementet. (2019). *Læreplan i matematikk (MAT01-05). Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020*. <https://www.udir.no/lk20/mat01-05?lang=nob>

- Kunnskapsdepartementet. (2017). *Rammeverk for grunnleggende ferdigheter: Digitale ferdigheter som grunnleggende ferdighet*. Udir.  
<https://www.udir.no/laring-og-trivsel/rammeverk/rammeverk-for-grunnleggende-ferdigheter/2.1-digitale-ferdigheter/>
- Magnusson, S., Krajcik, J., & Borko, H. (1999). Nature, sources, and development of pedagogical content knowledge for science teaching. I J. Gess-Newsome & N. G. Lederman (Red.), *Examining pedagogical content knowledge: The construct and its implications for science education* (s.95-132). Dordrecht: Springer Netherlands.
- Mertens, D. M. (2019). *Research and Evaluation in Education and Psychology: Integrating Diversity with Quantitative, Qualitative, and Mixed Methods* (5. utg.). Sage
- Mishra, P. & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, 108(6), 1017-1054.<https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Moen, T. (2013). Sosiokulturell teori Vygotsky i teori og praksis. I R. Karlsdottir & I. D. Hybertsen (Red.), *Læring, utvikling, læringsmiljø: En innføring i pedagogisk psykologi*. (s.) Fagbokforlaget
- Niess, M. L., Ronau, R. N., Shafer, K. G., Driskell, S. O., Harper S. R., Johnston, C., Browning, C., Özgün-Koca, S. A., & Kersaint, G. (2009). Mathematics teacher TPACK standards and development model. *Contemporary Issues in Technology and Teacher Education* 9(1), 4-24 <https://www.learntechlib.org/primary/p/29448/>
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1–17. <https://doi.org/10.1080/20004508.2019.1627844>
- Postholm, M. B., & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen damm akademisk.
- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G., & Vøll, L. O. (2016). *Teknologi og programmering for alle - En faggjennomgang med forslag til endringer i grunnsopplæringen - august 2016*. Utdanningsdirektoratet. <http://hdl.handle.net/11250/2414205>



Stake, R. E. (2006). *Multiple case study analysis*. Guilford Press.

Rehman, A. A., & Alharthi, K. (2016). An introduction to research paradigms. *International journal of educational investigations*, 3(8), 51-59.

[https://www.researchgate.net/publication/325022648\\_An\\_introduction\\_to\\_research\\_paradigms](https://www.researchgate.net/publication/325022648_An_introduction_to_research_paradigms)

Wadel, C. (2014). *Feltarbeid i egen kultur*. Cappelen Damm akademisk.

Wachira, P. & Keengwe, J. (2010). Technology Integration Barriers: Urban School Mathematics Teachers Perspectives. *Journal of Science Education and Technology* 20, 17–25 <https://doi.org/10.1007/s10956-010-9230-y>

Wong, G. K.-W., & Cheung, H.-Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming.

*Interactive Learning Environments*, 28(4), 438–450.

<https://doi.org/10.1080/10494820.2018.1534245>

## Tabeller og figurer

Figur 1	TPACK rammeverket	Koehler & Mishra, 2009
Figur 2	Visuell beskrivelse av når læreres tanker og forståelse går mot integrasjon gjennom TPACK	Niess et al., 2009
Figur 3	Sammenligning av lærernes svar	
Figur 4	Kunnskapskomponenter hos lærerne	
Tabell 1	Ulike observatørroller	Postholm & Jacobsen, 2018
Tabell 2	TPACK analyseverktøy	Basert på TPACK rammeverk av Koehler & Mishra, 2009 (Figur 1)
Tabell 3	Lærer 1 sine svar	
Tabell 4	Lærer 2 sine svar	
Tabell 5	Lærer 3 sine svar	

## 8 Vedlegg

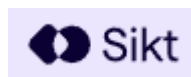
### Vedlegg 1 - sikt godkjenning

## Vurdering av behandling av personopplysninger

Referansenummer  
742099

Vurderingstype  
Standard

Dato  
11.01.2024



#### Tittel

Masteroppgave - Multiple casestudie av hvordan passer lærerens uttalte fokusområder i arbeid med programmering i matematikkundervisningen med TPACK modellen?

#### Behandlingsansvarlig institusjon

Universitetet i Agder / Avdeling for lærerutdanning

#### Prosjektansvarlig

Said Hadjerrouit

#### Student

Henrik Snilsberg

#### Prosjektperiode

01.01.2024 - 30.05.2024

#### Kategorier personopplysninger

Alminnelige

#### Lovlig grunnlag

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

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 30.05.2024.

#### Meldeskjema

#### Kommentar

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

**FØLG DIN INSTITUSJONS RETNINGSLINJER** Det er institusjonen du er ansatt/student ved som avgjør hvordan du må lagre og sikre data i ditt prosjekt og hvilke databehandlere du kan bruke. Husk å bruke leverandører som din institusjon har avtale med (f.eks. ved skylagring, nettspørreskjema, videosamtale el. ). Personverntjenester legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

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

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

Lykke til med prosjektet!

# Intervjuguide

1. Hvordan har du lært deg programmering?
  1. Skulle du ønske du kunne mer om programmering? Hvordan vil du lære det?
2. Hvilken utdanning har du?
  1. Hvilken erfaring har du innen matematikk?
  2. Skulle du ønske du hadde mer kunnskap innen matte? Hvordan vil du lære det?
3. Hvilken rolle spiller programmering i din undervisning?
  1. Er det slik du ønsker at det skal være?
  2. Hvor mange undervisningstimer med programmering har dere på skolen?  
Hvor mange har dere hatt i år?
4. På hvilken måte går du frem for å tilfredsstille de kriteriene som står i læreplanen om algoritmisk tekning?
  1. Er det noen felles fremgangsmåte for dette på trinnet/ skolen/kommunen?
  2. Føler du at du får dekket alle kompetansemålene som omhandler programmering i læreplanen?
    1. *Hvis ikke*, hvilke deler og hvorfor?
  3. Hvordan knytter du praksisen opp mot det som står i overordnet del?
    1. Digital ferdighet: bruke prog. Til matematiske problemer
    2. Kritisk tenkning
5. Hvilket syn har du på programmering som verktøy/hjelpemiddel?
  1. Hvilke muligheter og utfordringer ser du ved dette?
    1. Er det noen pedagogiske praksiser som virker mer effektive enn andre?
      1. isåfall hvilke?
  2. Hva opplever du elever synes er mest utfordrende?
    1. Hvilke løsninger bruker du for å hjelpe disse elevene?
      1. Føler du elevene opplever mestring ved bruk av gitt løsning?

6. · **Forklar TPACK hvis ukjent:** Hvor integrert føler du at programmering, pedagogikk (undervisningsmetoder), og matematikk er i din undervisningspraksis?
7. Hvilke muligheter og utfordringer har du møtt når du prøver å kombinere alle tre elementene i TPACK?
8. Kan du gi eksempler der en endring i en av de tre elementene (programmering, pedagogikk, eller matematikk) krevde justeringer/endring i de andre?
9. Hva er din tilnærming til TPACK? Hvilke ressurser eller strategier bruker du for å utvikle TPACK?
10.
  1. Tror du synet ditt har endret seg om 10 år, i såfall hvordan?
  2. Tror du andre lærere er enige med deg? Hvorfor/hvorfor ikke?
11. Hvis du hadde manglet en av kunnskapene, hvilken ville du manglet og hvordan hadde du løst det i en klasseromssituasjon?

## Vedlegg 3 - observasjonsguide

# Observasjonsskjema

Generelle observasjoner:

---

---

---

---

---

---

---

---

Tecnological Content Knowledge	
Tecnological Pedagogical Knowledge	

Pedagogical Content Knowledge	
Content Knowledge	
Pedagogical Knowledge	
Tecnological Knowledge	

**Spørsmål som dukker opp til intervju:**

---



---



---



---

**TPACK:**



## Vedlegg 4 - Samtykkeskjema

# Informert samtykke

Vil du delta i forskningsprosjektet/ masteroppgaven?

«Hvordan vektlegger læreren pedagogisk, teknologisk og fagkunnskap?»

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke hvordan lærerens uttalte fokusområder i arbeid med programmering i matematikkundervisningen passer med TPACK modellen.

I dette skrevet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

### **Formål**

Formålet med oppgaven er å undersøke hvordan lærerens uttalte fokusområder i arbeid med programmering i matematikkundervisningen passer med TPACK modellen. TPACK er en utviklet modell bestående av teknologisk kunnskap, pedagogisk kunnskap, og kunnskap om faginnhold, der hypotesen er at alle tre kategorier må være oppfylt for at elevene skal få utbytte av undervisningen. I forskningsprosjektet skal vi analysere hvordan matematikklærere selv vektlegger de forskjellige kunnskapene i planlegging, og ønsket kunnskap, når de gjennomgår undervisning i programmering. I tillegg skal det observeres en time med undervisning i programmering, der formålet er å kartlegge når og hvordan de forskjellige kunnskapene anvendes i praksis.

Ut ifra dette har vi skissert frem 2 forskningsspørsmål:

*«Hvordan vektlegger læreren pedagogisk, teknologisk og fagkunnskap?»*

*«Hvordan er det tiltenkt at grenene i TPACK skal komme frem i undervisningen?»*

### Masteroppgave

Informasjonen som samles inn i intervjuet og observasjonen skal brukes til analyse i en masteroppgave som leveres til Universitetet i Agder. Informasjonen skal ikke brukes til videre forskning.

### **Hvem er ansvarlig for forskningsprosjektet?**

Universitetet i Agder, avdeling for lærerutdanningen er ansvarlig for prosjektet.

Veileder i prosjektet: Said Hadjerrouit

Forskere i prosjektet: Henrik Snilsberg og Soroush Aghakouchaki

### **Hvorfor får du spørsmål om å delta?**

Undersøkelsen skal baseres på ca. 3 matematikklærere på ungdomsskole som underviser i programmering. Mulige informanter og kontaktinformasjon er funnet ved hjelp av avdeling for lærerutdanning ved Universitetet i Agder.

Hva innebærer det for deg å delta?

Dersom du ønsker å delta innebærer det deltakelse i et intervju med lydopptak på ca. 20-30 minutter, og observasjon av 1-2 matematikktimer, hvor informasjonen vil benyttes i besvarelsen av problemstillingen oppgitt over. Intervjuet vil inneholde spørsmål angående din mening om matematikkundervisningen med hensyn til programmering, og hvordan dette utarter i praksis. Observasjonen består av 2 studenter som noterer ned observasjoner av lærer i undervisningssammenheng.

Lydopptaket vil senere bli transkribert, der informanten blir referert til som «lærer 1», «lærer 2» eller lignende.

### **Det er frivillig å delta**

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

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

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket. Personlig informasjon som navn og epostadresse er kun delt med ansvarlige for innhenting av informanter ved Universitetet i Agder, veileder og forskere i masteroppgaven.

Intervjuets innhold og oppgaven som helhet vil være anonymisert. Informanter blir referert til som «lærer 1», «lærer 2» eller lignende, og skolen blir referert til som «en ungdomsskole i Agder» eller «skole 1», «skole 2» og lignende. Personopplysningene behandles konfidensielt og i samsvar med personvernregelverket. Notater og opptak fra intervjuet oppbevares utilgjengelig for andre enn observatørene. Dersom det kommer frem personopplysninger i intervju vil disse kodes slik at de kun kan gjenkjennes av oss.

Dersom det er aktuelt å skildre lærerens interaksjoner med enkeltelever eller elevgrupper gjøres dette anonymt, der utseende, navn og kjønn utelates fra observasjonsskjemaet.

Det sendes et informasjonsskriv til alle informanter, som så får ansvar for å dele ut skrivet til elevene i klassene som skal observeres.

### **Samtykkeskjema**

Dette samtykkeskjema inneholder navn, og vil dermed bevares i låst skap. Skjemaet skal kun oppbevares i papirform og blir ikke digitalisert.

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

Prosjektet vil etter planen avsluttes ved innlevering av masteroppgave 24. Mai 2024.

Etter prosjektslutt vil datamaterialet med dine personopplysninger anonymiseres, der lydopptak slettes og samtykkeskjema makuleres.

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

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Universitetet i Agder har Sikt – Kunnskapssektorens tjenesteleverandør vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter

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

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

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

- Avdeling for lærerutdanningen ved Universitetet i Agder, veileder Said Hadjerrouit, Tlf: 38141793.

- Student 1: Soroush Aghakouchaki Tlf: 951 81 514 mail:  
[Soroagha@gmail.com](mailto:Soroagha@gmail.com)
- Student 2: Henrik Snilsberg Tlf: 400 88 299 mail:  
[henriksnils@gmail.com](mailto:henriksnils@gmail.com)

Hvis du har spørsmål knyttet til vurderingen som er gjort av personverntjenestene fra Sikt, kan du ta kontakt via:

- Epost: [personverntjenester@sikt.no](mailto:personverntjenester@sikt.no) eller telefon: 73 98 40 40.

Med vennlig hilsen

*Prosjektansvarlig*

*Student 1*

*Student 2*

Said Hadjerrouit

Soroush Aghakouchaki

Henrik Snilsberg

-----  
-----

## Samtykkeerklæring

Jeg har mottatt og forstått informasjon om masteroppgave prosjektet «Hvordan vektlegger læreren pedagogisk, teknologisk og fagkunnskap?» og har fått anledning til å stille spørsmål. Jeg samtykker til:

- Å delta i intervju med lydopptak. Jeg samtykker også til at mine svar kan oppbevares og behandles frem til prosjektet er avsluttet.
- Å delta i observasjon av undervisning
- at lydopptak lagres frem til prosjektslutt, til transkribering

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

Signatur\_\_\_\_\_

-----

(Prosjektdeltager, dato)

## Vedlegg 5 - Infoskriv til elever og foresatte

Hei alle elever og foresatte.

Onsdag 24. Januar 2024 skal vi (2 studenter fra Universitetet i Agder) observere og notere ned observasjoner av deres lærer i en matematikktime i deres klasse. Grunnen er at vi skal innhente informasjon er for å skrive masteroppgave som skal leveres til våren. Vi kommer derfor til å være til stede under matematikkundervisningen. Vi skal ikke delta eller være aktive i selve undervisningstimen. Vi skal *ikke* ha noen form for lydopptak eller video/bilder, så dere eller deres barn vil ikke bli eksponert på noen måte.

### Hva skal vi observere?

Som nevnt tidligere skal vi kun observere læreren, og vi kommer ikke til å bemerke eller nevne elever i våre notater. Vi skal se på hvordan læreren bruker sin teknologiske, pedagogiske og faglige kunnskap til å undervise elevene i programmering.

Masteroppgaven som helhet vil være anonymisert. Informanter blir referert til som «lærer 1», «lærer 2» eller lignende, og skolen blir referert til som «en ungdomsskole i Agder» eller «skole 1», «skole 2» og lignende. Personopplysningene behandles konfidensielt og i samsvar med personvernregelverket. Notater vil kun være tilgjengelig for observatørene (oss to studenter) og veileder. Alt av observasjoner og notater vil makuleres etter prosjektslutt (sommer 2024).

Dersom det er aktuelt å skildre lærerens interaksjoner med enkeltelever eller elevgrupper gjøres dette anonymt, der utseende, navn og kjønn utelates fra observasjonsskjemaet.

Eksempelvis: «læreren får spørsmål av to elever: «Hva skal jeg gjøre nå?»»

Håper dette går fint for dere. Det er bare å sende oss en email dersom det er noe dere lurer på, så skal vi svare så raskt vi har anledning.

Med vennlig hilsen

Soroush Aghakouchaki      Email: Soroua18@student.uia.no

Henrik Snilsberg      Email: Henriksn@student.uia.no

## Vedlegg 6 - Data av intervju og observasjon

### Intervju:

Kunnskapskomponent	Lærer 1		Lærer 2		Lærer 3	
	Kan ikke	Kan	Kan ikke	Kan	Kan ikke	Kan
Frekvensen av komponenter	20	49	15	51	19	34
1. TK	9	9	4	10	3	7
2. PK	1	3	0	9	2	3
3. CK	1	8	5	9	2	5
4. TPK	1	4	0	4	1	2
5. TCK	4	4	4	3	7	8
6. PCK	2	8	2	9	1	3
6. TPACK	2	13	0	7	3	6

### Observasjon student 1:

Kunnskapskomponent	Lærer 1	Lærer 2	Lærer 3
Frekvensen av komponenter	17	25	20
1. TK	1	3	1
2. PK	2	2	3
3. CK	1	2	3
4. TPK	2	5	2
5. TCK	3	5	0
6. PCK	4	5	8
6. TPACK	4	3	3

### Observasjon student 2:

Kunnskapskomponent	Lærer 1	Lærer 2	Lærer 3
Frekvensen av komponenter	30	30	30
1. TK	3	3	3
2. PK	3	3	6
3. CK	5	4	2
4. TPK	5	4	7
5. TCK	4	5	4
6. PCK	6	7	5
6. TPACK	4	4	3

# Vedlegg 7 - Transkripsjoner

## Intervju 1

**student 1:** intervjuer

**student 2:** kontrollør

**informant 1:** intervjuobjekt

**00:00**

Ja, vi begynner egentlig bare fra toppen med litt sånn hvor det startet hos du.

Hvordan har du lært deg programmering?

**00:13**

Når jeg studerte for 30 år siden, så hadde jeg et kurs i noe som heter informatikk. Og der var det litt programmering. Eller litt, ja, det var litt programmering. **1c**

**00:24**

Så der startet det hos deg

**00:25**

ja. Absolutt. Så jeg hadde vel for så ved et rent programmeringskurs også.

Sånn at jeg har på en måte denne strukturen av programmeringsspråk, hvordan det bygges opp, hvordan kommandoer du gir til maskinen, hvordan den svarer og sånn.

Det lærte meg litt da.

Så jeg vil jo si at jeg hadde sånn sitt programmering da, på skolen. Og det er jo veldig hjelpsomt nå, at jeg har hatt det. **1c**

**00:53**

Du føler da at du har er tilstrekkelig for det du skal bruke det til her?

**00:56**

Ja, det føler jeg. Men, jeg vet ikke hvor mye jeg kan bare prate i vei her. **1c**

**01:01**

Vær så god.

**01:03**

For du kan si at det som jeg synes, altså det jeg husker veldig godt fra da jeg tok programmering selv, det er at da hadde jeg jo noen kurs der jeg, altså en hel vinter hadde jeg jo flere programmeringsfag. Og da husker jeg veldig godt at det er ikke sånn at denne selve den ideen om hvordan du på en måte snakker til datamaskinen, det er ikke sånn at du får den inn på første forelesning på en måte. Det er noe som modne..., hvertfall jeg opplevde veldig at det ble sånn modning over tid at jeg skjønnte etter hvert, liksom hvordan faktisk dette foregår. Sånn at det er ganske langt ifra det å bare lese et program og se "ja men dette programmet skjønner jeg kan kjøre", til å kunne lage det programmet selv. Og det tenker jeg er veldig sånn i forhold til når vi gjør det her på

skolen, at de lærere som ikke har vært borte i det hele tatt, har en utrolig lang vei å gå. Og det at de som skal klare det, det er litt sånn naivt på en eller annen måte å tro at alle lærere skal få det til. Og bare sånn for å ta et kjapt eksempel på det, nå skal jeg ha en annen lærer dere skal snakke med etterpå, men jeg og han har programmert mye sammen. Og han hadde ikke programmert noe før, og jeg hadde hatt programmering for 30 år siden. Og jeg ser bare på når jeg samarbeider med han, den prosessen det er. Det gikk veldig lenge, han kunne se programmene jeg laget og skjønne programmene, men han kunne ikke gjøre det selv. Men han tok ut tida da, han brukte jo masse ettermiddager, alle programmene vi var enige om å kjøre, lagde han hjemme selv, han brukte masse tid på det, og ha på en måte lært seg det. Men det er på en måte ikke en quick fix, det tar tid. **1c, 1c, 1a**

**02:30**

Ja, for det er jo en ting man kan ha sett, men vi må komme litt tilbake til det senere.

**02:36**

Ja, nå svarte jeg sikkert på veldig mye, men akkurat det der, i forhold til hva som er viktig for meg å formidle, det er noe med at dette er en ganske basic ting, fordi når noen bestemmer at alle mattelærere i Norge skal programmere, så er det noen som aldri har sett det før, og det er ikke bare å gjøre det altså. Det er et helt eget fag, det er ikke matematikk, det er et annet fag som du trenger å faktisk jobbe med over tid. **1c, 3a**

**02:59**

Helt ny teori.

**03:01**

Ja, og som jeg sa til dere, vi har fått hjelp fra universitetet, fire forelesninger, og så er det en travel hverdag. Jeg skulle ønske at de lærere som var med på det fikk gjort mer mellom hver gang, men så rekker de ikke det. Og da er det, jeg skal ikke si det er bortkastet, men det kreves uendelig mye mer for å beherske et nytt fag. **1c**

**03:18**

For det ble jo liksom at man måtte ha litt eget initiativ som lærere.

**03:22**

Ja, det er du helt nødt til.

**03:26**

Du sa litt om at du hadde tatt egne programmeringskurs, hva slags utdanning har du innenfor det med lærer og yrke, holdt jeg på å si?

**03:34**

Jeg tok fag, så jeg tok matematikk, og så tok jeg egnetlig kristendom, og så tok jeg PPU. Og så var

det på en måte sånn at jeg er et hovedfag i matematikdidaktikk, så da var det sånn at jeg hadde først nok matematikk, og så måtte jeg ha et fag til, som da var kristendom for min del, så jeg kunne ta PPU, og så var PPU også noe jeg måtte ha for å kunne gå inn på det hovedfaget på en måte. Så har jeg hovedfag i matematikdidaktikk. **2c, 3c**

**03:58**

Også har du undervist en del i matematikk?

**04:01**

Ja, og så er det litt sånn at jeg begynte å jobbe på skole 1, så er det sånn at jeg tok over jobben til en realfagslærer, og så fikk jeg matematikk og naturfag. Men naturfag har jeg ikke mer enn det jeg har fra videregående. **6c**

**04:12**

Nei, ok. Da er vi på... Skulle du ønske du hadde noe utvidet kunnskap innenfor matte, eller føler du at der er strekke...

**04:23**

I matte til å jobbe på ungdomstrinnet så føler jeg at jeg kan mer enn nok. Det har jeg ikke noe behov for. **3c**

**04:29**

Nei, fint. Hva slags rolle spiller programmering i undervisningen din?

**04:34**

Den spiller en... Ja, altså, den spiller en ganske viktig rolle. Det er jo noe sånn sett på en måte... Vi gjør det litt fordi det kommer på eksamen. Det er det veldig enkle svaret. Vi har fått beskjed om å gjøre det, det kommer på eksamen. Vi er nødt til å... Samvittigheten i forhold til elevene er jo at vi må rigge dem på det. Så ser vi selvfølgelig at det har noen veldig positive sider. Det er noen ting som vi mener at programmering gjør at elevene lærer bedre. Men det er nok sånn at vi er ganske selektive med hvilket tema vi tenker at vi skal bruke programmering på. Så for eksempel i dag fikk dere være med når vi holdt på med geometri, og det tenker vi er et veldig fint tema for da får vi både programmert, samtidig får vi jobbet mye med formlene for volum, formlene for areal omkrets, hvis det var det. Det har vi gjort tidligere da. Så vi får på en måte jobbet mye med matematikken selv om vi programmerer. Det som er vanskelig på huset er å få en lik standard i alle klasser, fordi at læreren er forskjellig, noen tør nesten ikke ta i det. Mens noen av oss har litt mer kunnskap om det. **1c, 3c, 7c, 5c, 1a**

**05:45**

At man er litt mer komfortabel med å bruke det.

**05:47**

Ja, litt sånn. Elevsiden her er jo på en måte, da kan vi sikkert diskutere i timen etterpå, men det er jo ganske mange elever som synes det er ganske

gøy. Det som kanskje, sånn hvis vi skal si noe med hovedinntrykk, er kanskje at guttene liker det bedre enn jentene. Og det er jo ikke alle ting på skolen som er sånn. **2c**

**06:10**

Kan jeg bare skyte inn, bruker dere som regel bare scratch?

**06:13**

Vi har på en måte valgt det hos oss.

Og så har vi på en måte spurt og spurt og spurt veldig mange plasser.

Hva er det noen, altså om noen kan fortelle oss hvilke språk vi skal bruke.

Det er klart de eksamensoppgavene som har kommet til nå, de er veldig sånn algoritmiske. Det er ikke et program, det er bare en slags algoritme. Så skal de på en måte kjenne igjen programmeringsstrukturen. Og det kan de like fint kjenne igjen om det er scratch eller om det er Python eller hva det er. For de skal ikke la av et program selv. Det har ikke vært sånn enda. Så har jeg skjönt at det vil komme kanskje. Og jeg vet at mange tenker at det vil komme Python på ungdomstrinnet. Så jeg er litt spent på det. For jeg tror på en måte at bøygen for de lærere som aldri har programmert. Enda større på Python enn det på scratch. Så du får midt i strekk i feltet. Noen elever er sjanseløse, for de har ikke lærere som kan det. Så vi har valgt bare scratch hos oss, foreløpig. **1c, 3c, 5a, 1a**

**07:09**

Du nevnte jo det i stedet for at det tok litt tid å forstå algoritmene...

**07:14**

Jeg opplevde det selv.

**07:16**

Føler du at det er den delen av det de vil ha frem i læreplanen? Når de gir oppgaver på eksamen hvor det handler om algoritmer, men ikke nødvendigvis programmering?

**07:26**

Både og, synes jeg.

For jeg synes på en måte at de ...

Jeg vet ikke, for de eksemplene...

Nå har ikke det vært eksamen så veldig mange ganger for det ene ble avlyst. Altså i ny læreplan. Den første ble avlyst med korona. Så vi har vel hatt én eksamen, tror jeg. Og så har vi sett to eksempelsett. Jeg føler litt de oppgavene som har kommet der er hverken fugl eller fisk. Det er litt sånn noen blokker og noe "hvis" ditt. Og så "ja, nei". Altså jeg vet ikke. Jeg tror ikke du trenger å ha programmert veldig mye for å skjønne hva... Jeg vet ikke. Det er ikke sånn dønn at dette er programmering, synes jeg da. Men det er vel kanskje sånn de må gjøre det i starten. **5a**

**08:12**



Litt tilbake til det med undervisningen. Hvor mange skoletimer har du hatt om programmering? Eller sånn mye snitter dere på i løpet av et år?

08:21

Det er ganske vanskelig å si.

08:24

Kommer kanskje litt an på temaer og sånn?

08:26

Det kommer litt an på temaene. Men jeg kunne kanskje tenke meg at i 9. klasse, der har vi ganske ...

Jeg har sett at vi klarer 6-8 timer, kanskje. Jeg tror ikke det er mer enn det. 7c

08:43

Nei.

08:44

Vi har på en måte også prøvd å ta det med når vi har sånn fagdag, eller tentamen. Så kommer vi til våren til å ha med det. Vi hadde det til jul. En oppgave i programmering. Da må de jo lage program selv og sånn. Noen finner jo noe som ligner i disken sin, selvfølgelig. Men de får en oppgave de skal lage. Og de kommer til å få en enda mer utfordrende ting til våren. Men det kommer helt sikkert til å bli det også til våren. At de faktisk må programmere. Ikke bare se en algoritme, de skal lage et program og sende det til oss. 5c, 7c

09:16

Fint. Vi har jo pratet litt om det der med algoritmisk tenkning. Hvordan går dere fram på en måte for å tilfredsstille det som kommer fra UDIR angående det? Tenke i systemer og løse problemer?

09:33

Vi gjør det lite sånn i forhold til at vi lager algoritmer på tavla. Vi gjør det stort sett med hjelp av scratch. Vi bruker blokkene i scratch og gjør det på den måten. Foreløpig så har det vært veldig sånn, de programmene dere så i dag, at du hiver inn noen variabler og så regner programmet ut noe. 7c

09:50

Men det er klart at vi tenker også at denne "hvis"-setningen, den kommer vi til å bruke en del tid på. Det er kanskje den vi først og fremst opplever at de har en av variantene på eksamen. Det er en forutsetning hvis det skjer, så det, og hvis ikke så noe annet. Men jeg tror nok at vi kan få til en løkke som spinner ti ganger. Vi kommer nok til å gjøre noen av de tingene også, sånn at de får litt av den programmeringsideen. 5c

10:18

Du sa litt om hvordan dere gjør felles på trinnene og på skolen, at det var litt vanskelig å samkjøre siden det var litt ulike (nivåer).

10:26

Det jeg kan si til det er at vi er jo en skole der vi stort sett jobber på trinn. Det betyr at. Nå er jeg i ledelsen så jeg har bare en matteklasse, det var den som dere var oppe i, men i utgangspunktet så er det at de fleste lærere har bare matematikk på et trinn. Det betyr jo at det blir litt kulturen på det trinnet. Det trinnet jeg har vært på, jeg har vært sammen med han som heter "lærer 2" mange år, så vi har på en måte lagt opp og da smitter det på hele trinnet. Litt fordi vi har flere klasser og litt fordi alle gjør det samme. Så det er nok andre trinn som på en måte har litt annen inngang på det, litt mindre. Men det vi for eksempel gjorde i fjor, bare for å ta et eksempel, i fjor hadde vi tiende klasse der det var flere lærere som ikke hadde programmert så mye. Da valgte vi som skole å gi de noen kurs. Så vi la det kurspakke der vi fikk lærere som hadde programmering litt under huden til å kjøre noen få timer i hver klasse, så vi sikret at de kunne det som kom på eksamen. 1a, 7c

11:24

Ja, veldig fint. Når det kommer til kompetansemålene som handler om programmering i læreplanen, du nevnte i forbiarten at dere følte at dere fikk dekket over det ganske greit.

11:35

Jeg føler jo kanskje det, vi gjør det. Det tror jeg, i alle fall de programmeringstingene vi har gjort, med for eksempel den gjengen som jeg sendte ut sist i tiende, så tenker jeg at de gjorde det. Det er jo litt sånn at når du møter gamle elever så sjekker du litt. Det er jo litt yrkesskade. Men det har jeg fått litt bekreftet, at de kunne i alle fall det de måtte kunne når de begynte på videregående. Det har jeg fått tilbakemelding på. 7c

12:08

Veldig fint. Det står jo litt om digitale ferdigheter i læreplanen. Føler du at dere jobber konkret inn mot de, eller føler du at det kommer med?

12:19

Det kommer litt med på kjøpet, vil jeg si. Det er klart at i gamle dager så hadde vi litt en tanke om at... jeg husker det var en periode vi hadde at de fikk et kurs i både Excel (og geogebra?), da de kom. Men elevene har jo en del av den type læringsmål også på barnetrinnet. Scratch er mange elever som har vært borte, for eksempel på barnetrinnet. Regneark kan mange kikket på. Så de kan jo en god del fra før. Så jeg tror nok hos oss er det mer sånn at vi nå bruker de verktøyene inn mot fag der vi trenger det. Ja. 3c, 7c

13:01

Veldig fint. Hva slags syn har du på programmering som verktøy eller hjelpemiddel?

13:08

Jeg har jo... Jeg mener jo at noen ting i matematikken tror jeg de kan lære bedre ved hjelp av programmering. Og der har jeg noen konkrete

eksempler på det. Samtidig er det noe som jeg tenker er litt vanskelig i forhold til at man bare innfører programmering, og så gjør man alle skal gjøre det, og så er det mange lærere som ikke har kompetanse til det. Du ber egentlig om at dette skal bli veldig lite likt fra skole til skole. Du ber egentlig om trøbbel, synes jeg. Og du forteller ikke hvilket program det er. **7c, 4a**

Du forteller veldig lite om hva du egentlig skal gjøre.

Enten at det er noe algoritme...

Jeg synes.. jeg er veldig kritisk til det. Men hvis jeg skal si noe om det jeg mener er den store gevinsten med slik som vi jobber, det er egentlig at de får en veldig god forståelse av variabelbegrepet. Så hvis jeg skal ta den rene matematiske tingen, som jeg tenker at dette er bra, så er det... For det er noe med at... Det er sånn det er kanskje i dag og, men det blir veldig fokus på variablene de skal sette inn. Så de får en veldig feeling på at en variabel er et tall. Og i dette tilfellet så er det brukeren som fører maskinen med noen tall, og så kommer det ut noe i andre enden. Sånn at... Og det er bare et veldig eksempel. Når vi begynte å programmere i høst, vi hadde et kurs i fjor for åttende, der brukte vi et par-tre timer kanskje, og når jeg på en måte i høst skulle vi gå tilbake og begynne med geometri, for vi hadde fått plangeometri i høst, og vi har romgeometri nå. Så tok vi noen enkle program der vi regnet under arealer og noe greier, og de er ganske kjappe, men vi må opprette variabler. Sånn at den første oppgavene der jeg så i dag med det rektangelet, som jeg hadde på veggen, det var noe de har sett før. Men en gang i høst, når jeg begynte å nevne programmering, og nå skal vi regne ut arealer av ting, så må vi opprette variabler i programmet. Så akkurat det føler jeg de får. Det er en god gevinst. **7c, 5c, 7c**

**15:03**

Fint, da svarte du på det neste spørsmålet mitt og, med utfordringer og muligheter. Veldig fint. Er det noen pedagogiske praksiser du ser fungerer godt når du skal jobbe med programmering? Altså sånne ting du gjør i klasserommet?

**15:17**

Ehm... Hva skal jeg si til det? Ja, altså vi prøver... Der har vi prøvd mye forskjellig. Altså i forhold til for eksempel sånn som i dag, så var det noen som kunne se starten av programmet, og så skal de gjøre det ferdig. Men det blir litt mer du kan tilpasse litt til ulike behov. Men noen elever har veldig mye igjen for det. At de får en idé om hvordan de skal starte. Noen ganger i scratch, det du kan gjøre i scratch, er at du kan lage en oppgave som du deler, der du bare har dratt ut alle blokkene til dem, slik at de får... Alle henter den samme fila på en måte, og så kan de sette sammen et program der de ikke må finne alle blokkene selv. Alle ligger i bildet, og så skal de bare sette opp rekkefølgen. Men det er litt sånn nybegynner ting, tenker jeg. Det gjorde vi i fjor. Og så synes jeg de tar det ganske kjapt. Så det er

noe som jeg ikke opplever så viktig nå lenger. **4c, 4c**

**16:14**

Har du noe med å gi til litt mer åpne oppgaver? At de bare får lage litt det de vil, og så ser vi hva de kommer ut med?

**16:24**

Ikke så veldig mye. Og det er nok... Jeg vet ikke. Opplevelsen min er jo litt at hvis du skal liksom... Noen vil profitere på det, mens noen driver bare med helt andre ting. Så han bytter katten ut med et helikopter, og så er det et helikopter som flyr rundt. Og det er jo gøy nok det. Men vi er nok litt sånn, spesielt meg og han "lærer 2" at vi har litt en tanke om hva vi har lyst til å gjøre. Så det er kanskje... Og så handler det nok litt om kanskje hvordan man jobber i matematikken ellers og. Og der er nok vi mattelærere litt ulike, tror jeg. Og jeg vet jo på en måte denne utforskningen og alt med det sånn. Så er det sånn at spesielt sånn som med det trinnet vi har nå, de er... De har vi måtte tatt på en litt annen måte. De hadde selv veldig lav selvillit i matte når de kom. De svarte veldig dårlig på de tester vi hadde når de kom. Så det grunnleggende matematiske har de veldig... Ja, har vi måtte jobbe mye med. De hadde også veldig dårlig arbeidsmoral. Ikke arbeidsmoral kanskje, men utholdenhet. Så lenge som de satt og jobba i dag i timen dere var, ikke sjangs i fjor, Der måtte vi jobbe virkelig med at de skulle kunne sitte litt konsentrert i 20 minutter. Det var liksom masse tid bare på selve det å kunne sette seg ned og faktisk... Så det var veldig mye sånne ting. Og da blir sånne åpne oppgaver vanskelig på et vis. Så det er ikke at det er mulig, vi kunne sikkert ha gjort det og sånn, men de støter på problemer hele tiden fordi de har liksom ikke verktøyene i orden på en måte. Bare et helt konkret eksempel, et litt banalt eksempel da, men vi diskuterte i fjor hvordan vi skulle finne 10% av noe. En sånn muntlig diskusjon i klassen. Og så går det ganske lang tid før vi blir enige om at vi kan dele på 10. At en 10% er å dele på 10. Og da tenker jeg kanskje at ok, ungdomsskole, da er vi i mål. Men det var vi ikke, for hvordan deler vi på 10? Det er jo et problem. For det er heller ikke sånn automatisert, eller hva man skal si. Så det er noe med at med den gjengen her så stopper det så veldig fort opp. Fordi det som du tenker at de skulle kunne, det kan de ikke helt. Og det gjør på en måte at vi er nok blitt veldig sånn, her styrer vi. Fordi det er så mange ting vi må... Føler vi da. Men så vi har synes at med den gjengen her har åpne oppgaver vært vanskelig. **2a**

**18:55**

Jeg så at du delte ut oppgavene på papir, og ikke sånn digitalt. Er det noen spesiell grunn til det, eller var det bare det praktiske og enkelt?

**19:03**

Det er... Nei, det er nok litt sånn vane. Jeg gjør litt begge deler. Men jeg har nok mest gjort det sånn, litt fordi at når de da sitter med skjermen, spesielt når vi skal programmere, så sitter de med skjermen.

Og det at de skal bytte bilder hele tiden, noen av de sliter med det. Så spesielt når vi jobber med dataprogrammering, så er det veldig greit at de kan ha to skjermer, et papir og en skjerm. Så det er kanskje hovedgrunnen til det. **2c, 4c**

**19:38**

Når elevene setter seg fast på en oppgave, hvordan føler du at det fungerer å hjelpe dem videre?

**19:46**

Ja, det er jo et stort problem på en måte. Og det handler egentlig om at, sånn som i timen i dag, så går jeg ganske mye rundt. Det er jo ikke sånn at jeg sitter foran og venter på at det er behov for meg. Og det gjør på en måte også at det er alltid noen som sitter i kø og trenger hjelp. Og det gjør liksom at den tids"slotten" jeg har på hver elev er ganske kort. Og det er et problem, sant? For da er det veldig fort å bare gi dem det, sant? Hjelpe dem over det hinderet. Og det er kjempevanskelig. Og jeg har ikke noe sånn... Jeg kan jo alle teori her, jeg vet jo på en måte at det er ikke det de lærer mest av. **6c**

**20:31**

Nei, men hvis du skal rekke alle på en time så...

**20:34**

Ja, også er det sånn at det som skjer er jo på en måte at da begynner han å slutte å jobbe, og da begynner han å prate om noe annet, og da går han på do. Det er jo det som skjer, sant? Men det er ikke det at de må ha det helt stille, og de må gjerne sitte og jobbe sammen. Sånn som i dag så ser dere at de bytter litt plass og de satt seg sammen og sånt. I dag fungerer det kjempefint, for jeg tror nesten det var programmering på skjermen hele tiden. Det var ikke sånn at folk var inne på andre ting, i hvert fall i stor utstrekning da, sikkert noen. Men da jobber de i hvert fall, tenker jeg. **7c**

**21:08**

Ja, fint. Nå er vi litt over på slik teori som vi skal jobbe med. TPACK, er det noe som er kjent for deg?

**21:16**

Nei.

**21:17**

Det er et rammeverk som sier at hvis du har kunnskap om fag, ditt fag altså, som er matematikk i dette tilfellet, om pedagogikken og pedagogiske praksiser, og om teknologien, altså det teknologiske som i dette tilfellet er programmering, at de tre utfyller hverandre for å få best mulig læring. Føler du at du har noe spesielt fokus på de tre her, eller hvilke av de tre vil du si at du fokuserer særlig på i undervisning?

**21:55**

Jeg fokuserer mest på de to første, minst på den teknologiske. Det er helt åpenbart. Og så er vi nok ganske nøye med, nå snakker jeg mest om meg selv, men jeg er ganske nøye med, det er noe med,

om det gjelder programmering, altså Scratch, eller om det gjelder for eksempel Excel, eller om det gjelder GeoGebra, uansett hva det gjelder, så opplever jeg at det er på en måte viktig å gjøre ganske mange ting som har med forståelse før du løfter det inn i de programmene. For det er klart at de programmene hjelper veldig mye, men det som jeg er veldig redd for er at man får til ting, for eksempel i GeoGebra, som man egentlig ikke forstår. Du får det til, for du vet hva du skal trykke på, du vet hva du skal skrive, men du skjønner ikke egentlig hva den kurven betyr. Sånn at funksjonslære for eksempel er veldig typisk, at det er du nødt til å jobbe ganske mye med funksjonsbegrepet, før du løfter deg inn i GeoGebra. For GeoGebra er veldig lettvin, du gjør det fort, du får laget masse grafer og holdt på, og du kan selvfølgelig bruke noe av det til å lære funksjonslære, men den linken er utrolig viktig. Så det kommer til slutt for meg. **1a,6c,5a,7c**

**23:12**

Ja, fint. Da svarer du egentlig litt på det med utfordringen ved å kombinere de tre. At det kan bli litt operasjonelt og ikke ...

**23:22**

Jeg synes veldig. Og dette ser jeg jo liksom ... Jeg har jo noen gutter som har vært igjennom type videregående skole, og jeg ser det veldig på dem, at de kan få det opp i GeoGebra, men de vet jo ikke hva det betyr. Så hva er greia med den da? Hvorfor går den sånn? Veldig tydelig, altså. For det er klart at ... Ja, sånn at ... Jeg tror det er viktig at det kommer til riktig tid. Tror jeg. **7c**

**23:56**

Så har vi et spørsmål som heter «Hva er din tilnærming til TPACK?»

Hvilke ressurser eller strategier bruker du for å utvikle TPACK?» Men det er jo kanskje litt vanskelig å svare på hvis ikke du har jobbet konkret med det.

**24:07**

Ja, det er jo egentlig veldig vanskelig å svare på i forhold til TPACK da. Men jeg tenker jo på det litt i forhold til de der ... Altså både Excel og GeoGebra og Scratch. Hvis det ikke er de tre programmene vi bruker, så er det sånn at da jeg begynte å jobbe, så brukte vi kanskje ingen av de. Så det er jo sånn at vi ... Det er jo ingen tvil om at det er ... Har mye igjen i matematikken og ha de programmene tilgjengelig. Det er jo ikke det tvil om, tenker jeg. Og så er jo spørsmålet bare «Når skal du liksom innføre det?» Når skal du bruke det, og hvor mye skal du bruke det? Det er jo mer det det går på, tenker jeg da. Så jeg synes jo vi forholder oss til oss og den digitale siden Men akkurat det første i min utdannelse er jo de første to jeg har egentlig fokusert på. **4c, 6c**

**24:49**

Ja, for det er jo ofte de to (gestikulerer til ped og fag) som kommer sammen. Og så har den tredje liksom trengt seg litt på de siste årene.

24:54

Ja. Og så er det jo på en måte litt sånn ... Men det er jo på en måte litt det da, for det er klart at ... Det som er en diskusjon ganske lenge er jo litt det der med ... Egentlig for til alle disse programmene som blir innført, er jo på en måte litt ... Ja, men vårt pensum ... Vi trenger jo ikke de programmene, i forhold til vårt pensum bare, på en måte. Og så er det en diskusjon da, for da lager du oppgaver. Det er ikke sånn at du sier at vi trenger geogebra for å kunne løse pensum på ungdomsskolen i funksjonslære liksom. Men vi lager heller oppgaver som gjør at ... Det er mer sånn, hvilke oppgaver kan vi bruke det til? Du får en heller motsatt tankegang. At det blir programmene som blir de viktige, og ikke hva de skal lære. **7a**

25:38

At det føles litt påtvingt nesten.

25:39

Ja, litt sånn. Og så kan man selvfølgelig se at det er noen positive ting med det. Men det er litt sånn ... Det som mange lærere opplever er at pensum i matte på ungdomstrinnet er så svært. At alle sånne ting gjør jo bare at du utgjør ... Du får aldri vekk noe, du bare fyller på. Og det er en utfordring, time messig. Det er jo ikke så mange timer vi har. **6a**

26:02

Nei.

26:03

Og hvis vi skal bygge forståelse og alt det der, så er det grense for hva ...

Kjenner dere pensum på ungdomstrinnet i forhold til videregående, for eksempel?

26:11

Har kikket på noe av det.

26:13

For jeg synes jo på en måte ... Hvis vi tenker det en tiendeklassing vi skal kunne på eksamen, da. Det er ikke det at det er så sykt vanskelig, men det er utrolig mange ting. Mens hvis du for eksempel tar ... **3c**

26:23

Det er veldig breitt.

26:24

Ja, veldig breitt. Hvis du tar t-matematikk for eksempel, min opplevelse er jo at det er jo selvfølgelig et høyere nivå, men det handler litt om det samme på en vis måte. Det er litt sånn ... Det henger litt mer sammen. Her er det litt i alle retninger. Så det er litt sånn rart. **3c**

26:41

Ja. Ofte kan jo det med programmering, at det har blitt innført så lenge etterpå, gjøre at det havner litt

på siden, føler du det? At det blir en egen boble på siden av matematikken.

26:51

For veldig mange tror jeg det blir sånn. Og bare sånn i forhold til det vi gjorde på tiende i fjor, så var det jo sånn. De har ikke programmert, vi må gi dem noe så de kan klare eksamen. Så det blir et kurs for at vi skal kunne klare å få til det som venter oss på en eksamen, liksom. I stedet for at det blir et verktøy vi bruker. **7a**

27:09

Har du sett det åpne seg noen nye muligheter ved at du har fått programmering inn i matematikken?

27:16

Ja, for så vidt jeg har ... I forhold til det med variabelt begrepet har jeg sett det. I forhold til for eksempel det med sannsynlighet, terningkast for eksempel, har vi jo mange muligheter i programmering. **5c**

27:27

Mange simuleringer.

27:28

Ja, massevis. Så det er absolutt, vil jeg si det, at det var kanskje to ting jeg hvertfall kom på veldig sånn kjapt. Kanskje de mest åpenbare tingene.

27:41

Tror du synet ditt kommer til å endre seg om ti år?

27:45

(ler) Vet du, da er jeg er så gammel at jeg ikke vil gjøre noe nytt.

Det er problemet. Nei, men jeg er litt spent. Jeg er veldig spent på hva som skjer, kanskje først og fremst knyttet til programmering. Hvordan det faktisk blir for jeg... sånn som det er nå, så vet jeg det er vanvittig strekk i feltet. Jeg vil tippe det er ungdomsskoler som ikke nesten har noen som programmerer. Noen gjør det sikkert kjempe masse. Noen synes det er gøy å gjøre det fordi det er gøy og de kan det og alt. Men noen har ikke peiling. Og det er klart, elevene blir jo da, liksom prisgitt etter hvilken lærer de har. Og sånn tror jeg det kommer til å være lenge. Det tror jeg. Og jeg ser det vi ser veldig tydelig, og det er jo at de eldste, jeg skal ikke si de, men jo eldre du er som lærer, jo tøffere er det på en måte dette å sette seg inn i. Fordi det er jo litt med IT-kompetanse generelt og selvfølgelig med den genrasjonen. Men det er liksom ikke bare for de å gjøre det. De må faktisk lære seg dette. **1a, 1a**

28:51

Tror du andre lærere er enige med deg, at de har samme synspunkt?

28:58

Nei, jeg tror nok ikke alle ser verdien kanskje av det. Det tror jeg ikke. Og jeg tror ganske mange tenker at programmering er noe annet enn

matematikk, egentlig. Sånn at det er ikke så mye gevinst i matematikken med at vi programmerer. Det er liksom en ting til vi skal gjøre. Jeg tror det er mange som føler sånn. **1a, 3a**

**29:20**

Da har jeg et siste spørsmål på blokka. Hvis du hadde manglet en av kunnskapene, altså av de hovedkunnskapene i en TPACK, med pedagogisk, innhold eller teknologi hvordan ville du gått frem for å tilegne deg det du følte du manglet?

**29:42**

Nei, jeg vet ikke. Hva skal jeg si til det da? Altså, det jeg kan si er at her på huset da, hvis, nå er jo matematikkfaget liksom brenner litt for da, så hvis jeg på en måte opplever at her på huset mangler vi noe, sånn at bare sånn det vi nå gjorde med at vi fikk liksom til at vi fikk hentet noe støtte fra universitetet for å hjelpe oss med programmering, det er fordi at vi som skole tenker at dette trenger vi. Så hvertfall på systemnivå, så vil jeg tenke at det har vi observert og sett og prøvd å gjøre noe med. Om det hadde vært det samme om vi hadde på en måte manglet mer sånn den... Jeg vet ikke, hver gang vi velger å læreverk så har vi en diskusjon som handler om noen av de tingene, for de ulike læreverkene har litt forskjellig inngang til, ja, litt i forhold til på en måte hvilke læringssyn man har. Noen er veldig på sånn utforskning og inquiry, mens andre læreverk er mer på å repetere masse oppgaver. Du ser det jo litt på det. Og da har vi jo på en måte en diskusjon i kollegiet på dette, og det er jo ikke tvil om at mange lærer liker best litt sånn repetere mange oppgaver type bøker. Også er det noe vi ikke jobber med sånn... Vi jobber ikke så mye med det i kollegiet. Vi har jo samme mattebok, men bare for å ta det og, vi har en mattebok, vi valgte mattebok som har litt sånn inquiry-utgangspunkt. Og sånn som det trinnet vi har nå, som mangler så mye verktøy til å utforske, så er det sykt vanskelig. Fordi det er to ting som skjer. Det ene er selvfølgelig at de mangler det matematiske oppgaver vi, det andre er at de oppgavene, altså de mattebøkene som legger vekt på mye inquiry, de blir en del tekst i. For det blir jo på en måte lite ferdigstilte stykke, og det blir mye at det blir et problem du skal løse, og de klarer liksom ikke det å lese. For noen er det en kjempebok. **1c, 6c**

**31:49**

Det blir litt sånn uangripelig.

**31:51**

Ja, for det å bare sette seg inn i og lese, hva er det de mener vi skal gjøre her, de får liksom ikke til det å kode av hva de egentlig skal gjøre. Kjempvanskelig, synes jeg. **6a**

**32:00**

Det er jo absolutt spennende.

**32:02**

Derav også mange ark som blir delt ut, så det er jo det. **6c**

**32:06**

Jo, det blir jo sånn da.

**32:09**

Så det er nok en vane.

**32:10**

Kan jeg spørre litt sånn tilleggsspørsmål? Pleier dere å hente mye hjelp fra internett sånn, å finne oppgaver eksternt?

**32:22**

Nå tør jeg faktisk ikke si fra alle lærere hvordan det er. Altså, en periode hadde vi noe som heter ... Hva heter det da? Vi hadde en sånn der leksebank. Eller, vet du, vi hadde en sånn der ... Hva heter det for noe da Vi hadde et sånt nettsted der det var en haug med oppgaver. Der elevene hadde en konto som de kunne logge seg på, og gjøre masse algebra, masse diit og datt. Det brukte mange ganske mye. Spesielt til lekser, det var veldig lett å gi. Du kunne gi leksene i itslearning og bare lenke de opp til dette stedet. Men utover det så tror jeg kanskje det er litt sånn tilfeldig at man klipper inn noen figurer og sånn hvis man trenger det. Men jeg tror ikke folk liksom søker nettet til så mye oppgaver egentlig. Nei, det tror jeg ikke. Men jeg ser for min egen del så lager jeg en god del selv. Ja, ganske mye faktisk. **6c, 6c**

**33:18**

Men da er jeg fornøyd med alt det her. Tusen hjertelig takk for tiden.

## Intervju 2

**Student 1:** kontrollør

**Student 2:** Intervjuer

**Informant 2:** Intervjuobjekt

**00:00**

For å sikre at vi ikke mister noe kjører vi bare dobbelt i tilfelle en streiker.

**00:05**

Jeg har mistet noe selv på ikke å ha dobbelt.

**00:09**

Vi har en intervjuguide som jeg kommer til å intervju mest. Så skyter du bare «Student 1» inn hvis han har noen spørsmål. Så prater du egentlig bare fritt. Du trenger ikke svare bare ut fra spørsmålet. Vi vil gjerne ha alt du kommer på. Det er helt nydelig.

**00:24**

Ja, så bra.

**00:26**

Først, hvor har du lært deg programmering?

**00:29**

Det er i all grad selvlært og gjennom kollegaen min, at vi har sittet og jobbet sammen. **1c**

**00:37**

Skulle du på noen måte ønske at du hadde mer, eller er det noe du føler du savner?

**00:44**

I starten da dette ble introdusert, så var det et program som het GeoGebra, som var introdusert i grunnskolen. Det måtte vi også drive med selvlæring. Jeg trives egentlig godt med det, med å jobbe selv og finne ut av ting. Men samtidig kjente jeg på det med å endelig mestre godt GeoGebra, for så å få introdusert enda et nytt program du aldri hadde møtt før. Og så skulle du prøve å undervise i dette, og sørge for at elevene fikk god nok kompetanse til å kunne bruke det hensiktsmessig. Det merket jeg at det begynte å bli tyngre.

Men likevel så var det ingen lovnader om kurs i fremtiden. Det var egentlig bare at dette skal dere kunne. Elevene skal begynne å programmere. Det er de nye læreplanmålene Så da var det noen av oss som bare, javel, da får vi bare gjøre som vi gjorde og sette i gang og begynne å jobbe med programmering og først og fremst scratch. **1c, 1a, 1c**

**01:55**

Hadde du vært med på kurs hvis det hadde blitt tilbudt?

**02:00**

Ja, 100 prosent. Det vil jeg si. Det er ikke noe tvil om at hvis det hadde vært et kurs som kunne ha begynt å forklare litt hvordan du skulle programmere og hva tanken bak det var, så hadde du forstått mer tidligere, og det tror jeg hadde dratt nytte av. **1c**

**02:22**

Ja. Så litt mer til det generelle. Hvilken utdanning er du?

**02:28**

Jeg er adjunkt med tilleggsutdanning. Jeg har fire år, og så har jeg to år etter det. Så jeg har seks år. **6c**

**02:37**

Er det da den hvor du har matte, norsk, engelsk, eller har du noen andre?

**02:42**

Jeg er gammel skole jeg, så jeg har grunnskolen 1-10. Det som kaltes ALU, tror jeg det heter. Allmennlærerutdanning.

**02:52**

Ja.

**02:53**

Etter L97.

**02:55**

Ja. Hvilke erfaringer har du i matematikkfaget?

**03:01**

Erfaringer i ...

**03:03**

Ja, sånn i utdanning og eventuelt kurs eller jobb?

**03:07**

Jeg har 75 studiepoeng i matematikk, sånn at allmennlærerutdanningen den ga oss, for å kunne undervise i 1-10, sa at du skulle ha 30. Så jeg har tatt 45 etter det. **3c, 3c**

**03:23**

Ja.

**03:24**

For å få mer kompetanse.

**03:26**

Ja. Og så har du en del år som mattelærer?

**03:30**

Ja, har jobbet ganske mange år som matematikklærer. Nærmer seg vel rundt 16 år. Jeg tror det er 15 år, og så har jeg vel 16 år som lærer. **6c**

03:43

Ja. Noe du skulle ønske du hadde mer kunnskap om innen matematikk hvis du ser vekk fra programmering da?

03:51

Nei, egentlig ikke. Jeg synes det er matematikkfaget generelt, jeg har jo ganske mange studenter som kommer inn og tar over undervisningen, spesielt de to realfagene mine, naturfag og matte. Og da ser jeg jo at det er flinke faglige studenter som kommer, men samtidig så er det dette læringstrykket, alt det som elevene skal gjennom. Hvis de skulle tatt over min klasse, og kjørt igjennom det de tenkte de skulle få til, så hadde de ikke kommet gjennom mer enn 50-60 prosent av det de skulle. Og det er jo litt basert på, det er ikke fordi at de gjør noe dårligere eller noe sånt, det handler jo i stor grad om at det er mye kreativt. Det er jo veldig lite sitte og jobbe og produsere oppgaver. Det er jo gjerne jobb i grupper, klare å løse den oppgaven her på tavla, drive å ha sånn tavle... disse her du limer opp på veggen. 6c, 6a

04:57

Åja. Sånn vertikal tavle?

04:59

Ja, som de limte på de ulike delene av veggen, og så jobber de da i grupper.

Da fikk de kanskje i løpet av en time produsert tre-fire oppgaver. Og det kan jo være litt snaut, samtidig som at ikke alle elevene får bidratt i en gruppe. Så mye positivt og kreativt i den nye matematikken, med at det er utforskende, og i stor grad driver å lære på egen hånd og driver med dybdelæring, men samtidig så føler jeg ikke at elevene får terpet nok på en del av, sånn at det sitter disse ulike oppgavene. 6a

05:47

Litt lage repetitive oppgaver?

05:50

Ja, at det gjerne kan sitte litt. De må gjerne få lov til å utforske alt det her.

Men jeg ville hatt den delen som et tillegg til, i stedet for at den tar fullstendig overhold, og at den repeteringen forsvinner litt vekk, sånn at de får ikke den mengdetreningen som veldig mange elever trenger. Vi ser det jo nå, nå som vi har kjørt over, at det er mange elever, jeg har aldri hatt det trinnet jeg har nå, det har aldri vært så langt nede på resultatene når det gjelder nasjonale prøver. Så det er jo en tilbakemelding om at det kan være mange faktorer som spiller inn, spesielt det med korona. Men nå begynner det å bli en stund siden. 3c

06:42

Litt tilbake til det med programmering. Hvordan rolle føler du det har i undervisningen din i løpet av et år? Hvordan katalogiserer du det?

06:54

Det er litt sånn, både og. Jeg synes det er veldig fint med tanke på det vi holder på med nå, som har med plangeometri og romgeometri. Da synes jeg det er veldig bra. Det er jo det å prøve å implementere det mer i de ulike temaene, som vi kanskje kunne ha vært flinkere til. Samtidig så kommer det jo statistikk og synlighet, som vil gjøre at du kan få brukt det i en annen variant. Det kommer vi helt sikkert til å gjøre. Pytagoras ikke minst. 5a, 7c

07:30

Er du fornøyd med det du gjør nå, med tanke på det med programmering?

07:37

For min del er jeg fornøyd. Den kompetansen jeg gir elevene mine, tenker jeg er fin nok med tanke på det videre løpet, altså videregående. Og tilbakemeldinger fra tidligere elever er at de gjerne har ligget minst like langt fremme som sine medstudenter. Og har fått den introen av det. Det er ikke alle som har skjønnet det. Det gjelder jo også her på Huset, som har gått all in på det med programmering. At de heller sitter og venter litt og håper at det går litt av seg selv. 5a, 1a

08:24

Føler du det at dere måtte ta litt tak i det selv? At du ikke får bestemt at her er pakka du skal gjennomføre? At du heller må finne ut av selv hva det egentlig går ut på, og hvordan kan vi gjøre det beste i klasserommet?

08:37

Ja, det er det liten tvil om. Nå har vi fått ganske god hjelp fra universitetet. Og det har gjort at vi er egentlig ganske godt skolerte. Mange av oss har fått en fin introduksjon til spesielt ulike varianter innenfor Scratch, og samtidig en god innføring i hvordan Python kan brukes.

Men det er veldig opp til oss selv. Vi må nesten bare prøve å vurdere ut fra tema og oppgave, hva som kan passe inn. Og vi skulle gjerne hatt sånn at det der, her kan det være lurt å jobbe med det, innenfor det temaet. Og dette innenfor det temaet, for eksempel. 1c, 1c

09:30

Jeg skjønte at dere bruker Scratch hovedsakelig. Er du fornøyd med det, eller vil du gjerne teste noen andre?

09:37

Jeg er jo veldig fornøyd med det. Scratch, jeg synes jo det er lett gjenkjennelig for elevene, når de typisk skal se på det jeg har skrevet på tavla istad. Og da kan de relatere akkurat den volumformen. Det finner jeg jo også der. Og hvis jeg bare flytter litt på den overflate arealformelen her, så ser jeg at det stemmer jo overens. Den tekstbaserte til Python er jo bedre på mange måter. Samtidig som at det ikke blir så visuelt for elevene. Men den gjør jo ikke nødvendigvis de samme feilene. 1c, 4c

10:24

Det kan jo bli litt abstrakt med bare tekst og tall.

10:27

Det kan det også, spesielt for elevene.

10:30

Så merker man jo også i timen at det kan by på noen utfordringer med begrensningene som Scratch har for eksempel, men det finnes jo måter å jobbe rundt det. Så det synes jeg jo løste seg.

10:45

Det gjør ofte det, og mange av elevene finner dette ut selv. Og han som fant ut først, han ligger mellom tre og fire i matte. Han er den som ikke fant det ut, som satt bak, som hadde veldig lyst til å finne det ut. Han fikk seks nå i termin en. 2c

11:04

Ja, så det er ikke nødvendigvis å korrelere med karakterer.

11:08

Nei, så det var egentlig litt gøy, akkurat det der. At han gikk da rundt etterpå og viste det til elever som på kortet har en høyere karakter i matematikkfag. 2c

11:21

Sånn cirka hvor mange timer tror du du har i året, hvor du har en form for programmering.

11:28

Ja, da tenker du på i undervisningen, hvor mange timer som legges ned her. Et sted rundt kanskje 6 til 8 timer. 1c

11:44

Ja.

11:45

Det vil jeg tro, opp mot fagdagen, ja. Jeg vil tenke rundt det.

11:49

Også har dere klokketimer og ikke tre kvarter.

11:53

Det har vi.

11:54

Stemmer, det er viktig å få med. Så litt over til noe annet. Hvordan går du frem for å tilfredsstille de kravene som er i læreplanen med algoritmisk tenkning? Har du noen plan på hvordan du skal inn på det?

12:11

Jeg har ikke sånn spesiell plan rundt det. Men vi har jo det med å, litt som det var på tavla i sted, med at elevene skal forstå hvilke algoritmer de skal bruke. Men vi prøver egentlig å se på temaet og plassere passende oppgaver som vi vet de kan løse ved hjelp av programmering da. Det er egentlig det som

hele tiden er baktanken, sånn at de skal se at det kan lønne seg å bruke programmering på enkelte oppgaver. 7c

12:59

Synes du at du har noen felles fremgangsmåte på trinnet, på skolen eller i kommunen?

13:06

Nå er jo jeg litt privilegert, for jeg og han som har hatt disse timene nå har jo jobbet veldig mange år sammen. Og dermed hatt en litt lik tanke rundt hva som er fordelene, hensikten med programmering. Og samtidig er disse ulempene, og ikke nødvendigvis at alle disse planmålene blir helt nøyaktige. Formuleringene er jo ganske utydelige i ny og ne. Hvordan man skal bryte ned disse målene, hva det er de (UDIR?) egentlig mener. Gjør vi dette for å lære programmering, eller gjør vi dette for å lære matematikk? Og så det var jo mye av det vi lurte på. Og spesielt oss to har nok prøvd å løse de ulike problemene som har dukket opp, og hva som har vært ment med disse kompetansemålene. 3c, 3a, 5a

14:16

Føler du at du får dekket alle kompetansemålene som handler om programmering, som står i læreplanen?

14:27

Det blir jo litt det med den algoritmiske tankegangen. Vi føler jo at vi er innom mange av de, også prøver vi å la disse kompetansemålene, for de er jo litt sånn vide, og du kan plassere dem inn litt forskjellige plasser. Så i og med at de ikke er så konkrete, så tenker jeg at vi får dekket inn de målene på en eller annen måte i løpet av et semester. Det gjør jeg. 3c

15:00

Er det litt samme som du tenker på overordnet del? At det bare er ganske paraply, og så tenker du at det går du gjennom litt, så får du dekket det? Er du innom og kikker?

15:12

Ja, kanskje litt. Jeg tror jo at vi er innom det vi skal, og at vi får jobba gjennom disse kompetansemålene på en god og riktig måte. Det gjør jeg egentlig, ut fra hvordan vi tolker det. 3c

15:33

Det er veldig bra. Litt tilbake til programmeringen, hvordan syn har du på det som et hjelpemiddel eller verktøy i klasserommet?

15:45

Jeg synes det er bra. Jeg synes det er veldig fint at det har blitt introdusert til elevene. Det gjør matematikkundervisningen litt annerledes. Det gjør at elevene kan sitte og lage og konstruere ting som de ikke er vant med fra før. Før var det jo GeoGebra og regneark som var de digitale verktøyene til matematikken. Så programmeringen, litt sånn "åh, nei, nå enda er det en ny ting du skulle begynne



med". Når du har sett hvor raskt elevene tar det, hvor flinke de er, bare på det å skape et program gitt denne romgeometrien, så har jeg jo elever i den andre klassen som har allerede tatt og laget et program som regner ut overflate og volum av alle disse geometriske figurene som vi har hatt til nå. Det uten å ha fått spesiell opplæring i det, men har utforsket mye av programmet selv. **1c, 7c, 7c**

**16:59**

Det er jo veldig gøy.

**17:03**

Og det ser vi at mange av disse treerne i matematikkfaget har høy kompetanse i programmering sammenlignet med ellers i matematikkfaget. **2c**

**17:13**

På motsatt side, ser du noen utfordringer ved programmering?

**17:18**

Ja, jentene opplever dette tyngre. Det er jo ikke sånn nødvendigvis over hele linja, men veldig mange av de svakere guttene har opplevd et løft på dette temaet. **2c**

**17:32**

Samtidig som jentene, sånn som et par av disse flinke jentene, blant annet det ene tilfellet isted, hun er flink altså, men hun sier at dette skjønner jeg ikke. Mens andre som har betydelig lavere matematikkarakter enn henne, kan si at dette forstår jeg mer av. Han ene som var nedi hjørnet der, gjør ingenting i matematikk. Han var inne og spilte litt, så jeg, men han jobba gjennom både det jeg gjorde på tavla og de to første oppgavene på dette arket. Han gjør virkelig ingenting. **2c, 4c**

**18:13**

Har det noe med måten matematikkfaget bygget på med at du kommer ganske langt med å lære deg å bare gjennomføre prosesser? At det er litt sånn operasjonelt av og til?

**18:29**

Det kan godt hende, men jeg tror mange opplever at det er, altså at du er litt mer kreativ, at du har muligheten til å flytte og prøve ut og feile, gøy å se. Du får en tilbakemelding med en gang, programmet ditt fungerer ikke liksom, her er det noe galt, se om du kan fikse det. Da er det mange av elevene som bare kan velge å gi opp, mens noen ser at, ok, jeg prøver å flytte sånn og sånn. Det funket. Var det jeg har gjort annerledes nå, kontra det jeg egentlig hadde som utgangspunkt? Det at de lærer på den måten, det tenker jeg er veldig bra for disse elevene, samtidig som de som har egentlig vært sånn flinke pike hele tiden og gjort det læreren sa, når de da skal prøve å finne ut hva har jeg gjort feil. Vi har gitt noen av elevene disse programmene her. Her har du et program, det er noe feil med det. Da er det vanskelig. Og det er selvfølgelig noen jenter som synes, ja, men dette skal jeg klare å få til. Mens noen opplever at, oi, jeg har ikke peiling på

hva jeg gjør i dette programmet, hva er dette for noe? Så de er ikke på sporet i det hele tatt med å finne feil. **2c, 4c,**

**19:50**

Er det noen av de praksisene du føler fungerer bedre på klasser generelt? Nå er det jo først på tavla, og så gir ut ark, og så jobber de selv. Er det noen du har funnet ut har fungert godt?

**20:06**

Noe som jeg har opplevd at elevene liker best, eller som jeg gir mest tilbakemelding på, det er gjerne å lage program der jeg demonstrerer. Ok, dette fungerer, men hvis jeg gjør sånn så fungerer det ikke. Kan dere fikse det? Det treffer de veldig. Men igjen, da er det jo veldig mange av guttene som hopper på og synes dette er uansett nivå, og synes dette er veldig gøy. Mens jentene har da gitt tilbakemelding om at det har vært litt tyngre å prøve. Men over det hele så vil jeg si at det er en bra oppgave type. **5c, 5c, 6c**

**20:51**

Er det noe du føler elevene synes er veldig utfordrende med det? Hva er det vanligste problemet?

**21:00**

Ja, det er jo gjerne disse løkkene da. "Hvis" for eksempel, å skjønne bruken av det, og det er jo gjerne det som har gjort at de har stoppet opp, og det er jo den type oppgaver elevene har fått som har vært utgangspunktet for noe som ikke fungerer. **3a**

**21:29**

Hva må jeg gjøre her hvis dette skal skje, så må dette være gjeldende. Og det har vært vanskelig for dem.

**21:40**

Altså sånne logiske tankerekker

**21:42**

Ja, veldig.

**21:44**

Hva pleier dere å gjøre for å hjelpe dem til å forstå de der løkkene?

**21:48**

Prøve å bryte det ned til et språk som gjør at disse blokkene, at jeg kan snakke rundt det. Hva vil du? Spørre elevene. Hva vil du at programmet skal gjøre? Ok, prøv å kladder litt. Hva tenker du er det neste programmet nå skal gjøre? Hva vil du? Skal vi hele tiden prøve å konstruere det for å komme ut med et så enkelt program som mulig, men man ser jo ofte at det blir noen flere blokker enn nødvendig, men likevel så kommer vi på en eller annen måte i mål. Så det er det å ha en slags oppskrift som du, de som fikk til det her nå, de sitter jo gjerne og kladder, de sitter og skriver, mens de som opplevde at her får jeg lykke til, de prøvde å ta det til hodet.

Prøvde ikke å skrive ned hva det er du har lyst til å lage. Hva vil du at programmet skal gjøre? **6c**

**22:47**

Skrive ned en kladd og så få det inn etterpå

**22:50**

Du lager en liten kladd for deg selv.

**22:52**

Så til litt det mer tekniske som vi jobber med i oppgaven vår. Vi har en modell som heter TPACK. Det er basert på tre sånne bobler med pedagogikk, faginnhold og teknologi. Så er på en måte målet å kombinere alle i en time. Hvordan, eller. Føler du at du klarer å få inn alle de tre delene i en time, med faginnhold, med programmering og med pedagogikk?

**23:26**

Jeg tenker jo det. Sånn som jeg hadde det i sted, så pedagogikken er jo veldig vid. Men det med å prøve å formidle dette ut til elevene og vise dem hva programmering går ut på, og samtidig få inn matematikkfaget. **7c**

**23:51**

Det, tenker jeg, kommer ganske tydelig frem på en sånn oppgave som dette.

Så er det jo ulikt hva elevene sitter igjen med. Det er det. Men jeg håper og tror at flere av elevene opplevde at dette med volum er mer visuelt, at man forstår det bedre. At du nå faktisk har et program som regner ut volum av alle verdens rektangulære prizmer, for eksempel. Og det er jo litt kult for mange. Mens andre var jo der at de ikke helt koblet dette med programmering og matematikkfaget, eller faginnholdet. At de sa, men hva slags lengde skal det ha? Ja, det er jo det som varierer. Men da skjønner de at "å ja!" det er det de egentlig håper på. Den reaksjonen der. Det varmer et matematikkherte hver eneste gang du hører «å ja» sånn genuint. **7c, 6c, 6c**

**25:03**

Føler du det er noe ekstra utfordring du har fått når du prøver å implementere teknologi eller programmering i forhold til pedagogikken? Er det noe vanskeligere der?

**25:20**

Ja, det kan jo bli den der «hvorfor skal vi lære dette?» Det er jo en ting. Men jeg synes jo det å forsvare programmering inn i matematikken og som en del av faget, det synes jeg egentlig er ganske greit. Det er jo en annen måte å vise dette på. Etter hvert som man har klart å lære seg å bruke programmering inn i faget, så har det egentlig bare blitt positivt for min del. Jeg skjønner at elever synes det kan være litt vanskelig, spesielt hvis de kommer med litt sånn ulike kompetanser fra de ulike barneskolene som de får fra. Men da sørger vi rett og slett for at vi starter fra scratch og jobber med scratch. Så det går vi nøye gjennom slik at vi

har med alle oss og får heller tilpasse underveis til de som er langt fremme. **6c, 5c,**

**26:20**

Merker du at det er noe mer eller mindre tidkrevende å bruke programmeringen?

**26:25**

Man skulle jo tro det. Men det var ganske mye mer tidkrevende når GeoGebra kom, vil jeg si. Det som har skjedd er at de har fjernet konstruksjon. Så konstruksjon er jo ikke noe du bruker i matematikken lenger. Og når det erstattet av med programmering, litt sånn opp i opp, for du brukte ganske mye tid på å lære elevene til å holde med passereren. Så det tenker jeg ikke er så veldig krevende lenger. **3c, 2c**

**26:56**

Har du noen eksempler der en endring av en av de tre elementene, enten programmering, pedagogikk eller matematikk, krevde at du måtte justere på det andre?

**27:08**

Justere på det andre, altså innenfor de tre?

**27:12**

Ja.

**27:17**

Nei, altså ...

**27:22**

Det er jo godt nok, hva er det? Hvis du ikke har noe så ...

**27:25**

Ikke som jeg kommer på nå, men som sagt, jeg var jo småskeptisk i starten, men det var jo basert på egen kompetanse. Men når jeg da kastet meg ut i det og forsto at dette kunne brukes som et godt verktøy inn mot matematikkfaget, og la vekk den stoltheten om at jeg skulle kunne alt best. For elevene, de raste jo forbi meg med en gang. Altså den kompetansen enkelte av elevene fikk i løpet av første og andre time med programmering, den gjorde at jeg bare måtte erkjenne at jeg ikke best, jeg kan ikke alt. Så det funket fint for meg å heller bruke de elevene som ressurser i undervisningen. **1a, 2c, 4c**

**28:17**

Ja. Er det noen måter du prøver å tilnærme deg å kombinere faginnhold, teknologi og pedagogikk i timen? Er det noen strategier du bruker for å prøve å utvikle det?

**28:31**

Nei, det er jo egentlig bare å henge med og prøve å se om det er noen gode ideer, der man har noen kreative opplegg for eksempel, som man kan bruke i undervisningen. Jeg synes jo det er veldig gøy med utfordrende og utforskende oppgaver som gjør at elever kan mestre på sitt nivå. Så dersom man

har en bank der den type oppgaver er tilgjengelig, så er det jo fort gjort at man kan gi det til elevene og oppleve at de synes matematikkfaget er gøy, selv med programmering. 7c, 6c

29:19

Sånn, hvis du spoler ti år frem i tid, tror du at du har endret noe syn på det da?

29:24

Nei. På programmeringen? Nei, det tror jeg ikke. Vi ser nå, i utdanningsvalg så er det en film som heter Jobs, sikkert en gammel kompis av dere, Steve Jobs Men han snakker jo der om en del programmering, og dette er jo tidlig 80-tallet, og det at elevene skal skjønne at det gjør vi, og det har vi gjort hele tiden, og det er egentlig det som er fremtiden. Det synes jeg er fornuftig, egentlig. Så jeg tror ikke at det forsvinner helt, altså mitt syn på programmering. 1c,

30:12

Tror du andre lærere er enige med hvordan du fremstiller programmering?

30:18

Noen. Jeg tror mange synes det er skummelt, litt som jeg sa i starten, at den følelsen av at du får trøkt over noe som du hverken kurset sier, for opplæring, som er ganske utydelig. Det var jo ingen som kunne svare på spørsmålene våre. Ingen. Hva betyr disse målene her? Alle bare sendte det bare videre, fikk aldri svar. Det gjorde at du egentlig bare, ja vel, da får vi begynne å tolke det selv da. De har jo egentlig bare sittet, og så, dette høres bra ut, og så sender vi dette ut, og sånn, ok. Men da tar vi og gjør en tolkning av dette på vår egen måte. Vi skjønnte jo at dette var kommet for å bli, så da måtte vi jo bare hive oss uti det, og håpe at det vi gjør nå,

det er riktig. Og det var jo også litt tilbakemeldingen, at det var ikke så langt fra sannheten, og da opplevde vi som hev oss uti at, ok, dette kan vi jobbe videre med. Vi velger denne strategien her. Det ser ut å funke. Men samtidig var det jo noen lærere som tenkte at dette kan jeg ikke. Også, som kastet inn håndkle, og at dette blir forvirrende, det blir for mye. Det var kanskje vanskelig bare det med Excelen i sin tid. Nå kom GeoGebra, som var i all verden her, så skal vi ikke bruke millimeterark og tegne opp linjer lenger. Også kom plutselig programmeringen. Det ble ganske mye for matematikken. Jeg tror ikke det er noen andre fag som har fått trykket ned så mye digitalt over seg, som de ikke hadde kompetanse i. 3a, 3a, 3c, 3a

32:11

Ja. Det er supert. Litt siste spørsmål. Hvis du hadde manglet en av de kunnskapene, teologi, eller matkunnskap, eller programmeringskunnskap, hvilke føler du er på en måte minst viktige, og hvordan tror du du hadde løst det i klasserommet?

32:27

Nei, det er veldig lett å svare på. Jeg hadde jo fjernet programmeringen. Jeg hadde det. Og grunnen til det er jo at jeg liker å lære. Pedagogikken må jeg ha i bunnen, uansett hvilken form av pedagogikk. Samtidig som programmering er en del av matematikkfaget. Og da ville jeg heller kunne de 95 prosent andre. 1a, 2c, 3c

33:05

Det var veldig bra det. Da er jeg fornøyd, hvis ikke du har noe mer?

33:08

Nei, jeg har ikke noe å tilføye. Tusen takk for tiden din.

### Intervju 3

Student 1: kontrollør  
Student 2: Intervjuer  
Informant 3: Intervjuobjekt

00:00

Ja, jeg er klar.

00:04

Ja, så vi skal spørre litt om programmering, spesielt i matematikk. Så vi skal begynne med hvordan har du lært deg programmering?

00:14

Jeg har først og fremst lært programmering gjennom et 30-timers kurs som Oslo kommune tilbøy alle mattelærere når den nye læreplanen kom. Men også, jeg kunne litt fra før, sånn bare som jeg har gjort på egen hånd. 1c

00:30

Var det frivillig kurs, eller var det...?

00:33

Ja, det var det. Det var jo en frivillig, ja. Det var det helt sikkert. Men jeg tror alle mattelærere var med på det.

00:40

Føler du at du har nok kunnskap da? Skulle du ønske at det var mer tilbud?

00:47

Nei, jeg føler i utgangspunktet at jeg i hvert fall sånn som faget ser ut nå, så synes jeg jeg kan nok. 1c

00:56

Men det skal jo sies også at jeg syntes nok ikke det med en gang jeg var ferdig med det kurset. Men etter at jeg har hatt det kurset og liksom undervist en runde med programmering, så tenker jeg at nå kan jeg nok. Og jeg kan ikke så mye, men jeg kan et stykke lengre enn det elevene forventes å få til. For foreløpig har det ikke vært veldig skremmende det som har kommet på eksamen, for eksempel. 1c, 3c

01:23

Nei, ikke sant. Så bare litt mer generelt, hvilken utdanning er det du har?

01:28

Jeg har allmennlærerutdanningen.

01:31

Sånn fireårig?

01:32

Ja, fireårig. Fra 2002 til 2006. Og så har jeg tatt 30 studiepoeng i tillegg med matematikk. Og

fordypningen min var naturfag og samfunnsfag, når jeg tok lærerutdanningen. 3c

01:49

Føler du at du da har fått nok kunnskap innen matematikk?

01:54

Nei, vet du hva, det synes jeg går fint. Men nå har jeg også jobbet i 17 år, så alt til sammen synes jeg er nok. Da har jeg 60 studiepoeng i matematikk til sammen. Det synes jeg holder for det jeg holder på med. 2c, 3c

02:12

Så litt mer inn, spisset mot programmering. Hvordan rolle føler du det spiller inn i matematikkundervisningen?

02:20

Der jeg jobbet før, så var det integrert i nesten hvert tema. Da fulgte vi progresjon til A-univers, den virker veldig fornuftig, men i nesten hvert tema var det litt programmering knyttet til det. Så først var det en bolk hvor det var bare programmering for å sette elevene i stand til å programmere noe som helst. Og så kom det drypp med programmering etter hvert. Men nå, her jeg jobber, har jeg ikke hatt noen ting programmering med elevene på 8. trinn enda. Nå kommer det nå snart da, men det drives litt annerledes. Det er litt annen kultur her. Jeg tror det er mange lærere som er usikre på programmering. 5c, 5a

03:09

Hvordan ønsker du at den programmerings biten skulle være? Skulle du gjerne hatt mer her eller?

03:14

Jeg er faktisk ikke helt sikker. For det har ikke blitt landet enda, hva skal være programmering i skolen. Veldig ofte i matematikk har vi mattelærere brukt eksamen som en veileder for hva samfunnet forventer at man kan innenfor de ulike tingene. Sånn som eksamenene har vært så langt, så har det vært fullstendig overkill det jeg holder på med i programmering for elevene. Men nok ikke i forhold til læreplanen altså. Så jeg er litt usikker. Men jeg tror vi kommer til å måtte ta en ordentlig periode med programmering her på skolen, i og med at vi ikke har hatt noe foreløpig. Jeg ville gjerne heller hatt det sånn. Jevnt og trutt inn i alle de forskjellige delene. 5c, 4c, 5a

04:04

Nå nevnte du litt at dere ikke har hatt programmering her enda. Men sånn ca. hvor mange undervisningstimer med programmering har du hatt per år?

04:13

Åh, det synes jeg er kjempe vanskelig å svare på. Det hadde vært lett å svare på i den forrige jobben

min. Hvis det var sånn at vi hadde en bolk hvor vi hadde programmering, så var vi ferdig med det. Nå har vi lagt ligninger i seks uker, og så lar vi det ligge og så gjør vi noe annet. Jeg synes det er helt umulig å svare på. Men la oss si at vi hadde sånn som sist gang i fjor, siste runde jeg var igjennom, så hadde vi tre uker med tre timer matematikk i hver. Sånn ni timer da, som er sånn intro-programmering, og så etter det så er det, hva skal vi si, til sammen ti minutter hver uke, for det du har litt sånn heltid. Det er helt umulig å si. Det å sette et timetall på det. **1c, 5c**

**05:17**

Ja, men det er mer enn godt nok det. Så programmering er veldig tett knyttet opp til det med algoritmisk tenkning. Hvilke måter har du tidligere gått frem for å tilfredsstille de kriteriene i læreplanen om algoritmisk tenkning?

**05:32**

Tenker du før jeg begynte å bruke programmering?

**05:36**

Ja, eller underveis også.

**05:39**

Ta spørsmålet i gang til, for nå skal jeg stille meg inn riktig.

**05:43**

Ja, ja. Så på hvilken måte har du gått frem i løpet av året for å tilfredsstille de kravene som står i læreplanen om algoritmisk tenkning?

**05:56**

Jeg gjør det ikke på noen annen måte enn i programmering. **5c**

**06:01**

**Nei**

**06:02**

Jeg tror ikke, jeg kommer ikke på noe annet. Kanskje noe innenfor funksjoner. Nå føler jeg meg litt dum da, som ikke har veldig algoritmisk tenkning. Innenfor funksjoner har jeg ofte lagt opp undervisningen av funksjon... eller intro til funksjoner som en slags programmeringsoppgave. Du kommer med en, jeg vet ikke om det har vært brukt i funksjonsmaskiner? **7c, 6a**

**06:36**

Ja, men som en boks.

**06:38**

Ja, ikke sant? Her kommer det en input, og så skjer det samme hver gang så får du en output etterpå, som jo kanskje da handler litt om algoritmisk tenkning. Det er i hvert fall det eneste som jeg kan se for meg. **6c**

**06:55**

Det passer veldig fint.

**06:58**

Kan jeg for å si en ting til?

**07:00**

Jaja, bare kom med det.

**07:01**

Kanskje også i forbindelse med figurtall. Figurtall er også litt den samme, opplever jeg i hvert fall litt samme strategi for hvordan man går frem for å nøste opp hvilke ulike deler av denne figuren er satt sammen, knytter det til trekantall, eller kvadrattall, eller whatever man gjør for noe. Kanskje det er også noe. **6c**

**07:28**

Bruker du gjerne uttrykket variabel da, eller er det da ofte mer sånn ukjent?

**07:34**

Nei, jeg bruker ordet variabel egentlig. Fra og med en gang jeg introduserer algebra, så bruker jeg det som variabel. Jeg gjør det. **3c**

**07:44**

Føler du at du har fått noen felles fremgangsmåte på hvordan man bruker programmering eller algoritmisk tenkning i skolen eller kommunen?

**07:54**

Nei, det er ikke min opplevelse. Fremdeles opplever jeg at hver lærer står liksom på sin tue, og de lærerne hos oss gjetter... Her gjetter jeg på det blir, for jeg tror kanskje det er bare meg på åttende trinn som har noe særlig programmering. Så jeg gjetter på vi kommer til å legge det opp med at jeg tar hele trinnet... et eller annet sånt. Så jeg synes ikke vi har kommet så langt egentlig igjen. Jeg tror det er mange lærere som er kjempe usikre på det jeg skulle undervise i programmering. **3a, 1a**

**08:21**

Da føler du at du står ganske fritt til å velge akkurat hva du vil, egentlig da?

**08:26**

Ja, absolutt. **3c**

**08:28**

Føler du at du får dekket de kompetansemålene som er i programmering, i læreplanen?

**08:39**

Da tror jeg... jeg skal si at... jeg svarer ja. Også første gang jeg begynte å undervise programmering, da hadde vi nettopp vært gjennom utvalget en ny lærebok. Da valgte vi at vi går for dette læreverket her. Og da var en av de grunnene til at vi valgte det, var fordi vi synes det svarte veldig bra på kompetansemålene i programmering blant annet. Men etter det, så stoler jeg bare på læreboka. Fordi jeg på en måte gjorde et tidspunkt

den jobben med å sjekke om jeg syntes det svarte til læreplanen. Så jeg tror jeg gjør det. **7c**

**09:22**

Du svarte jo tidligere at du følte at du hadde gjort overkill med tanke på eksamen.

**09:28**

Ja, absolutt. Men så er det det, ja, den er jo litt sånn, nei jeg synes ikke den læreplanen i programmering er veldig, den er ikke veldig tydelig. For det er veldig mange valg man må gjøre selv, når man har lest den læreplan. Ok, skal vi programmere tekst, eller skal vi bruke blokk? Og det har vært masse diskusjoner om hvilke språk bruker folk. Det er jo ingen av de tingene der som er landet. Det er jo på samme måte som vi i praksis har landet at hele Norge bruker Excel, hele Norge bruker GeoGebra, som verktøy. Men det har vi jo ikke landet på i programmeringsverdenen enda, ikke det hele tatt. **1c, 5a**

**10:08**

Nei, men det var egentlig neste spørsmål, så fikk du svart på det. Hvordan knytter du praksis med programmering opp mot det som står i overordnet del? Tenker du noe over det?

**10:23**

Ikke noe, og så føler jeg meg litt sånn anarkistisk, ikke noe mer enn at, jeg er litt sånn usikker på hvor vi vil ende med den programmeringsbiten. Det har jo vært diskusjon hele tiden også. Det skulle jo være programmering i alle fag, det var jo tanken i begynnelsen, når de startet å arbeide med innføringen. Og så ble det liksom, nei det skal ikke egentlig være det alikevel. Så nå er det jo, er det vel egentlig bare matematikk og naturfag som det eksplisitt står noe om programmering? Så jeg tenker kanskje at jeg ikke er helt sikker på om jeg tenker at det er så relevant i forhold til den overordnede delen. Men de som har laget den læreplanen har sikkert tenkt litt lenger frem enn meg. Så det kan godt være at om 15 år til da, så er vi veldig glad for at alle elever i Norge har litt programmering. Men jeg ser ikke det helt enda altså. I alle fall ikke nå når du kan bare få en KI til å skrive programmet til deg. Det er i hvert fall utfordrende. **5a,**

**11:35**

Godt poeng. Hvordan syn er det du har på programmering som verktøy eller som hjelpemiddel i matematikken?

**11:44**

Ja, det synes jeg er veldig kult. Jeg elsker det. For det er veldig ofte jeg ser, hvis du knytter det sammen med at utforskning har fått en mye større plass i matematikk læreplanen enn det hadde tidligere. Når jeg jobber nå med utforskningsoppgaver, altså matematisk problemløsning, så tenker jeg veldig ofte at «Faen heller, kunne jeg programmert løsninger på dette spørsmålet?» Vi hadde jo faktisk her om dagen en

problemløsningsoppgave som vi gjorde i hele personalet knyttet til matematikk, hvor jeg bare var irritert for at jeg ikke hadde en PC. For jeg hadde lyst til å prøve å se om jeg kunne programmere meg fram til... For det var jo en generalisering vi skulle prøve å gjøre da. Vi ser at hvis du legger sammen sånn og sånn og sånn, så får du alltid ni. Uansett, du får alltid ni. Da ble jeg sånn «Faen heller, hvorfor det?» «Jo, dette her bør jeg kunne ha programmert.» Så jeg synes innføring av programmeringen hører veldig bra sammen med det utvidet fokus på utforskning og problemløsning i matematikken. **7c**

**13:01**

Nå svarte du litt på det for så vidt, men det er noen spesifikke muligheter som åpner seg ved bruken av programmering?

**13:09**

Ja, det er det. Det er akkurat det å kunne bruke det som problemløsning når de kommer et stykke videre i matematikken. Og så er jo utfordringen at. Jeg har jo også undervist i programmering-valgfag, og der sitter jo "flinkisene". De som elsker å programmere, de velger programmering i valgfag. Og der har jeg sittet med noen veldig flinke elever, og kanskje har jeg vært borti et par av de som har kommet langt nok i utviklingen av modningsmessig og faglig og alt sånt, til å kunne se at man kan løse sånne typiske problemer med programmering. Jeg tror ikke det er så mange på ungdomsskolen, ellers, som kommer langt nok, verken i matematikkforståelsen eller programmeringskunnskapene, til å kunne bruke det i faktisk problemløsning. Men det er det som jeg ser på som den store fordelene, særlig på litt sånn universitetsnivå, eller kanskje videregående og eller... Jeg har ikke undervist på videregående, men der må det være kjempegøy. **7c, 7a**

**14:19**

Helt supert. På motsatt side, er det noen utfordringer du ser med programmeringen?

**14:25**

Det er veldig mekanisk. Det er det. Det er på en måte innført en ting til i matematikkfaget, som mange elever synes er veldig vanskelig. Vi har jo ulike deler av matematikken, som er mer abstrakt enn andre. Jeg vet jo at når jeg lærer på åttende trinn og introduserer algebra, så er det et par som jeg mister. Og det ser jeg at programmering er også et sånn tema. I motsetning til geometri da. Og når programmering kom inn, det som gikk ut, var jo blant annet konstruksjon. Og det mener jeg er riktig, for det er ingen som konstruerer med passer og blyant lenger. Så det er helt fint at de gjorde det. Men man tok vekk et konkret tema fra matematikken og puttet inn et som oppleves som veldig abstrakt for mange elever. Så det er den store utfordringen. Mange av de elever som synes matematikk er utfordrende, synes i alle fall at programmering er utfordrende. **4a, 3a**

15:40

Har du da noen spesifikke praksiser eller utdanningsmetoder som du føler hjelper bedre?

15:47

Ja, blokk. Ferdig snakket. Det er bare ikke noen tvil. Så det har alltid vært det. Jeg synes det er et dilemma. Når jeg har undervist i programmering, så har jeg undervist med utgangspunkt i Python. Og tenkt at jeg gjør det, for det er det som er fremtiden. Da får de holde på med den blokken på barneskolen, og når de kommer på ungdomsskolen så går vi over til tekst og så tut og kjør. Men jeg ser det at for mange av de svakere elevene, så bør de aldri røre tekstprogrammering. De bør være i blokk.

5c

16:22

Da svarte du litt på det forsåvidt, føler du at elevene får noe mestring når de får gitt den løsningen ved blokk?

16:32

Ja, det synes jeg. I blokk. Jeg har hatt programmering, og så har jeg skjønnet at her har jeg et par elever som ikke er med på notene i det hele tatt. Så har jeg tatt det helt fra scratch. Her på skolen bruker de campus-inkrement, men der jeg jobbet sist var det Kikora. Kikora hadde en blokkinnlæringsløp, så det ble kjempemotiverende for dem å redde programmeringen deres. Jeg husker ikke helt hva spørsmålet var for noe. 6c

17:16

Nei, det er helt fint. Du må bare prate. Vi skal gå litt mer over i det tekniske. Vi skal skrive om en analysemetode som heter TPACK. Har du brukt det før?

17:28

Nei, det har jeg ikke.

17:30

Det er tre forskjellige temaer. Du har teknologisk kunnskap, pedagogisk kunnskap og faglig kunnskap. Hvis du oppnår alle tre, kan du ha en god undervisningstime.

17:47

Ja, hos meg?

17:49

Ja, for lærerne. Hvor integrert føler du at teknologi, pedagogikk og matematikk sammen er i undervisningspraksisen din?

18:08

Hva var de tre? Teknologi? Da tenker jeg på kodinga.

18:13

Ja, selve kodingen.

18:16

Også matematikkfaget som fag.

18:18

Ja, og pedagogikk.

18:23

Det er et vanskelig spørsmål. Da spør jeg nesten hvor fornøyd jeg er med det. Kan jeg lese spørsmålet?

18:33

Ja, det står nederst der.

18:36

Jeg tror at det som er mest utfordrende for meg å få integrert godt nok, er den pedagogikk-biten. For jeg har ikke erfaringen enda. Jeg tenker at jeg kan programmeringene godt nok, og jeg synes at vi har klart å knytte undervisning i programmering til matematikk. I og med det som jeg snakket om, at når vi hadde figur-tallene, så skulle de programmere de 100 første figur-tallene. At vi klarte å bruke og knytte det til matematikken. Så jeg tror den utfordring min er den pedagogikk- og undervisningsmetoden-biten. For nå har jeg gjort det en gang, og da var jeg helt ... Det var første gang. Så jeg tenker at det er det som er dårligst integrert. Og så er jeg litt spent på hvis det nå gjør det sånn at vi tar fire uker med programmering, og så går vi bare videre. Da er jeg veldig spent på om jeg i det hele tatt får integrert matematikk noe særlig 2a, 1c, 5c, 2a, 5a

19:53

Ja, at det bare blir programmering...

19:55

Men for all del, alle eksemplene som brukes og de oppgaver med gjør er jo matematikkoppgaver. Det er jo det vi løser når jeg underviser i programmering, så jeg får det med. Men jeg er veldig usikker på om jeg klarer å knytte det nok til matematikken fremover. 5c, 7a

20:15

Ja, helt supert det. Ja, nå hopper vi litt over den, for det svarte du jo på. Har du et eksempel der du føler at du måtte justere på hva du fokuserte på i undervisningen? Altså av de tre, at det ble mindre fokus på programmering og mer inn mot pedagogikk, for eksempel?

20:41

Ja, det som jeg i alle fall har et eksempel på, det er at det veldig fort blir fokuset skjøvet over på selve programmeringen. For det er det som er den store utfordringen for elevene. Vi har holdt på med et eller annet. Vi har holdt på med programmering, vi jobbet bare med programmeringsverktøy som de i utgangspunktet kunne. Det er jo alltid sånn at elever kan jo ikke, selv om man har lært det bort. Men så ender vi veldig opp med at det de sliter med er programmeringen. Vi kommer ikke helt til matematikken. Det er forståelig at fokuset til

elevene, "jeg får ikke til programmeringen", da kommer matematikken i andre rekke. **7a**

**21:27**

Det tenker jeg er det som jeg har opplevd mest.

**21:30**

Ja, så det skifter mye mer mot programmering.

**21:32**

Ja, mot programmering.

**21:33**

Og fokuset er mest det tekniske.

**21:36**

Ja, rett og slett. Og så, ja, det har vært interessant å se hvor flinke må de bli i programmering før den biten går av seg selv. Det er jo helt forskjellig. I en klasse har du noen som aldri trenger å tenke på programmeringen, som bare trenger å tenke på det matematiske. Men det var vel det som er min erfaring. Det er den retningen det dessverre går av og til. **5a**

**22:03**

Hvis du føler at det er noe som du ikke kommer gjennom, for eksempel programmering, som du så nå, hvordan prøver du å endre for å få det til?

**22:17**

Tenker du det er noe teknisk bit i programmering som ikke de ...? Nei, spør en gang til. For da tenker du fremdeles om den tredelingen.

**22:25**

Ja, stemmer ...

**22:26**

Ok, ta spørsmålet i gang til, for nå er jeg koblet på.

**22:28**

Du nevnte at det ofte er mer fokus på programmering, og at det mister litt matematikken. Er det noen spesielle ting du prøver å gjøre forskjellig til neste time, for eksempel, og fokusere på for å utvikle.. sånn du kommer tilbake til å få inn matematikken?

**22:48**

Jeg skjønner hva du mener. Nei, jeg tror ikke jeg har noe svar. Jeg kan ikke huske å ha noen erfaringer med det. For det ble bare sånn at du brannslukker den timen og prøver å hjelpe de som slet med programmeringen. Og så er du på en måte videre neste gang, det er det klart. For det var bare den timen som var satt av til å holde på med programmering, samtidig som vi jobbet med å programmere noe innenfor funksjoner eller ligninger eller whatever. Så det var enklere å gjøre i denne ... Nei, jeg vet ikke, jeg synes det er vanskelig å svare på spørsmålet. For det er jo alltid sånn at du underviser i et eller annet tema i matematikken, og så har du en forventning om at ... Eller du har

egentlig ikke det, men du har for en forhåpning om at neste gang vi kommer til noe hvor de trenger disse verktøyene som du lærte da, så kan de de verktøyene. Men det er jo ikke sånn det fungerer. Det er jo sånn med ligninger også, ikke sant? Går du to uker og så plutselig de trenger en ligning, så kan de ikke det lenger. Men jeg tror ikke jeg har klart på en måte å gjøre noe med det. Nei, det tror jeg bare jeg har. Shit, denne timen ble hektisk. Nå ble det mye løping. Neste gang er det heldigvis tilbake til funksjoner igjen. **7c, 5a**

**24:02**

Ja, men det er helt supert det. Da er vi kanskje ferdig med de vanskeligste tenkespørsmålene. Men generelt det vi har snakket om nå, tror du noe av synet ditt har endret seg i ti år fram i tid?

**24:18**

Ja, det er et godt spørsmål. Eh... Jeg håper jo da, for nå er vi fremdeles i en 3-pack pakken? (TPACK)

**24:31**

Både det og programmering generelt.

**24:34**

Det som jeg håper, det er at hele skoleløpet gjør jobben sin. For det er jo programmering, jeg har jo en sønn på andre trinn, de har jo litt programmering. Og hvis det er litt programmering på andre trinn, så kan jeg kanskje komme dit da, når de begynner på ungdomsskolen. At de har såpass godt grunnlaget at jeg i større grad kan bruke programmering som verktøy for å løse matematiske problemer. Enn å bare drive opplæring i programmering. For det har det vært så langt. Og nå er det, hvor lenge er det siden nå? Ja, hvor lenge er det siden læreplanen ble... fire år siden? Og så, la oss si de tre første var helt kaos da. Og ble brukt til opplæring av personell, for det var jo ingen som var klar for programmering i 2020. Og så, la oss si det er om fire år til. Så er det kanskje sånn at de som jeg får på ungdomsskolen, har hatt liksom, ok programmering gjennom hele skoleløpet. Det gleder jeg meg til å se. For hvis de da kommer og er ganske flink til blokk, fra barneskolen, så er det lettere å ta overgangen til tekstprogrammering, og i mye større grad kunne bruke det som problemløsningsverktøy da. Det håper jeg. **7c**

**25:57**

Tror du at de fleste andre lærere som også har programmering er enige i det?

**26:03**

Jeg opplever meg egentlig som litt mer positiv enn den gemene hop. Det er litt mye motstand, fordi det er litt... Jeg synes jo ikke det er vanskelig å lære seg, men jeg skjønner jo at for veldig mange så er det en bøyg å komme seg over og lære seg. For det er jo egentlig et helt nytt fag. Det er jo ingen... Vi mattelærere har jo egentlig ikke noen bedre forutsetninger for å lære programmering enn det samfunnsfagslærere har. Mener jeg i hvert fall. Så det er mange som er skeptiske og som er liksom



sånn, uff, vi tar det liksom fire uker før eksamen, så tar vi det og så håper vi at det går greit. Jeg tror det er fremdeles en del sånn motstand i lærer-Norge mot programmering. **1a, 5c, 1a**

**26:53**

Det er jo ikke slik at du har noen insentiver for å lære seg det på egen hånd.

**26:58**

Nei, nei, det har ikke vært det. Og det der 30 timers kurset som vi tok, det var også ganske krevende. Jeg tror hvis det hadde vært vekk på... Det var jo fem ganger seks timer, digitalt. Og jeg tror hvis det hadde gått glipp av en av de, så hadde jeg vært helt fucked. Ja, ja, det mener jeg. Det mener jeg virkelig. Fordi jeg kunne egentlig ikke så mye fra før. Og det var helt sånn, jeg har aldri vært så konsentrert i hele mitt liv for å forstå noe som det programmeringen var. Og jeg gjorde faktisk, jeg er veldig sånn, jeg gjorde veldig mye, jeg gjorde mange av øvelsene og sånn. Det var liksom en sånn flink student da. Så det var akkurat på kanten om det var nok det 30 timers kurset altså. Jeg skulle gjerne tenkt mer og gjort noe mer. Jeg skulle gjerne tatt en bachelor. Og begynt å jobbe i forlaget. **1c**

**27:58**

Vet du forresten om det er noen tilsvarende som har vært igjennom her.

**28:05**

Det har jeg ikke peiling på. Det vet jeg ikke.

**28:09**

Nei, helt supert. Da er det bare et siste spørsmål igjen.

**28:13**

Litt tilbake til de tre blokkene med kunnskaper, teknologi, matematikk og pedagogikk. Hvis du på en måte hadde manglet en av de, enten programmeringsferdigheter eller pedagogikken, hvilken føler du på en måte er lettest å dekke over? Hvordan tror du du ville løst det?

**28:36**

Jeg tror det er lettest å dekke over pedagogikken. Du kommer deg ikke i mål uten programmering. Jeg kan ikke skjønne. Alternativet ville jo vært å bare gi elevene en bok på en måte. Her, lykke til. Eller et læringsløp på campus-inkrement. For det kjenner jeg til selv. Når jeg har undervist i programmering

valg, så har noen elever... For det første er det mange som blir mye flinkere enn meg. Det er helt fint, naturlig. Og så er det noen som har valgt et annet programmeringsspråk enn det jeg har. Da kan jeg på en måte ikke hjelpe dem. Når de står fast i et eller annet... La oss si at de har jobbet med javascript. Når de da har kommet et lite stykke, så kjenner jeg ikke syntaksen i det hele tatt. Det står helt fast. Så jeg tror det at... Da er du lost. Jeg tror det. Jeg er jo klart. Hvis ikke du er pedagog, hva gjør du da? Jeg vet ikke. **2c**

**29:38**

Det må jo være et pedagogisk gen i hver lærer uansett, må det ikke det da?

**29:43**

Det må det. Og for all del... Jeg har jo vært inne og undervist i tysk, selv om jeg ikke kan kvekk med tysk. Så hvis jeg hadde vært... Hvis du er en flink pedagog, da... Så skulle du jo komme et stykke på vei i programmering.

Men da hadde jeg jo valgt å ikke programmere med tekst. Og så hadde jeg heller valgt å bare se på... Du kan jo bare programmere på papir, for eksempel. Ut fra hva som foreløpig ser ut til å være på eksamen, så hadde du vært til å tolke programmering... Altså flytskemaer og sånne ting. Og det kan man jo jobbe med, selv om man ikke kan noe syntaks, da. Rett og slett. **2c, 4c**

**30:21**

Rett og slett om tankegangen.

**30:23**

Ja, det gjør jo det. Det gjør det.

**30:26**

Supert. Har du noe mer?

**30:28**

Jeg har ikke mer på blokken nå

**30:32**

Da er vi veldig fornøyde med det.

**30:34**

Ja, bra.

**30:36**

Tusen takk for tiden

## Vedlegg 8 - Observasjoner

### Observasjonsskjema Lærer 1, Student 1

#### Generelle observasjoner:

Stort klassisk klasserom, 1 lærer, sykdom

TPACK – Bruke \*2 «underst» til slutt for at det skal bli riktig «regnerækkefølge».

TPACK – «her må vi etter regne for å se hvor det går galt»

Problemløsning sammen med elev. Satte seg inn i tankeprosessen. – 2 Hvis og liste istedenfor variabel. Prøvde på noe logisk uten kunnskap om blokkens funksjon.

Tecnological	Operatør blokk (grønn)
Content	«regnerækkefølge» og stabling av operatører.
Knowledge	3.14 og ikke 3,14 for pi
Tecnological	Får hjelp i nivå 1
Pedagogical	Kan sjekke om program virker
Knowledge	Kan hjelpe hverandre (jentejeng), har gjort dette før. Prøver uten hjelp  «hva tenker de er best». Desto mindre input, desto mindre tull  «er du med på den logikken?» (IT)
Pedagogical	Ber elevene fortelle hvordan man kan løse
Content	Laget en test for programmets kvalitet
Knowledge	Viser visuelt ved i brette et ark hvordan man kan regne overflate med de formlene de allerede kan.  Oppsummerer ikke i plenum, men spør hvor langt de har kommet
Content	REP. Areal rektangel, volum og overflate
Knowledge	

Pedagogical Knowledge	Ulike startsteder «hva skal jeg gjøre» Nå må du gjøre det du skal Avslutte timen med en gåte
Tecnological Knowledge	Lagre, repetisjon variabler, hva gjør de ulike blokkene.

### Spørsmål som dukker opp til intervju:

Felles kjøreregler, kurs, utdanningsdirektoratet, læreplan

### TPACK:

Siden dette er siste, trenger de å lage egen variabel.

Går det an å bare sette inn svarverdien?

«du har laget en variabel her uten å gi den nye verdi, da regner den som null. Derfor blir svaret feil»

## Observasjonsskjema Lærer 1, Student 2

### Generelle observasjoner:

Klassisk klasserom, sitter 2 & 2. 1 klokketime undervisning, 1 lærer, 16 elever (pga sykdom)

Begynner med smartboard

Lærer presenterer, gir oppgaver, så jobber elever selvstendig

Frihet i valg av vanskelighetsgrad, presser ikke elever til for vanskelig nivå

Ikke «hele» tall i svar, eks  $Volum = 20,609$

Tecnological	Bruker powerpoint til fremvisning av intro.
Content	Lærer Viser en metode for utregning , åpen og oppfordrer til å prøve alternative løsningsstrategier.
Knowledge	Lærer vet oppgave er løselig med verktøyet scratch. Viser forståelse mellom programmet og volum.
Tecnological	Presentasjon av innhold.
Pedagogical	Viser både teknologiske ferdigheter + formler
Knowledge	Går rundt og hjelper elever i gang med oppgaver – lite venting på hjelp Vet rekkefølgen, forklarer godt retning på algoritmene Mye hjelp i starten med blokkprogrammeringen
Pedagogical	Starter med hva timen skal inneholde. ->Presenterer som scratch
Content	Lager volum-/overflateoppgaver
Knowledge	Figur på tavle, spør elever, så viser formel til figur Lager sylinder med ark, forklarer med laget figur Gir god hjelp på vanskelige oppgaver, men etterlater elever til å prøve selv. Elever prøver (og får for det meste til) Sjekker hvor langt elevene kom -> Ingen positiv/negativ bemerkning av innsats
Content	Tilsynelatende kan innhold av scratch + volum/overflate
Knowledge	Viser svar, kan fasit og fremgangsmåte Bruker ukonsekvente benevninger, eks: $6m * 4cm = 72$
Pedagogical	Ber elever lukke cromebook for forklaring
Knowledge	Oppfordrer elever til «vanskelige» oppgaver, skaper trygghet om timen Rask å hjelpe, effektiv rundt. Skryter av elever som får til. Ingen opprop – kan alle navn
Tecnological	Ber elever lagre i mappe
Knowledge	Klar med fremgangsmåte, utklipp og powerpoint digitalt Vet forskjell mellom komma og punktum i formel

### **Spørsmål som dukker opp til intervju:**

Pleier dere å jobbe med konkrete oppgaver, eller er det flest åpne oppgaver?

Ville du gjort noe annerledes mtp større klasse?

### **TPACK:**

Blander fremgangsmåte, kunnskaper og hjelp sammen godt

Kan tilsynelatende hjelpe og forklare «alt» effektivt. Ingen elever blir sittende lenge uten hjelp.

Setter krav til at elevene må jobbe

Gir ikke rene svar, får eleven til å tenke over selv hva som må gjøres, gir hint om nødvendig

# Observasjonsskjema Lærer 2, Student 1

## Generelle observasjoner:

Likt klasserom, mer tavlebruk. Starter kun på tavle

Tverrfaglig spansk

Jobber generelt, ikke spesifikt

Ca. 30 minutter gjennomgang. Konfiskerer brusboks etter gjennomgang.

Snakket lavt når læreren hjalp elevene, vanskelig å høre.

Få spørsmål når elevene begynte å jobbe.

Er alt i TCK også TPACK når vi er i klasserom?

Tecnological	«hvilke verdier trenger vi?»
Content	Hvordan får vi disse operatørene til å gange 3 ganger.
Knowledge	Hvordan gange sider i prismet for overflate og volum på en måte hvor det virker generelt. Variabelen gjør at svaret blir null. Variabelen er ikke satt. Regnerekkefølge og begrensninger i scratch
Tecnological	Hvordan kan jeg starte programmet mitt.
Pedagogical	Gjør det så ryddig og enkelt som mulig.
Knowledge	«Det er helt riktig, dette vil ikke du få til nå, men kanskje du forstår hvorfor» Du gjør så alt riktig – bare pass på at... Du trenger ikke kalle variabelen lengde hvis det blir for langt, her kan du kalle den L. «Det skjedde ikke noe, hva må jeg gjøre?»

Pedagogical Content Knowledge	Gå gjennom formler. Spør elever om forklaringer «Vær presis» Si hva det faktisk er og ikke slurv. Hva kan du om pi nå? At alle sider i en kube er like lange Klackle for å hjelpe med å holde orden. Kan snu om her. Ikke viktig (ganging) Er pi en variabel? Gi dag er det 3.14 og endrer er det... «Jeg hørte elev si at «jeg bare kopierte fra koden»».
Content Knowledge	Parantesregler og felles faktor Viser med tegninger hvordan formelen for volum og overflate fungerer.
Pedagogical Knowledge	Komme opp og peke
Tecnological Knowledge	«Opp med scratch» Vise sett variabel blokken Feilsøker når operatøren står på feil sted (tid)

### Spørsmål som dukker opp til intervju:

Ingen

### TPACK:

Hvordan ville feil metode gitt deg rett svar her?

«ikke umulig» regnefølge i TCK. Algoritmisk fremgang

Kladden og bruken av parenteser, Hvordan får vi operatørene til å gjøre dette ved å bruke variablene.

# Observasjonsskjema Lærer 2, Student 2

## Generelle observasjoner:

20 elever, 1 lærer, klassisk klasserom

Tavleundervisning

Åpner scratch -> gjennomgang

Elever jobber selv på samme ark som i 1. klassen

Tecnological Content Knowledge	Bruker uttrykk «variabel», «side»  Bruker riktig blokker, retter raskt opp egen feil.  Sjekker både med hoderegning og ved teknologi -> dobbeltsjekker alt  Sier til elev «da tar du den variabelen(på scratch) og setter den til O».  Forstår godt både teknologi og innhold.  Først innhold og formel, så teknologi
Tecnological Pedagogical Knowledge	Bruker elever til gjennomgang av scratch. Elever følger steg for steg med lærer. Viser hvordan funksjonen fungerer underveis.  Understreker forskjell på variabel og svar. Spør mange elever underveis  Sjekker at alle henger med.  Nevner regnerekkefølge, Eks: så gange eller gange så pluss.
Pedagogical Content Knowledge	Tavle – Viser formler på tavla, begynner på noe annet etterpå  Bruker elever til å forklare, grundig gjennomgang  Viser godt hvilke sider av prisme som regnes  Gir forslag til forenkling brøk, eller ikke vekt på det.  Forventer at elever kan formler for volum og overflate av alle gitte figurer  Lærer anbefaler å teste med lave tall for å sjekke ved hoderegning  Beskriver pi med humor, gjør det lett og forståelig



Content	Kan $G \cdot H$ i prisme
Knowledge	Korrekte benevninger og fagterminologi Beskriver at pi ikke er en variabel, men gitt tall
Pedagogical	Stiller krav til opp med hendene, spør varierende. Også de som ikke rekker opp hånden.
Knowledge	Går rundt og hjelper, ser alle elever Spøkefull relasjon med elevene Tillater litt småprat når han snakker og under selvstendig jobbing
Tecnological	Vet hvor ting i scratch er. Bruker blokker på en god måte.
Knowledge	Kort bom mellom boble til sek og sekund -> retter opp egen feil raskt Rask kopiering i stedet for unødvendig tid, viser forståelse av effektivitet og teknologien Vet snarveier og mappelagring osv.

### Spørsmål som dukker opp til intervju:

Ingen

### TPACK:

Viser til løsninger både gjennom hoderegning og scratch

Bruker fagterminologi og eksempler i forklaring

Inkorporerer geometri og scratch på en trygg og enkel måte

Nøye på planlegging eller kladding av formler før scratch.

# Observasjonsskjema lærer 3, Student 1

## Generelle observasjoner:

Stort rom, cromebook med hjem, Kritt og smart tavle

Elev med lite strøm

Kontekst: Time 2 med phyton, mattetime 1 lærer, 1 assistent

Trinket.io – viser resultat underveis: koding

Ingen «regnestykker» de skal løse/ generalisere eha. Programmering skal bruke det de lærte sist og i dag neste gang til å programmere

Tecnological Content Knowledge	
Tecnological Pedagogical Knowledge	Veileder prosessen programmering,  ikke matte i dette tilfellet
Pedagogical Content Knowledge	«Ja, du fikk til den første» « du får  nå å bruker variabel. Så kan du få den til å printe det»  Stor tekst, viser eksempler siste minutt så de som ikke husker kommer i mål med oppvarming  «Du har stor b her og her»  Tilbake til kommando, komma  #kommentarer: koden for å forklare funksjonen, en god kode praksis  Viser elever følger  Gå igjennom regneartene i phyton

Content	Samleark med kommando og syntaks
Knowledge	Koble variabel og input Eksempler på bruk av input fra matteboka matematisk 8
Pedagogical	Se alle elevene
Knowledge	Vise når man er klar med å vippe ned skjerm Bruke begge tavler til å huske og utfylle
Tecnological	Anvende variabel
Knowledge	

### Spørsmål som dukker opp til intervju:

#### TPACK:

Oppvarmingsoppg. Bruk

kommandoark til å printe noe VHA variabel

Løser opp i feil på tavla, desimal og regneart

# Observasjonsskjema Lærer 3, Student 2

## Generelle observasjoner:

1 lærer, 1 assistent, 16 elever

Begynner med PP/word

2. time med programmering – mye tid på oppstart

Elever kan ikke mye koding – mye tid på gjennomgang

Tecnological Content Knowledge	Vet/viser hvordan å regne med variabler i programmet Kan at program kan spørre, får til å lage kode og knytte svar til variabel Kobler kodingen av programmet til å lage en kalkulator
Tecnological Pedagogical Knowledge	Ber elever prøve å kode med en gang, hjelper alle som trenger individuelt Bruker stor skrift på tavlen, så alle elever bak kan se Ber elevene lukke skjerm under gjennomgang Sier til elev: « det er stor A der *peker* og liten a der *peker*, da blir det forskjellige variabler og programmet skjønner det ikke” Forklarer desimaltegn i phyton er «.» og ikke «,» Stegvis forklarer og viser programmeringen
Pedagogical Content Knowledge	Skriver koder på word sammen med elevene (tavle) Forklarer elever om variabler, nøye forklart så alle er med Forklarer at man kan få program til å spørre om ting Har word dokument med koder som oppdateres for hver kode de gjennomgår Forklarer forskjell på heltall og desimaltall i trinket
Content Knowledge	Kan koder nødvendig til phyton Bruker «addisjon», «subtraksjon», osv

Pedagogical Knowledge	<p>Begynner felles gjennomgang m. tavle</p> <p>Hjelper elever, ser rundt om alle får det til</p> <p>Får elever til å prøve koder underveis</p> <p>Passer på at elever forstår, før neste steg blir introdusert.</p> <p>Spøker med elever i timen – skaper relasjon med elevene</p> <p>Avslutter timen med skryt til elevene for bra arbeid gjennom timen.</p>
Tecnological Knowledge	<p>Bruker trinket- vet det fungerer</p> <p>Kan tegn &amp; hvor de finnes</p> <p>Ser feil selv. Retter opp raskt</p>

### Spørsmål som dukker opp til intervju:

Hva gjelds som teknologi og faginnhold når programmering er selve målet?

#### TPACK:

Hjelper elever med

innlogging -> gir tips for å få programmet til å funke

Viser fremgangsmåte av multiplikasjon på tavle, bruker variabler som a og b til å regne ut. Spør elever underveis om svar.

Bruker matematiske begreper i forklaringen av teknologien, der elevene får repetert begrepene gjennom timen.