



Universitetet
i Stavanger

FAKULTET FOR UTDANNINGSVITENSKAP OG HUMANIORA

MASTEROPPGAVE

Studieprogram:
Masteroppgave i matematikk,
Grunnskolelærerutdanning 5-10

Vårsemesteret 2023

Forfatter: Eirik Særsland

Veileder: Sean Martin

Tittel på masteroppgaven: Programmering i matematikk: En studie av hvordan programmering kan endre lærerrollen og bidra til god matematikkundervisning.

Engelsk tittel: Programming in mathematics: A Study on How Programming may Cause a Change in Teacher's Role and Promote Effective Mathematics Teaching.

Emneord: Lærerperspektiv, lærerrolle, matematikkundervisning, MKU, profesjonsutvikling, programmering, programmeringsdidaktikk, TPACK, teknologi i skolen.

Antall ord: 34 791
+ antall vedlegg/annet: 4 021

Stavanger, 15.05.2023.

FORORD

Dette prosjektet markerer avslutningen på min studietid på lærerstudiet ved Universitetet i Stavanger. Og spesielt masterprosjektet har vekket noe i meg. «Hva har skjedd med deg, Eirik? Du er ikke til å kjenne igjen.». Studievennene mine har ikke kjent meg igjen i arbeidet med denne masteroppgaven. Det samme gjelder for så vidt for kona også. Det er sagt med glimt i øyet og et lurt smil, men det er nok også litt sannhet i det. Masterprosjektet har vært massivt, tidkrevende og omfattende. Og jeg har likt det. En kan jo tolke utsagnene deres som at jeg ikke har glødet for oppgaver før. Jeg velger å ikke se det slik. Og jeg vet de ikke mener det slik. Det er bare det at et prosjekt som dette, med dypdykk i et selvvalgt tema og mulighet til å prate med ulike lærere, bli kjent med yrket som venter og faktisk forske litt – jeg har likt det. Og det har motivert meg. Resultatet er det største dokumentet og oppgaven jeg noensinne har produsert og levert. Men for meg er kunnskapen og erfaringen jeg selv har tilegnet meg det viktigste.

Å jobbe med dette prosjektet har gjort meg mer ydmyk. Jeg blir ydmyk overfor yrket jeg nå er i ferd med å gå inn i med alt som hører til. Mest av alt ser jeg frem til å møte menneskene jeg skal lære så mye av. Kolleger, elever, ledere og foreldre. Alle kommer til å spille en vesentlig rolle i mitt liv og i min utvikling.

Avslutningen på dette prosjektet gjør meg også takknemlig på mange områder. Det gjør meg takknemlig for muligheten til å studere i velferdsstaten Norge. Jeg er takknemlig for familie som støtter meg. Min kone og mine barn gir meg både tid til studiene og nødvendige avbrekk med uvurderlige øyeblikk og en meningsfylt hverdag. Mamma og pappa har hele tiden vist interesse for studiene mine og for min utvikling. Jeg er takknemlig for studievenner som har hjulpet meg å utvikle meg, og som jeg også har fått være med å hjelpe gjennom studiene. Refleksjoner, øving, latter og pauseunderholdning minnes med stor glede. En siste takk vil gå til min veileder, Sean Martin, som med sin erfaring har hjulpet meg med nye vinklinger der jeg har stått fast, viktige refleksjoner rundt prosjektet og motiverende tilbakemeldinger. Samtalene våre har vært matnyttige og bekreftende på at prosjektet går i en positiv retning.

Eirik Særsland

15.05.2023, Stavanger

SAMMENDRAG

Gjennom læreplanen Kunnskapsløftet 2020 er det besluttet at programmering skal være implementert i matematikkundervisning på grunnskolen. Forskning viser at lærere må ha kunnskap om faginnhold for god kvalitet på undervisningen og før innføringen av Kunnskapsløftet hadde de fleste lærere lite forkunnskaper i programmering. Forskning viser også at med programmering i matematikk endres lærerrollen. Studiens problemstilling er derfor: *Hvordan opplever lærere at programmering fremmer god matematikkundervisning og hvordan er læreres rolleforståelse med programmering i matematikkundervisning?*

Studien baseres på en flermetodestudie med semistrukturerte intervjuer og semistrukturerte spørreskjemaer av fem lærere. Datamaterialet studeres og analyseres gjennom en konvensjonell innholdsanalyse og en teoridrevet innholdsanalyse. Studiens funn kan ikke generaliseres, men resultater fra spørreskjemaene indikerer at programmering kan fremme kvaliteten på matematikkundervisning på noen områder. For at programmering skal fremme kvaliteten i undervisningen stilles det krav til læreren. Spesielt kreves det at programmeringsarbeidet knyttes til matematikk.

Resultatene fra denne studien viser at flere av lærerne har lav selvtillit på egne programmeringsferdigheter og at de finner det utfordrende å knytte programmering til matematikk naturlig. Likevel er lærerne positive til å undervise programmering i matematikk. Dette henger sammen med at programmering innlemmes gradvis i skolen og at digitale læreverker tar for seg grundige leksjoner. Lærerrollen endres med dette til å kunne sammenlignes med en trenerrolle. Verdifull tid kan nå prioriteres på andre områder av undervisningen og som øker kvaliteten på den. Lærerne får anledning til å bedre kvaliteten på undervisningen ved å bruke en større del av planleggingstiden på å finne gode problemløsningsoppgaver, bli kjent med ulike løsningsstrategier og planlegge hvordan de vil knytte programmeringsarbeidet til matematikk naturlig.

INNHOLDSFORTEGNELSE

FORORD.....	I
SAMMENDRAG.....	II
INNHOLDSFORTEGNELSE	III
OVERSIKT OVER TABELLER.....	VI
OVERSIKT OVER FIGURER	VII
1 INNLEDNING.....	1
1.1 Bakgrunn for valg av problemstilling	1
1.1.1 Å lære av lærere	2
1.1.2 Formålet med prosjektet.....	2
1.1.3 Motivasjon for prosjektet	3
1.2 Problemstilling	3
1.3 Oppgavens struktur	4
2 TEORI.....	5
2.1 Hva er god undervisning?	5
2.1.1 Hva er god matematikkundervisning?.....	6
2.1.2 Lærerens faglige kunnskap.....	8
2.1.3 Noe supplement fra studier på norske lærere	9
2.1.4 Programmeringsdidaktikk	9
2.1.5 Lærerens rolle.....	12
2.2 Programmering.....	14
2.2.1 Hva er programmering?	14
2.2.2 Hvorfor lære programmering?	16
2.2.3 Hvilke ferdigheter kreves for å mestre programmering?	18
2.3 Programmering i Kunnskapsløftet 2020	22
2.3.1 Relevans, dybdelæring og sammenheng	22
2.3.2 Algoritmisk tenkning i Kunnskapsløftet 2020	24

2.3.3 Eksamensoppgaver med programmering	25
2.4 Teoretiske rammeverk	28
2.4.1 MKU	28
2.4.2 TPACK	30
3 METODE	32
3.1 Oversikt over forskningsdesign	33
3.2 Overveielser ved datainnsamlingen	34
3.2.1 Flermetodestudie	34
3.2.2 Semistrukturert intervju	35
3.2.3 Semistrukturert spørreskjema	38
3.2.4 Studiens deltakere	41
3.3 Analysemetode	42
3.3.1 Analyse av spørreskjema	42
3.3.2 Analyse av intervjuer	44
3.4 Forskningsetiske perspektiver	47
3.4.1 Samtykke	47
3.4.2 Konfidensialitet	48
3.4.3 Søknad til Kunnskapssektorens tjenesteleverandør (Sikt)	48
3.5 Studiens kvalitet	49
4 RESULTATER	51
4.1 Resultater fra spørreskjema	51
4.1.1 Matematisk rikdom	51
4.1.2 Matematikkfaglig opptak og tilbakemelding	52
4.1.3 Feil og unøyaktighet	53
4.1.4 Elevdeltakelse i matematiske resonnement og meningsskapning	54
4.2 Resultater fra intervju	56
4.2.1 Lærernes tanker om hva som kjennetegner god matematikkundervisning	56

4.2.2	Lærernes reaksjoner til implementering av programmering i LK20.....	57
4.2.3	Lærernes programmeringsferdigheter og profesjonsutvikling	58
4.2.4	Hva er lærernes forståelse av programmering i matematikk i dag?	62
4.2.5	Hvordan undervises programmering i matematikk?	67
4.2.6	Hvordan lærerne mener at programmering bør undervises.....	70
4.2.7	Hva sier lærerne om den endrede lærerrollen i matematikkundervisningen med programmering?	72
5	DISKUSJON	74
5.1	Om programmering og matematikk	74
5.1.1	Algoritmisk tenkning og problemløsning	75
5.1.2	Motivasjon og engasjement.....	75
5.2	Programmering kan fremme kvalitet i undervisningen.....	76
5.2.1	Elevdeltakelse i matematisk resonnement og meningsskaping.....	77
5.2.2	Matematikkfaglig opptak og tilbakemelding	78
5.3	Lærerferdigheter for god undervisning	79
5.3.1	Grunnleggende ferdigheter.....	79
5.3.2	Ulike strategier og tilknytning.....	80
5.3.3	Horisontkunnskap og langsiktig koherens	81
5.3.4	Utvikling av lærerferdigheter	82
5.4	Lærerrolle i endring.....	83
5.4.1	Programmering endrer lærerrollen	83
5.4.2	Lærer i en digital verden	85
5.5	Hvordan undervise programmering	86
5.5.1	Utforskende matematikkundervisning	86
5.5.2	Visuell matematikk	88
6	Avslutning	90
6.1	Svar på forskningsspørsmål	90

6.2 Avsluttende kommentar	92
6.3 Implikasjon for videre forskning	94
7 LITTERATURLISTE	95
VEDLEGG	101

OVERSIKT OVER TABELLER

Tabell 1: Basic skills og applied skills fra Casner-Lotto & Barrington, 2006, s. 9.	17
Tabell 2: Eksempler på kompetansemål som omhandler programmering (Kunnskapsdepartementet, 2019).	23
Tabell 3: Oversikt over kompetansemål som omhandler programmering på grunnskolen (Kunnskapsdepartementet, 2019).	24
Tabell 4: MKU-rammeverket med dimensjoner og indikatorer. Hentet fra Opsvik og Skorpen (2014).	29
Tabell 5: Forskningsdesign	33
Tabell 6: Kvale og Brinkmanns kvalitetskriterier for et intervju skrevet inn i tabell. (2019, s. 194).	35
Tabell 7: Eksempel på tilpassing fra observasjonsskjema til påstand.	39
Tabell 8: Eksempel på forkorting av påstand fra observasjonsskjema.	42
Tabell 9: Fusjonering av to kategorier til én påstand.	42
Tabell 10: Eksempel på motsatt poenggiving av likt svaralternativ.	43
Tabell 11: Eksempel med svar, kommentar og poengscore.	44
Tabell 12: Overskrifter generert fra intervju gjennom konvensjonell innholdsanalyse.	45
Tabell 13: Beskrivelse av innhold i teknologisk innholdskunnskap og kode.	46
Tabell 14: Oversikt over resultat av programmeringens innflytelse på matematisk rikdom. ..	51
Tabell 15: Oversikt over resultat av programmeringens innflytelse på matematikkfaglig opptak og tilbakemelding.	52
Tabell 16: Oversikt over resultat av programmeringens innflytelse på feil og unøyaktighet. .	54
Tabell 17: Oversikt over resultat av programmeringens innflytelse på matematiske resonnement og meningsskaping.	55

OVERSIKT OVER FIGURER

Figur 1: Undervisningskunnskap i matematikk (UKM). Oversatt fra Ball et al. (2008). Hentet fra Opsvik og Skorpen (2014).....	8
Figur 2: Iterativt arbeid foregår i sykluser. Hentet fra Stenseth et al. (2019) s. 8.....	11
Figur 3: Den algoritmiske tenkeren. Figuren er tilpasset fra Barefoot Computing (UK) og hentet fra Udir (2019).....	25
Figur 4: Prøveeksamensoppgave (10. trinn) 2023 som omfatter programmering. Hentet fra udir.no	26
Figur 5: Eksempel på god elevbesvarelse på prøveeksamensoppgave. Hentet fra udir.no.....	26
Figur 6: Prøveeksamensoppgave (10. trinn) 2023 som omfatter programmering. Hentet fra udir.no	27
Figur 7: TPACK-modellen av Matthew Koehler og Punya Mishra, 2009, gjengitt med tillatelse fra utgiveren © 2012 fra tpack.org	30

«Når lærerne forstår sine egne læringsprosesser og sin faglige utvikling, bidrar det til selvstendighet og mestringsfølelse. Opplæringen skal fremme lærernes motivasjon, holdninger og læringsstrategier, og legge grunnlaget for læring hele livet ...

... Ved å reflektere over egen og andres læring kan lærerne litt etter litt utvikle bevissthet om egne læringsprosesser. Lærere som lærer å formulere spørsmål, søke svar og uttrykke sin forståelse på ulike måter, vil gradvis kunne ta en aktiv rolle i egen læring og utvikling. Gjennom arbeid med faglige utfordringer vil lærerne få kunnskap om hvordan de lærer og utvikler seg i faget. Dypere innsikt utvikles når lærerne ser sammenhenger mellom kunnskapsområder, og når de behersker et mangfold av strategier for å tilegne seg, dele og forholde seg kritisk til kunnskap.

Til tross for lærernes egeninnsats og bruk av læringsstrategier vil enkelte ha utfordringer med å lære. Årsakene er ofte mange og sammensatte. Ambisjonen om å utvikle evnen til livslang læring hos alle lærere krever derfor en bred tilnærming fra skolen.»

Hentet og omskrevet fra Kunnskapsdepartementet, 2017

1 INNLEDNING

Gjennom fagfornyelsen (Kunnskapsdepartementet, 2019) blir matematikklærere satt til å skulle undervise i et helt nytt tema: programmering. OECD er en internasjonal organisasjon som arbeider med blant annet myndigheter og politikere om en bedre politikk for utvikling av en bærekraftig økonomisk vekst og bedre liv. Organisasjonen er blant de som ønsker en endring i skolens undervisning for å holde tritt med den digitale utviklingen. I OECD sin rapport fra 2019 står det følgende:

«The digitalisation of economies and societies increases the need to develop a set of digital skills in school. As student assessment rarely measure digital competencies, there is little evidence on how to best develop these skills. Nevertheless, countries need to make sure they implement a consistent approach throughout the school years, focusing on what needs to be learnt, such as computational thinking, rather than on specific computer use or software skills that can quickly become obsolete.» (OECD, 2019, s.179)

Utdanningsdirektoratet ser viktigheten av at elever lærer å kunne programmere, og i 2020 blir Kunnskapsløftet 2020 (LK20) innført hvor programmering blir integrert i matematikkfaget. I LK20 finnes det momenter av programmering direkte og indirekte i kompetansemålene for matematikk på hvert eneste trinn på grunnskolen fra 2. klasse.

1.1 Bakgrunn for valg av problemstilling

Ball et al. (2008) sier at lærerens faglige kunnskap påvirker muligheten for å skape god matematikkundervisning. Nå er programmering inne som et nytt tema i matematikk med koding, løkker og algoritmer. Programmering er lagt på matematikklærere og skaper samtidig mange spørsmål og stor undring. Lærerne har ingen utdanning i programmering, og det er derfor begrenset med kunnskap om programmering blant lærere, samt begrenset erfaring med å integrere programmering i matematikkundervisningen (Kaufmann & Stenseth, 2021). Dette gjør det interessant å undersøke hvordan lærere løser det å undervise i programmering uten den faglige kompetansen om temaet. I tillegg viser forskning til at programmering gjør at hele lærerrollen er i endring i undervisning med programmering (Forsström & Kaufmann, 2018). Hvordan ser dette ut i praksis?

1.1.1 Å lære av lærere

Skolen er et sted der læring forekommer på mange plan. Ikke bare blant elevene, men også blant lærerne. I den overordnede delen av Kunnskapsløftet 2020 står det at elevene gjennom skoleløpet skal lære å lære. Metakognisjon kalles dette. Det er både interessant og nyttig å endre perspektiv i ny og ne. I forkant av dette kapitlet står en omskrevet versjon av kapittel 2.4 i den overordnede delen av LK20. Alle steder der originalen skriver «elevene» er skiftet ut med «lærerne» og teksten endrer perspektiv. Ved å personliggjøre teksten på dette viset kan en lettere sette seg inn i et elevperspektiv. Med programmering befinner mange lærere seg plutselig igjen i rollen som elev. Det kan oppleves både positivt utfordrende, men kanskje også litt skummelt.

Læreryrket krever at lærere er gode på å tilpasse seg. Noen ganger må de venne seg til nye klasser, elever eller kolleger, andre ganger er det nye fag eller nytt faginnhold. Læring omhandler også å tilpasse seg og å være åpen for ny forståelse. Selv om mange lærere ikke kan alt fra før, er lærere tilpasningsdyktige og vant med endringer. Denne gangen må lærerne venne seg til en ny læreplan. Med ny læreplan skjer det flere endringer. Elever og lærere er satt i en posisjon som er annerledes i dag enn den har vært tidligere. Lærere må forstå og anerkjenne endringene, og det kan være nyttig for lærere å forsøke å sette seg inn i elevenes nye posisjon, og ikke anse elevrollen som den samme som den var ved ens egen skolegang. Dette vil kunne føre til at en får forståelse for at en må endre tilnærmingen til rollen sin og kanskje også endre en selv.

Å se til dyktige og erfarne mennesker for å tilegne seg kunnskap er ofte fornuftig, og det er snart gått tre år siden LK20 først trådte i kraft. Denne studien vil gå nærmere inn på hvordan lærere har arbeidet og tilpasser seg for å lære noe de ikke nødvendigvis har dyp kunnskap om fra tidligere. Masterprosjektet vil studere hvor utviklingen med programmering i matematikk står i dag, og undersøke hva lærere ønsker at dette skal lede til.

1.1.2 Formålet med prosjektet

Den teknologiske verdenen utvikles i rekordfart. Dette medfører at forskning raskt består av utdatert innhold, og det er behov for kontinuerlig oppdatert forskning. Målet med denne studien er å undersøke hvordan ulike lærere i dag opplever at programmering kan benyttes for å fremme matematikkundervisningen, og hvordan de ser på sin lærerrolle med programmering integrert i læreplanen. Prosjektet har et særlig fokus på lærernes egne oppfatninger og erfaringer med hvordan programmering kan bidra til kvalitet i undervisningen, hvordan man kan utøve undervisning med programmering og hvordan dette påvirker deres rolle som lærer. Det er ikke

et mål å vurdere om lærerne har kvalitet i sin undervisning eller om de besitter gode programmeringsferdigheter, selv om dette er komponenter som kan være av betydning for resultatet. Det overordnede målet med prosjektet er å bidra til større forståelse av matematikklæreres hverdag og praktisering av det nye temaet «programmering», samt å undersøke hvordan lærere opplever at rollen deres er i endring.

1.1.3 Motivasjon for prosjektet

Enhver forskning starter med et problem og nysgjerrighet (Christoffersen & Johannessen, 2012). Det er nysgjerrigheten til selve problemet, hvor det er noe vi ikke forstår, noe vi undrer oss over eller noe som vi ønsker å vite mer om, som motiverer til forskning. I lang tid har jeg vært interessert i matematikk. Skoleløpet har hjulpet meg slik at jeg har fått utfordret meg og utviklet matematikkunnskapene mine. Jeg har forstått at jeg er motivert for faget, mye fordi jeg kjenner at matematikk er noe jeg mestrer. Mestring gjør at jeg kjenner på stor glede av å bruke matematikkunnskapene mine og ikke minst viktigheten av at jeg, og andre rundt meg også, burde bruke og lære dette. Derfor vil jeg trives i rollen som matematikklærer. Men hva om jeg ikke hadde mestret matematikk? Ville jeg da vært motivert for å undervise dette faget? Hva om jeg liker matematikk, mestrer mye, men ikke alt? Hvordan skal jeg da arbeide for å gi elevene grundig undervisning i tema jeg selv ikke mestrer? Selv har jeg lite forkunnskaper om programmering, og grunnskolelærerutdanningen jeg har gått har heller ikke hatt et fokus på at nyutdannede lærere skal kunne programmere. Min motivasjon for dette prosjektet ligger i spørsmålene som er stilt over, samt et ønske om å tilegne meg kunnskap om programmering og undervisning av programmering.

1.2 Problemstilling

Holme og Solvang (1996) forteller at det er gitte krav til en problemstilling og peker ut tre momenter som må gjenspeiles i problemstillingen. Disse er at den skal være spennende, fruktbar og enkel. I dette ligger det at ny erkjennelse skal være både mulig og ønskelig, det skal være faglig videreutvikling, avgrenset og presist (Holme & Solvang, 1996). Med bakgrunnen forklart i forrige delkapittel og Holme og Solvangs (1996) kriterier som fundament for bearbeidelse av forskningsspørsmål, har denne studien fått følgende problemstilling:

Hvordan opplever lærere at programmering fremmer god matematikkundervisning og hvordan er læreres rolleforståelse med programmering i matematikkundervisning?

Problemstillingen består av to forskningsspørsmål som skal besvares. Den første delen av problemstillingen benytter blant annet ordene «opplever» og «god matematikkundervisning».

Med «opplever» vil det bli undersøkt hvordan lærere erfarer og oppfatter gjennom deres øyne at situasjonen er. Det blir en viktig del av denne studien å identifisere hva forskning sier god matematikkundervisning innebærer. Dette blir utdypet i delkapittel 2.1. Etter å ha definert hva god matematikkundervisning innebærer vil det bli undersøkt hvordan lærere opplever at programmering kan fremme ulike faktorer ved god matematikkundervisning. Det andre forskningsspørsmålet er inspirert fra Forsström og Kaufmann (2018) sin forskning om at programmering endrer lærerrollen. Med rolleforståelse menes læreres forståelse av arbeidsoppgaver knyttet til matematikkundervisning. Studien har undersøkt gjennom intervjuer hvordan lærere selv forklarer hvordan de opplever at programmering kan tilføre og endre på arbeidsoppgaver som igjen endrer matematikkundervisningen.

1.3 Oppgavens struktur

- Kapittel 2 Teori. Forskning om hva god matematikkundervisning bør inneholde og kompetanse læreren bør besitte. Tidligere forskning på hvordan programmering bør undervises, hva programmering er og hvilke ferdigheter en behøver for å kunne programmere. Programmering i LK20, eksamensoppgaver med programmeringsinnhold og teoretiske rammeverk.
- Kapittel 3 Metode. Forskningsdesign og overveielser med valg av metode. Beskrivelse av studiens deltakere og fremgangsmåte med tilpasninger av rammeverk og analysemetode. Avsluttes med forskningsetiske perspektiver og beskrivelser av kvaliteten av denne studien.
- Kapittel 4 Resultater. Funn fra spørreskjema og intervjuer presenteres.
- Kapittel 5 Diskusjon. Resultater fra denne studien diskuteres i lys av teori med mål om å finne svar på forskningsspørsmålene.
- Kapittel 6 Avslutning. Konklusjon av denne studiens svar på forskningsspørsmålene, avsluttende kommentar og implikasjon for videre forskning.

2 TEORI

Undring og refleksjon hjelper oss som mennesker til å ville utforske for å forstå verden rundt oss. Det å kunne dra nytte av andres erfaringer, forståelser og utforsking gjør at en kan begynne sin undring og refleksjon fra et utviklet perspektiv. Ny utforsking basert på utviklede teorier gjør at samfunnet og verden vi lever i fortsetter å utvikles. Når en driver forskning, er det nødvendig å kontekstualisere forskningen sin og se den i lys av tidligere forskning på feltet (Postholm & Jacobsen, 2018). Dette kapitlet kontekstualiserer studien og gir nyttig bakgrunnskunnskap før vi går inn i forskningsdelen av studien. Teorikapitlet bygges opp av det tidligere forskning sier god matematikkundervisning innebærer og er, og hvilke ferdigheter en lærer må ha for å gjennomføre god matematikkundervisning. Dette belyses både i internasjonal og norsk kontekst. Etter dette vinkles kapitlet mot undervisning av programmering: Hva forskning sier om undervisning av programmering, forventninger til lærernes kompetanse og hvordan lærerens rolle er i endring på grunn av dette. Videre går vi dypere inn i temaet programmering. Vi ser på hva programmering er, hvorfor mennesker bør lære seg å programmere og hvilke ferdigheter som kreves for å kunne programmere. Avslutningsvis ser vi på hvor LK20 implementerer programmering gjennom fagfornyelsen, hvordan dette fremstilles og hvordan eksamensoppgaver vektlegger programmering.

2.1 Hva er god undervisning?

PISA (Programme for International Student Assessment) måler 15-åringers kompetanse i lesing, matematikk og naturfag og er en internasjonal undersøkelse som gjennomføres hvert tredje år. På generelt grunnlag velger PISA-undersøkelsen å måle god undervisning gjennom tre faktorer: Tydelig struktur og klasseledelse, støttende lærer med klare mål og kognitivt utfordrende oppgaver (Olsen, 2013). TIMSS (Trends in International Mathematics and Science Study) er en annen internasjonal undersøkelse som måler elevens kompetanse i matematikk og naturfag. I tillegg samles det inn informasjon om elevene, lærerne og skolene. I TIMSS måler en generell undervisningskvalitet gjennom fire svært like dimensjoner som i PISA. Disse er god klasseromsledelse, støttende lærer, tydelige intensjoner og kognitive utfordringer (Bergem et al., 2016). God undervisning er derfor interaksjoner og graden av oppnåelse av disse tre og fire omfattende dimensjonene. I denne studien ser vi nærmere på den faglige kvaliteten av undervisningen i matematikk.

2.1.1 Hva er god matematikkundervisning?

Douglas A. Grouws er en pensjonert professor med fokus på matematikkundervisning og har vært en sterk bidragsyter til forskning innenfor undervisning og læring i matematikk. James Hiebert er en pensjonert professor i matematikk. Han var leder for etterarbeidet av videoanalyser fra TIMSS-undersøkelsen i 1999 der blant annet undervisningen i seks høypresterende land ble grundig studert. Forskningen hans har hatt et stort fokus på forbedring av matematikkundervisning. Hiebert og Grouws har sammen skrevet et kapittel i «Second handbook of mathematics teaching and learning» fra 2007. Kapitlet omhandler forholdet mellom undervisning og læring i matematikk og forfatterne ønsker å finne ut hva som kjennetegner effektiv matematikkundervisning. Forfatterne konstaterer at mangelen på undervisningsteorier gjør det krevende å si hva som er god undervisning (Hiebert & Grouws, 2007). Det er flere grunner til at det er vanskelig å definere hva som er god undervisning. En av utfordringene er at ulike undervisningsmetoder egner seg til ulike læringsmål (Hiebert & Grouws, 2007). En annen grunn er fordi det ofte er så uendelig mange faktorer som samhandler med hverandre i en og samme undervisningsmetode (Hiebert & Grouws, 2007). De konkluderer likevel med at god matematikkundervisning omhandler å ha effektivt læringsutbytte (Hiebert & Grouws, 2007). Mange forskere benytter ordlyden «effektiv matematikkundervisning», men det finnes ingen universell definisjon på hva det er. Likevel kan det forklares som undervisning som fremmer forståelse og læring i matematikk (Fauskanger, 2017).

For å vite hvordan en bør undervise, er det nødvendig å vite hva som er målet med undervisningen. Mange ser det naturlig å se på målet med matematikkundervisningen som utvikling av *matematisk kompetanse*. Mona Nosrati er førsteamanuensis ved Matematikk-senteret og Kjersti Wæge er leder ved Matematikksenteret. Ifølge Nosrati og Wæge (2015) består matematisk kompetanse av fem komponenter: forståelse, beregning, anvendelse, resonnering og engasjement. Cuoco et al. (1996) er tidlig ute med å oppfordre til en revolusjon av matematikkundervisningen med sin artikkel, «Habits of mind». De ønsker å gå bort fra en passiv, instrumentell undervisning og insisterer på at undervisningen må bygge opp om at elevene skal finne mønstre, beskrive de matematiske prosessene de gjennomgår og at elevene skal være utforskende og oppfinnsomme (Cuoco et al., 1996). Hele fokuset i undervisningen er på å utvikle den relasjonelle forståelsen, og når det skjer utvikler elevene også sin instrumentelle prosedyrekunnskap (Nosrati & Wæge, 2015). Hiebert og Grouws (2007) forteller at matematikkundervisningen skal legge til rette for at elevene skal ha *opportunity to learn*, som innebærer at undervisningen legger til rette for å kunne lære og utvikle *skill efficiency*

(ferdigheter) og *conceptual understanding* (relasjonell forståelse). Ferdighetsbegrepet kommer fra læringsteorier fra psykologen Robert M. Gagné og omhandler raske og nøyaktige beregninger av matematisk innhold. En kan sammenligne det med god instrumentell forståelse som ofte utvikles gjennom tradisjonelle undervisningsformer med læreren som introduserer tema, viser eksempler og lar elevene løse og repetere lignende oppgaver, ofte med lik strategi (Nosrati & Wæge, 2015). For å utvikle relasjonell forståelse må en fokusere på sammenhenger mellom ideer, fakta og prosedyrer (Hiebert & Grouws, 2007). Det kan gjøres blant annet ved å bruke en mer moderne tilnærming for undervisning gjennom en undersøkende matematikkundervisning (Nosrati & Wæge, 2015). Dette innebærer å la elevene få streve med problemløsningsoppgaver (Hiebert & Grouws, 2007).

Mathematical Quality of Instruction (MQI) er et kvantitativt rammeverk som er utviklet av Heather Hill i samarbeid med forskerkolleger ved University of Michigan og Harvard University. Rammeverket baserer seg på undervisningsteorier, forskning på effektiv undervisning av matematikk og analyser av flere hundre amerikanske lærere (Harvard University, u.å.). MQI er utviklet med et perspektiv der det matematiske arbeidet som foregår i klasserommet er i fokus, og ikke klasseromsklima, pedagogikk eller bruk av generelle undervisningsstrategier. Kvaliteten av undervisningen vurderes ut ifra scoren av fem domener (Harvard University, u.å.). Disse er *Common core-aligned student practices*, *working with students and mathematics*, *richness of mathematics*, *errors and imprecision* og *classroom work is connected to mathematics*. «Common core-aligned student practices» omhandler elevenes engasjement til å stille spørsmål med matematisk innhold, matematiske forklaringer og om elevene blir spurt om å grunngi deres besvarelser. «Working with students and mathematics» går ut på om læreren lytter til, tolker og forstår elevenes matematiske ideer, som dermed gjør at læreren raskt kan rette opp i misoppfatninger som ellers ville ledet til feil svar. «Richness of mathematics» inkluderer to elementer, meningsskapning og matematisk praksis. Meningsskapning omhandler å se sammenheng mellom ulike matematiske ideer (f.eks. brøk og forhold) eller flere representasjoner av samme idé (f.eks. tallinje og figurer). Matematisk praksis inkluderer mulighet for ulike fremgangsmåter, men det gis anerkjennelse for effektivitet og enkelhet. Dette fører til at elever blir bedre på å generalisere. «Errors and imprecision» er feil og mangler ved lærerens forståelse av innholdet eller unøyaktighet ved presentasjoner og forklaringer. Til slutt tar «Classroom work is connected to mathematics» for seg om aktiviteter i undervisningen har et matematisk poeng eller ikke. Dette rammeverket er utgangspunktet for et av rammeverkene denne studien er analysert gjennom.

2.1.2 Lærerens faglige kunnskap

Det er grundig dokumentert at lærerkompetansen og læringsutbyttet hos elevene har en sammenheng (Kunnskapsdepartementet, 2014) og søkelyset har endret seg betydelig siden Lee Shulman (1986) omtalte det manglende fokuset på faginnhold i undervisning som «the missing paradigm». Ball et al. (2008) og forskermiljøet ved University of Michigan utviklet en modell av kunnskapen lærere trenger (Figur 1) for å skape god matematikkundervisning (Mathematical Knowledge for Teaching, MKT) basert på Shulman (1986) sin belysning av behovet for mer fagdidaktikk i lærerutdanningen. Modellen deler opp Shulmans innholdskunnskap i flere underkategorier som Fauskanger et al. (2010) har oversatt til norsk. Fauskanger et al. (2010) kaller modellen for undervisningskunnskap i matematikk (UKM) og oversetter kategoriene til allmenn fagkunnskap, spesialisert fagkunnskap, kunnskap om den matematiske horisonten, kunnskap om faglig innhold og elever, kunnskap om faglig innhold og undervisning og kunnskap om læreplan og pensum. Svakheten til modellen er at den ikke tar for seg all praktisk kompetanse som utøves i klasserommet (Opsvik & Skorpen, 2014).



Figur 1: Undervisningskunnskap i matematikk (UKM). Oversatt fra Ball et al. (2008). Hentet fra Opsvik og Skorpen (2014).

Shulman (1986) har også inspirert Ma (2020) til hennes høyt anerkjente forskning, «Knowing and Teaching Elementary Mathematics». Hun introduserer *Profound Understanding of Fundamental Mathematics* (PUFM) som er noe som kjennetegner gode lærere og som er egenskaper lærere må ha for å undervise på høyt nivå. PUFM innebærer fire egenskaper den gode læreren har: *connectedness* (kobler sammen ulike matematiske emner), *multiple perspectives* (kjenner til ulike løsningsstrategier), *basic ideas* (kjenner de elementære, grunnleggende reglene i matematikk) og *longitudinal coherence* (kjenner til hva elevene skal

lære videre). De viktigste faktorene for at en lærer kan tilegne seg PUFM er *grundige forberedelser, stram struktur* og en *tydelig læreplan* som følges nøyaktig og som ikke er åpen for tolking (Ma, 2020). I tillegg forteller Ma (2020) at kinesiske lærere utvikler seg matematikkfaglig gjennom ukentlige faggruppemøter med diskusjon av læremateriell, de lærer av elever og de lærer av å arbeide med fagstoff på fritiden sin.

I OECD-rapporten om den digitale verden vi møter står det følgende: «*There is a need to provide high-quality training to teachers on how best to integrate technology in their pedagogical practices. Teachers are less likely than other tertiary-educated graduates to perform well in problem solving in technology-rich environments*» (OECD, 2019, s. 30). Det er altså et stort ønske om å få kompetanseheving av lærerne for at kvaliteten på undervisning av programmering skal bli bedre. Samtidig er det uklart, og en omfattende diskusjon, hvilken kompetanse lærerne behøver for å avgjøre hvordan programmering best kan integreres i lærerutdanningen (Kaufmann & Maugesten, 2022).

2.1.3 Noe supplement fra studier på norske lærere

I norsk kontekst er det gjennomført undersøkelser av hva matematikklærere i norsk skole generelt mener er ingrediensene i god matematikkundervisning (Fauskanger, 2016) og hvilken kunnskap som er nødvendig for effektiv matematikkundervisning (Fauskanger, 2017). Studiene viser at norske lærere anser elevrespons som den viktigste faktoren for undervisning av høy kvalitet (Fauskanger, 2016) og fremhever egenskaper ved læreren, som engasjement og holdninger, heller enn fagkunnskap og undervisningskunnskap som mest nødvendig for å gjennomføre effektiv matematikkundervisning (Fauskanger, 2017).

2.1.4 Programmeringsdidaktikk

Norge er blant de siste europeiske landene til å innføre programmering i de nasjonale læreplanene. Det er derfor nyttig å se til andre land for å undersøke hva som er god programmeringsdidaktikk. Crick (2017) har gjort en omfattende undersøkelse av eksisterende forskning om Computer Science (CS) etter at det ble innført ny læreplan i England i 2014. Crick (2017) skulle undersøke progresjonen som var skjedd i undervisningen av CS og identifisere hvilke områder som behøvde videre utforskning. Han belyser at det har blitt stilt de samme forskningsspørsmålene om programmeringspedagogikk i 30 år, uten at det er bevist hvilken undervisning som er mest effektiv (Crick, 2017). Selv om det er mye forskning som er gjennomført, er det et behov for mer forskning på gode verktøy for undervisning av programmering. En annen ting Crick (2017) belyser, er spørsmålet om hvordan man underviser

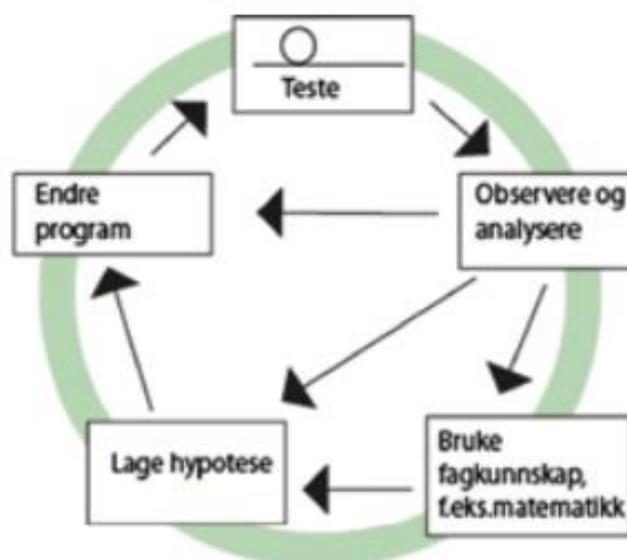
Computational Thinking (CT) (mer utdypet i delkapittel 2.2.3) effektivt. Undervises hvert delement hver for seg, eller undervises alle delkomponentene på samme tid?

Waite (2018) fikk oppdraget med å følge opp Cricks (2017) funn med enda mer litteraturgjennomgang. Waite (2018) viser til at ny forskning indikerer at en pedagogikk som inkluderer veiledet utforskning, målrettede oppgaver og kreativ og åpen problemløsning gir en effektiv læringsplattform. Det er likevel behov for å bevise at dette gjelder. Det kommer frem av litteraturforskningen at det er tendenser til å være fordelaktig å benytte fysiske komponenter i programmeringen, men at det er lite forskning på hvilke fysiske komponenter som bør benyttes og hvilke undervisningsopplegg som er mest effektive. Et viktig aspekt ved fysisk programmeringsundervisning er finansieringen som kreves. For å rettferdiggjøre finansieringen som må til, behøver lærere derfor pedagogiske opplegg for å utnytte potensialet til de fysiske elementene og for å maksimere investeringen. Spillutvikling viser seg å være svært motiverende blant elever, men det er uklart hvordan det gir en varig motivasjon hos elevene (Waite, 2018). «Unplugged»-aktiviteter er aktiviteter som kan kobles til ferdigheter som kreves for å programmere, men som ikke krever datamaskin. Selv om dette er populære aktiviteter blant lærere, er det alt for lite forskning på området til å si om det har læringseffekt. Waite (2018) gjør funn av at det er gode muligheter for å bygge videre fra blokkbasert programmering til tekstbasert programmeringsspråk.

Øistein Gjøvik og Joakim Høyland er universitetslektorer på NTNU og har programmering som interesseområde. De har sammen skrevet boken *Kloss for kloss: blokkprogrammering for lærere*, og belyser at de ulike rollene programmering kan ha i matematikk preger synet på hvordan undervisningen kan ta form (Gjøvik & Høyland, 2022). Skal det undervises *med, gjennom* eller *i* programmering? Hvis en tolker at elevene skal lære programmering for å kunne programmere, er det hensiktsmessig med undervisning *i* programmering. Dette involverer å lære elevene om løkker, vilkår, variabler og funksjoner, samt det konkrete programmeringsspråket med dets styrker og svakheter. Med denne type undervisning vil det være nyttig å reflektere over hvordan man som lærer kan legge opp til at elevene skal få oppgaver der det naturlig blir et behov for funksjoner eller variabler (Gjøvik & Høyland, 2022). En kan også vurdere det dit at elevene skal benytte programmering som et verktøy og et program som skal hjelpe elevene med å løse oppgaver. Da er det snakk om undervisning *med* programmering. Dette kan f.eks. være et kalkulatorprogram for utregning av en størrelse. Gjøvik og Høyland (2022) sier læreren bør ha reflektert over hva eleven må kunne om de ulike matematiske ideene til programmet i forkant av denne typen undervisning. Andre ganger kan

det være ønskelig at elevene skal få utforsket i programmeringsspråk. Da foregår undervisning *gjennom* programmering, og læreren må vurdere om han selv eller elevene skal være den som lager programmet det skal undervises gjennom. Et eksempel på undervisning gjennom programmering kan være når elever skal utforske sammenhenger mellom variable størrelser og areal. Undervisning i, med eller gjennom programmering er et interessant aspekt, fordi det sier mye om hvordan ulike lærere, og andre, kan tolke og utøve den samme læreplanen på ulikt vis.

I mange situasjoner er det ønskelig å strukturere undervisningsforløp etter bestemte modeller. En modell en kan arbeide ut ifra er PRIMM-modellen. Denne modellen består av fem trinn: predict, run, investigate, modify og make. PRIMM-modellen er utviklet for undervisning i programmering slik Gjøvik og Høyland (2022) ser det fordi det utelukkende handler om å forstå algoritmen programmet er bygd opp av. Stenseth et al. (2019) har laget en annen modell for undervisningsforløp. I sin forskning omtaler de en oppgave der elevene har fått et halvferdig program som må bearbeides. Dette gjør at matematikkundervisningen skjer *gjennom* programmering. Forfatterne viser til sin modell (Figur 2) for *iterativt arbeid* og argumenterer for at elevene må inkludere aspektet med å bruke sin matematiske fagkunnskap når de løser programmeringsoppgaver for å forsvare programmeringens posisjon i matematikk. Forfatterne sier at det blir et stort ubenyttet potensial i elevsamarbeidet dersom elevene ikke beveger seg ut fra den *intuitive syklusen*, som består av testing, observasjon med analyse og endring av program, og som kun fører til en individuell, gjentakende «prøve og feil»-metode (Kaufmann et al., 2022).



Figur 2: Iterativt arbeid foregår i sykluser. Hentet fra Stenseth et al. (2019) s. 8.

Koehler og Mishra (2009) har videreutviklet Shulmans idé om pedagogisk innholdskunnskap (PCK) fordi de mener at i hjertet av god teknologisk undervisning kreves tre kjernekomponenter: innhold, pedagogikk og teknologi. De har så utviklet et rammeverk som kan analysere om undervisning får frem interaksjonen mellom alle disse tre komponentene. Rammeverket kaller de for TPACK. Denne studien har benyttet dette rammeverket i deler av analysene, og er dypere forklart i delkapittel 2.4.2. Kaufmann og Maugesten (2022) har forsket på læreres tanker om integrering av programmering i matematikkundervisningen. Blant annet undersøker forskerne de ulike elementene TPACK består av og vurderer hvilke områder lærere oppfatter programmering som mest nyttig. Resultatene viser at lærerne ikke så programmering mer nyttig inn i emnet geometri og i problemløsning enn i andre emner.

2.1.5 Lærers rolle

«Hva som «teller» som kunnskap, har endret seg, og hva som er lærers rolle, har dermed også endret seg» (Michaelsen, 2019, s. 143). Mye av grunnen til dette er at læring ikke lenger bare er noe som skal tilegnes, men deltakelse står sentralt. Lærers kompetanse er debattert og Johansen (2020) sier at, *«Mange lærere har mangelfull kompetanse innen programmering»* og legger til at lærerne har lite kjennskap til hvordan programmering og matematikk kan integreres. Den store utfordringen med programmering er som med teknologi generelt, utviklingen skjer i en fart lærere ikke klarer å følge. Koehler og Mishra (2009) sier at, *«By their very nature, newer digital technologies, which are protean, unstable, and opaque, present new challenges to teachers who are struggling to use more technology in their teaching»* (s. 61). Ny teknologi er altså så komplekst og avansert fordi det er brukbart på så mange områder, den er ustabil gjennom at det alltid er i endring og teknologien er ugjennomsiktig gjennom at den indre funksjonen er skjult for brukeren. På mange måter kan en si at lærere har for mange muligheter med noe de ikke har oversikt over, og heller ikke vet funksjonen til. Lærere har med dette utilstrekkelig kompetanse i faget og klarer ikke å henge med på utviklingen (Koehler & Mishra, 2009). På grunn av disse utfordringene er mer forskning på temaet høyest nødvendig.

Forsström og Kaufmann (2018) har gjort en litteraturgjennomgang av 15 artikler for å undersøke programmering i matematikkundervisning. Det kommer frem av gjennomgangen at lærerrollen har endret seg med programmeringens inntreden i matematikkundervisningen. Gjennomgående for artiklene som er undersøkt viser det seg at samarbeidslæring blant elevene står sentralt. Lærers rolle er derfor i større grad enn tidligere å være en veileder og støttespiller i elevsamarbeidet. I tillegg skal læreren operere som en konfliktløser når elevene stopper opp i problemløsningen. En siste rolle er å bygge opp om et klasseromsklima hvor respekt for

medelever står sentralt og hvor elevene lytter og kan endre på ansvarsoppgaver og roller innad i sine læringsgrupper (Forsström & Kaufmann, 2018). Dette er i tråd med Mason (2016) sine tanker om å skape en *conjecturing atmosphere* i et problemløsende klasserom. Mason (2016) peker på viktigheten av lærere som er bevisst sin *teachers lust*, sitt ønske om å hinte elever videre, og være gode på å vurdere når en skal la elevene «stange hodet i veggen» og når en skal avsløre hint som fører elevene videre. Stenseth et al. (2019) sier at behovet for læreren i størst grad er i planleggingen av oppgaver og for å sette elevene i gang, samt å veilede underveis i samarbeidet. I tillegg skal læreren legge opp til en planlagt prosess der elevene sammen med læreren kan kvalitetssikre hvert steg av programmeringen. At utvikling av metakognisjon er positivt for problemløsning er ikke nytt, og ble tidlig belyst gjennom George Pólyas (2014) nivåer av problemløsning tilbake i 1945 (Se forklart i delkapittel 2.2.3). Nosrati og Wæge (2015) ser på hvordan læreren kan fremme dette i en undersøkende matematikkundervisning. De sier at en viktig del av den undersøkende matematikkundervisningen er den avsluttende delen der elevene sammen med læreren diskuterer ulike løsningsmetoder, og der læreren leder diskusjonen og hjelper elevene med å se hvordan de ulike løsningene henger sammen og relasjonen til læringsmålet (Nosrati & Wæge, 2015).

Selv om lærerens rolle endres mot mer veiledning med programmering inn i matematikkfaget (Forsström & Kaufmann, 2018; Stenseth et al., 2019), sier forskningen at lærerne fremdeles må opparbeide seg gode programmeringsferdigheter (Bocconi et al., 2018; Kaufmann & Maugesten, 2022). Lærere må lære hvordan programmering henger sammen med matematikk og undervisning. De må få erfaring i hvordan programmering kan brukes på ulike matematiske emner (Kaufmann & Maugesten, 2022). I tillegg vil undervisning av programmeringsferdigheter bedres når alle lærere har dypere forståelse av programmering og når den er blitt en integrert del av den pedagogiske tilnærmingen som benyttes i klasserommet (Bocconi et al., 2018).

2.2 Programmering

I denne delen av teorikapittelet tar vi for oss programmering. Det blir gitt en grunnleggende innføring i hva programmering er, og det blir belyst hvorfor det er nyttig å kunne programmere. Til slutt blir det utdypet hvilke ferdigheter en må ha for å mestre å programmere.

2.2.1 Hva er programmering?

Nasjonal digital læringsarena sier det er vanlig å forenkle definisjonen av programmering til at «*datamaskinen instrueres til å utføre ulike oppgaver*» (Lindsø, 2020). Forsström og Kaufmann (2018) har også med det teknologiske aspektet når de sier at programmeringsevnene fører til at, «*the computer can perform specific tasks, solve problems, and support human interactions*» (Forsström og Kaufmann, 2018, s. 19). Stenseth et al. (2019) vektlegger i større grad prosessen og sier at programmering er evnen til å analysere, forstå og løse problemer gjennom en iterativ prosess. Senter for IKT i utdanningen definerer i sin rapport fra 2016 programmering på denne måten:

Programmering, slik det er brukt i dette dokumentet, omfatter mer enn å bare skrive programkode som kan kjøres på en datamaskin, det inkluderer også prosessen med å komme fram til denne koden. Det vil si prosessen fra å identifisere et problem og tenke ut mulige løsninger på problemet, til å skrive kode som kan forstås av en datamaskin, og å feilsøke og kontinuerlig forbedre denne koden (Sevik, 2016, s. 9).

Selv om flere inkluderer prosessen i definisjonen av programmering, er det de ulike programkodene som får det elektroniske apparatet til å fungere når det er aktivert. For at en datamaskin skal kunne forstå de ulike kodene, må det benyttes et *programmeringsspråk* datamaskinen forstår. Programmeringsspråk er bygd opp av noe som kan sammenlignes med andre språks grammatikk, men til forskjell fra de språkene vi kjenner vil programmet gjøre feil dersom en ikke følger syntaksen (Rossen, 2019). Ord er alltid entydige og stammer gjerne fra matematikk eller logikk.

Det finnes flere tusen programmeringsspråk, og det er vanlig å skille mellom lavnivåspråk og høynivåspråk av programmering (Rossen, 2019). Lavnivåspråk er normalt ikke mulig å lese for et menneske. Assembler er et lavnivåspråk hvor forholdsvis enkle operasjoner, som å addere to tall, behøver svært mange linjer med kommandoer for en endelig kode (Rossen, 2019). Høynivåspråk er leselig, og benyttes for å gjøre utregninger og prosessere data uten at en trenger å vite noe om maskinkoden. De komplekse og avanserte kommandoene i lavnivåspråkene er

abstrahert og en får et mye mer lettleseilig programmeringsspråk. Python er et eksempel på et høynivåspråk. Fordi Python er brukervennlig og uttrykkskraftig (Dvergsdal, 2019) kan det både benyttes som et enkelt verktøy for mindre programmer og for større nevrale nettverksmodeller som for eksempel driver ChatGPT. Det medfører også at Python er et naturlig valg av programmeringsspråk i skolen. Ved utvikling av programmer med høynivåspråk oversettes programmet til maskinkode (lavnivåspråk) gjennom en *kompilator* slik at det kan bli lest og kjørt av datamaskinen (Rossen, 2019). Andre programmer oversettes gjennom å tolkes. I tillegg til lav- og høynivåspråk er det utviklet et *fjerdegenerasjonsspråk*. Et program i et fjerdegenerasjonsspråk forteller hva som skal bli gjort, imens det overlater til datamaskinen å avgjøre hvordan (Rossen, 2019). Fjerdegenerasjonsspråk er antatt å være enklere å benytte uten spesiell programmeringskunnskap.

Tekstprogrammeringsspråk krever nøyaktighet og kan virke abstrakt, så en av hovedgrunnene til at noen programmeringsspråk prioriteres i skolen over andre kan være deres visuelle natur (Kaufmann & Maugesten, 2022). Visuelle programmeringsspråk kan være med på å hjelpe elevene med å se relevansen til deres hverdag (Grover & Pea, 2013) eller fungere som motiverende faktor (Haraldsrud et al., 2020). Blokkprogrammering er et eksempel på visuell programmering. Blokkbasert programmeringsspråk er blokker med koder som settes sammen til en algoritme der strukturen i programmet er viktigere enn nøyaktigheten av kodene. Dette fordi klosser som ikke hører sammen ikke kan settes sammen (Gjøvik & Høyland, 2022). Scratch er det aller mest utbredte blokkprogrammeringsverktøyet (Haraldsrud et al., 2020) og har som mål å introdusere programmering for nybegynnere. Utviklingen av blokkprogrammeringsspråk som Scratch har gjort det mulig å introdusere programmering allerede i grunnskole og barnehage (Kaufmann & Maugesten, 2022) og det er gode muligheter for å bygge videre fra blokkbasert programmering til tekstbasert programmeringsspråk (Waite, 2018).

Selv om vi ofte forbinder programmering med datamaskiner, apper og nettsider kan programmering like gjerne omhandle fysisk programmering (Sevik, 2016). Fysisk programmering er et annet eksempel på visuell programmering, og Lego Mindstorms er et eksempel på fysisk programmering. Lego Mindstorms går i all hovedsak ut på å bygge ulike roboter av Lego. Til forskjell fra ordinær Lego inkluderer Lego Mindstorms sensorer for lys, høyttalere, avstandsmålere, motor og batteri. Alt styres fra en app hvor du kan programmere roboten din til å gjøre akkurat det du vil med en vanskelighetsgrad som kan tilpasses. First Lego League er årlige arrangementer hvor deltakere bygger ulike roboter og konkurrerer om hvem

som har den beste roboten. Lego Mindstorms viser seg å kunne være positivt for noen elevgruppers utvikling av STEM-ferdigheter (Forsström & Kaufmann, 2018; Hussain et al., 2006). STEM kan ansees som ferdigheter i noen realfagsrettede områder, og står for science, technology, engineering og math. Utviklingen av disse ferdighetene forutsetter at elevene har en lærer som støtter de med sin dypere kunnskap innenfor temaet det undervises i, som for eksempel kobling av motorer i Lego Mindstorms.

2.2.2 Hvorfor lære programmering?

«*As technology changes, so do the skills people need to thrive in work and life*» (OECD, 2019, s. 178). Det er liten tvil om at programmeringsferdigheter er viktige og at deres betydning vil øke i fremtiden (Kaufmann & Maugesten, 2022). I nærmest alle typer jobber blir arbeid endret på grunn av mulighetene digitalisering gir verden (OECD, 2019). Nyere forskning viser også til at en må ha en grunnleggende forståelse av programmering og algoritmisk tenkning (se delkapittel 2.2.3) når kunstig intelligens er på vei til å bli en del av hverdagen vår (Tamborg et al., 2022). Med disse pågående endringene, vil automatisert arbeid ikke lenger være et arbeid mennesker behøver å gjøre. Det krever andre ferdigheter for å klare seg i arbeid og i hverdagen i årene som kommer enn det som har vært tidligere.

«*Are they really ready to work?*» av Casner-Lotto og Barrington (2006) er en dyptgående studie der fire globale aktører, The Conference Board, Corporate Voices for Working Families, Partnership for 21st Century Skills og Society for Human Resource Management, har gått sammen og grundig studert og lagt frem deres perspektiver på krav til fremtidens arbeidere. Studien bygger på blant annet *21st Century Skills* som er et rammeverk av ferdigheter en rekke organisasjoner og bedrifter ser på som helt nødvendige for å kunne klare seg i det 21. århundre. I studien listes det opp ulike eksempler på hvilke ferdigheter dette kan omhandle (Tabell 1). Det skilles mellom grunnleggende ferdigheter (basic skills) og anvendte ferdigheter (applied skills).

Basic skills	Applied skills
English Language (spoken)	Critical Thinking/Problem Solving
Government/Economics	Oral Communications
Reading Comprehension (in English)	Written Communications
Humanities/Arts	Teamwork/Collaboration

Writing in English (grammar, spelling, etc.)	Diversity
Foreign Languages	Information Technology Application
Mathematics	Leadership
History/Geography	Creativity/Innovation
Science	Lifelong Learning/Self Direction
	Professionalism/Work Ethic
	Ethics/Social Responsibility

Tabell 1: Basic skills og applied skills fra Casner-Lotto & Barrington, 2006, s. 9.

De grunnleggende ferdighetene er ferdigheter som vanligvis blir undervist i utdanningsløpet. De anvendte ferdighetene er ferdigheter som bygger på de grunnleggende ferdighetene. Det er kognitive ferdigheter og sosiale ferdigheter som anvendes i blant annet jobbsituasjon. De øverste lederne i hver av de fire organisasjonene oppfordrer sterkt til å arbeide for å integrere de anvendte ferdighetene i utdanningsløpet:

«The education and business communities must agree that applied skills integrated with core academic subjects are the “design specs” for creating an educational system that will prepare our high school and college graduates to succeed in the modern workplace and community life. These skills are in demand for all students, regardless of their future plans, and will have an enormous impact on our students’ ability to compete» (Casner-Lotto & Barrington, 2006, s.7).

Blant de anvendte ferdighetene sier Gjøvik og Høyland (2022) at programmering fremmer logisk tenkning og problemløsningskompetanse. Haraldsrud et al. (2020) forklarer at programmering gir digital dannelse og er med på å sikre fellesskapet i samfunnet. I bunn og grunn handler digital dannelse om å forstå hvordan den digitale verden fungerer. I likhet med hva Casner-Lotto og Barrington (2006) presiserer i sitatet over, sier Haraldsrud et al. (2020) at programmeringen ikke bare er for de som skal bli programmerere, men for alle, og at det er en nødvendighet med forståelse av de grunnleggende byggesteinene i det digitale samfunnet.

Mennesker bør derfor lære seg å programmere for å være med på samfunnets utvikling. Programmering omfatter å anvende grunnleggende ferdigheter for å skape noe nytt eller løse et

problem. Ferdigheter i programmering gir mulighet til å bli en attraktiv arbeidstaker som er konkurransedyktig uavhengig av fremtidsplaner og ambisjoner.

2.2.3 Hvilke ferdigheter kreves for å mestre programmering?

Flere europeiske land har innlemmet programmering i sin nasjonale læreplan for å utvikle elevenes «computational thinking» (Balanskat & Engelhardt, 2015; Grover & Pea, 2013). En annen grunn til at europeiske land inkluderer programmering i matematikk, er for å styrke elevenes problemløsningsferdigheter (Balanskat & Engelhardt, 2015). I dette delkapittelet tar vi for oss hva som inngår i hver av disse anvendte ferdighetene, og beskriver noen likheter ved dem.

Computational Thinking

Wing (2006) definerer Computational thinking (CT) gjennom ulike karakteristikk. Dette innebærer blant annet å tenke som en datavitenskapsmann med et menneskelig problemløsningsperspektiv og er en kombinasjon av matematisk og teknisk tenking. CT er så sentralt og viktig at Wing (2006) oppfordrer til å anse det som en grunnleggende ferdighet, og dermed likestilt med lesing, skriving og regning. Wing (2006) skriver at dersom en benytter de ferdighetene mennesker er best på sammen med de ferdighetene maskiner er best på, så kan vi finne løsninger på problemer og designe system vi aldri før har kunnet gjøre på egenhånd. Med tiden har det utviklet seg ulike definisjoner av CT. Grover og Pea (2013) sier at en allment akseptert definisjon av CT er at det omhandler mestring av noen definerte ferdigheter:

- Abstraksjon og mønstergeneralisering (Inkludert modellering og simulering)
- Systematisk prosessering av informasjon
- Symbolsystemer og representasjoner
- Algoritmiske forestillinger
- Strukturert dekomponering av problemer (modularisering)
- Iterativ, rekursiv og parallell tenking.
- Betinget logikk
- Effektivitet- og ytelsesbegrensinger
- Feilsøking og systematisk feildeteksjon.

Bocconi et al. (2018) forteller at det er en bred forståelse i de nordiske landene at CT er noe mer enn programmering. Blant annet inngår særlig tre av de anvendte ferdighetene fra 21st. Century Skills: *problemløsning*, *logisk tenking* og *kreativitet* (Bocconi et al., 2018). I en

litteraturgjennomgang av forskere ved UiO fra 2019 er det laget en oversikt over ulike syn på CT (Dolonen et al., 2019):

- CT som forståelse av verden rundt oss gjennom at CT er hovedmodell for kommunikasjon, samarbeid og analyse av problemstillinger.
- CT som programmeringskompetanse med vekt på syntaks og semantikk.
- CT som algoritmisk tenkning og problemløsning gjennom dekomponering.
- CT som bred digital ferdighet med kompetanse om verktøy og tilpasning av verktøy for å løse et problem.

CT kan derfor anses som nyttig både i og utenfor programmering. I den norske læreplanen er Computational thinking oversatt til *algoritmisk tenkning* (Utdanningsdirektoratet, 2019b). Utdanningsdirektoratet (2019b) forklarer at algoritmisk tenkning går ut på å løse problemer på en strukturert og fornuftig måte. I begrepet ligger det også en teknologisk forståelse. Gjennom å bruke sin teknologiske kompetanse skal en la en datamaskin løse alt eller deler av et problem. Dette blir utdypet i delkapittel 2.2.3. Med mange definisjoner for det samme begrepet kan det ofte bli mer forvirring enn nødvendig. Bergqvist (2022) sier at det faktisk kan føre til en misoppfatning om at CT bare kan fremmes gjennom koding på en datamaskin. Det er forskning som viser til at CT også kan fremmes på andre måter (Berg, 2021; Knuth, 1974) som for eksempel gjennom analog programmering, altså programmering uten datamaskin.

Problemløsning

I OECD-rapporten fra 2019 påpekes det at lærere med gode problemløsningsferdigheter i teknologirike miljøer har en positiv påvirkning på elevers prestasjoner i problemløsning og matematikk med datamaskin (OECD, 2019). Problemløsning er en sentral del av blant annet programmeringens feilsøking som skjer i den iterative prosessen (Se figur 2) (Stenseth et al., 2019). Jean Piaget (1896-1980) innfører begrepet *skjema* i hans forskning mot å finne frem til kunnskapens struktur (Imsen, 2014). Et skjema er individets forståelse av et fenomen som kommer som et resultat av tolkning og sansing av fenomenet. Dette igjen, fører oss videre til to andre begreper Piaget innfører: *assimilasjon* og *akkomodasjon*. Assimilasjon handler om at nye inntrykk tilpasses gamle skjemaer individet allerede har. Når vi ikke kan forklare inntrykk med skjemaene vi allerede har, behøver vi å reorganisere skjemaene som ikke er tilstrekkelig. Akkomodasjon er å forandre på de kognitive strukturene vi har dannet oss slik at de blir sanne. I noen situasjoner holder det å reorganisere, utdype eller utvide et gammelt skjema, imens andre ganger må vi lage helt nye skjemaer. Begge deler er akkomodering. Å omorganisere og forandre for å skape forståelse er å lære (Imsen, 2014). På denne måten ligner hjernen en datamaskin.

Programmering går i stor grad ut på å lage algoritmer. Noen ganger fungerer ikke programmet som vi trodde det skulle gjøre. Da har programmereren en viktig oppgave i å finne ut hvorfor programmet ikke fungerer. Det vil i disse situasjonene være nødvendig med en reorganisering, utdyping eller utviding. Når et program endres blir det gjort en feilretting, og problemet kan bli løst.

George Pólya (1887-1985) ønsket å produsere en generell strategi for å løse problemer, og med sin systematiske fremgangsmetode i møte med kognitivt utfordrende problem anses han av mange som den mest innflytelsesrike matematikeren på 1900-tallet (Taylor & Taylor, 1993). Metoden består av fire enkle trinn: forstå problemet, legg en plan, gjennomfør planen og se tilbake (Pólya, 2014). For å forstå problemet bør en stille seg gode spørsmål. For eksempel «Hva blir jeg spurt om?» eller «Har jeg noe jeg kan assosiere problemet til?». Å utarbeide en plan omhandler å velge en strategi i møte med problemet. Skal en bare bruke «prøve-og-feile»-metoden eller kan en kanskje se noen mønstre? Kanskje kan det være nyttig med en hjelpefigur? Når en iverksetter planen kreves det at en har de nødvendige ferdighetene, samt å vurdere underveis om strategien fungerer. Til slutt ser en tilbake på det en har gjort. En metakognisjon der en evaluerer arbeidet og vurderer hva som har fungert godt og hva som kunne fungert bedre. «Hva om en endrer på den?» eller «Hva om jeg heller prøver sånn?».

Bruder (Liljedahl et al., 2016) forteller at for å være en god problemløser er det nødvendig med *mental agility*. Dette innebærer de fem egenskapene *reduksjon*, *reversibilitet*, å kunne *se problemet fra flere sider*, å kunne *endre synsvinkel* og *overføring av gamle strategier*. Reduksjon er det samme som å abstrahere. Du fjerner unødvendig informasjon og korter ned algoritmer. Hjelpefigurer, tabeller og andre visuelle og informative hjelpemidler er ofte et verktøy som benyttes for å redusere problemet. I matematikken benyttes redusering ofte i for eksempel algebra når elevene skal «forenkle uttrykkene». I tekstoppgaver handler det om å luke bort informasjon du ikke behøver for å løse problemet. Reversibilitet betyr at når en har løst et problem så klarer en å trinnvis bevege seg tilbake til problemets start igjen med gyldige operasjoner. Stenseth et al. (2019) omtaler som sagt dette som en del av lærerens rolle i programmeringen, nemlig en planlagt prosess hvor en kvalitetssikrer stegene i lag med elevene. Bruder (Liljedahl et al., 2016) sammenligner det med en person som leter etter en nøkkel de har lagt bort. En strategi for å finne de kan være ved å trinnvis gå gjennom hvor en har vert. Å se problemet fra flere sider omhandler å ta hensyn til flere aspekter ved et gitt problem på en og samme tid. En forholder seg til gjeldende prinsipper, og selv om en idé kan møte på motstand har du den alltid i minne for et mulig gjennombrudd i problemløsingen. Eksempler på dette kan

være prinsipper om symmetri eller formlikhet for beregning av areal. Å kunne endre synsvinkel omhandler å ta et skritt tilbake og forsøke å se etter nye mulige retninger å gå, vurdere nye fremgangsmåter og perspektiver. Dette for å unngå å bli sittende helt fast i et spor som ikke leder noen vei. Å overføre gamle strategier omhandler å benytte tidligere erfaringer og gjenkjenne de i nye kontekster. Det kan for eksempel være å se mønster en kjenner fra tidligere som «trekantall» eller Fibonaccitalle.

Det er mange likheter mellom CT-ferdigheter og Pólyas dimensjoner for problemløsning (Denning, 2017), samt Bruders definisjon av mental agility. Selv om problemløsning inneholder ferdigheter man behøver for å kunne programmere, advarer Kaufmann og Stenseth (2021) mot å anta at dette er ferdigheter man tilegner seg naturlig og automatisk ved å drive med programmering. Fordi tiden lærerne har med elevene i klasserommet er så ekstremt verdifull, blir det derfor viktig å vurdere hvor mye tid en skal bruke på å tilegne elevene programmeringsferdigheter nok for deres behov (Bergqvist, 2022). Med en bredere tolkning av programmering kan problemløsning være en enklere vei for å implementere arbeid med programmeringsferdigheter i matematikkundervisningen (Bergqvist, 2022).

2.3 Programmering i Kunnskapsløftet 2020

11. oktober 2016 fatter Stortinget gjennom Stortingsmelding nr. 28 at det skal vedtas nye læreplaner gjennom en fagfornyelse (Meld. St. 28 (2015–2016)). Fagfornyelsen er prosessen med å fornye læreplanene. Ulike utvalg har argumentert for og redegjort hva som bør være innholdet. Resultatet er Kunnskapsløftet 2020 (LK20). LK20 er den siste innførte læreplanen, og med de største endringene av innhold siden læreplanen for Kunnskapsløftet 2006, LK06. Den nye læreplanen tredde i kraft høsten 2020 og er gradvis blitt innført for de ulike trinnene. Gjøvik og Høyland (2022) sier at hensikten med revisjonen av læreplanen kan oppsummeres ved relevans, dybdelæring og sammenheng.

2.3.1 Relevans, dybdelæring og sammenheng

Relevansen for implementeringen kan i stor grad ses i sammenheng med mye av innholdet i delkapittel 2.2.2. Verden er i forandring og skolen må henge med. 21st Century Skills (forklart i delkapittel 2.2.2, se tabell 1) viser til hvilke ferdigheter det bør arbeides mot å inkludere i opplæringen. Blant de største endringene i fagfornyelsen presenteres dybdelæring. Dybdelæring vil si å gradvis utvikle kunnskap og varig forståelse av både innhold og læringsmetode (Utdanningsdirektoratet, 2019a). Dette gjelder i enkelte fag, men også på tvers av fag. Programmering kan være med på å styrke nettopp dybdelæring ved at elevene får utforske kjente og ukjente konsepter på flere måter, også på tvers av fag (Haraldsrud et al., 2020). Implementeringen av programmering i matematikken og dybdelæring i samme læreplan har hatt sine kritikere ifølge Gjøvik og Høyland (2022). Dette fordi mange mener det er vanskelig å koble programmering med matematikk, og at programmeringen bare blir enda et tema å undervise i. Dette leder oss over til sammenheng der programmering har et stort potensial. Da LK20 tredde i kraft ble det klart at den inkluderte tre tverrfaglige temaer. Disse er folkehelse og livsmestring, demokrati og medborgerskap og bærekraftig utvikling. I tillegg ble programmering innført i fire fag, gjennom matematikk, naturfag, musikk og kunst og håndverk. I motsetning til hva som har vært praktisert tidligere gir programmering mulighet for at et tverrfaglig samarbeid oftere kan inkludere matematikkfaget (Gjøvik & Høyland, 2022). Tabell 2 viser eksempler på kompetansemål som omhandler programmering i de ulike fagene for 7. og 10. klassinger. Programmering kan derfor gi mulighet for å drive tverrfaglig arbeid, spesielt på tvers av disse fire fagene.

	Matematikk	Naturfag	Musikk	Kunst og håndverk
7. trinn	Bruke programmering til å utforske data i tabeller og datasett	Utforske, lage og programmere teknologiske systemer som består av deler som virker sammen	Bruke teknologi og digitale verktøy til å skape, øve inn og bearbeide musikk	Bruke programmering til å skape interaktivitet og visuelle uttrykk
10. trinn	Utforske matematiske egenskaper og sammenhenger ved å bruke programmering	Bruke programmering til å utforske naturfaglige fenomener	Skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder	Utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter

Tabell 2: Eksempler på kompetansemål som omhandler programmering (Kunnskapsdepartementet, 2019).

Sammenheng er også viktig innad i matematikkfaget. Da omhandler det å koble kunnskap fra ulike temaer i arbeid med programmering. Med den formelle læreplanen blir det innført kompetansemål som kan knyttes til programmering på alle trinn fra 2. trinn og til 10. trinn (Se tabell 3) og matematikkfaget er det faget som har fått flest nye kompetansemål som spesifikt nevner programmering.

Trinn Kompetansemål

2. trinn	Lage og følge regler og trinnvise instruksjoner i lek og spill
3. trinn	Lage og følge regler og trinnvise instruksjoner i lek og spill knyttet til koordinatsystemet
4. trinn	Utforske og beskrive strukturer og mønstre i lek og spill Lage algoritmer og uttrykke dem ved bruk av variabler, vilkår og løkker
5. trinn	Lage og programmere algoritmer med bruk av variabler, vilkår og løkker
6. trinn	Bruke variabler, løkker, vilkår og funksjoner i programmering til å utforske geometriske figurer og mønstre
7. trinn	Bruke programmering til å utforske data i tabeller og datasett
8. trinn	Beskrive og generalisere mønstre med egne ord og algebraisk Utforske hvordan algoritmer kan skapes, testes og forbedres ved hjelp av programmering

9. trinn	Simulere utfall i tilfeldige forsøk og beregne sannsynligheten for at noe skal inntreffe, ved å bruke programmering
10. trinn	Utforske matematiske egenskaper og sammenhenger ved å bruke programmering

Tabell 3: Oversikt over kompetansemål som omhandler programmering på grunnskolen (Kunnskapsdepartementet, 2019).

Når lærere, skoleledere, forskere, foreldre eller andre leser det formelle læreplandokumentet, tolkes, analyseres og forstås dette på ulikt vis. John I. Goodlad omtaler dette som den oppfattede læreplanen (Engelsen, 2015). Tolkningen fra lærerne er utgangspunktet for undervisningen det legges opp til. En kan legge opp til undervisningsøkter som tar for seg et og et kompetansemål, eller en kan arbeide overordnet med fokus på læring og dermed på tvers av kompetansemålene. Som nevnt i delkapittel 2.1.4 skriver Gjøvik og Høyland (2022) om undervisning *i, med* og *gjennom* programmering. Undervisningsøkter med mål om å se sammenheng mellom matematiske emner skjer i størst grad når en arbeider med oppgaver på tvers av ulike matematikkemner. Dette kan ta form gjennom at en benytter programmering for å utforske et annet matematisk emne. Et eksempel er hvis elevene skal undervises i emnet sannsynlighet og statistikk. Da foreslår Gjøvik og Høyland (2022) at en kan programmere Monty-Hall-eksperimentet i Scratch. Eksperimentet går ut på at det er en premie bak en av tre dører. Deltakeren velger så en dør den tror premien er bak. Deretter åpnes en annen dør som ikke inneholder noen premie, og deltakeren får tilbudet om den vil bytte dør eller stå ved den han plasserte seg ved fra start av. På dette viset får elevene arbeide med betinget sannsynlighet gjennom programmering og kommer kanskje frem til at det er å foretrekke å bytte dør. Dersom en slår sammen kompetansemål og arbeider overordnet med temaer er det mulighet for å undervise *gjennom* eller *med* programmering slik eksempelet over viste.

En følge av at det kan være utfordrende å koble programmering til matematikk er at programmering ofte blir undervist separat og som et eget emne. Dette kommer av at undervisningsøkter med et og et kompetansemål for øyet gjør det naturlig å ikke legge vekt på sammenheng mellom emner. Undervisning om programmering blir dermed gjennomført som undervisning *i* programmering der målet er å lære seg programmeringsferdigheter.

2.3.2 Algoritmisk tenkning i Kunnskapsløftet 2020

For å definere det omfattende begrepet algoritmisk tenkning viser Utdanningsdirektoratet (2019b) til «Den algoritmiske tenkeren» (Figur 3) som er et ideal å arbeide mot å være lik. Den består av ulike nøkkelbegreper som hovedsakelig inngår i de anvendte ferdighetene i tabell 1 (I delkapittel 2.2.2) og arbeidsmåter. Av den kommer det frem at elevene skal kunne bruke *logikk*

gjennom å analysere og forutse, *forstå algoritmer* steg-for-steg og gjennom regler, *dekomponere* problemene til mindre deler, *finne mønstre* ved å finne og bruke likheter, *abstrahere* og fjerne unødvendige detaljer og *evaluere* programmet gjennom vurderinger. Alle disse nøkkelbegrepene skal forekomme gjennom ulike arbeidsmetoder. Av kjerneelementene kan vi lese at, «*Elevene skal legge mer vekt på strategiene og framgangsmåtene enn på løsningene*» (Kunnskapsdepartementet, 2019). Dette skal skje gjennom at elevene får fikle, skape, feilsøke, holde ut og samarbeide (Utdanningsdirektoratet, 2019b).



Figur 3: Den algoritmiske tenkeren. Figuren er tilpasset fra Barefoot Computing (UK) og hentet fra Udir (2019).

Utdanningsdirektoratet henviser til resultatene fra en rapport av 30 studier om hvordan bruk av Scratch kan lære elevene algoritmisk tenkning. I rapporten kommer det frem at elevene får synliggjort flere av sine algoritmiske evner gjennom bruk av Scratch og lærere kan evaluere hvordan elever løser problemer ved hjelp av algoritmisk tenkning (Fagerlund et al., 2021). Det stilles spørsmål ved om Scratch alene er et godt nok verktøy for å utvikle den algoritmiske tenkningen.

2.3.3 Eksamensoppgaver med programmering

Fordi koronapandemien inntraff samtidig som implementeringen av ny læreplan er det ikke vært gjennomført eksamener på grunnskolen med LK20 som læreplan. Likevel er det åpent tilgjengelig eksempler på eksamener og veiledning til disse, slik at elever og lærere skal kunne bli kjent med hvordan eksamen legges opp etter ny læreplan. Figur 4 er hentet fra prøveeksamen 2023. Løsningsforslaget til oppgaven er i figur 5.

Oppgave 8 er et eksempel på en oppgave hvor eleven skal kunne utforske matematiske egenskaper og sammenhenger ved å resonnerer, argumentere og kommunisere egne løsninger, vise forståelse for andres resonnerement og løsninger, og vurdere modeller, tekster og løsninger kritisk.

Kompetansemål fra 10.trinn: utforske matematiske egenskaper og sammenhenger ved å bruke programmering

Oppgave 8

Bildet viser et dataprogram.

$$a = 4$$

$$b = 5$$

Gjenta b ganger

Tegn et linjestykke med lengde a

Snu $(360 : b)$ grader til høyre

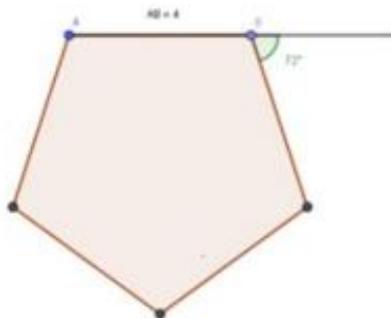
- Forklar hva som skjer når programmet blir kjørt.
- Tegn figuren og sett riktige mål på figuren din.

Figur 4: Prøveeksamensoppgave (10. trinn) 2023 som omfatter programmering. Hentet fra udir.no

Oppgave 8

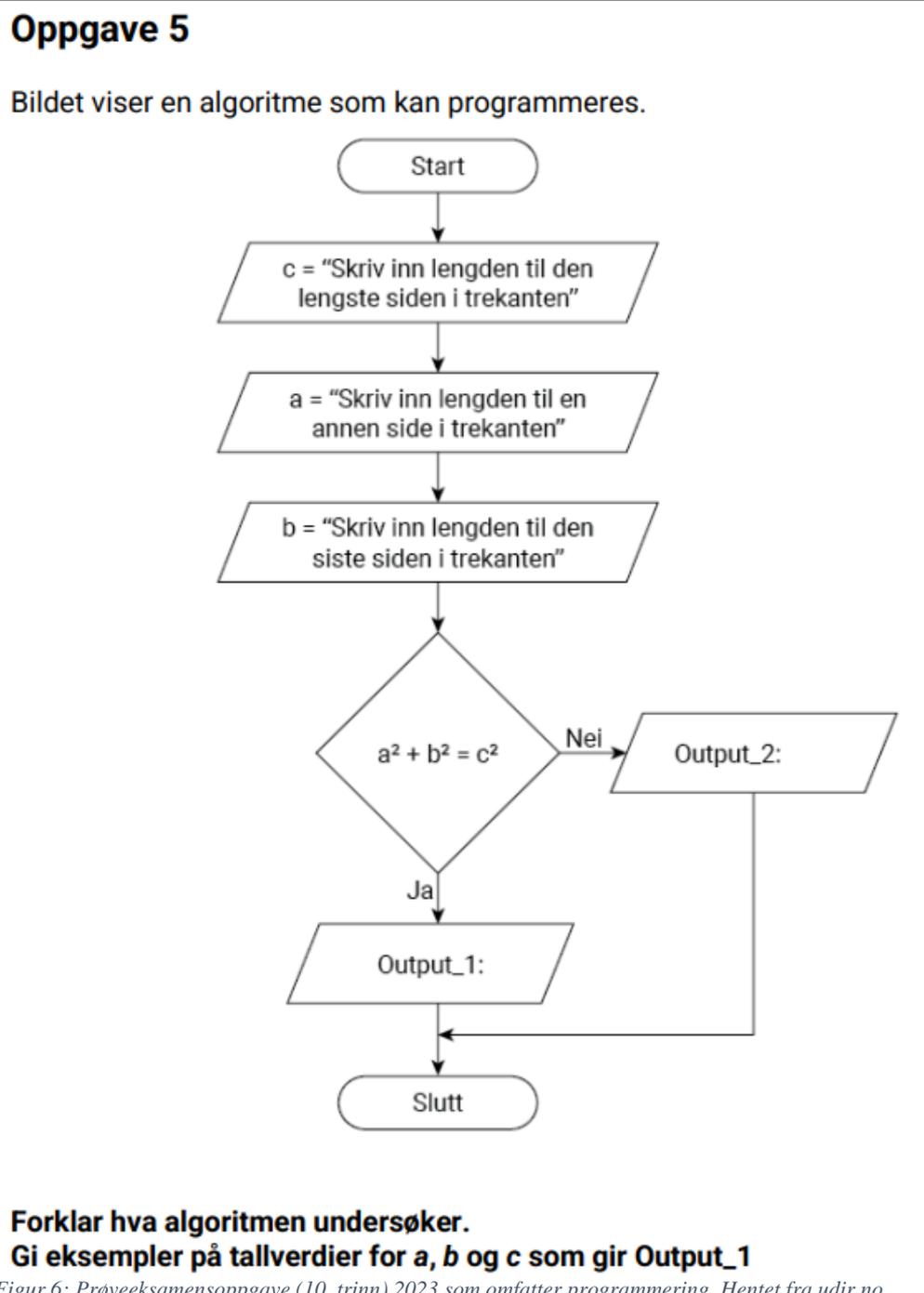
- Programmet setter først variablene a til 4 og b til 5.
Deretter kjører programmet en løkke som gjentar noe fem ganger.
Programmet tegner en linje på fire og deretter gjennomføres en rotasjon på 72-grader mot høyre.
Dette gjentar programmet fem ganger.
Programmet har nå tegnet en femkant.

b)



Figur 5: Eksempel på god elevbesvarelse på prøveeksamensoppgave. Hentet fra udir.no

Følgende prøveeksamensoppgave var relatert til programmering i 2022:



En løsning på oppgaven kan være:

Programmet ber en skrive inn de tre sidene i en vilkårlig trekant med den lengste siden først. Programmet sjekker deretter om Pytagoras' setning gjelder. Hvis det gjelder, er trekanten rettvinklet. Dersom man taster inn 5, 4, 3 kan Output_1 være, «Trekanten er rettvinklet». Dersom man taster inn 4,2,1 Kan Output_2 være, «Trekanten er ikke rettvinklet».

2.4 Teoretiske rammeverk

I denne siste delen av teorikapittelet redegjøres det for de ulike rammeverkene som blir benyttet ved denne studien. Det første rammeverket som beskrives er matematisk kvalitet i undervisning (MKU) før vi siden tar for oss TPACK-rammeverket.

2.4.1 MKU

Frode Opsvik og Leif Bjørn Skorpen er førstelektorer i matematikk ved Høgskulen i Volda og har vært aktive i forskningsprosjektet «Kvalitet i opplæringa, KIO». Prosjektet har som mål å undersøke hvordan kvalitet i undervisningen blir forstått, praktisert og opplevd. Opsvik og Skorpen (2014) omdanner gjennom forskningsprosjektet Mathematical Quality of Instruction, MQI, til et kvalitativt rammeverk de kaller for matematisk kvalitet i undervisningen, MKU. Det opprinnelige MQI-rammeverket er et kvantitativt rammeverk som måler kvaliteten i undervisningen (Opsvik & Skorpen, 2014). MQI-instrumentet videreutvikles og forbedres stadig, og forskning viser at det er en sammenheng mellom læreres matematiske kunnskap, MKT (se delkapittel 2.1.2), og den matematiske kvaliteten i undervisningen (Hill et al., 2008). I MQI-analysene blir lærerne tildelt en score i form av et tall etter om de tilfredsstillt hver indikator i liten, middels eller høy grad. Opsvik og Skorpen (2014) sitt argument for transformeringen av MQI-rammeverket er blant annet at det kan virke som at læreren får en karakter på arbeidet sitt, noe som kan oppleves ubehagelig. Konklusjonen blir at det er uaktuelt med et kvantitativt rammeverk i deres forskning. Likevel er Opsvik og Skorpen (2014) tydelige på at de mener indikatorene kan være tjenlige også til å undersøke matematikkundervisninger kvalitativt.

Som en del av omdannelsen av det kvantitative MQI-rammeverket til et kvalitativt MKU-rammeverk oversetter Opsvik og Skorpen (2014) hver dimensjon og alle indikatorene de er bygd opp av til norsk. Disse kommer frem i tabell 4 på neste side. Omdannelsen blir til MKU-rammeverket som består av fire dimensjoner. Disse er *matematisk rikdom*, *matematikkfaglig opptak og tilbakemelding*, *feil og unøyaktighet* og *elevdeltakelse i matematiske resonnement og meningsskaping* (Opsvik & Skorpen, 2014). Opsvik og Skorpen (2014) forklarer hvert av de fire overordnede dimensjonene. Matematisk rikdom ser på dybden av det matematiske innholdet og har et fokus på meningen bak fakta og prosedyrer. Matematikkfaglig opptak og tilbakemelding omhandler i hvilken grad læreren forstår og responderer på elevenes matematiske forklaringer. Feil og unøyaktighet omhandler lærerens upresise språk, upresise notasjoner, feil elevsvar som ikke blir korrigert eller lærerens manglende presisjon i undervisning av faginnholdet. Elevdeltakelse i matematiske resonnement og meningsskaping

ser på tegn som indikerer at elevene er involvert i kognitivt aktiverende arbeid i klasserommet. Denne studien tar sikte på å analysere gjennom en tilpasset versjon av MKU-rammeverket. Tilpasningen i denne studien gjøres fordi MKU-rammeverket er laget for å hente inn datamateriale gjennom observasjon, imens denne studien benytter andre datainnsamlingsmetoder. Opsvik og Skorpen (2014) presiserer at det er viktig å ha med seg at, «*MKU-apparatet er konstruert for å løfte fram den matematiske kvaliteten ved undervisninga, og ikkje er meint å fange den universelle didaktiske eller pedagogiske kvaliteten ved undervisninga*» (s. 114). Det er altså en enda mer kompleks oppgave læreren står overfor ved undervisningen enn bare det som fanges opp av MKU. Et eksempel er dimensjonen, «god klasseromsledelse» som PISA (Olsen, 2013) og TIMSS (Bergem et al., 2016) undersøker. Likevel kan MKU si noe om kvaliteten av matematikkundervisningen som er nyttig når en skal vurdere om programmering fremmer god matematikkundervisning.

1. Dimensjonen <i>matematisk rikdom</i> er sett saman av følgjande indikatorar:		
1.1	<i>Kopling og samanbinding</i>	Kople og binde saman ulike omgrep, idear, prosedyrar og representasjonsformer.
1.2	<i>Forklaringar</i>	Gi matematisk meining til omgrep, idear, prosedyrar eller løysingsmåtar.
1.3	<i>Fleire løysingsmåtar og prosedyrar</i>	Bruke fleire løysings- og tenkemåtar, strategiar eller prosedyrar for eitt og same problem.
1.4	<i>Generalisering</i>	Utvikle/utleie generaliseringar av matematiske samanhengar eller prosedyrar på grunnlag av konkrete døme.
1.5	<i>Matematisk språk</i>	Bruke matematisk terminologi og notasjon, og matematiske omgrep på ein presis, naturleg og konsistent måte.
2. Dimensjonen <i>matematikkfagleg opptak og tilbakemelding</i> er sett saman av indikatorane:		
2.1	<i>Oppklaring av vanskar, feil og misoppfatningar</i>	Det vert arbeidd med det matematiske innhaldet i elevane sine vanskar, feil og misoppfatningar.
2.2	<i>Opptak og tilbakemelding</i>	Læraren forstår, byggjer vidare på eller gir tilbakemelding på elevane sine matematiske forklaringar, spørsmål og resonnement.
3. Dimensjonen <i>feil og unøyaktigheit</i> er sett saman av indikatorane:		
3.1	<i>Større matematiske feil eller manglar</i>	Løse ei oppgåve feil, bruke omgrep på feil måte, gløyme eller utelate viktige matematiske sider ved det ein fokuserer på.
3.2	<i>Unøyaktig språkbruk</i>	Uppresis bruk av matematisk språk i munnleg eller skriftleg form, eller uppresis bruk av kvardagsspråk når ein forklarar matematiske idear.
3.3	<i>Uklart innhald</i>	Det matematiske innhaldet blir uklart presentert.
4. Dimensjonen <i>elevdeltaking i matematiske resonnement og meiningskaping</i> er retta mot elevdeltaking i aktivitetar av typen:		
4.1	<i>Elevforklaringar</i>	Elevar forklarar idear, prosedyrar eller løysingsmåtar med fokus på forståing av det matematiske innhaldet.
4.2	<i>Elevspørsmål og påstandar</i>	Elevar stiller spørsmål, kjem med påstandar eller motpåstandar knytt til det matematiske innhaldet.
4.3	<i>Elevane sitt kognitive engasjement</i>	Elevar deltek i resonnering og gjennomfører kognitivt utfordrande arbeid som til dømes å danne samanhengar mellom ulike representasjonsformer, omgrep eller framgangsmåtar.

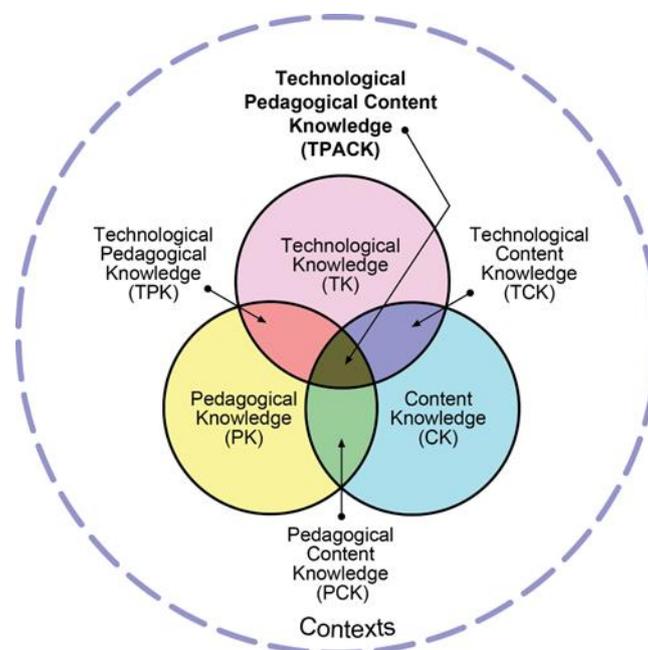
Tabell 4: MKU-rammeverket med dimensjoner og indikatorer. Hentet fra Opsvik og Skorpen (2014).

MKU-rammeverket er et nyttig fokuseringsredskap på egen undervisning (Opsvik & Skorpen, 2014). Dette fordi en blir bevisst hvilke av indikatorene som er aktivisert på en positiv eller

negativ måte, samt om de ikke er til stede i det heletatt. Dette presiserer forfatterne også, «Poenget er å vise den urøynde observatøren nokre sentrale moment som han/ho kan vere merksam på, og som dette omgrepsapparatet gjev oss eit språk til å reflektere rundt» (Opsvik & Skorpen, 2014, s. 112).

2.4.2 TPACK

Koehler og Mishra (2009) har generert en modell, TPACK (Figur 7), som beskriver interaksjonen mellom læreres pedagogiske kunnskap, innholdskunnskap og teknologiske kunnskap. Modellen er en videreutvikling av Shulmans (1986) modell av pedagogical content knowledge. TPACK er et nyttig verktøy i forskning som ser på læreres teknologiske kunnskap og profesjonsutvikling med en teknologisk kontekst (Maugesten et al., 2022).



Figur 7: TPACK-modellen av Matthew Koehler og Punya Mishra, 2009, gjengitt med tillatelse fra utgiveren © 2012 fra tpack.org

Modellen tar for seg relasjonen mellom læreres pedagogiske kunnskap (PK), innholdskunnskap (CK) og teknologiske kunnskap (TK) for undervisning. PK er lærerens grunnleggende kunnskap om prosessen og aktiviteter eller metoder av undervisning og læring (Koehler & Mishra, 2009). CK omfatter læreres kunnskap om konsepter, teorier, ideer, organisatoriske rammeverk, kunnskap om bevisføring og bevis, samt etablerte praksiser og tilnærminger for å utvikle slik kunnskap (Koehler & Mishra, 2009). TK er definert som kunnskap om ny teknologi og hensiktsmessig bruk av den i arbeid eller i dagliglivet (Koehler & Mishra, 2009). Kunnskap om programmeringsspråk inngår i TK (Maugesten et al., 2022). Mellom disse komponentene

befinner det seg kombinasjonskomponenter av de tre gjeldende typer kunnskap. Kombinasjonen av PK og CK er pedagogisk innholdskunnskap (PCK) og er omtalt av Shulman (1986). PCK omhandler pedagogisk kunnskap som er anvendelig for spesifikt innhold. En lærer som kjenner til ulike representasjoner av et tema, kjenner til misoppfatninger av emnet og kobler læreplankunnskap til undervisningen har god PCK. Kombinasjonen av PK og TK er teknologisk undervisningskunnskap (TPK). TPK er kunnskapen om fordeler og ulemper ved representasjon gjennom teknologiske verktøy og kunnskap om hvordan teknologiske verktøy kan benyttes for undervisning (Koehler & Mishra, 2009). TPK er spesielt viktig fordi mange programmer ikke er tiltenkt direkte som et undervisningsverktøy, så kreativitet og et åpent sinn for nytenking styrker TPK. TPK kan derfor forstås som hvordan lærerne bruker sin pedagogiske kunnskap til å undervise i programmering (Maugesten et al. 2022). Kombinasjonen av TK og CK er teknologisk innholdskunnskap (TCK). TCK er forståelse av måten teknologi og innhold påvirker og begrenser hverandre (Koehler & Mishra, 2009) og kan forstås som lærernes refleksjoner over hvordan programmering er relatert til matematisk innhold (Maugesten et al., 2022). Lærere må ha en dyp forståelse av hvordan emnet kan endres ved bruk av teknologier, hvilke teknologier som er best egnet for læring i et gitt fag og hvordan faginnholdet kan endre teknologien (Koehler & Mishra, 2009). Koehler og Mishra (2009) oppsummerer hvordan kombinasjonene av alle disse kunnskapene sammen legger til rette for effektiv undervisning:

«TPACK is the basis of effective teaching with technology, requiring an understanding of the representation of concepts using technologies; pedagogical techniques that use technologies in constructive ways to teach content; knowledge of what makes concepts difficult or easy to learn and how technology can help redress some of the problems that students face; knowledge of students' prior knowledge and theories of epistemology; and knowledge of how technologies can be used to build on existing knowledge to develop new epistemologies or strengthen old ones.» (Koehler & Mishra, 2009, s. 66)

Archambault og Barnett (2010) har gjort en grundig undersøkelse av TPACK-rammeverket, og konkluderer med at det kan være komplisert å bruke fordi det er uklare skiller mellom komponentene. De sier, *«From the practitioner data contained within this research, it seems that from the onset, measuring each of these domains is complicated and convoluted, potentially due to the notion that they are not separate»* (Archambault & Barnett, 2010, s. 1661). Det vil være nyttig å definere enda tydeligere hva som inngår i de ulike komponentene av TPACK-rammeverket.

3 METODE

Hvordan går vi frem for å hente inn informasjonen vi behøver for å besvare problemstillingen vår? Og er informasjonen vi innhenter gjort på en slik måte at den gir et fullgodt svar på problemstillingen? Metodekapittelet skal besvare disse spørsmålene og gi en grundig innføring i hvordan studien er gjennomført, hvilke overveielser som er gjort og gi informasjon som er relevant å informere om for denne studien. Forskning blir delt inn i kvantitativ og kvalitativ forskning. I kvalitativ forskning går en i dybden på et tema, og en arbeider med relativt få informanter (Tjora, 2017). Det er denne formen for forskning jeg vil gjøre i dette prosjektet. I etterkant av datainnsamlingen, gjennom å studere resultatene, er en bedre rustet til å si om metoden har vært en god måte å undersøke problemstillingen på. Selvkritikk og åpenhet rundt resultatet og metodebruk er med på å gi forskningen utvikling. Uavhengig av metodevalg har Hellevik (1999, s.14-16) beskrevet seks normer å betrakte som idealer for all forskning. Idealene er:

- Overenstemmelse med virkeligheten som høyeste sannhetskriterium
- Systematisk utvelging av data
- Mest mulig nøyaktig bruk av data
- Aktive anstrengelser for å avkrefte egne forhåndsoppfatninger
- Presentasjon av resultatene som tillater kontroll, etterprøving og kritikk
- En forsøker å gjøre forskningsvirksomheten kumulativ

Disse normene er holdepunkter i overveielser ved etiske dilemmaer og arbeid med reliabilitet og validitet av denne studien. Normene kan identifiseres i beskrivelsen av forskningen i dette og forrige kapittel.

Forskerens førforståelse

Gadamer er kjent for blant annet sitatet sitt om at «*forståelse er avhengig av visse fordommer*» (Kvale & Brinkmann, 2019, s. 74). Hellevik (1999) forteller at psykologer har påvist at mennesker stadig overser eller fordreier informasjon på grunn av deres forhåndsoppfatninger. Det blir derfor nødvendig for meg å tydeliggjøre noen tanker jeg har hatt før forskningen begynte, for at jeg selv og lesere av forskningen kan inkludere dette når en kritisk vurderer forskningsresultatene validitet.

All forskning kan ikke føre til revolusjonære funn, men all forskning kan bidra med noe til forskningsområdet. Det er en del av å gjøre forskningsvirksomheten kumulativ. Siden jeg er mindre erfaren med forskningsarbeid kan jeg sikte på å støtte opp om teori med praktiske

eksempler, i tillegg til at jeg kanskje kan peke på interessante funn for videre forskning. Jeg har stor tro på at dette vil bli et lærerikt prosjekt, både for meg og involverte rundt prosjektet. Min førforståelse er at programmering er krevende å undervise i matematikk, fordi mange lærere har lite erfaring og begrenset opplæring gjennom yrket. Jeg har en forventning om at prosjektet skal gi meg møter med lærere som har en dypere innsikt på dette temaet enn meg, og at jeg med prosjektet mitt kan få mulighet til å få et større innblikk i hvordan lærere arbeider profesjonelt og kreativt for å undervise programmering i matematikkundervisningen sin. Jeg har en forventning om at prosjektet kan inspirere lærere og meg selv til å mestre undervisning av programmering.

3.1 Oversikt over forskningsdesign

Tabell 5 gir et oversiktlig bilde av hvordan forskningsdesignet til dette masterprosjektet ser ut.

Datamateriale	Lyddopptak og tilhørende transkripsjoner av fem semi-strukturerte lærerintervjuer, samt semistrukturerte spørreskjema fra fire av lærerne som ble intervjuet.	Kapittel 3.2
Analysemetode	Studien er analysert gjennom en kombinasjon av teoridrevet og konvensjonell innholdsanalyse. MKU-rammeverket er bearbeidet fra et observasjonsskjema til et semistrukturert spørreskjema. Intervjuene er hovedsakelig analysert gjennom en konvensjonell innholdsanalyse, med et moment av teoridrevet innholdsanalyse ved bruk av TPACK. Med dette som grunnlag kan jeg gjennom egen forskning utvikle kunnskap basert på deltakernes unike perspektiver, samt vurdere, bekrefte, utfordre og supplere eksisterende teori knyttet til god matematikkundervisning med programmering (Fauskanger & Mosvold, 2015).	Kapittel 3.3
Teoretisk rammeverk	Studien tar utgangspunkt i teori om hva god matematikkundervisning med programmering er, og hvilke elementer som bør integreres. MKU-rammeverket (Opsvik & Skorpen, 2014) er et rammeverk for å analysere faglig kvalitet i undervisning. TPACK (Koehler & Mishra, 2009) er en modell for å undersøke læreres kunnskap om interaksjonen mellom pedagogikk, faginnhold og teknologi.	Kapittel 2.4

Tabell 5: Forskningsdesign

3.2 Overveielser ved datainnsamlingen

I denne studien vil det bli undersøkt hvordan situasjonen er i skolen med programmering i matematikkundervisningen. Spesielt interessant er det å se lærernes perspektiver på et tema som de er satt til å måtte forholde seg til, men som ikke var en del av yrket deres da de utdannet seg og skrev under på arbeidskontrakten deres. Målet med studien er derfor å få innsikt i hvordan lærere opplever det å undervise i matematikk med programmering som et nytt element. En slik studie vil betraktes som fenomenologisk (Christoffersen & Johannessen, 2012; Postholm & Jacobsen, 2018). Fenomenologiske studier har som mål å forstå verden mennesker lever og handler igjennom deres erfaringer og bevissthet (Postholm & Jacobsen, 2018). Med en fenomenologisk tilnærming vil studien kunne beskrive noen læreres erfaringer og forståelse med programmering i matematikkundervisning. For dette må ulike utforskningsstrategier veies opp mot hverandre.

3.2.1 Flermetodestudie

Det finnes mange muligheter for innsamling av datamateriale. For forskning hvor formålet er å få erfaringer, tanker og synspunkt egner intervjuer seg best (Christoffersen & Johannessen, 2012; Silverman, 2020; Thagaard, 2013). Det ble derfor raskt konkludert med at denne studien skulle bygges på intervjuer som datamateriale. Samtidig var det et stort ønske om å se hvordan lærerne utøvde programmering i praksis, samt å få en enda dypere innsikt i lærernes reflekterte tanker og holdninger til hvordan programmering kan fremme matematikkundervisningen. Derfor ble både observasjon av undervisning og spørreskjema vurdert. Å kombinere flere forskningsmetoder kan gjøre at styrkene til de ulike metodene utnyttes og samtidig at en får dekt svakhetene deres (Bryman, 2012). Det er likevel uenighet blant forskere om dette er fornuftig. Silverman (2020) for eksempel forteller at du alltid bør søke å bare bruke én metode, og at dersom du behøver å kombinere metoder er det et tegn på at du ikke har mestret innsnevringen av problemet du utforsker. Likevel kan bruk av forskjellige metoder hjelpe til med å svare på forskjellige spørsmål. Kvalitative metoder handler om å finne ut hvordan noe skjer eller oppleves, imens kvantitative metoder undersøker hvor mye, i tallform, noe konkret oppleves (Kvale & Brinkmann, 2019). Det ble besluttet å gjennomføre spørreundersøkelser blant lærerne i denne studien. Observasjon ble uaktuelt da lærerne i denne studien ikke hadde planer om å benytte programmering i undervisningen sin før mot slutten av tidsrammen denne studien har.

3.2.2 Semistrukturert intervju

Et intervju kan kategoriseres etter hvor tilrettelagt det er på forhånd. Strukturerte intervjuer er intervjuer hvor tema, spørsmål og rekkefølgen av spørsmålene er fast (Christoffersen & Johannessen, 2012). Ustrukturerte intervjuer er åpne intervjuer hvor det stilles åpne spørsmål om et gitt tema, og bærer preg av en samtale (Christoffersen & Johannessen, 2012). Intervjuer må imidlertid ikke være enten eller. Et intervju med en overordnet *intervjuguide*, men hvor intervjuet beveger seg litt frem og tilbake etter intervjuerens ønske kalles et semistrukturert intervju (Christoffersen & Johannessen, 2012). Spørsmål som ikke er nedskrevet kan stilles, men alle fastsatte spørsmål skal stilles, og det skal være en lik ordlyd i ulike intervjuer (Bryman, 2012). Det er ingen fasit for hva som gir best kvalitet på intervjuene, men denne studien velger å benytte en semistrukturert tilnærming.

Kvale og Brinkmann (2019) sier at det ikke finnes entydige kvalitetskriterier for forskningsintervjuer, men velger å trekke frem seks punkter som bør inngå for kvalitet i et intervju (Tabell 6). De tre siste punktene henviser i størst grad til det ideelle intervjuet. Videre konkluderer Kvale og Brinkmann (2019) med at et godt intervju også er avhengig av forskerens håndverksmessige dyktighet som omfatter for eksempel kunnskap om forskningstemaet og balansen mellom tydelighet og åpenhet, vennlighet og kritikk, og som hele tiden er bevisst sensitiviteten i forhold til relasjonen mellom forsker og intervjudeltaker samt etiske aspekter.

Nr.	Beskrivelse
-----	-------------

1	I hvilken grad fås spontane, innholdsrike, spesifikke og relevante svar fra intervjupersonen?
2	Jo kortere intervjuerens spørsmål er og jo lengre intervjupersonens svar er, desto bedre.
3	I hvilken grad følges spørsmålene opp fra intervjuerens side, og hvordan klargjøres betydningen av de relevante delene av svaret?
4	Idealintervjuet blir i stor grad tolket mens det pågår.
5	Intervjueren forsøker i løpet av intervjuet å verifisere sine fortolkninger av intervjupersonens svar.
6	Intervjuet er «selvkommuniserende» - det er i seg selv en fortelling som ikke krever særlig ekstra kommentarer og forklaringer.

Tabell 6: Kvale og Brinkmanns kvalitetskriterier for et intervju skrevet inn i tabell. (2019, s. 194)

Utvalgsstørrelse

For en uerfaren forsker kan det være utfordrende å velge ut antallet informanter til masterprosjektet sitt. Litteratur viser også til at det er ulike tanker om akkurat dette. Christoffersen og Johannesen (2012) råder til en utvalgsstørrelse fra 10-15 informanter, og med begrenset tid og økonomi, som i studentprosjekter, kan antallet være i underkant av 10 stk. Dalland (2015) henviser til at det kan være nok med en, to eller tre intervjupersoner. Kvale og Brinkmann (2019) sier at det er vanlig med alt mellom 5 og 25 intervjuundersøkelser. Med ulikt utvalg kommer naturligvis ulike fordeler og ulemper. Dalland (2015) forteller at det ikke er nødvendig med mange intervjuer fordi hensikten skal være å gå i dybden, og ofte får en mye informasjon fra hvert enkelt intervju. Christoffersen og Johannesen (2012) sier derimot at prinsippet, uavhengig av målgruppen, bør være at en samler inn data til det ikke lengre er ny informasjon å hente. Kvale og Brinkmann (2019) sier at det er en tendens til at kvalitative undersøkelser som regel har for mange eller for få intervjupersoner. Problemet med for mange intervjupersoner er, som Dalland (2015) også nevnte, at en ikke får gått i dybden på alle intervjuene, og med for få intervjupersoner vil det være vanskelig å generalisere (Kvale & Brinkmann, 2019). I senere tid er det et inntrykk av å være en fordel å ha et mindre antall intervjuer og heller bruke tid på forberedelser og analyse av intervjuene (Kvale & Brinkmann, 2019). I denne studien ble det besluttet å intervju fem lærere. Det ble gjennomført ett intervju med hver av de fem lærerne. Kriteriene for at lærerne skulle kunne delta i undersøkelsen var at de var praktiserende matematikklærere som har matematikkundervisning.

Delaktighet i intervjuene

I planleggingen av forskningen måtte jeg velge mellom å gjøre intervjuene selv eller om jeg skulle gå sammen med medstudenter med lignende tema som denne studien. Det er fordeler og ulemper ved begge sider. Ved å gå sammen med medstudenter kan vi fordele informanter oss imellom og anskaffe en større mengde datamateriale. Med mer data vil jeg igjen ha mulighet for flere ulike tolkninger som kan trekkes med i diskusjonen. Likevel landet jeg på at jeg kommer til å stå for alle intervjuene som gjøres selv. «*Idealintervjuet blir i stor grad tolket mens det pågår*» (Kvale & Brinkmann, 2019, s. 194). Svakheten dette trekker med seg er at det vil også gå mer tid til transkripsjoner og til gjennomføring av selve intervjuene. Styrken dette har ligger i blant annet Kvale og Brinkmanns (2019) utsagn over. I tillegg har jeg allerede valgt å gå for en kvalitativ forskning. Å delta i intervjuet bringer med aspektet med nærhet, tolkning og en menneskelig interaksjon ved bruk av alle sanser. Å være til stede i intervjusituasjonen, å få tolke og analysere allerede i øyeblikket og kunne grave dypere der jeg finner interessante

utsagn er en stor del av kvalitativ forskning og veier tyngre enn mengde med datamateriale i denne studien.

Intervjuguidene

Intervjuguiden vil bli en generell guide med spørsmål som skal hjelpe meg som intervjuer med å gjennomføre et ryddig intervju. I prosessen med å produsere intervjuguiden sier Bryman (2012) at det er noen grunnleggende elementer som en bør tenke over. Det bør være en oversiktlig emneinndeling for god flyt i intervjuet, spørsmål bør vinkles mot problemstillingen i studien, språket bør tilpasses den som blir intervjuet, en bør unngå ledende spørsmål og innhente både generell og spesifikk bakgrunnsinformasjon om deltakeren. Bryman (2012) anbefaler også å gjennomføre noen prøveintervjuer for uerfarne intervjuere. Intervjuguiden i denne studien ligger ved som vedlegg 4. Den består av spørsmål som legger opp til å bli bedre kjent med deltakeren, refleksjoner læreren har om god matematikkundervisning, erfaringer med undervisning av programmering og lærerens forståelse av forholdet mellom programmering og matematikk.

I arbeidet med intervjuguidene har det vært et mål å ha åpne introduksjonsspørsmål, for å få intervjupersonenes spontane tanker om temaet. I tillegg sier Kvale og Brinkmann (2019) at spørsmålene bør være korte og enkle. Med åpne spørsmål ble det viktig å forberede seg på å stille oppfølgingsspørsmål som åpnet for redegjørelse og for å kunne ta tak i interessante utsagn og tanker. Det kunne også bli relevant å stille inngående spørsmål der en utforsker grundigere intervjupersonens refleksjoner. Spørsmål som «kan du utdype?» eller «har du eksempler?» kan benyttes, både for å sikre at jeg tolker svarene riktig og for å få personlige eksempler. Et interessant område denne studien undersøker er læreres opplevelse av undervisning med programmering. Til dette vil det være hensiktsmessig å benytte seg av spørsmål om hvordan intervjupersonene opplever blant annet kollegers erfaringer, ferdigheter og kunnskap. Dalland (2015) sier at forkunnskaper om informant og tema er nødvendig for å kunne avgjøre på kort tid om en skal gå i dybden på besvarelser, gi respons eller gå videre. Om nødvendig kan en heller velge å øke antall intervjuer i ettertid dersom det viser seg å bli for spinkel informasjon.

Lydopptak og transkripsjon

Med intervju som forskningsmetode, og flere intervjuer som skal gjennomføres, er det nyttig å gjøre opptak av intervjuene. Videre transkriberes intervjuene så de blir i tekstform. Bryman (2012) peker på flere fordeler ved å ta opp og transkribere intervjuer. Blant annet omhandler det at man skal inkludere alt som blir sagt, og ikke miste praktiske og viktige detaljer. Det tillater også grundigere undersøkelser av hva som blir sagt, og gir mulighet for å gå igjennom

intervjuene flere ganger. I tillegg styrker det validitet gjennom at en kan åpne opp for offentlig gransking og analyser av andre forskere og gir også mulighet for at dataen kan gjenbrukes på andre måter enn den opprinnelige forskerens tiltenkte problemstilling.

Når intervjuer transkriberes er det viktig at det blir gjort riktig. I dette prosjektet står jeg alene om transkripsjonen, men dersom det er flere ulike mennesker som skal transkribere blir det ekstra viktig med en mal for hvordan dette skal gjøres. Det opprettes dermed en *transkripsjonsnøkkel* som forklarer hva ulike tegn betyr og hvordan det skal transkriberes. Blant annet må man ta stilling til om man skal inkludere alle lyder som kommer frem, hvordan pauser skal komme frem og om det skal transkriberes på dialekt eller ikke. Jeg har valgt å produsere og legge ved transkripsjonsnøkkelen i dette prosjektet for å gi innsikt i hvordan ulike tegn skal tolkes i transkripsjonene. Transkripsjonsnøkkelen er vedlagt som vedlegg 5. Nøkkelen er produsert i samråd med erfarne forskere gjennom Mathematics Education Research Group 2022, (MERG2022), et forskningsprosjekt som var del av faget MGL4122 i masterprogrammet Matematikdidaktikk ved Universitetet i Stavanger, høsten 2022. Transkripsjonene ble skrevet inn i en mal som er vedlagt som vedlegg 6, og som indikerte kommentarnummer i intervjuet, tidspunkt i intervjuet, hvem som uttalte seg, diskursen og eventuelle kommentarer. Intervjuene er grundig gjennomgått og i resultatdelen av studien er det hele tiden et mål å få frem hva deltakeren som intervjues ønsker å formidle. Fordi jeg står alene med transkripsjonsarbeidet ble det en viktig etisk vurdering å dele resultatdelen av studien med lærerne som ble intervjuet. De har fått lese gjennom studien før den ble ferdigstilt, de er blitt fortalt hvilket «pseudonym» de er i studien og de har kunnet komme med tilbakemeldinger på om de er mistolket. På det viset sikres det at lærerne ikke fremstilles feil, og det er gjort én korrigerende på bakgrunn av tilbakemeldingene.

3.2.3 Semistrukturert spørreskjema

Spørreskjemaer benyttes som regel i kvantitative undersøkelser. Fordi denne studien er av kvalitativ art, ble det derfor nødvendig å gjøre justeringer som gjør at spørreskjemaene får en mer kvalitativ tilnærming. Spørreskjemaer med både prekodete og åpne svar kalles for semistrukturert spørreskjema (Christoffersen & Johannessen, 2012). Spørreskjemaet er dermed tilpasset og får mange likheter med et strukturert intervju. Strukturerte intervju er bygd opp med et fast tema og fast rekkefølge, men med åpne spørsmål (Christoffersen & Johannessen, 2012). Deltakeren får med dette mulighet til å besvare med egne ord.

De semistrukturerte spørreskjemaene i denne studien ble tilsendt de ulike deltakerne og ble gjennomført uten forskerens tilstedeværelse. En svakhet ved dette, og generelt med

spørreskjemaer, er at en mister deltakernes spontane reaksjon og de første assosiasjonene deltakerne gjør seg ut fra spørsmålene. I stedet blir resultatene fra de semistrukturerte spørreskjemaene i større grad reflekterte, bearbejdede og gjennomtenkte tanker og meninger lærerne har. Dette styrker studien gjennom at spørreskjemaet gir en dypere innsikt i deltakernes holdninger og ikke bare deres første tanker. I prosessen med å utarbeide et spørreskjema informantene mine skulle besvare var det nødvendig å gjøre overveielser angående spørsmålene, svarmulighetene og poenggivingen.

Spørsmålsformulering

Formuleringen av spørsmål, eller påstander, er av stor betydning i et spørreskjema. En bør forsøke å stille mest mulig konkrete spørsmål (Christoffersen & Johannessen, 2012). Gjennom spørsmålene måtte jeg identifisere om deltakerne mente at programmering ville fremme god matematikkundervisning. Det ble derfor tatt utgangspunkt i observasjonsskjemaet Opsvik og Skorpen (2014) brukte i sin forskning for god matematikkundervisning og hver dimensjon og dens underkategorier ble omskrevet til påstander i et spørreskjema. Hver av påstandene skulle entydig lede til én kategori. Tabell 7 viser et utdrag av hvordan omformuleringene skjedde.

<i>Matematikkfaglig opptak og tilbakemelding</i>	<i>Observasjonsskjema (Opsvik & Skorpen, 2014)</i>	<i>Påstand i spørreskjema</i>
<i>2.2 Opptak og tilbakemelding</i>	Læreren forstår, bygger videre på eller gir tilbakemelding på elevane sine matematiske forklaringar, spørsmål og resonnement	I programmering kan jeg forstå, bygge videre på og gi tilbakemeldinger på elevens matematiske forklaringer og spørsmål.

Tabell 7: Eksempel på tilpassing fra observasjonsskjema til påstand.

Lukkede svar med kommentarfelt

En av vurderingene som ble gjort om spørreskjemaene var at det skulle bli benyttet lukkede svaralternativer. Lærerne fikk mulighet til å besvare påstandene med ett av fem alternativer. De skulle på en gradert skala med fire alternativer, fra «Helt enig» til «Helt uenig», svare det alternativet som samsvarte best med deres oppfatning. Blant forskere er det uenighet om en skal inkludere et nøytralt alternativ (Christoffersen & Johannessen, 2012). Det var uaktuelt i denne studien å ikke inkludere et nøytralt femte alternativ, «Vet ikke/Ingen formening», da «ingen formening» er en formening om at programmering verken fremmer eller svekker dimensjonen. Christoffersen og Johannessen (2012) kan også vise til at forskning sier at en undersøkelse med fem trinn har høyest reliabilitet.

En annen vurdering som ble gjort var å inkludere obligatoriske kommentarfelt til hver påstand. Dette er begrunnet med at det var viktig for studien å få innsikt i lærerens tolkning og resonnering av hver påstand. Det obligatoriske kommentarfeltet gjorde spørreundersøkelsen semistrukturert og at den tilnærmet seg en kvalitativ undersøkelse. Gjennom påstandene var det heller ikke mulighet for deltakerne til å få forklart hva som lå i ordlyden eller formuleringene. Lærernes assosiasjoner, tolkninger og forståelse var individuelle, og de kunne blitt besvart annerledes om de hadde forstått påstandene på en annen måte. Dette ville vært et problem når svarene skulle analyseres og kommentarfeltet styrker derfor muligheten til å forstå hva som ligger i besvarelsene. Det ble besluttet at undersøkelsen kun skulle inneholde 12 påstander for at undersøkelsen skulle ha et kvalitativt preg. Med dette grepet ble spørreundersøkelsen begrenset, og deltakerne kunne prioritere å fylle kommentarfeltene og grunngi svaret sitt. Deltakerne ble også motivert til å eksemplifisere og inkludere assosiasjonene deres i kommentarfeltet.

Poenggiving

Ofte kan det være ønskelig med partall antall alternativer i spørreskjemaer. Da tvinges deltakeren til å ta stilling mot en side. Dette var ikke et ønske i denne studien, da programmering ikke behøver å påvirke noen av dimensjonene for kvalitet i matematikkundervisningen. Det ble vurdert at med de fem ulike kategoriene ville det bli naturlig å gi positive poeng for de positivt rettede svarene, imens det var negative poeng for de negativt rettede svarene. Dette medførte at for alle kategoriene ville svaralternativet «helt enig» gi 2 poeng, imens «litt uenig» gir -1 poeng. Unntaket var i kategori tre hvor påstandene er formulert motsatt. Det vil si at det vil være «helt uenig» som underbygger Opsvik og Skorpens (2014) dimensjon for god matematikkundervisning og som vil tilsvare 2 poeng.

Frivillig deltakelse

Deltakerne i studien har gitt sitt samtykke til å delta i denne studien. Likevel står de fritt til å trekke seg når som helst dersom de ønsker dette (Se delkapittel 3.4). Da spørreskjemaene ble utlevert valgte en av deltakerne å avstå fra å besvare dette. Begrunnelsen hans var at det ble for vanskelig å svare på påstandene da deltakeren ikke hadde gjennomført undervisning med programmering enda.

3.2.4 Studiens deltakere

I studien er det intervjuet fem lærere fra to forskjellige skoler. Alle lærerne underviser eller har undervist i matematikk. Selv om undersøkelsen inneholder både kvinnelige og mannlige deltakere, velger jeg å alltid omtale deltakerne som hankjønn for å overholde konfidensialiteten med krav og løfte om anonymisering av de ulike deltakerne.

Lærer A

Første deltaker er lærer på 9. trinn og har alltid undervist på ungdomsskolen. Læreren har arbeidet på samme skole i 13 år, og fikk noe tilfeldig matematikk som undervisningsfag tidlig, selv om læreren har språkfag som hovedfag. Deltakeren har stor interesse for data og arbeidet i databransjen før læreryrket stod for tur. Deltakeren forteller selv at det var på tide med programmering inn i den nye læreplanen og beskriver seg selv som et datamenneske som lagrer alt digitalt.

Lærer B

Andre deltaker har arbeidet som lærer i 23 år. Læreren underviser flere klasser i matematikk fordelt på hele ungdomsskolen. Han har alltid undervist på ungdomsskolen og har videreutdanning og mastergrad som lærerspesialist i matematikk.

Lærer C

Tredje deltaker var lærer i syv år før han gikk over til å bli en del av skoleledelsen. Deltakeren har tross sitt ansvar som skoleleder hatt matematikkundervisning i 15 år. Deltakeren har 90 studiepoeng i matematikk og alltid undervist på ungdomsskole.

Lærer D

Fjerde deltaker har arbeidet i skolen i elleve år. Læreren har 60 studiepoeng i matematikk og har arbeidet både på barneskole og ungdomsskole.

Lærer E

Femte deltaker har jobbet mest på ungdomsskolen i sine 14 år som lærer, men også noe på barneskole. Læreren har 70 studiepoeng i matematikk. Læreren har i alle år undervist i matematikk og har en ekstra stor interesse for blant annet First Lego League.

3.3 Analysemetode

I dette kapitlet blir det presentert hvordan intervjuene og spørreskjemaene ble analysert gjennom rammeverkene TPACK og MKU, samt en konvensjonell innholdsanalyse av intervjuene. Rammeverkene ble presentert i delkapittel 2.3. Alle trinn er grundig presentert med eksempler for størst mulig reliabilitet for studien.

3.3.1 Analyse av spørreskjema

Med Opsvik og Skorpen (2014) sitt MKU-rammeverk som inspirasjon ble det generert påstander ut ifra de ulike dimensjonene. Hver påstandene skulle kunne kodes entydig til de originale dimensjonene og underkategoriene. Påstandene ble besluttet å avkortes, både i lengde og innhold, så det skulle være enklest mulig for deltakerne å forstå påstanden. Utfordringen med denne avkortningen var at det ville gå på bekostning av noe av innholdet i dimensjonene. Et alternativ kunne være å ha et spørsmål for hver del av innholdet av underkategoriene. Dette ville medført mange flere spørsmål. Fordi dette er en kvalitativ forskning, ble det besluttet å ha en påstand som representerte underkategorien, også fikk noe av innholdet stå ubesvart.

<i>Matematisk rikdom</i>	Observasjonsskjema (Opsvik & Skorpen, 2014)	Påstand i spørreskjema
<i>1.2 Forklaringer</i>	Gi matematisk mening til omgrep, idear, prosedyrar eller løysningsmåtar.	Programmering gir meg god mulighet til å forklare ulike matematiske begrep og prosedyrer?

Tabell 8: Eksempel på forkorting av påstand fra observasjonsskjema.

Tredje dimensjon er *feil og unøyaktighet*. Denne dimensjonen inneholder tre underkategorier, hvor 3.1 (Større matematiske feil eller mangler) og 3.3 (Uklart innhold) ble besluttet at inneholdt så likt innhold at de ble fusjonert til én påstand.

<i>Feil og unøyaktighet</i>	Observasjonsskjema (Opsvik & Skorpen, 2014)	Påstand i spørreskjema
<i>3.1 Større matematiske feil og mangler</i>	Løyse ei oppgåve feil, bruke omgrep på feil måte, gløyme eller utelate viktige matematiske sider ved det ein fokuserer på.	Med programmering er det raskt for at en utelater/ikke fokuserer på de matematiske sidene.
<i>3.3 Uklart innhold</i>	Det matematiske innhaldet blir uklart presentert.	

Tabell 9: Fusjonering av to kategorier til én påstand.

De ulike dimensjonene består av et ulikt antall underkategorier. *Matematisk rikdom* består av fem underkategorier, *opptak og tilbakemelding* består av to underkategorier, *feil og unøyaktighet* består av to underkategorier og *elevdeltakelse i matematiske resonnement og meningsskaping* består av tre kategorier. Dette medfører at i spesielt andre (opptak og tilbakemelding) og tredje (feil og unøyaktighet) dimensjon har hver påstand betydelig større påvirkning på resultatet enn hva som gjelder for første dimensjon (matematisk rikdom) som består av fem påstander. Dette må tas hensyn til ved analysen av resultatene.

Poenggingen gjøres på en slik måte at hvert av svaralternativene gir en poengscore. La meg presisere at poengene er noe deltakerne gir og ikke noe de får. Dette fordi analysen er av programmering og ikke av lærerne. Dersom de er «helt enig» i en påstand, tilsvarer dette at de gir underkategorien 2 poeng. Hvis de er «litt enig» gir de 1 poeng, hvis de er «litt uenig» gir de -1 poeng og hvis de er «helt uenig» gir de -2 poeng. En kan forestille seg at når deltakerne svarer «vet ikke/Ingen formening», så kunne det i noen tilfeller vært riktig å annullere spørsmålet for deres besvarelse. Fordi alle underkategoriene er en del av MKU-rammeverket må de likevel inkluderes, og det vil bli gitt en score på 0 dersom de svarer dette. Spørsmålet regnes derfor med i utregningen av den gjennomsnittlige poengscoren deltakeren gir. Poenggingen i dimensjon 3 er motsatt av de andre dimensjonene. Dette medfører at for eksempel «helt uenig» gir 2 poeng og «helt enig» gir -2 poeng.

Lærer B

	Påstand	Svar	Score
<i>1.2 Forklaringer</i>	Programmering gir meg god mulighet til å forklare ulike matematiske begrep og prosedyrer?	Litt uenig	-1
<i>3.2 Feil og unøyaktighet</i>	Med programmering er det raskt for at en benytter et mer hverdagslig (upresist matematisk) språk.	Litt uenig	1

Tabell 10: Eksempel på motsatt poenggivning av likt svaralternativ.

Den totale scoren som oppnås deles på antall underkategorier for å beregne et gjennomsnitt av poeng deltakeren gir programmeringsundervisning i dimensjonen. Gjennomsnittet vil bli benyttet for å få en oversikt over hvilke dimensjoner deltakerne mener programmering fremmer i størst grad, og hvilke dimensjoner programmering svekker eller ikke har innvirkning på.

Studien er en kvalitativ studie, og det ble derfor besluttet å legge ved et obligatorisk kommentarfelt til hver påstand i spørreskjemaet. De individuelle kommentarene skal gi en forklaring, et eksempel eller en utdypning av deltakerens assosiasjoner og tanker rundt deres egen besvarelse. Dette gir mulighet for å utforske ulike syn på like påstander, og til å supplere poengscorene med argumenter. Jeg ser på dette som en styrke ved undersøkelsen.

*Elevdeltakelse i
matematiske
resonnement og
meningsskaping*

Lærer E

*4 Elevspørsmål
og påstander*

Påstand	Svar	Kommentar fra deltaker	Score
Programmering gir elevene mulighet til å stille spørsmål og lage hypoteser med matematisk innhold.	Litt uenig	Spørsmål og hypoteser skrevet som en matematisk formel kan lettere testes ut av et program.	-1

Tabell 11: Eksempel med svar, kommentar og poengscore.

3.3.2 Analyse av intervjuer

Intervjuene med lærerne i denne studien har blitt transkribert, og transkripsjonene ble benyttet for å analysere innholdet. Dette ble gjort hovedsakelig gjennom en *konvensjonell innholdsanalyse*. En konvensjonell innholdsanalyse kjennetegnes ved at forskere dykker ned i datamaterialet grundig i sin helhet og gjennom gjentatte gjennomlesinger identifiseres viktige tanker og begreper (Fauskanger & Mosvold, 2015). Dette leder til at koder genereres før de grupperes i meningsfulle kategorier eller overskrifter. Konvensjonell innholdsanalyse ble vurdert nyttig til denne studien da det var et ønske om å utforske innholdet i intervjuene.

I prosessen med generering av koder ble likt tematisk innhold kodet med en unik fargekode. De samme kodene gjaldt for alle intervjuene. Videre ble utsagn med lik fargekode samlet før det ble gitt meningsfulle overskrifter. Det ble vurdert å lage kategorier av de ulike temaene, men fordi intervjuene inneholdt lærernes tanker og erfaringer ble det mer naturlig å lage grupperende

overskrifter i stedet, da korte kategorinavn kan bli for upresist i forhold til innholdet. Resultatet ble sju overskrifter som kommer fram i tabell 12.

Nr.	Overskrift
1	Lærernes tanker om hva som kjennetegner god matematikkundervisning
2	Lærernes reaksjoner til implementering av programmering i LK20
3	Lærernes programmeringsferdigheter og profesjonsutvikling
4	Hva er lærernes forståelse av programmering i matematikk i dag?
5	Hvordan undervises programmering i matematikk?
6	Hvordan lærerne mener at programmering bør undervises
7	Hva sier lærerne om den endrede lærerrollen i matematikkundervisningen med programmering?

Tabell 12: Overskrifter generert fra intervju gjennom konvensjonell innholdsanalyse.

Analysene har også bestått av en mindre del *teoridrevet innholdsanalyse*. Dette innebærer at etablerte kategorier, modeller eller rammeverk fra teori eller forskning benyttes. I arbeidet med transkripsjonene, sammen med analyser gjennomført ved de opprinnelige intervjuene, kommer det frem utsagn fra lærerne som peker på deres og andre læreres kunnskap om ulike komponenter som er definert i rammeverket TPACK. I denne studien ble det forsøkt å definere og kategorisere ulike underkategorier av TPACK. Denne delen av analysen var en krevende prosess fordi som Archambault og Barnett (2010) sier er grensene mellom komponentene uklare. Det ble grundig utforsket hvordan Koehler og Mishra (2009) definerer de opprinnelige kategoriene og dynamikken mellom de, samt utforsket hvordan andre forskere hadde beskrevet sin forskning analysert gjennom TPACK (f.eks. Maugesten et al., 2022). Resultatet ble et dokument med hovedkategoriene fra TPACK og underkategorier av disse forsøkt definert. Følgende er et eksempel på hvordan kategorien *Teknologisk innholdskunnskap* ble definert gjennom tre underkategorier:

<i>Teknologisk innholdskunnskap</i>	Kode	Forklaring
5.1	TC1	Kunnskap om hvordan programmering er relatert til matematisk innhold.
5.2	TC2	Kunnskap om at valget av teknologi både gir muligheter og setter begrensninger for matematisk innhold som kan undervises. På samme måte kan ulikt matematisk innhold begrense hvilke typer teknologier som kan brukes.
5.3	TC3	Kunnskap om hvordan emnet kan endres ved bruk av teknologier, hvilke teknologier som er best egnet for læring i matematikk og hvordan faginnholdet kan endre teknologien.

Tabell 13: Beskrivelse av innhold i teknologisk innholdskunnskap og kode.

Å finne utsagn som kunne kodes direkte til underkategoriene ble utfordrende, spesielt fordi intervjuene ikke hadde vektlagt spørsmål for å identifisere eller utforske lærernes ferdigheter. Spørsmål og oppfølgingsspørsmål ble heller stilt med en hensikt om å prøve å få innblikk i deres tanker, erfaringer og refleksjoner rundt temaet programmering og undervisning i programmering. For eksempel var det få spørsmål som kunne få frem lærernes kunnskap om matematikkfagets identitet. I tillegg kunne lærerne flere ganger prate om for eksempel ulike teknologiske program uten at de forklarer om de har kunnskap om programmet. Kombinasjonen av uklare grenser mellom komponentene i rammeverket og et intervju som ikke hadde som mål å identifisere TPACK-komponentene skaper unøyaktigheter i kodingen. Likevel ble kodingen benyttet for å ha en generell oversikt og forståelse av lærernes kunnskaper før jeg diskuterte resultatene som kom frem av intervjuene. TPACK egnet seg derfor ikke alene for å besvare denne studiens forskningsspørsmål, men ble benyttet som et ledd i analysen av intervjuene.

3.4 Forskningsetiske perspektiver

Å drive forskning skal gjøres på en måte som er etisk forsvarlig overfor alle involverte parter i prosjektet. Det er ikke vanskelig å forstå viktigheten av å overholde disse etiske normene om en dukker inn i litteraturen. Som regel vies det av store deler, og av og til hele kapitler, til temaet (f.eks. Dalland, 2015, kap. 6; Kvale & Brinkmann, 2019, kap. 4; Silverman, 2020, kap. 6; Thagaard, 2013, kap. 1,4). Det stilles ulike krav til egenskaper og opptredener gjennom forskningsperioden. Thagaard (2013) peker på redelighet og nøyaktighet som krav til forskere. Tjora (2017) vektlegger tillit, konfidensialitet, respekt og gjensidighet som nødvendige aspekter i kontakten med informanter og Kvale og Brinkmann (2019) går i dybden på filosofien og forbinder det med situert skjønn. Uavhengig av hva en kaller det og hvilke egenskaper en vektlegger er poenget deres å meddele viktigheten av å drive etisk forsvarlig forskning.

Den nasjonale forskningsetiske komité for samfunnsvitenskap og humaniora (NESH) har ansvar for å utarbeide nasjonale forskningsetiske retningslinjer. Formålet med retningslinjene er å gjøre rede for hensyn og forpliktelser forskere har overfor deltakere, i tillegg til at de utdyper hvilket ansvar forskeren har for å ivareta forskningsetikken (NESH, 2021). Å gjøre rede for prioriteringer i forskningsprosessen er et kjennetegn på god forskningspraksis (NESH, 2021). I 2021 presenterte NESH den femte utgaven av forskningsetiske retningslinjer gjeldende for samfunnsvitenskap og humaniora (NESH, 2021). Den bygger på de generelle forskningsetiske retningslinjenes prinsipper (NESH, 2019) som er respekt, gode konsekvenser, rettferdighet og integritet. To sentrale forskningsetiske prinsipper er informert samtykke og konfidensialitet.

3.4.1 Samtykke

Et av kravene NESH stiller til forskning er at deltakelse til forskningen skal være frivillig (2021). Det forutsetter at forskeren gir tilstrekkelig informasjon som spesifiserer studiens formål, metode, hvordan resultatene vil brukes, og følger av å delta (NESH, 2021). Deltakelse skal derfor ikke komme av press. Dette kan være en utfordring i mange situasjoner, for eksempel dersom det blir gitt uttrykk for at det kan få negative konsekvenser hvis en ikke velger å delta, eller ved at samtykke samles inn av en autoritetsperson (NESH, 2021). Deltakerne i denne studien fikk et informasjonsskriv (vedlegg 3) hvor det kommer tydelig frem at deltagelsen kan trekkes tilbake eller avslås uten begrunnelse, og uten at det vil få noen negative konsekvenser.

3.4.2 Konfidensialitet

Konfidensialitet innebærer at personlige opplysninger som hovedregel skal være avidentifisert, og formidling av forskningsmaterialet som hovedregel skal være anonymisert (NESH, 2021). Alle lydopptak har blitt bevart konfidensielt på krypterte minnepenner. For å ivareta anonymiteten har alle deltakerne blitt omtalt som «lærer» etterfulgt av en unik bokstav i transkripsjonene og i denne studien. Stedsnavn og personer som omtales blir erstattet med klammeparenteser med kort forklaring, som f.eks. [by i Norge] eller [kollega]. Dette var for å forhindre eventuelle problem knyttet til personvern og opplysninger om deltagernes geografiske lokasjon. I tillegg er diskursen i transkripsjonene normalisert til bokmål og alle deltakerne omtales som hankjønn, uavhengig av opprinnelig kjønn.

3.4.3 Søknad til Kunnskapssektorens tjenesteleverandør (Sikt)

I forkant av datagenereringen ble det søkt om tillatelse for datainnsamling ved Kunnskapssektorens tjenesteleverandør (Sikt) (se vedlegg 1). Dette ble gjort før intervjuene ble gjennomført og spørreskjemaene ble besvart. Der ble det bekreftet at behandlinger i prosjektet er i samsvar med personvernlovgivning, og at opplysninger bevares konfidensielt og følger personvernforordningen (se vedlegg 2).

3.5 Studiens kvalitet

Validitet i et forskningsprosjekt ser på studiens kvalitet og er en nødvendighet for å kunne anerkjenne og bygge videre på forskningen. Den avgjøres av troverdigheten til resultatene, om de er sanne og om slutningene en trekker er gyldige (Dahlum, 2021). Begrepet deles av noen i to undergrupper, ytre og indre validitet. Yin (2018) velger derimot å utvide til de fire følgende komponentene for å vurdere om forskningen er av god kvalitet.

Konstruktvaliditet går ut på å bevise gjennom flere ulike metoder hvordan situasjonen er (Yin, 2018). Argumentasjonen blir både sterkere og mer troverdig om en både studerer besvarelser fra spørreskjemaer og intervjuer lærere om deres matematikkundervisninger med programmering enn om en kun studerer intervjuer. Ulik metodebruk hjelper for å benytte styrkene til ulike metoder, og samtidig minske svakheter. Det ble benyttet flere ulike datakilder i denne studien, deriblant litteraturgjennomgang, semistrukturerte spørreskjemaer med kvalitativ tilnærming og semistrukturerte intervjuer for å hente inn relevant informasjon om domenet som forskes på. Både spørreskjemaene og intervjuene har sin styrke i at jeg får erfaringer og opplevelser fra primærkilder. Ut fra resultatene bygges en kjede av bevis for å kunne konkludere (Yin, 2018). Dette er et viktig trinn i validiteten av forskningen for å vise til at fenomenet og forskningen ikke er konstruert av forskerens subjektive holdninger og kommer frem i diskusjonskapittelet i denne studien.

Indre validitet går ut på om det er en sammenheng mellom funnene og hypotesen. Noen hypoteser kan stemme veldig godt med funn som blir gjort, men det behøver ikke å være argumentet som er årsaken til resultatene. Det kan være andre årsaker til at resultatene forekommer og funnene blir ugyldige. Forskere gjør mange slutninger, og slutningene gjøres ofte basert på antagelser (Yin, 2018). I diskusjonskapittelet er det forsøkt å studere alle sider av de ulike sakene som er kommet frem av studien, både de som lærerne nevner, de forskning nevner og også om det er sider som ikke er belyst. Den grundige oppveiningen av argumenter for og imot hverandre er belyst for leserne i diskusjonen og konklusjonen kommer frem i avslutningskapittelet. Dette er med på å styrke den indre validiteten.

Ytre validitet omhandler hvorvidt en kan avgjøre om funnene er generaliserbare for andre som tilfredsstillende kravene for utvalget i forskningen. Denne studien har undersøkt fem lærere fra to ulike skoler. Postholm og Jacobsen (2018) sier at med fenomenologiske studier kan generalisering finne sted mellom lignende kontekstuelle situasjoner. «Hvordan»-spørsmål, som i dette prosjektet, er også hjelpelig for å kunne generalisere (Yin, 2018), men med et

begrenset utvalg og få særegne kontekstuelle kjennetegn ved deltakerne blir det ikke mulig å generalisere funnene fra denne studien. Det kan ikke konkluderes med at programmering fremmer matematikkundervisning eller hvordan programmering bør undervises, selv med lignende kontekstuelle situasjoner. En nødvendig presisering om matematikkundervisning er også at undervisning er mer kompleks enn hva MKU-rammeverket fanger opp (Opsvik & Skorpen, 2014). Det grundige arbeidet i forberedelsesfasen, hvor det er oppsøkt teori som er relevant inn mot denne studien, har likevel styrket den ytre validiteten. Studien sier samtidig kun noe om hvordan disse fem lærerne opplever at programmering fremmer matematikkundervisning og hva disse fem lærerne ønsker at programmering skal tilføre matematikken.

Reliabilitet er konsistens og stabilitet i målingene (Svartdal, 2020). Reliabilitet er tett knyttet til validiteten av prosjektet og målet er å minimere bias (feil og avvik). Med høy reliabilitet skal forskningen kunne gjennomføres på nytt av en annen forsker, og resultatet skal bli det samme. Intervjuene i denne studien er blitt analysert gjennom en konvensjonell innholdsanalyse. En stor utfordring med dette er at det kan være utfordrende å identifisere nøkkelkategorier, og samarbeid mellom forskere i analyseprosessen er en forutsetning (Fauskanger & Mosvold, 2015). Fordi dette er et masterprosjekt hvor rammene er at studien skal gjennomføres individuelt har dette ikke blitt gjort slik det anmodes om, og reliabiliteten svekkes av dette. Likevel har deltakerne i studien fått utdelt studien før den ble ferdigstilt som et aktivt grep for å unngå feiltolkning av lærernes holdninger og forklaringer. Resultatene ble på grunn av utfordring med identifisering av nøkkelkategorier fremstilt under overordnede overskrifter.

For oppnåelse av høy reliabilitet bør en som forsker utarbeide en protokoll og en database til studien (Yin, 2018). Disse skal inneholde en trinnvis oversikt av gjennomførelsen av prosjektet, samt alt datamaterialet som er samlet gjennom prosjektet. Databasen skal være åpen for andre forskere, så de selv kan kontrollere og trekke slutninger fra resultatene. Beskrivelse av datainnsamlingsprosessen i denne studien er beskrevet i delkapittel 3.2 og datamaterialet er blitt lagret på en ekstern database som en ekstern part kunne ha mulighet til å inspisere om ønskelig. Dette styrker reliabiliteten av studien.

4 RESULTATER

I dette kapittelet tar jeg for meg resultatene fra studien som er gjennomført. I første del av kapittelet (4.1) presenteres resultatene fra spørreundersøkelsen som omhandler de fire dimensjonene som kommer frem av MKU-rammeverket. I andre del (4.2) tar vi for oss lærerintervjuene som har vært analysert gjennom en tematisk innholdsanalyse og gjennom TPACK-rammeverket.

4.1 Resultater fra spørreskjema

Resultatene fra det semistrukturerte spørreskjemaet presenteres gjennom MKU-rammeverkets fire dimensjoner som ble fremstilt i delkapittel 2.4.1. Vi ser på de ulike poengscorene som hver dimensjon er blitt gitt, samt kommentarene lærerne har lagt ved. Før resultatene fra spørreskjemaene presenteres vil jeg presisere at kommentarene fra spørreskjemaet er vel så viktig som tallscoren som kommer ut av spørreskjemaet. Tallene som kommer ut av undersøkelsen gir likevel en god indikasjon på hvilke elementer lærerne i denne studien mener programmering i størst grad fremmer.

4.1.1 Matematisk rikdom

I dimensjonen matematisk rikdom (Tabell 14) kommer det frem av spørreundersøkelsen at det er ulike formeninger om programmering har positiv innflytelse. Spesielt lærer E trekker opp snittet. Lærer B velger å ikke vurdere mer enn en av underkategoriene. Læreren forklarer dette i kommentarfeltene med at han har for lite erfaring med programmering til å ha noen formening om påstandene. Det kommer frem av resultatene at underkategori 1.3 (Flere løsningsmåter og prosedyrer) er den kategorien det er størst enighet om at programmering er med på å fremme. Lærerne i denne studien er generelt litt positive til at programmering fremmer dimensjonen, men i for liten grad til å kunne konkludere med det.

Matematisk rikdom

		Lærer A	Lærer B	Lærer C	Lærer D	Lærer E	Gj.snitt.
1.	Matematisk rikdom	-	0,20	0,20	0,40	1,20	0,50
1.1	Kobling og sammenbinding	-	1	1	-1	2	0,75
1.2	Forklaringer	-	0	-1	1	2	0,50
1.3	Flere løsningsmåter og prosedyrer	-	0	1	1	2	1,00
1.4	Generalisering	-	0	0	1	-1	0,00
1.5	Matematisk språk	-	0	0	0	1	0,25

Tabell 14: Oversikt over resultat av programmeringens innflytelse på matematisk rikdom.

Lærer E er helt enig i påstand 1.1, og forklarer, «*Matematiske formler/algoritmer bryter ned matematiske begrep i variabler og faste ledd. De kan omformuleres, og skrives om, og gir en output til hver input, slik som et program vil gjøre*». Læreren kobler med dette det matematiske emnet «funksjoner» til programmering og mener programmering fremmer muligheten for arbeid med dette.

Underkategori 1.3 viser seg å gjøre størst utslag i tabell 14. Lærer D sier følgende til påstanden, «*Det er ofte problemløsning, og dersom koden ikke fungerer må en feilsøke, eller prøve på nytt*». Læreren assosierer påstanden til problemløsningsoppgaver som kan gi elever utfordrende situasjoner når strategien deres stopper opp. Da vil en kunne oppleve at en må feilsøke eller begynne helt på nytt med nye strategier. Lærer C ser på programmering som et verktøy når han sier at, «*Dersom man i tillegg til vanlige løsningsmetoder, kan løse problemet ved hjelp av programmering vil man ha flere løsningsmetoder enn hvis man ikke behersker programmering*».

Lærer E forklarer til 1.4 at, «*Programmering kan ta tid og fokus fra det konkrete tema/ matematiske oppgaven man jobber med. Prosedyre/bruk av program kan avspore eleven fra det faglige innholdet*». I denne forklaringen peker læreren på at det teknologiske aspektet ved programmeringen kan oppsluke elevens oppmerksomhet, så fokuset ikke blir på matematikken.

4.1.2 Matematikkfaglig opptak og tilbakemelding

I dimensjonen matematikkfaglig opptak og tilbakemelding (Tabell 15) er det en klar positiv trend blant lærerne om at programmering fremmer matematikkfaglig opptak og tilbakemelding. Resultatet for begge underkategoriene tilsier at lærerne i denne studien er «litt enig». Det er grunn for å vurdere resultatene med en viss skepsis fordi det kun er to underkategorier av dimensjonen og hvert svar blir veldig utslagsgivende. Likevel ser vi at samtlige lærere er positive (med unntak av 2.2 hos lærer D) samtidig som at det er få «vet ikke»-besvarelser. Vi kan dermed fastslå at det oppfattes blant lærerne i denne studien at programmering fremmer matematikkfaglig opptak og tilbakemelding.

Matematikkfaglig opptak og tilbakemelding

		Lærer A	Lærer B	Lærer C	Lærer D	Lærer E	Gj.snitt.
2.	Matematikkfaglig opptak og tilbakemelding	-	1,00	1,50	0,00	1,50	1,00
2.1	Opplæring av vansker, feil og misoppfatninger	-	0	2	1	1	1,00
2.2	Opptak og tilbakemelding	-	2	1	-1	2	1,00

Tabell 15: Oversikt over resultat av programmeringens innflytelse på matematikkfaglig opptak og tilbakemelding.

Lærer C er helt enig i at programmering kan benyttes for å arbeide med matematisk innhold som elevene har vansker med (2.1). Kommentaren hans indikerer likevel at han selv ikke besitter fasiten på hvordan dette skal gjennomføres. Han skriver, «*De fleste temaer kan nok arbeides med innen programmering, men jeg har ikke nok erfaring til å komme med noe mer konstruktivt*». Kommentaren hans gir derfor en pekepinn på at han har stor tillit til at programmering har et stort potensial, gjennom at det kan benyttes i mange emner i matematikk. Likevel kreves det erfaring for å vite hvordan man kan benytte programmeringen i de ulike emnene. Lærer E kommer med et eksempel fra egen undervisning når han kommenterer, «*Jeg bruker for eksempel glidere i Geogebra til å konkretisere vekst til funksjoner/tegning av diverse grafer, samt formler i regneark*» (2.1). Læreren viser til at han visualiserer for elevene med det digitale verktøyet Geogebra. På denne måten brukes programmeringen som et verktøy, og ikke som en ferdighet elevene skal tilegne seg.

Når det gjelder underkategori 2.2 er det ulike oppfatninger blant lærerne i denne studien, selv om de fleste er positive til programmering. Lærer B kommenterer, «*Jeg har troen på at programmering kan gi meg innblikk i elevene sin forståelse og gi meg mulighet til gode tilbakemeldinger til elevene*». Læreren viser til at programmeringen kan gi han som lærer innblikk i elevenes forståelse og dermed kan han gi gode tilbakemeldinger. Lærer D mener ikke nødvendigvis at programmeringen gir han mulighet for å gi gode tilbakemeldinger til elevene. Han kommenterer, «*Jeg synes ikke jeg får gitt annen tilbakemelding i programmering enn ellers når jeg går rundt og hjelper. I programmering vil mange ganger programmet gi beskjed når det er feil og de må prøve en annen fremgangsmåte*». Lærer D sier med dette at fordi programmet stopper så får elevene selv tilbakemelding på at noe er feil. Det blir ikke gitt tilbakemelding på hva som er feil eller hvor feilen ligger. Dette må i så fall læreren ha kunnskap om og ha satt seg inn i for å kunne hjelpe med.

4.1.3 Feil og unøyaktighet

Dersom en bare ser på tallscoren dimensjonen feil og unøyaktighet (tabell 16) er tildelt kan det ved første øyekast se ut som at lærerne ikke mener programmering fremmer dimensjonen. Snittscoren er på 0,13. Likevel, ved nærmere utforskning, kommer det frem at alle lærerne i denne studien har en formening om de ulike underkategoriene, og at de faktisk er uenige. Det er spesielt interessant å se på lærer B og lærer E som har svært ulike oppfatninger av om programmering fremmer dimensjonen. Lærer C og lærer D er derimot helt enige seg imellom. Totalt sett ser vi at underkategori 3.1 tildeles en negativ score fra lærerne. Dette indikerer at lærerne mener det matematiske innholdet kan bli borte i programmeringen. I underkategori 3.2

kommer det frem at lærerne ikke mener språket blir mindre faglig rettet av å programmere i undervisningen.

Feil og unøyaktighet

		Lærer A	Lærer B	Lærer C	Lærer D	Lærer E	Gj.snitt.
3.	Feil og unøyaktighet	-	2,00	0,00	0,00	-1,50	0,13
3.1	Større matematiske feil eller mangler	-	2	-1	-1	-2	-0,50
3.2	Unøyaktig språkbruk	-	2	1	1	-1	0,75
3.3	Uklart innhold	-	-	-	-	-	-

Tabell 16: Oversikt over resultat av programmeringens innflytelse på feil og unøyaktighet.

Imens lærer B kommenterer til kategori 3.1, «Jeg tror ikke programmeringstemaet har større risiko enn andre emner i faget» for å utelate de matematiske sidene, stiller lærer E seg mer skeptisk. Han kommenterer, «Det kommer helt an på hva slags programmering/oppgaver man bruker. Min erfaring med MicroBit og Lego Mindstorms er at eleven gjerne vil haste videre til neste oppgave, uten å dvele ved de matematiske sidene ved oppgavene ...». Læreren peker på at engasjementet blant elevene og motivasjonen deres til å fullføre er større enn ønsket om å undre og fundere. Lærer E kommenterer likevel videre at, «Noen ganger klarer man likevel å "lure inn" matematikk, slik at elevene ubevisst vil velge nylærte algoritmer og løsningsmetoder neste gang de møter på lignende utfordringer». Når det gjelder bruk av matematisk språk (3.2) opplever Lærer E at, «Elevene har ikke som mål å bruke presist matematisk språk - de vil løse oppgaven for å komme videre. Som lærer må man bevisstgjøre elevene til å ordlegge seg presist for å unngå misforståelser og utydelige svar». Lærer C mener ikke at dette er et problem når han sier, «Tror tvert imot at programmering kan bidra til å skjerpe språket».

4.1.4 Elevdeltakelse i matematiske resonnement og meningskaping

I siste dimensjon, elevdeltakelse i matematiske resonnement og meningskaping (Tabell 17), kommer det tydelig frem at programmering har innflytelse. Samtlige besvarelser er positive (med unntak av et par «vet ikke/ingen formening»). Et gjennomsnitt for hele dimensjonen på 1,33 tilsier i stor grad at det er enighet om at programmering fremmer elevdeltakelse i matematiske resonnement og meningskaping. Spesielt er underkategori 4.3 fremtredende med nært full score fra de fire lærerne.

Elevdeltakelse i matematiske resonnement og meningsskaping

		Lærer A	Lærer B	Lærer C	Lærer D	Lærer E	Gj.snitt.
4.	Elevdeltakelse i matematiske resonnement	-	1,33	1,67	1,00	1,33	1,33
4.1	Elevforklaringer	-	2	1	1	0	1,00
4.2	Elevspørsmål og påstander	-	0	2	1	2	1,25
4.3	Eleven sitt kognitive engasjement	-	2	2	1	2	1,75

Tabell 17: Oversikt over resultat av programmeringens innflytelse på matematiske resonnement og meningsskaping.

I underkategori 4.1 kommenterer lærer D at, «Elevene kan øve seg på å forklare hvorfor de har valgt en bestemt fremgangsmåte. Det kommer ikke av seg selv. Men lærer må gi i oppgave å forklare løsningsmetode». Læreren peker dermed på at det er en læreroppgave å hjelpe elevene å resonnerer for å fremme elevenes metakognisjon og selvbevissthet. Lærer E kommenterer til kategori 4.2 at, «Spørsmål og hypoteser skrevet som en matematisk formel kan lettere testes ut av et program». I denne kommentaren ligger det at programmering kan benyttes som et verktøy for å undersøke spørsmål eller hypoteser.

Underkategori 4.3 slår sterkest ut av samtlige underkategorier i de fire dimensjonene i MKU. Til påstanden kommenterer lærer B at, «Jeg tenker at når en jobber med kognitivt utfordrende arbeid så kan programmering være en egne arbeidsmåte, uten at jeg har god erfaring med dette. For noen elever er programmering i seg selv kognitivt utfordrende» og lærer D kommenterer, «De må tenke flere steg fremover for å lage en kode. De må tenke logisk og kjenne til kommandoene i programmet». Alle lærerne er samstemte i at programmering er kognitivt krevende og at elevene får grublet og utfordret seg.

4.2 Resultater fra intervju

Resultatene fra intervjuene blir presentert i dette delkapittelet. Intervjuene er blitt analysert i en grundig prosess som har inkludert analyse gjennom TPACK og konvensjonell koding. Med et utforskende blikk på datamaterialet, som består av transkripsjonene fra lærerintervjuene i studien, er det blitt generert overskrifter som skal gi en ryddig oversikt over hva de ulike lærerne har omtalt og reflektert rundt den samme tematikken. Alle lærerne i denne studien har ikke nødvendigvis omtalt alle tematikkene. Alt som er beskrevet i delkapittelet er basert på utsagn fra studiens fem deltakere.

4.2.1 Lærernes tanker om hva som kjennetegner god matematikkundervisning

Lærerne i denne studien har mange ulike, men også en god del like tanker om hva som er med på å skape en god matematikkundervisning. Helt sentralt blant alle lærernes besvarelser kommer det frem at gode matematikkøker ofte er gjenkjent gjennom høy elevaktivitet og stort læringsutbytte blant elevene.

For å legge til rette for elevaktivitet er det vesentlig å få elevene i gang så fort som mulig med å gruble, utforske og tenke (I3_010; I4_016; I5_016). Det bør være et mål for lærere å motarbeide den tradisjonelle undervisningen hvor elevene er passive lyttere til heller å undervise med en mer praktisk tilnærming (I3_010). Problemløsningsoppgaver eller «grubliser» er ofte noe som engasjerer (I4_020) og det kreves grundig planlegging for at en skal finne gode og tilpassede utforskningsoppgaver til elevgruppen (I4_018). Problemløsningsoppgaver er oppgaver hvor elevene bruker verktøykassen sin (ferdighetene sine) for å løse en problemstilling hvor de ikke får introdusert en fremgangsmåte (I3_014; I3_064). Problemløsningsoppgaver fungerer dårlig for de elevene som ikke har nok verktøy i verktøykassen sin (I1_032; I3_068). Derfor bør problemløsningsoppgaver løses gjennom samarbeidslæring, og det bør være en etablert og strukturert samarbeidskultur med et positivt læringsmiljø blant elevene (I2_014; I3_010). Når elevene er engasjerte, aktive og samarbeider i undervisningen kan det føre til gode matematiske samtaler som gjør at elevene selv kommer med ulike løsningsmetoder som diskuteres og på denne måten produserer matematiske regler og algoritmer (I2_014). Det er likevel ikke gitt at samarbeid alltid bør være løsningen.

Lærer B: *«Jeg har en klasse på tiende der det er veldig godt klassemiljø, men et helt krasjet læringsmiljø. Jeg har aldri sett et så dårlig snitt, når jeg ser på karakterene fra niende, så der (.) det blir ikke læring hvis det er samarbeid hos de, for da blir det snakking.»* (I2_018)

Læringsutbyttet er tett tilknyttet elevenes mestring (I5_018) og det er større muligheter for at elevene kjenner på mestring om læreren liker det matematiske emnet det arbeides i (I5_020).

Et annet moment som kjennetegner god matematikkundervisning, omhandler hvordan en klarer å gjøre faginnholdet relevant for elevene. Når undervisningen gjøres relevant for elevene fører dette til økt motivasjon hos elevene (I1_022). Relevans kan omhandle å vise til nytten av ferdigheter i emnet, gjennom å referere til yrkesgrupper som behøver ferdigheter i emnet (I1_018), og å kunne produsere oppgaver som elevene kan relatere til sin hverdag (I5_022). Hvilke emner det undervises i kan ha betydning for hvor enkelt det er å knytte fagstoffet til verden utenfor skolen (I5_022).

4.2.2 Lærernes reaksjoner til implementering av programmering i LK20

Det kommer frem av intervjuene at da den nye læreplanen ble iverksatt, kjente flere av lærerne på stress. Ny læreplan involverte at kompetansemålene ble endret på, så det oppstod flere krevende situasjoner. Stofftrengsel og mangel på undervisning av innhold fra tidligere år førte til at prioriteringer måtte gjøres og beslutninger ble tatt. For den ene læreren førte dette til at programmering ble bortprioritert som igjen førte til at læreren og kollegene hans begynte å kjenne på stress og frykt. Spesielt før det var avklart om det ble eksamen for 10. trinn i 2022. Han forteller at,

«... Vi stresset veldig fordi at vi tenkte, ja, vi skal ha programmering, men vi skal òg (.) det forrige trinnet hadde jo gammel og ny læreplan, sånn at vi (.) det var jo ting på åttende trinn vi ikke hadde snakket om som de måtte kunne, for eksempel potenser. Det hadde ikke vi undervist de i, så det var en del ting fra åttendeklasse i ny læreplan vi måtte ta i 10. i fjor. Så da valgte vi å ta vekk programmeringen og så fokusere på litt mer, sånn som vi vurderte da, at det var viktigere at de kunne potenser enn enda mer programmering ...». (I2_024)

En annen lærer sier at lite forkunnskaper fører til stress, «... Også når det kom, da, så ble jeg litt stressa for jeg hadde 10. klasse, også var det sånn, oi, nå må jeg ha programmering også hadde vi ikke lært noe om programmering ...» (I4_028). En tredje lærer har derimot en helt annen reaksjon, og forteller at det er godt på tide med programmering inn i skolen. «... Og når dette her med programmering begynte å komme inn skolen, så tenkte jeg, «dette var på tide» når du tenker på utviklingen i samfunnet ...» (I1_008). Det kommer også frem av intervjuene at det nok er noen fordommer som fremstiller programmering skumlere enn hva det har vist seg å være i senere tid, «... Jeg tror kanskje det er det man tenker når man snakker om

programmering. At man sitter med (.) i Matrix så (ler) er det sånn, kanskje grønne tall, da ...» (I5_058).

4.2.3 Lærernes programmeringsferdigheter og profesjonsutvikling

Deltakerne i denne studien ble spurt både om hvordan de selv opplevde sine programmeringsferdigheter, samt andre læreres programmeringsferdigheter. Nærmest samtlige av deltakerne i denne studien forteller at de selv har for lite ferdigheter i å programmere. En lærer forteller at, «... *det er ikke noe som jeg kan fra før. Så når vi kommer så langt så må jeg sette meg inn i det og jobbe meg gjennom alt dette her ...» (I1_066).* En annen synes programmeringsferdigheter er relativt. Han sier han selv ikke har noen utdannelse innenfor temaet, men samtidig at han er blant de med mest kompetanse i programmering blant kollegene (I3_026). En tredje lærer forteller at han føler programmeringsferdighetene lærerne besitter er interesseavhengig (I4_048).

Samtlige lærere i denne studien er kjente med å bruke det digitale læreverket Campus Inkrement og forteller at læreverket er en svært god støtte til undervisning av programmering. I tillegg gjør bruken av digitale læreverker at kravet til programmeringsferdigheter hos lærerne ikke nødvendigvis behøver å være så høye:

Eks. 1: «... *Men Campus er jo på en måte da et veldig godt støtteverk, for det er video (.) altså de (læreverket) tar på en måte for seg instruksjonen, så hvis lærerne følger disse leksjonene, så bygger de jo opp egen kompetanse. ...» (I3_028).*

Eks. 2: «... *For eksempel innenfor programmering så ligger det jo egne program i Campus, så vi bruker leksjonene der. Så det er veldig trygt og godt for meg som ikke er så trygg i programmering selv. Å lene meg på Campus. ...» (I2_022).*

Eks. 3: «... *Vi har tilgang til det her Campus Inkrement frem til jul, også har jeg liksom skrytt litt av det opplegget som ligger der, fordi du må egentlig ikke kunne så mye. Fordi du kan kjøre en video, og så kan de gjøre tilhørende oppgaver. Det eneste du må kunne er hvis elevene spør om hjelp, så må du jo kunne hjelpe de. ...» (I4_056).*

En kan stille seg spørsmålet om dette digitale læreverket, Campus Inkrement, kan gjøre at hvem som helst kan undervise matematikk, og at en ikke lenger behøver å være matematikklærer for å kunne utføre jobben. Til det spørsmålet svarer en av lærerne, «... *Jeg tror nok ikke hvem som*

helst kan gå inn og nødvendigvis gjøre en god time, men hvem som helst kunne gått inn og hatt en time med det ...» (I3_048). Læreren tror derfor det kreves mer enn instruksjonsferdigheter for å skape god undervisning. Å være matematikklærer gir i seg selv ikke et fortrinn når programmering skal undervises. «Jeg tror nok ikke mattelærerne har noen fordel når det kommer til kapittelet om programmering. For det er, det er så programmeringsrettet ...» (I3_050). Campus legger opp til at det er programmeringsferdigheter som skal undervises, og ikke nødvendigvis matematikkferdigheter, «... Så i begynnelsen er det veldig tydelig at det er programmeringsferdigheter de krever og der har nok ikke mattelærerne noen fordel ...» (I3_050).

Et nødvendig tema å gå inn på er lærernes ferdigheter sammenlignet med elevenes ferdigheter. En lærer sier at, «*Det vi opplever i noen fag er at elevene blir veldig fort mye flinkere enn lærerne i programmering» (I1_068) imens en annen sier at «... det er alltid noen som liksom virkelig har det, og ser dette her og suser fra. Så kan de gjerne suse fra lærere også. Et par elever i hver klasse er det ofte sånn ...» (I5_070). Ofte kan det være lettere for en lærer å akseptere at han ikke kan teknologi, heller enn å si at han ikke kan matematikk:*

«... Også tror jeg kanskje òg det (.) jeg vet ikke om det føles lettere for lærerne å innrømme at, «jeg er så gammel og dette har dere vokst opp med, så programmering» ikke sant? Der er dere flinkere enn meg, mens matematikken som, sant, du skal være en matematikklærer og si at en elev er flinkere enn deg sitter gjerne lenger inne ...» (I3_062).

Den samme læreren påpeker også at det er viktig med lærernes spesialkompetanse på matematikken og i de ulike faglige emnene, selv om elevene kanskje er dyktigere til selve programmeringen:

«... Men vi ser jo at selv om de kan bruke verktøyene, så er lærerne likevel spesialister på fagene og på emnene. Så jeg tenker selv om elevene har hatt programmering på barneskolen, så har de gjerne ikke (.) Så vil jeg tro at programmeringen og blir i større grad faglig rettet og at lærerne blir spesialister på programmering innen matematikk, selv om gjerne elevene er bedre å programmere enn lærerne, så håper jeg at i faget så trumfer likevel lærerne elevene ...» (I3_058).

Profesjonsutvikling og kursing

Det kommer frem av intervjuene med lærerne at det kreves matematiske ferdigheter i programmering. Likevel er det noen basisferdigheter i programmeringen som må ligge til grunn før en kan bruke og undervise programmering effektivt. Som nevnt i delkapittel 4.2.2 kjente flere lærere på å ha for lite ferdigheter i grunnleggende programmering, og som det ble avsluttet med i samme delkapittel kommer det også frem at det er nettopp disse grunnleggende programmeringsferdighetene som kreves i undervisningen. Profesjonsutvikling blir dermed et tema, for profesjonsutvikling omhandler blant annet hvordan lærerne lærer. Gjennom intervjuene fikk lærerne mulighet til å kommentere hvordan de foretrakk å utvikle sine programmeringsferdigheter og undervisning av dette. Som nevnt i avsnittene over kan elever raskt gå forbi lærerne, og det er støtt og stadig omtalt et behov for kompetanseheving blant lærerne. De fleste lærerne er positive til kurs for å utvikle programmeringsferdigheter (I1_152; I2_088; I5_132). Og dette til tross for ulike erfaringer med kursing:

Intervjuer *Har dere hatt kursing eller sånt i programmering?*

Lærer *(ler) Ja, men det var jo, ja, jeg kan si ja, vi har hatt kurs, ja.*

Intervjuer *Ja, fungerer det?*

Lærer *Nei.*

Når det kommer til hvordan lærerne ønsker kursing er det ulike formeninger. Læreren med utsagnet over presiserer videre at opplevelsen kan ha vært et enkelttilfelle han og kollegene opplevde som lite givende, og foreslår heller andre måter å angripe kursing av lærere på. Han sier at, «... jeg tror hvis man får satt av tid på kalenderen for å sitte å pusle med det ...» og videre at, «... det krever mest bare at du gidder å sette deg ned og fikle med det selv, altså ...» (I5_132). En annen lærer ønsker å bli kurset med ferdige opplegg som legger til rette for differensiering og ulike fremgangsmetoder på en god måte (I1_158). En av lærerne ønsker kursopplegg som lar lærerne få utforske sammen i grupper:

«... Kanskje hvis vi hadde fått tid, at da har alle mattelærerne (.) skal sitte sammen og løse noen oppgaver, eller (.) Det trengte ikke være en foreleser. Det er jo nesten kjekkere å bare prøve seg fram på programmeringen (.) Kanskje mer sånn workshop-aktig.» (I4_114)

Flere av lærerne snakker om at de har kolleger som er dyktige i programmering som de kan støtte seg til dersom de behøver hjelp (I2_042; I3_050). Det er også ønskelig å samarbeide på

tvers av skoler dersom det er dyktige fagfolk i kommunen eller på andre skoler (I1_174). En siste lærer i denne studien forteller om en erfaring med kursing fra elever som hadde gått valgfag med programmering. Dette er en erfaring han ønsker å belyse at fungerte veldig godt blant annet fordi det gav utbytte til både lærerne og elevene som fikk trening i å lære bort og stor motivasjon ved å snu om på lærer-elev-rollen. Han forteller at det forgikk som følgende,

«... det var en lærer som holdt kurset, delte ut et sånt hefte med sånn steg for steg til oss på mikrobot, og så stoppet det jo fort opp for en del av oss og da gikk 3-4 elever rundt og hjalp oss. Også bare, «hvor er jeg nå i dette heftet», også kunne de gå rundt til lærerne og si, «Nå må du lese. Hva står det?»» (I2_102).

I denne studien blir det påpekt at kursing ikke må skje uten mål. En lærer sier at det er viktig å opptre behersket og at en ikke skal stresse med å kaste mange kurs på lærerne for at de skal bli dyktige til å programmere. Han sier at, *«... Hvis på en måte en har tida til hjelp, og ikke dette her skal være klart fra august av, så tenker jeg at det absolutt er realistisk å komme der, men jeg tenker at da må vi tenke lenger tidshorisonnt ...» (I3_044).* Med å «komme der» henviser læreren til «dit vi ønsker å være», og omtaler i denne situasjonen lærernes ferdigheter i å koble matematikk og programmering. Han legger blant annet til grunn at for hvert år som går vil det utdannes nye lærere som har et ekstra år med programmering i ryggsekken fra utdanningsløpet sitt og fastslår at, *«... at det tar tid å snu en skute, og kanskje spesielt innenfor det digitale ...» (I3_044).* Dette samsvarer med en annen lærer som ikke erfarer at det har vært et felles mål å få alle lærere til å lære seg programmering (I4_092).

Det kommer frem av disse sitatene at det å sette av tid er en faktor som må prioriteres. En av lærerne sier at dette er en del av det fastslåtte arbeidet en har som lærer. Han sier, *«Lærere har aldri nok tid, men på papiret så er det satt av tid til det. Til å sette seg inn i nye (.) altså faglig utvikling» (I5_126).* LK20 introduserer mye nytt og kommer med store endringer. Dette gjør at å prioritere utvikling av programmeringsferdigheter, det å sette seg ned og fikle for å lære seg programmering på egenhånd, ikke er like lett (I4_118; I5_132). En lærer forklarer at det ikke er et hovedsatsningsområde på programmering etter at det kom ny læreplan:

«Det er en del av pakka. Vi har vært inne (.) Det er en egen kompetansepakke som Udir tilbyr, og den har vi vært inne og jobbet i. Så sånn sett, så har vi jo hatt spesifikt fokus på dette med programmering, men jeg vil ikke si at det er et av hovedsatsningsområdene våre innenfor utviklingsarbeid. Det er det definitivt ikke, men vi har vært innom det ...» (I3_032).

4.2.4 Hva er lærernes forståelse av programmering i matematikk i dag?

Det er gått snart tre år siden LK20 sin inntreden i skolen, og fornyelsen er en prosess. Lærerne i denne studien har fått bearbeidet sine reaksjoner av programmeringens inntog, de fleste har fått erfare kurs og undervisning med programmering og lærerne sitter dermed med andre holdninger til hva programmering i skolen er i dag enn hva det var i 2020. Gjennomgående er det et ønske om å være positiv til programmering i matematikk. En av lærerne forteller at det virker riktig å implementere programmering i matematikk. Likevel stiller han noen spørsmål han selv undrer over:

«... Jeg synes det passer inn, egentlig, ja. Også er det om (.) Altså de (elevene) synes det er kjekt, men hva skal de bruke det til senere? Hvorfor skal alle lære programmering? Er det nyttig for alle, eller kommer det til å bli nyttig i framtida? At alle kan dette her? Eller er det liksom for spesielt interesserte, eller sånne datafolk?» (I4_074).

Flere av lærerne ser hensikten av programmering gjennom arbeidsmarkedets behov for gode problemløsere (I1_018; I1_144; I5_062). Helt konkret er hensikten med skole bl.a. utdanning til mange ulike yrker, og programmering har et tankesett som er nyttig å ha med seg. «Grunnskolen skal utdanne sykepleiere, advokater, yrkeskriminelle, vaktmestere, rørleggere, ikke sant? Ingeniører. Så noen av dem, ja. Men tankesett tror jeg er bra å ta med seg» (I5_062). Det er stor tillit blant lærerne til at programmering har en plass i matematikkfaget. Likevel synes flere at det er vanskelig å se hvordan man kobler matematikk og programmering. En lærer blir spurt om han kunne koble programmering til andre kompetansemål enn der det er nevnt direkte programmering på ungdomsskolen, og sier at,

«... selv har jeg lyst til å være positiv og si at, «det er det». Men jeg må si at jeg klarer ikke å se her på sparket, på en måte, hvordan en kunne ha gjort det, men det tipper jeg det er kompetente folk som vil være i stand til. Som har på en måte en dypere programmeringsforståelse og kombinert med god matteforståelse som klarer å trekke sammenhenger ...» (I3_076).

Likevel begynner lærerne å se noen sammenhenger i ulike settinger (I3_076; I4_064; I5_120). En utfordring kan være at oppgaver blir veldig kunstig rettet mot det matematiske. «Nei, jeg tror det burde være fullt mulig, altså. Å kombinere det. Men det kan jo være litt søkt, selvfølgelig, i noen settinger» (I5_114). Og lærernes lave selvtillit til egen faglig kompetanse gjør det utfordrende å utvikle gode oppgaver selv. «... Så jeg ser jo at det er noen sammenheng, men

jeg føler kanskje at det er litt søkt. Men det er jo kanskje bare fordi jeg ikke kan nok selv ennå. Jeg vet ikke» (I2_058). En av hensiktene med programmering skal være effektivisering, og en av lærerne sier at dette er krevende på grunn av liten kompetanse om programmering:

«Nei, jeg tenker hvis man skal bruke sånn digitalt redskap, så bør det jo være for at det gjør det enklere, sant? ... Så det må være en hensikt med å bruke en sånn digital ressurs. At når du har lært deg å bruke den, så går det mye raskere. Og jeg føler ikke jeg har kommet så langt i programmering at jeg ser helt den nytten» (I4_080).

En lærer forteller at det er ulik forståelse i hva lærere legger i det å programmere (I5_096). Han retter kritikk mot tidligere eksamensoppgaver som skal løses i Regneark/Excel. Han forklarer at oppgavene ofte er produsert med så enkel matematikk og så små tallmengder at det ville lønne seg å gjøre beregningene for hånd. Oppgaven tester dermed elevens programforståelse heller enn deres matematiske forståelse fordi, som han illustrerer, *«... du bruker ikke en svær fôrhøster når du skal slå plenen ...» (I5_140).* Selv om læreren kritiserer oppgavetypen anerkjenner han at ikke alt som læres i skolen er av praktisk nytteverdi med én gang når han sier, *«Men vi skal vel bygge grunnlag for videre levebrød».*

Lav selvtilitt på egen kompetanse gjør at lærere ofte søker til gode og ferdigutviklede opplegg som er tilgjengelige på nett eller via læreverker. *«... og der hadde jeg nok vært relativt hjelpeløs hadde det ikke vært for læreverket Campus som har en veldig god, eller veldig grundig bolk på det som har introduksjonsbiter og som steg for steg tilhørende oppgaver» (I3_022).* En annen lærer forteller også at det utvikles flere og flere gode og komplette pakker med opplegg gjennom læreverker (I5_038).

Det er en sentral forståelse blant lærerne at programmeringen har som mål å fremme en tankeprosess, en algoritmisk tenkning (I1_022; I4_062). *«... Men det er, altså det er tankesett, ikke sant? Jeg tenker det handler om å lære tankesett. Det å liksom (.) Ja, det er liksom algoritmisk tankegang ...» (I5_056).* En av lærerne fremhever derfor at problemløsning er det store paraplybegrepet og målet, og at programmering og matematikk fungerer som ulike verktøy, eller fremgangsmåter, for å løse nettopp problemer.

«Det er vel heller for å løse problemer. Altså ikke bare matematiske problemer, men generell problemløsning, da. At man kan bryte ned i mindre bolker og løsninger kan (.) man kan komme til en løsning på et problem på forskjellige måter. (.) Å lære noe for matematikken (.) Men matematikk er jo et verktøy for å

løse problemer, så det er nok ikke for å lære matematikken. Nei. Så jeg tenker det er problemløsning generelt, da» (I5_064).

Eksamen

Selv om en del lærere hadde høye skuldre på grunn av implementeringen av programmering i LK20, peker lærerne i denne studien på særlig én ting som har vært med på å ta bort stresset de følte på i starten. Nemlig nivået på prøveeksamenene som har vært. Det har ikke blitt avholdt eksamen etter ny læreplan ble innført av ulike grunner, og for den ene læreren i denne studien påvirket dette undervisningen hans. Han forteller at han måtte gå noen runder med seg selv når han valgte å droppe å repetere emnet programmering for klassen sin. Han stilte seg spørsmålet, «... *Repeterer vi ikke programmering nå, sånn som jeg burde gjort i tiende, fordi at det ikke blir en eksamen?»* (I2_048). Fokuset på prøveeksamen påvirket hva lærerne fokuserte på i undervisningen. Flere lærere forteller at nivået på oppgavene som omhandlet programmering medførte at lærerne hadde et inntrykk av å ha tilfredsstillende ferdigheter i programmering til å kunne undervise tilstrekkelig nivå. Den ene sier, «... *Men hvis du ser på de oppgavene som er rettet mot programmering på eksamen som har vært de siste årene, så er det ganske sånn magert, da ...»* (I5_050) før han fortsetter med, «... *For da tenker jeg at man legger jo undervisningen etter hva som er forventet ...»* (I5_052). En annen lærer sier, «... *jeg ble på en måte, kall det litt lettet eller skuffet, slash skuffet, da jeg så på eksamen at det bare var en sånn lett algoritme der en skulle se om stegene stemmer ...»* (I3_050). Han utdyper videre at fokuset på eksamen ofte legger en føring for hvordan vektingen av de ulike matematiske temaene skal tolkes:

Intervjuer *Så eksamen brukes litt som en sånn målestokk for hvor langt man bør være?*

Lærer *Ja, jeg tenker det at det (.) det blir alltid (.) det blir styrende til en (.) på en måte. Vi går alltid gjennom og ser, hva er det som er fokusområder her, så, ja. I hvilken grad det er riktig å gjøre, det vet jeg ikke, men jeg tenker at det i alle fall er greit å ha et øye med på hvordan eksamen ser ut og føler på en måte at den (.) altså det legger ikke føring på undervisningen, men når vi har eksamen friskt i minne, så vet vi hva som kreves, da. Også bruker vi det. Vi øver jo også jobber gjennom eksamensoppgaver med (.) alltid med 10. klasse i løpet av høsten, og da legger det litt føringer. Så er det klart når vi da (.) Hadde halve del 2 vært programmering, så hadde vi lagt tungt fokus på det. Når det da er*

én bitte liten oppgave som ikke egentlig krever så mye programmeringsferdigheter, det er mer en algoritme, som de må forstå, så (.) ja (.) så fokuserer vi heller på de andre tingene som eksamenen har vektlagt.

Siste halvdel av sitatet forklarer lettelsen og skuffelsen læreren har beskrevet tidligere. Det kan være ulike grunner til at nivået på prøveeksamensoppgavene har vært så lave. Som nevnt tidligere sier den ene læreren, «... *det tar tid å snu en skute, og kanskje spesielt innenfor det digitale*» og at det er bra og nødvendig å ta tiden til hjelp (I3_044). Dette indikerer at det lave nivået kan henge sammen med at det er så kort tid siden innføringen av ny læreplan, og prøveeksamennivået er et synlig bevis på at implementeringen skjer gradvis, også i vurderingen. Læreren presiserer for øvrig at det er fornuftig å gå rolig til verks. En annen lærer sier at det lave nivået kan være for å ikke skremme matematikklærerne fra matematikkfaget, «... *Om det er for å (.) jeg vet ikke, jeg (.) ikke skremme bort lærerne eller for å liksom senke teknologiangst eller programmeringsangsten, da, hos lærere ...*» (I5_050). Han forteller også at når lista blir lagt såpass lavt, så inspirerer det ikke til videre innsats i programmering (I5_138). Det blir uansett erfart at Campus Inkrement er en god støtte som går forbi hva som forventes av ferdigheter, og at lærerne har kompetanse nok til å undervise i temaet:

«... Men per nå, så tror jeg ikke det forventes, ut ifra eksamensoppgavene, at nivået skal være så høyt som Campus (.) altså på det nivået Campus legger seg på. Så i forhold til der eksamen er, så tror jeg lærerne våre er langt innenfor på en måte» (I3_028).

Forventningen om høye krav på eksamen skapte unødvendig stress for flere lærere som med tiden har sett at kravene til kunnskap i programmering ikke er høyere enn at de har kontroll på undervisning av emnet. En lærer forteller om prøveeksamen at, «... *Det er jo bare å kjenne til det der (.) tankesettet, da. Og det er ikke noe du trenger å bruke PC for, for så vidt ...*» (I5_140).

Dybdelæring

Et annet moment som kommer inn med ny læreplan er dybdelæring. Lærerne i denne studien ble spurt om implementeringen av dybdelæring kunne øke mulighetene for at lærere som er ekstra trygge i å programmere kan undervise enda mer i emnet, samt at lærere som ikke er like trygge på emnet kan droppe å undervise emnet så grundig fordi de kan gå dypere i andre emner. En lærer forteller at han syntes programmering har blitt noe urettferdig bortprioritert dersom lærere ikke har hatt tilgjengelig ferdige programmeringsopplegg når han sier at,

«Så jeg tror programmeringen er ofte blitt litt sånn stemoderlig behandlet i mat (.) selve matematikkfaget, da. For det er nytt for lærer. Sikkert nyere for lærer enn det er for elever. Altså som kvier seg for å hoppe i det, da. Hvis man ikke får en litt sånn ferdig (.) altså ferdigpakke (.) eller ferdig pakke presentert.» (I5_038).

En annen lærer er tydelig på at han synes programmeringsundervisningen er blitt påvirket av lærerens interesse for emnet, og at digitale læreverk, som Campus Inkrement som kan gi leksjoner, styrker mulighet for at elevene har like forutsetninger i klasserommet:

Lærer ... føler jeg at jeg har på en måte fått gjort ganske mye. Men hvis vi vil at det skal være likt, på en måte, for alle, uansett hvilken mattelærer de har eller, så burde de jo ha vært prioritert mer. Sånn, noen får mye, noen får ingenting kanskje.

Intervjuer For det går på interessen til mattelæreren nesten? Om man setter av mye tid til det, eller ikke?

Lærer Ja, for noen føler jeg er sånn at vi tar litt programmering, for det må vi jo, for det står jo i læreplanen, også blir det å finne et eller annet sånt ferdig opplegg på (ler). Også får de prøve seg litt på det, også ...

Intervjuer Mer for å forsvare det, enn (.)

Lærer ... Huket av det kompetansemålet.

Andre lærere tror at dybdelæring kan være et argument for bortprioritering, men at det også er andre faktorer som spiller inn på at noen lærere skyver fra seg å undervise emnet grundig. Én peker på usikkerheten til egne ferdigheter blant lærerne (I1_210), en annen peker på at andre fag også skal undervise programmering (I2_046) og en siste peker på den lave vanskelighetsgraden på eksamener (I5_084). Enn så lenge kommer det frem av intervjuene at forventningene til elevens forkunnskaper i programmering er lik null fordi det er så ulikt hva de har med seg, både for elever som skal opp i ungdomsskolen, og som skal opp i videregående skole (I2_070). En av lærerne forventer at dette endres til det helt motsatte i løpet av få år. Han sier,

«... Men det blir jo kjempekjekt og begynne med elever som har med seg dette i ryggsekken i fra barneskolen, og da tenker jeg, jeg tror ikke det blir en aktuell problemstilling engang, at en skal vekt... altså, da blir det dybdelæring i programmering vi holder på med her ...» (I2_050).

4.2.5 Hvordan undervises programmering i matematikk?

Gjennom intervjuene med lærerne i denne studien kommer det frem at programmering sees på som et eget emne i matematikk. Flere av lærerne sier de benytter læreverket Campus Inkrement i sin undervisning og at det kun benyttes blokkprogrammering. I tillegg merker studiens lærere at programmering trigger motivasjonen hos elevene.

Programmering som eget emne

Det har vært komplisert med programmeringsundervisningen frem til i dag. En lærer sier at kombinasjonen av arbeid i to ulike læreplaner gjorde at han ikke har klart å implementere det skikkelig godt enda (I2_024), en annen sier at elevene kommer inn i grunnskolen med såpass dårlige grunnleggende matematikkferdigheter at arbeid med dette må prioriteres (I1_032) imens en siste sier at undervisning av programmering er avhengig av om læreren er interessert eller ikke (I4_048). Det er likevel blitt undervist i programmering av flere av denne studiens lærere. Det kommer frem av intervjuene at de fleste av lærerne ser på programmering som et helt selvstendig emne (I1_198; I2_056; I4_100), og som lærerne har vansker med å se i sammenheng med andre emner (I1_190; I2_082; I3_076). Dette skjer blant annet fordi både elevene og lærerne fortsatt arbeider med å lære grunnleggende ferdigheter i programmering. En lærer forteller at de i en ungdomsskoleklasse begynte med oppgaver som var rettet mot 5. trinn fordi elevene hadde lite med seg fra barneskolen (I2_024). Den samme læreren peker på at han var utrygg på egne programmeringsferdigheter som begrunnelse for hvorfor han ikke koblet programmering til andre emner (I2_028). En annen lærer peker på at programmering har blitt gjennomgått raskt som eget emne kun for at elevene skulle være «safe» dersom de kom opp til eksamen (I5_050), og dette til tross for at flere lærere i denne studien sier de ønsker at programmering helst skal benyttes som et verktøy for å lære seg matematikk (I1_144; I5_094). Et problem med oppgaver som kombinerer programmering og matematikk er at oppgavene ofte blir formulert på en kunstig måte som virker søkt mot å lete etter matematikk i større problemer (I2_058; I5_092; I5_114). Det er en forståelse blant noen av lærerne i denne studien at noen emner er enklere å koble programmering til enn andre (I3_010; I5_096).

Campus Inkrement

Flere lærere i denne studien peker på at de har kunnet støtte seg til det digitale læreverket Campus Inkrement (I2_022; I3_018; I4_056). Dette er også med på å gjøre at programmering blir undervist som eget emne. Det oppleves at læreverket går grundig til verks i forhold til nivået på eksamen (I2_026; I3_022) og at lærerne selv utvikler grunnleggende programmeringsferdigheter underveis gjennom undervisningen (I2_078; I3_028). Gjennom det digitale

læreverket Campus tilbyr undervises faginnhold gjennom videosnutter. Dette kan medføre både gode og mindre gode konsekvenser. De fleste lærerne i denne studien liker likevel å undervise med dette. En lærer forteller at læreverket gir elevene mulighet til å se leksjonen så mange ganger de behøver og at elevene også kan se leksjoner hjemme, for så å komme og få hjelp og gjøre oppgaver på skolen (I2_012). Det er store forskjeller i programmeringsferdigheter blant elevene, blant annet fordi noen elever har gått valgfag som inkluderer programmering (I2_028). Campus gir rom for å arbeide på differensiert nivå med mulighet for blant annet off-piste-oppgaver som er ekstra utfordrende oppgaver som går på tvers av emner på skolen. En lærer sier at videoleksjonene gjør at han slipper å ha den dype kunnskapen av programmering (I4_048) og en siste lærer sier elevene får mye tid til å arbeide med differensierte oppgaver gjennom læreverket, og at elevene dermed får løst flere oppgaver på sitt nivå enn de ville gjort ellers (I3_062). En av lærerne forklarer dette om utviklingen av vanskelighetsgrad i Campus,

«... begynte med sykebil også når du da kommer nedover til de nederste vanskeligste oppgavene, og der var det mange elever som ikke kom, så videreutviklet de det til at da kan du lage blokker som løser stykker for deg. Regne ut arealet av figurer eller omkrets ...» (I3_040).

I tillegg gjør det digitale læreverket at alle klasserom får den samme leksjonen, og lærernes begrensede ferdigheter forhindrer dermed ikke elevenes mulighet for læringsutbytte. Det blir lik undervisning, uavhengig av hvilken lærer du har (I4_108). En annen positiv side ved digitale læreverk, er at de kan oppdateres kontinuerlig, i motsetning til lærebøker som er trykt på papir og som må fornyes i sin helhet (I3_046). Dette blir spesielt nyttig fremover i og med at fokuset på matematikkundervisningen stadig er i endring, og fordi matematikkundervisningen vil være helt annerledes om fem år (I3_046). Campus er ikke nødvendigvis så bra til å legge opp til utforskende matematikkundervisning (I3_066). Dette fordi det er raskt for å havne i et mønster hvor elevene ser leksjonene på video og arbeider med oppgaver selvstendig etterpå.

Blokkprogrammering

Lærerne i denne studien forklarer videre at de benytter blokkprogrammering. En lærer forteller at han syntes blokkprogrammering kan gjøres både avansert og heftig nok for nivået på grunnskolen (I3_022). Igjen så brukes blokkprogrammering mye i undervisningen fordi flere av lærerne i denne studien benytter Campus som bygger mye av leksjonene sine på blokkprogrammering. En av lærerne som har brukt andre metoder enn undervisning gjennom Campus sier at han har benyttet Scratch, men at det kan bli kjedelig å sitte foran skjerm og programmere «den katta» i lengden (I5_038). Han legger derfor heller opp til mer praktisk arbeid for elevene

og foretrekker blant annet Lego Mindstorms fordi da får elevene programmert noe håndfast (I5_042). Han forteller at det er mye positivt ved at, «... du får liksom et fast bevis på at programmet fungerer. Ikke bare det at (.) Et eller annet som skjer på skjermen, da. Jeg liker å få liksom noe praktisk ut av det» (I5_042). Han forteller også at blokkprogrammering er avansert nok fordi målet med programmeringen skal være å hjelpe elevene mot tankemåten programmering fremmer (I5_062). En annen lærer forteller at han har benyttet unplugged-aktiviteter. Han er noe usikker på om dette går som programmering:

«Også har vi hatt og noe, jeg vet ikke om det var programmering, men litt sånn på papir, at hvis (.) altså et sånt ark med forskjellige koder på, hvis sola er gul, skal den krysses over, så skal de på en måte få fram en kode med tall på, og da er det og litt sånn du må på en måte kunne forstå disse her instruksjonene eller kommandoene» (I4_070).

Motivasjon

Alle lærerne opplever at programmering trigger noe som skaper motivasjon og engasjement blant elevene. En måte å skape motivasjon på kan være ved at læreren slipper opp på kontrollen i klasserommet og opplyser om at han ikke har fasiten på oppgaver og problemer det arbeides med (I3_018). En lærer sier at undervisning med programmering kan undervises med konkurranser som motiverer: «... Så tror jeg det er bra med litt konkurranser og sånt, og få de litt hekta på at dette kan være noe spennende og noe kult ...» (I2_046). En annen lærer sier at motivasjonen kan komme av at man oppfatter at man gjør noe annet enn bare det faglige: «... Jeg tror det heller blir en sånn gøyale greie. En gøyale event at man sitter og trykker på noe som blinker og piper, også klarer man kanskje ikke å koble det til tema innen matematikk, da.» (I5_086). Det kommer frem at større prosjekter og konkurranser kan føre til at elever også engasjerer seg etter skoletid, noe som er en stor fordel for elevenes læring og som samtidig ikke tar av tiden til matematikkundervisningen (I2_046). En lærer sier at elevene ikke arbeider strategisk med matematikken i programmeringen, men at noen syntes det kan være artig å fikle litt likevel. Han forklarer,

«... Men i den programmeringen vi har nå, så er det jo masse geometriske figurer og mønster, så skal de få den her skilpadden til å gå og lage dette sporet, også blir de veldig fornøyde når de ser at de treffer. Også ser de, «Å, nei. Der flyttet den seg litt for lite. OK, da må den liksom, kanskje 80 skritt frem istedenfor 70». Også er de veldig fornøyde når det funker. Så kan de skifte farger også synes de det er kult. Også blir det (.) Ja. Jeg tror det er bare litt sånn at det er

nytt og spennende, enda. Kanskje. Men jeg vet ikke om de hadde tenkt at dette skal vi bruke for å løse matteoppgaver.» (I4_126).

Et interessant spørsmål er om programmeringen kan føre til at elever som ikke nødvendigvis er mest motivert for matematikk kan få økt motivasjon og prestasjon gjennom programmerings inntog i matematikk. Lærerne er uenige om dette. Imens én lærer sier at, «*det er noen som virkelig stråler ..., som gjerne ikke er så gode ellers, også ... er de skikkelig gode i programmering, så går de andre og spør, «kan du hjelpe med det»» (I4_066)* sier en annen at han sjeldent opplever gyldne øyeblikk med umotiverte elever og at, «*de har fjorten timer skjermtid resten av døgnet, så å gi dem en datamaskin der, det hjelper ikke altså. Det er ikke noe sånn at de blomstrer» (I5_074).* Han forteller at han generelt ser et mønster av at jentene suser i fra guttene i alle fag (I5_070) og programmering ikke vil hjelpe guttene som allerede har mye skjermtid på fritiden. En annen lærer opplever også å ha noen gutter som er mindre motiverte i matematikk. Han tror derimot at programmering kan gjøre dem motiverte for arbeid (I1_102). Han sier,

«... men jeg tenker akkurat i programmering, du tenker de der, særlig de der svake guttene kan bli motivert i matematikk når vi holder på med programmering. For akkurat det emnet er de interessert i, og som de kommer til å synes er ganske kjekt. Og hvis de får det til så kommer det til å øke motivasjonen blant de elevene i matematikk. ...» (I1_102)

4.2.6 Hvordan lærerne mener at programmering bør undervises

Lærerne forklarer hvordan de selv praktiserer undervisning med programmering i dag. Likevel er det ikke gitt at slik det praktiseres er slik de ønsker å undervise programmeringen. Lærerne ble derfor spurt om hvordan de ser for seg at matematikkundervisning med programmering bør utvikles i løpet av de neste fem årene og hvordan de ønsker å undervise programmering i matematikk. Svarene som kommer frem er at det er stor enighet om at programmeringen skal bringe inn et fysisk og konkret aspekt til matematikken. En av lærerne sier:

«... Så gjør det om til noe praktisk. DET tror jeg er viktig altså. Koble til sensorer. Koble til motorer (.) Eller servoer. Mange sensorer. At det ikke kun er noe som foregår på skjerm, for det (.) Det høres gammelt ut, men altså (.) De har nok tid med ting og tang, og alt det som de blir presentert for på skjerm utenfor skolen er mye mer fascinerende og flashy enn det vi klarer å møte dem

med. Men det der praktiske, DET! Der kan vi på en måte presentere noe nytt for dem, da.» (I5_150).

En annen lærer forteller:

«Jeg tenker jo at vi skulle hatt noe mer enn at de bare sitter digitalt med Ipad og jobbe på den, men at de kunne holdt på med mikrobit og fått til noen biler som kjørte eller disse her ballene (.) jeg vet ikke hva de heter engang. Men at det (.) at vi på en måte kunne konkretisert det mer og brukt ting som de fikk til å bevege seg. ... De elsker konkurranse, sånn at jeg skulle ønske at vi, ja, både at det er mer taktilt, at det var ting vi fikk til å bevege seg, som jeg sa, også (.) gjerne kjørt litt sånn at vi hadde konkurranser mot andre skoler også gjort litt ut av det. Det hadde vært kjekt.» (I2_074).

En siste lærer sier:

«... Men det jeg kanskje skulle ønske er at vi hadde hatt mer konkrete ting. Det er liksom, ja, vi programmerer litt på en skjerm, men det kjekke er når du kan få en robot til å gjøre noe, eller du kan få liksom noe til å bevege seg sånn som du har programmert. Det er jo mye kjekkere.» (I4_040).

Alle lærerne peker på at programmering gir mulighet for å gjøre noe mer praktisk i matematikkundervisningen. Selv om mye programmering skjer på skjerm, så sier lærerne at elevene må få et konkret og praktisk element å se på for å vurdere om de har løst problemet. Det kommer også et forslag om å benytte seg av konkurranser for å skape engasjement blant elevene. Til dette kan Lego fungere som en god metode (I5_100), da det allerede finnes konkurransebaserte oppgaver produsert av First Lego League (I5_076). Dette vil hjelpe elever å utvikle STEM-ferdigheter (I5_100). Utfordringen med dette vil være at det er svært tidkrevende, både for elever, men også for veiledere (I5_078). En annen utfordring læreren trekker frem er at elever ikke er like interesserte i Lego som de har vært tidligere. Han sier at, *«ungene har liksom ikke den der tekniske innsikten og LEGO-bakgrunnen, da. De bygger ikke LEGO, de kan Minecraft, men de bygger ikke LEGO lengre, da.» (I5_078).* En annen lærer viser til at det finnes andre mulige visuelle gjenstander å programmere også: *«... vi har noe sånn som heter Sphero-kuler her, så de kan programmeres sånn at de beveger seg, og tegner mønster og så videre, og det tenker jeg definitivt at en kan bruke programmering til. ...» (I3_076).*

En lærer sier det er viktig og arbeide på et nivå som lærerne henger med på. Derfor er det fornuftig å legge opp til undervisning med blokkprogrammering (I5_124). Han foreslår også at en ser på undervisning med programmering som undervisning av problemløsning, også bør heller matematikk og programmering være verktøy en kan benytte for å løse problemene (I5_064). Læreren sier at gode oppgaver er problemer som kan løses på ulikt vis, slik at elever må bearbeide løsningen sin ved å kutte ned og bruke færrest mulig steg etter å ha løst oppgaven én gang (I5_136).

4.2.7 Hva sier lærerne om den endrede lærerrollen i matematikkundervisningen med programmering?

Som i alle temaer må lærere være engasjerte for å være dyktige innenfor undervisning av et emne (I5_040). Med programmering inn i matematikkundervisningen er det implementert et emne som de fleste lærere ikke har utdanning i, og som igjen gjør det mer krevende å undervise bort (I4_028). I tillegg krever programmering en endring av undervisningsmetode slik at elevene i større grad skal få erfare og utforske emnet. Dette medfører at lærerne i denne studien ser at rollen deres som lærer er i endring:

«... jeg tenker lærerrollen har endret seg. Det har den. Også må vi tørre å stå i det. At vi har elever som kan ha kommet lenger enn oss. Samtidig så må vi ha kompetansen til å se hvordan de (.) selv de flinkeste elevene kan utvikle seg videre.» (I3_060)

I undervisning med programmering er det flere lærere som omtaler problemløsning som et sentralt moment (I1_020; I3_078; I5_062). En av lærerne sier at når en underviser matematikk med et fokus på problemløsning, er det en nødvendighet å slippe på den tradisjonelle rollen sin hvor det forventes at læreren skal ha fasiten hele tiden (I3_018). Han forteller at han opplever engasjement når han kan fortelle elevene at han ikke sitter med fasiten hele tiden:

«... for ofte så tenker jo elevene at vi sitter alltid med fasiten og er der med pekefingeren og nå er det feil og nå er det rett og så videre. Og det opplever jeg at de (.) altså da synes de det er spennende på en måte, at lærerne ikke heller vet. Og at vi på en måte (.) at vi er likestilte da, på en måte.» (I3_018).

Som nevnt tidligere er mange elever dyktige dataforbrukere, og ofte kan det teknologiske aspektet gjøre at elevene ligger langt fremme på området (I3_058). Noen ganger er elevene forbi læreren også (I5_070) og dette er med på å endre lærerrollen. Som nevnt før forteller en av lærerne at han tror noen lærere kan ha lettere for å akseptere å ligge bak elever på grunn av

det digitale (I3_062). Han sier videre at dette kan brukes positivt, spesielt sett i lys av at læreren bør gi fra seg den tradisjonelle rollen som «fasitmaskin». På spørsmål om hvordan læreren bør operere i disse situasjonene trekker læreren sammenligning til idrettsverdenen hvor treneren ikke er bedre enn utøveren, men likevel besitter kunnskap for videreutvikling. Han sier,

«En parallell er at, jeg vet ikke om det er en god parallell, men i idretten, så er det ikke sånn at treneren er bedre enn utøveren, men han klarer likevel å si hva utøveren skal videreutvikle for å bli bedre. Og litt der tenker jeg vi er i matematikk òg, og programmering.» (I3_058)

Bruken av Campus Inkrement som læreverk påvirker også rollen læreren har i matematikkundervisningen. Når leksjonene blir gjennomgått av læreverket får elevene en god innføring i fagstoffet (I3_048). Innføring av fagstoffet er ikke lengre en del av lærerrollen med Campus, så oppgaven til lærerne blir å vurdere når elevene kan ha nytte av en mer utforskende undervisning (I3_066), veilede (I1_038; I5_078) og presentere oppgaver og bistå inn i arbeidet med oppgaver og problem (I3_058; I5_136). Det å legge opp til gode problemløsningsoppgaver og gode undervisningstimer sier en lærer at blir enklere med de ulike pakkene som digitale læreverk tilbyr (I5_066). Som nevnt tidligere forteller en annen lærer at han likevel forventer at lærerne har en sentral rolle med å bistå elevene i å se koblingen til det matematikkfaglige innholdet (I3_058).

5 DISKUSJON

I analysene av resultatene ble det gjort flere interessante funn angående programmering som en del av matematikkfaget. I diskusjonskapittelet vil resultatene bli drøftet i lys av tidligere forskning og teori. Diskusjonskapittelet består av fem deler. Den første delen tar for seg fellestrekk og forskjeller mellom programmering og matematikk som er nyttig når en skal vurdere hvordan en bør drive matematikkundervisning med programmering. Delkapittel 5.2 belyser hvordan programmering kan fremme kvaliteten på undervisningen. Videre, i delkapittel 5.3, ser vi på hva som blir ansett som god undervisning i matematikk og hvilke ferdigheter som forventes av lærerne for dette. Delkapittel 5.4 drøfter hvordan programmering gjør at lærerrollen er i endring. I denne delen blir det naturlig å så nærmere på Campus Inkrement som digitalt læreverk fordi dette er en sentral del av undervisningen til flere av intervjupersonene i denne studien. Siste delkapittel omhandler hvordan matematikkundervisning med programmering bør gjennomføres ifølge lærerne i denne studien.

5.1 Om programmering og matematikk

I teorikapittelet ser vi at Koehler og Mishra (2009) sier at ny teknologi er så avansert fordi den er anvendelig, ustabil og ugjennomsiktig. Dette medfører at lærere sannsynligvis vil føle at de har for liten kompetanse i faget, selv når de har utviklet enda dypere ferdigheter innenfor programmering. Forskjellen på matematikk og ny teknologi er at matematikken omhandler noe som er synlig konstant. Den er bygd opp av entydige aksiomer, grunnleggende regler som gjennom deduksjon, altså bevisføring, danner grunnlaget for å bygge opp logiske slutninger. De logiske slutningene er gjeldende regler som ikke er ustabile og ei heller usynlige. Når det gjelder programmering er de elementære, grunnleggende reglene som gjelder skrevet på et lavnivåspråk som et menneske aldri vil klare å tolke. Selv høynivåspråk, som i det minste kan være leselige, er bygd opp av lavnivåspråk. Når et menneske ikke kan følge de logiske slutningene tilbake til de grunnleggende og fundamentale aksiomene som gjelder, vil de ikke kunne forstå oppbyggingen som hvis de hadde kunnet dette. Å forstå de grunnleggende ideene bak programmering er ikke som å forstå de grunnleggende ideene bak matematikk. Kunnskap om grunnleggende ideer er blant de viktige kompetansene Ma (2020) sier at kjennetegner dyktige matematikklærere. Ulik «grammatikk» i program gjør at programmeringskompetanse derfor ofte er knyttet til det ene programmet som benyttes. Slik en av lærerne retter kritikk mot eksamensoppgaver som skal løses i Excel, ville de samme problemstillinger oppstå med programmering. Skal elevene få programforståelse eller lære seg å bruke programmering fordi det er nyttig? Alt dette medfører at programmeringsferdigheter alene ikke kan være målet i

skolen, verken for lærere eller elever. En må heller søke å oppnå tankesettet som programmering krever. Samtidig er programmeringsferdigheter forutsetninger for arbeid med utvikling av tankesettet. En skal derfor ikke unngå å jobbe med programmeringsferdigheter, men en bør vekte algoritmisk tankegang tyngre. Alt dette viser at den forventede førforståelsen min om at programmering er krevende å undervise bort er berettiget.

5.1.1 Algoritmisk tenkning og problemløsning

Algoritmisk tenkning er tankesettet som behøves for å mestre programmering. Det kan forstås i ulike retninger (Dolonen et al., 2019) og innebærer svært mange ulike momenter som abstraksjon, dekomponering, logikk og feilsøking m.m. (Grover & Pea, 2013). Algoritmisk tenkning omhandler noe mer enn programmering, og går ut på å anvende basisferdigheter. Å mestre bruk av basisferdigheter beskrives som helt nødvendig for å klare seg i det 21. århundre. Bocconi et al. (2018) peker spesielt på problemløsning, logisk tenkning og kreativitet som anvendte ferdigheter programmering fremmer. Grunnskolen skal gi elevene et kunnskapsfundament uavhengig av yrkesretningen de velger og problemløsning ansees som én av flere anvendte ferdigheter. Det omhandler blant annet å være tilpasningsdyktig og løsningsorientert, og det kommer frem av intervjuene at en anvendt ferdighet som er nødvendig på så mange områder passer godt inn i skolen. Derfor bør en arbeide mot å utvikle dette tankesettet som både problemløsning og algoritmisk tenkning bygger opp om. Gjennom intervjuene i denne studien kommer det frem at problemløsning kan sees på som det store paraplybegrepet, altså det overordnede målet elevene skal oppnå. Med dette som utgangspunkt blir både matematikk og programmering ansett som ulike verktøy for å oppnå løsningen på ulike problemer. Når elever blir utfordret på å bruke «verktøykassen sin» for å løse problemløsningsoppgaver, kan dette både innebære bruk av tradisjonell matematikk og av programmering.

5.1.2 Motivasjon og engasjement

Forsström og Kaufmann (2018) undersøkte blant annet hvordan samarbeidslæring i programmering fremmer elevers motivasjon for å lære matematikk. De så at programmering var motiverende for mange elever. Likevel kunne de ikke generalisere og si at programmering øker motivasjon hos alle elever. Begrunnelsen var at rammene ble endret i så stor grad fra den vanlige undervisningen i tillegg til at det ikke lå til grunn at motivasjonen var langvarig. I Norge belyser flere lærere at elevengasjement og respons er blant de viktigste faktorene for å gjennomføre god matematikkundervisning med effektivt læringsutbytte (Fauskanger, 2016). Lærerne i denne studien legger vekt på lærerens oppgave med å skape elevrespons og aktivitet, noe som vinkler Fauskanger (2016) sine funn mot faktorer som lærerne kan påvirke. Lærerne i

denne studien mener programmering fremmer elevenes kognitive engasjement med programmering i matematikk. De sier at undervisning med programmering i matematikk har et større potensiale for å gjøre noe annet enn slik matematikk tradisjonelt undervises. I tillegg kommer det frem av intervjuene at lærerne opplever at programmering gir elevene større motivasjon og et større engasjement. Lærerne foreslår eksempler som større prosjekter og konkurranser som de tenker gir gode muligheter for at en opplever elevene mer motiverte for programmering enn hva gjelder matematikk ellers. Likevel er det viktig å ha med seg at målet med matematikk ikke skal være motivasjon alene, men at det ofte er en kobling mellom motivasjon, prestasjon og mestring.

Forsström og Kaufmann (2018) forteller at elever som vanligvis blir kategorisert med lav evne i matematikk kan gjennom en endret rolle i samarbeidslæringen i klasserommet komme opp med sofistikerte matematiske ideer. Det er uenighet blant lærerne i denne studien om programmering motiverer elever som vanligvis har utfordringer med matematikk. En lærer belyser at interessen for datamaskin kan motivere. En annen lærer påpeker at disse elevene har så mye skjermtid på fritiden at mer skjermtid ikke vil fungerer som motivasjon. Flere av lærerne i denne studien sier at «gyldne øyeblikk» der elever med utfordringer i matematikk glimter til under programmering heller er unntakstilfeller enn noe de ser regelmessig. En lærer forteller at målet for programmering i skolen på ingen måte er å forsøke å være mer stimulerende enn spillene disse elevene bedriver på fritiden sin. Som Forsström og Kaufmann (2018) beskriver så er det rolleendringen blant elevene i klasserommet, og ikke interessefeltet «datamaskin», som skal skape motivasjon. Lærerne må derfor se kvalitetene elevene besitter og være behjelpelig med å legge opp til at elevene kan få benytte disse i et samarbeid, for eksempel ved at dyktige databrukere kan bidra med teknologisk kunnskap inn i problemløsningen. Programmering gir gode muligheter for å drive med elevsamarbeid i klasserommet. Læreren legger til rette for et positivt læringsmiljø i samarbeid og elevene må selv ta initiativ til å bidra inn i samarbeidet.

5.2 Programmering kan fremme kvalitet i undervisningen

Gjennom denne studien kommer det frem at særlig to av dimensjonene i MKU-rammeverket kan fremmes av programmering. Disse er «opptak og tilbakemelding» og «Elevdeltakelse i matematiske resonnement og meningsskaping». Dimensjonen «matematisk rikdom» virker også til å bli påvirket positivt ifølge lærerne i denne studien, men i så liten grad at det vil være vanskelig å argumentere for at programmering har stor innflytelse. Det er stor uenighet om

dimensjonen «feil og unøyaktighet» styrkes eller svekkes blant lærerne i denne studien. Det gjør det vanskelig å generalisere og det kan være nyttig å undersøke dette nærmere.

5.2.1 Elevdeltakelse i matematisk resonnement og meningskaping

Dimensjonen lærerne i denne studien mener programmering i størst grad fremmer er «Elevdeltakelse i matematisk resonnement og meningskaping». Dimensjonen tar for seg om elevene er involvert i kognitivt aktiverende arbeid i klasserommet.

Lærerne i denne studien peker på at programmering fremmer det kognitive engasjementet i klasserommet. De sier at programmering er kognitivt krevende, og at elevene må beregne og tenke flere steg fremover når de lager koder. Elevene må blant annet kjenne til kommandoer og gjøre logiske resonnement. Videre anvender de ferdighetene sine og produserer det oppgaven etterspør. Alt dette er ferdigheter matematisk kompetanse består av (Nosrati & Wæge, 2015). «Mental agility» består av ferdigheter som kan sees i sammenheng med ferdighetene den algoritmiske tenkeren består av. Spesielt kan reduksjon sammenlignes med abstraksjon, reversibilitet med logikk og overføring av gamle strategier med å se mønstre. Som forklart tidligere sier Bruder (Liljedahl et al., 2016) at dette en ferdighet som gode problemløsere må ha. Fordi programmering, algoritmisk tenkning, «mental agility» og matematisk kompetanse krever mange like ferdigheter, kan en tenke seg at arbeid med én av disse komponentene utvikler alle komponentene. Bergqvist (2022) har vi også sett at foreslår å arbeide med nettopp problemløsning for å utvikle programmeringsferdigheter. Dette er spesielt interessant i lys av Kaufmann og Stenseth (2021) sin advarsel mot å anta at en automatisk tilegner seg problemløsningsferdigheter ved å drive med programmering. Advarselen kan komme av flere grunner. Én grunn kan være at komponentene omfatter områder som de andre komponentene ikke inneholder. En annen grunn kan omhandle fokuset i arbeidet. Fra resultatene av spørreundersøkelsen kan vi lese at en lærer erfarer at elevene ikke automatisk funderer over matematiske sider ved problem, men heller ønsker å haste seg videre til neste oppgave. Det er med dette grunnlag for å indikere at det er mulig å arbeide med programmering for å bli en dyktig problemløser, men det er en viktig oppgave for matematikklærere å hjelpe elevene til å se matematikk i problemene som undersøkes når det gjennomføres matematikkundervisning med programmering. Fordi problemløsningsoppgaver kan benyttes for å utvikle relasjonell forståelse (Hiebert og Grouws, 2007) er det potensiale for at programmering kan benyttes som et verktøy i problemløsningsoppgaver, og dermed, med god veiledning fra læreren og gode problemløsningsoppgaver, være med på å utvikle relasjonell forståelse i matematikk.

Som nevnt i teorikapittelet er deltakelse sentralt for læring. Engasjement er en måte å være deltakende i klasserommet på. I tillegg til å ha et potensiale for å utvikle problemløsningsferdigheter og matematisk kompetanse sier lærerne i denne studien at programmering kan bidra til å fremme elevdeltakelse og dermed også å etablere et positivt læringsmiljø. Lærerne i denne studien forteller at programmering kan legge til rette for at elevene får mulighet til å forklare de ulike strategiene de benytter. Med en lærer som er behjelpelig med å fremme en delingskultur av løsningsstrategier i klasserommet vil det kunne utvikles et positivt klasseromsmiljø med fokus på læring. Et slikt klassemiljø omtales av Mason (2016) som en «conjecturing atmosphere» og sentrale kjennetegn er at det hele tiden er åpent for forslag til strategier, og spesielt nye og effektive løsninger krediteres. Arbeidet med utvikling av en slik delingskultur er allerede belyst av Forsström og Kaufmann (2018), og de omtaler det som en del av den endrede lærerrollen. Programmering gir mulighet for utviklingsarbeid med lærings- og klassemiljøet. En av lærerne i denne studien presiserer likevel at samarbeid ikke bør skje uten mål og mening, da noen klasser kan ha et veldig godt klassemiljø, men samtidig ha et dårlig læringsmiljø der arbeid i timen kan være lite produktivt.

5.2.2 Matematikkfaglig opptak og tilbakemelding

En forutsetning for å utvikle seg effektivt er gode tilbakemeldinger. Matematikkfaglig opptak og tilbakemelding omhandler i hvilken grad læreren forstår og responderer på elevenes matematiske forklaringer. At læreren kan forstå elevenes utfordringer, misoppfatninger og vansker er viktig for å kunne hjelpe elevene med å utvikle seg. Det kommer frem i resultatkapittelet at lærerne mener at programmering fremmer dimensjonen om matematikkfaglig opptak og tilbakemelding. Fra lærernes forklaringer kan en skille deres tolkning av dimensjonen i to. Opptak og tilbakemelding til hele klassen og individuelt. Gjennom programmering kan læreren visuelt fremstille situasjoner for større forståelse av matematikk til en hel klasse. Å ha kunnskap om hvordan man kan fremstille ulike matematiske situasjoner er av Ball et al. (2008) omtalt som spesialisert fagkunnskap. En lærer forteller at programmering kan gi elever visuelle bilder gjennom glidere av funksjoner i Geogebra eller formler i Regneark. Dette er med på å avdekke kunnskapshull hos elever, og fremmer dermed den matematiske kvaliteten i undervisningen. En annen lærer tolker dimensjonen mer individuelt. Han sier, «*I programmering vil mange ganger programmet gi beskjed når det er feil og de må prøve en annen fremgangsmåte*». Lærere som skal kunne mestre dimensjonen med tilbakemelding må være dyktige på å sette seg inn i feilmeldinger program gir, slik at de kan hjelpe elevene med å rette opp i programfeil. Over tid vil læreren se hvilke feil som går

igjen hos elever og vil måtte gi gode tilbakemeldinger på dette. På dette viset utvikler læreren kunnskapen Ball et al. (2008) omtaler som kunnskap om faglig innhold og elever. Programmering kan gi læreren flere muligheter til å visualisere matematiske situasjoner for elever, og i tillegg gi lærere mulighet til å få innsikt i hvor elever har kunnskapshull. På dette grunnlaget kan lærere gi gode tilbakemeldinger for å hjelpe elevers utvikling.

5.3 Lærerferdigheter for god undervisning

Matematisk kunnskap står helt sentralt i all teori som omtaler hva som kreves for god matematikkundervisning. Ball et al. (2008) har utviklet en modell (MKT) som viser til hvilke ferdigheter en lærer må besitte for god matematikkundervisning. Av den kommer det frem at lærere blant annet må ha grunnleggende faglig kunnskap, horisontkunnskap og spesialisert fagkunnskap. Ma (2020) introduserer hva dyp forståelse av grunnleggende matematikk, PUFM, innebærer, der mange av de samme kunnskapene defineres: «ulike løsningsstrategier», «grunnleggende ferdigheter», «langsiktig koherens» og «tilknytning».

5.3.1 Grunnleggende ferdigheter

Som forklart i teorikapittelet vil Ma (2020) sine argumenter for «grunnleggende ferdigheter» og Ball et al. (2008) sine argumenter for «grunnleggende faglig kunnskap» innebære at lærerne bør ha en grunnleggende kunnskap i programmering. Lærerne må kjenne til løkker, vilkår, variabler og funksjoner. TPACK er et rammeverk som skal plukke opp om lærerne mestrer interaksjonen mellom de tre kjernekomponentene innhold, pedagogikk og teknologi som må til for god teknologisk undervisning. Gjennom intervjuene kommer det ikke tydelig frem hvilke kunnskaper lærerne har om interaksjonen mellom disse tre kjernekomponentene. Grunnen til dette kan være at komponentene er utfordrende å skille fra hverandre, slik Archambault og Barnett (2010) beskriver. En annen grunn kan være at intervjuene ikke legger opp til å identifisere kunnskapen lærerne besitter. Studien har heller ikke som mål å identifisere hvilke ferdigheter lærerne har. Likevel kommer det frem at lærerne har tilegnet seg en del grunnleggende kunnskap om programmering. Samtidig blir det tydelig at de har veldig lav selvtillit på egne programmeringskunnskaper. Dette kan henge sammen med at de grunnleggende programmeringsferdighetene ikke stikker særlig dypt, men er kommet i nyere tid gjennom fagfornyelsen og innføringen av LK20, hvor det er blitt arbeidet med undervisning i programmering som eget emne. Lærerne er samstemte i at de er positive til profesjonsutvikling og utvikling av ferdigheter i programmering. Gjennom intervjuene blir det også belyst at det er ulikt hvor interesserte lærerne har vært i programmering, og at undervisningen frem til nå også har vært preget av dette.

Da prøveeksamensoppgavene ble offentliggjort forteller lærerne i denne studien at de kjente på en blanding av lettelse og skuffelse. Nivået på oppgavene var for dem overraskende lavt. En kan stille seg spørsmål om programmering ikke er så stort og viktig som først antatt. Som en av lærerne forteller omhandler dette en gradvis implementering av programmering. «*Det tar tid å snu en skute*», sa han. Nivået på eksamen er et synlig bevis på at endringene som er nødvendig også gradvis blir implementert i vurderingssituasjonen.

5.3.2 Ulike strategier og tilknytning

Lærerne må ha kunnskap om «ulike løsningsstrategier» og «tilknytning» ifølge Ma (2020). Dette er i tråd med Ball et al. (2008) sin beskrivelse av behovet for spesialisert fagkunnskap. Kunnskap om tilknytning handler om å koble programmering til andre emner i matematikk, og også kunne koble andre emner til programmering. Ulike løsningsstrategier omhandler kunnskap om de ulike retningene elevene kan gå i møte med et problem. Å koble på det matematiske aspektet ved arbeid med programmering er helt nødvendig for å forsvare programmerings posisjon i matematikkfaget ifølge Stenseth et al. (2019). Lærerne i denne studien forteller at de definitivt ser at det er en kobling mellom matematikk og programmering, men at det er utfordrende å se hvordan det skal kobles til ulike kompetansemål. Flere av lærerne i denne studien sier at blant annet emnene geometri og funksjonslære er enklere å koble til programmering enn andre. Dette er i kontrast til hva som blir belyst i teorikapittelet at kommer frem av Kaufmann og Maugesten (2022) sin forskning. Som beskrevet i resultatkapittelet forteller flere lærere at når en skal legge opp til at programmering skal ha et fokus på matematikk, blir oppgaven eller problemet kunstig fokusert mot matematikk. Som påpekt tidligere er det likevel en nødvendighet med fokus på matematikk.

Av resultatkapittelet kom det frem at lærerne i denne studien peker på problemløsningsoppgaver som et sentralt holdepunkt for gode matematikkundervisninger. Dette underbygger Bergqvist (2022) sin forskning om at problemløsning kan være en god måte å tilegne elevene programmeringskunnskaper på. Som sagt i teorikapittelet må en være dyktig i algoritmisk tenkning og gode problemløsningsferdigheter kreves for å kunne mestre programmering (Balanskat & Engelhardt, 2015; OECD, 2019). Det er påpekt at det er mange likheter mellom algoritmisk tenkning og problemløsning. Blant Pólya sine fire trinn for problemløsning er spesielt siste trinn ekstra spennende å se nærmere på. Det går ut på å «se tilbake» på hva du har gjort. Pólya råder til å stille seg spørsmål som fremmer metakognisjon. Hva gjorde jeg? Kunne jeg gjort det annerledes? En blir på dette viset bevisst egen tankeprosess og læring. Dette tar en med seg til neste gang en møter problem av lignende art (Liljedahl et al., 2016; Pólya, 2014)

Evnen til dette er en nødvendig egenskap hos gode problemløsere. Til sammenligning omhandler algoritmisk tenkning blant annet feilsøking. I denne prosessen behøver man å gå frem strukturert. En må forstå programmet for å finne ut hvor feilen kan befinne seg. I forsøk på å forbedre et program kan en dekomponere og abstrahere programmet. Da spør en seg, er det noen momenter her som er unødvendige? Kanskje kan vi effektivisere formelen og forkorte den? Når programmer er komprimerte og fungerer, kan lignende abstraherende grep benyttes ved andre anledninger. I resultatkapittelet blir det belyst av lærerne i denne studien at programmering må bli benyttet for å effektivisere regneoperasjoner og prosesser. Hvis ikke forsvinner hele hensikten med programmering. PRIMM-modellen er en strukturert modell som er bygd opp etter samme strategi som Pólya foreslår for problemløsning, med «investigate» som en del av prosessen og som innebefatter mye av feilsøkingprosessen og «se tilbake»-momentet.

5.3.3 Horisontkunnskap og langsiktig koherens

Et tredje moment som lærerne må ha kunnskap om er langsiktig koherens og horisontkunnskap (Ball et al., 2008; Ma, 2020). Langsiktig koherens innebærer at det blir benyttet presist og riktig språk og notasjoner. I noen settinger kan dette være krevende, for eksempel når det gjelder kommunikasjon. Med et mindre utviklet ordforråd kan faktisk et upresist språk til tider være mer hjelpelig for å skape forståelse. Samtidig kan dette igjen føre til en konflikt når en skal forklare situasjoner ved et senere tidspunkt. Med riktige og presise ord og uttrykk kan elevene tilpasse ny kunnskap til skjemaene de allerede har ved assimilasjon. Dersom elevene har fått forklart regler gjennom upresist språk, kan det være behov for en stor reorganisering av elevenes kognitive struktur gjennom akkomodasjon. Dette er argumenter for at langsiktig koherens er viktig i matematikk.

Horisontkunnskap inkluderer at lærerne er klar over hvorfor det som undervises i dag er nyttig for elevene i videreutviklingen av emnet og faget. Det kommer ikke frem av intervjuene at lærerne i denne studien har horisontkunnskap innenfor programmering. Lærerne forteller at de benytter læreverker som har delt inn pensum i emner og etter ulike klassetrinn. Tidligere år har noen av studiens lærere måttet arbeide med oppgaver på et lavere trinn enn der elevene går, fordi elevene og lærerne har besittet et lavt ferdighetsnivå av grunnleggende programmeringsferdigheter. Som beskrevet i resultatkapittelet forteller lærerne at det forventes at de skal lære seg programmering samtidig som de skal undervise det de første årene etter implementeringen i LK20. Lærerne forventer at de har større kontroll på emnet når de har undervist noen år og kjenner bedre til hva læreverket som benyttes forventer av ferdigheter på de ulike trinnene.

Fordi noen av lærerne benytter digitale læreverk har lærerne tilgang til pensum forbi deres klasstrinn og opp på videregående trinn. Dette gir mulighet for å tilegne seg horisontkunnskap. Lærerne må selv sette av tid i sin arbeidshverdag til å prioritere å sette seg inn i dette, da det ikke er prioritert en kollektiv kompetanseheving i programmering på deres arbeidsplasser.

5.3.4 Utvikling av lærerferdigheter

Fra OECD rapporten fra 2019 ble det fastslått at det må arbeides for å gi lærerne opplæring av høy kvalitet i hvordan en best kan integrere teknologi i undervisningen. Lærerne i denne studien forteller at de er veldig positive til kompetanseheving, men samtidig at ingen opplever at det er en høy prioritet å få løftet hele lærerfellesskapets programmeringsferdigheter. Det kommer frem av intervjuene at det vil ta tid å få innarbeidet LK20, og at programmering er et av mange områder som må bearbeides og utvikles kompetanse innenfor. Flere av lærerne i denne studien sier likevel at dersom de selv ønsker å gå på kurs eller utvikle seg innenfor emnet, opplever de å ha ledelsens støtte i dette.

Som vi så i teorikapittelet pekte Ma (2020) på spesielt tre faktorer som bidrar til å utvikle ferdigheter og egenskaper for å bli en lærer som underviser på høyt nivå. Disse er gode forberedelser, stram struktur og en tydelig læreplan. Programmering inn i fagfornyelsen er en politisk avgjørelse (Kaufmann & Maugesten, 2022), og i Norge har det vært et fokus på at lærerne skal ha frihet i gjennomføringen av undervisningen sin, spesielt i læreplanene som har kommet etter Læreplanverket fra 1997. I L-97 var innholdet i større grad fastsatt. I dag står lærerne frie til å velge hva og hvordan det skal undervises. Kompetansemålene er tolkbare, og bare innenfor programmering kunne Gjøvik og Høyland (2022) fortelle om tre retninger en kan velge å se på undervisningen på. Det gjelder undervisning i programmeringsferdigheter, undervisning gjennom programmeringsverktøy og undervisning med halvprogrammerte program. I mye større grad må beslutningstakere spille lærere gode, heller enn å gi så stor frihet at kreativitet og tolkning står i veien for god undervisning. Dersom lærere skal kunne nærme seg det Ma (2020) kaller for dyp forståelse av grunnleggende matematikk bør en ha en mye mindre tolkbar læreplan. Med en stram struktur i oppbyggingen av undervisningen og en lite tolkbar læreplan vil lærere kunne bruke all sin forberedelsestid på å legge opp til undervisningsøkter der fokuset er på gode problemløsningsoppgaver. I planleggingen bør det prioriteres å bli kjent med ulike strategier og hvordan lærere kan koble sammen både ulike emner og elevsvar til matematikk. Kreativiteten til lærere blir da ikke lengre et hinder for gode undervisningsøkter, og lærerne kan legge alt sitt fokus i planleggingen på hvordan de ønsker å

legge frem problem, hvor problem kan utbedres og hvor lenge elevene skal arbeide med et og samme problem.

Det er av denne studien gjort interessante funn som peker i retning av at det arbeides med profesjonsutvikling blant lærerne i denne studien på de samme områdene Ma (2020) forteller at kinesiske lærere utvikler kunnskap om matematikk. Både Ma (2020) og lærerne i denne studien omtaler læring av kolleger som en helt sentral del av profesjonsutvikling. Å jevnlig møtes for å diskutere og studere læremateriell er viktig for å utvikle sin matematiske kunnskap. En annen metode som både kommer frem i forskningen til Ma (2020) og i disse intervjuene er at elevene kan benyttes for å lære seg matematikk. Som beskrevet i resultatkapittelet er det erfart kurs i programmering for lærere i denne studien med dyktige elever som ressurspersoner i kurset. Elevene får trene seg i å forklare og lede andre mennesker, og motivasjonen deres blir stor av å kunne snu om på lærer-elev-rollene en liten stund. Et siste område Ma (2020) sier at ferdigheter i matematikk utvikles er på fritiden. Som presentert i resultatkapittelet kommer det frem at lærerne som kan programmering har brukt mye tid på dette på egenhånd, og andre forteller at de nå blir nødt til å sette seg ned med dette på fritiden før de skal undervise emnet.

5.4 Lærerrolle i endring

Et samfunn i endring krever at skolen endres. I klasserom der det blir praktisert tradisjonell undervisning står læreren fremme ved kateteret og tavla og underviser pensum og gir oppgaver som han selv besitter fasiten av. Elevene er passive lyttere som iherdig forsøker å holde tritt med undervisningen. I nyere tid er det et stort fokus på at dette må endres. Først og fremst gjennom å ha en mer utforskende matematikkundervisning. Denne typen undervisning er elevfokusert og legger opp til utforskning, prøving og feiling (Nosrati & Wæge, 2015). Samtidig har den digitale verdenen som kontinuerlig er i utvikling gitt flere muligheter for hvordan undervisning kan praktiseres.

5.4.1 Programmering endrer lærerrollen

I større og større grad skal undervisningen innebære deltakelse, og ikke bare overføring av kunnskap. I undervisning med programmering er utforskning og samarbeidslæring ofte brukte undervisningsstrategier. Som presentert i teorikapittelet sier Forsström og Kaufmann (2018) at dette medfører at lærerrollen er i endring, og at læreren skal operere som en veileder i klasserommet heller enn å være en som underviser. I resultatet kommer det frem av intervjuene at lærerne i denne studien støtter denne tanken. De legger vekt på at det blir viktig for lærerne fremover å gi slipp på behovet for kontroll over alt som skal foregå i klasserommet. En av

lærerne sa, «*Vi må lære oss å ikke være redde for å gi fra oss kontrollen*». En trygg og god lærer lar elevene få utfolde seg og utforske, imens læreren i større grad er en behjelpelig støttespiller om elevene sitter fast, er uenige i samarbeidet eller er i villrede.

Lærerne i denne studien forteller at elevene utvikler seg i rekordfart innenfor emnet programmering. Mange elever er allerede svært dyktige i håndteringen av datamaskin. Dette gjør at de har gode forutsetninger til å arbeide og lære raskt om programmering. En av lærerne ser lærerens endrede rolle i klasserommet i lys av idrettsverdenen. Han sammenligner lærerens rolle med en trener. Han sier, «*I idrett er det ikke sånn at treneren er bedre enn utøveren, men han klarer likevel å si hva utøveren skal videreutvikle for å bli bedre*». I lagidretter omhandler trenerrollen også å sette sammen et team hvor de ulike ferdighetene gjør at laget presterer best mulig. En kan spille på de ulike utøvernes styrker, og sammen som lag oppnå mer enn hva individene klarer på egenhånd. Det vil nærmest være samarbeidslæring i form av synlig prestasjon.

Mange læreroppgaver er allerede diskutert i dette kapitlet. Lærere skal blant annet hjelpe elevene med å se matematikk i programmeringsoppgaver og -problem, de skal fremme et positivt læringsmiljø med samarbeidslæring og gi gode tilbakemeldinger. Stenseth et al. (2019) peker på at det blir en læreroppgave å velge ut gode problem og oppgaver som ligger i skjæringspunktet mellom programmering og matematikk. Lærerne i denne studien forteller at det er gode ressurser tilgjengelig på nett og i digitale læreverk. Lærere må likevel forberede seg grundig gjennom å ha kontroll på hvordan problemene kan utbedres, bygges på, differensieres og knyttes til matematiske emner. Som diskutert i delkapittel 5.2.2 er opptak og tilbakemelding en dimensjon lærerne i denne studien sier at programmering kan fremme. Et interessant tema å gå inn på er hvordan lærere er viktige med sine ulike oppgaver i møte med programmering gjennom denne dimensjonen. For eksempel vil programmet av og til gi tips om hvor feilen kan ligge. Da er det nødvendig at elevene forstår meldingen programmet gir, og klarer å forstå hvorfor programmet ikke fungerer. Gjør de ikke det er det helt nødvendig at læreren kan tolke og forstå feilmeldingen. Når elever klarer å få programmet til å fungere, og det ikke er behov for hjelp fra læreren, vil læreren likevel være en viktig autoritet som elevene vil oppsøke anerkjennelse hos ved å vise frem programmet til. Oppmuntring og evner til å stille spørsmål som får eleven til å dekomponere, gjennom å trinnvis forklare prosessen de har gjennomgått, er viktige læreroppgaver i disse situasjonene. Spørsmålene må legges opp til resonnering og vil være med på å hjelpe elevene med utvikling av deres metakognisjon. Elevene kan utfordres til

å effektivisere programmet om mulig gjennom dekomponeringen, for eksempel ved abstraksjon av overflødige komponenter eller forkorting og omformulering av algoritmer og formler.

5.4.2 Lærer i en digital verden

De fleste av lærerne som deltok i undersøkelsen kunne fortelle at de følte de ikke hadde «god nok» erfaring med programmering til å undervise det i matematikkundervisningen. Som vi har sett i resultatkapittelet rapporterer lærerne derfor at de setter stor pris på å kunne benytte Campus Inkrement som læreverkt. Noe av grunnen til dette forklares med at elevene kan få grundig innføring av pensumet gjennom læringsvideoer. Elevene får også mulighet til å se leksjoner om igjen, dersom noe er uklart. På Campus Inkrement tar læreverket seg av leksjonene. Dette gjelder også i emnet programmering, så lærerne forteller at de selv kan bruke tid på andre områder av undervisningen. Det er derfor ingen tvil om at, i likhet med programmeringens endring av lærerrollen, så endrer digitale læreverkt også lærerrollen. Lærerne behøver ikke å instruere og kan bruke kreativiteten sin på andre områder i undervisningsopplegget. Mange driver «omvendt undervisning», altså at elevene ser leksjoner hjemme og kan arbeide med oppgaver på skolen. Instruksjonsvideoene er profesjonelt produsert og hjelper lærere ved å gi god og nøyaktig instruksjon.

Fordi Campus tolker programmering som et eget emne blir det naturlig at lærerne i denne studien som benytter dette læreverket også tolker programmering som eget emne. Det er likevel viktig som lærer å vurdere om læreverket gir en god innføring i emnet, og også om en skal vurdere å gå bort fra læreverket i gitte emner for eksempel fordi en vil arbeide med undervisning av temaer gjennom programmering. Enn så lenge opplever lærerne i denne studien at Campus Inkrement gir dem ressurser til å undervise elevene langt forbi hva de behøver for å oppnå kompetansemålene og kravene ved nasjonal prøveeksamen.

Et spørsmål en kan stille seg er om Campus eller andre digitale læreverkt gjør at lærerjobben kan utføres av lærere uten matematikkompetanse. En av lærerne i denne studien forteller at hvem som helst kan gjennomføre en time med programmering i matematikk. Mye fordi Campus kan ta seg av leksjonen. Likevel legger det ikke grunnlag for at det blir god kvalitet på undervisningen. Som en lærer sa, «*Matematikklærerne blir spesialister på programmering innen matematikk*». Læreregenskaper som må til for undervisning på høyt nivå krever kunnskap om kobling av matematikk og programmering. Som forklart tidligere sier Ma (2020) at matematikklærere må kjenne ulike løsningsstrategier, være god på tilknytning mellom emner, kjenne grunnleggende ideer og ha langsiktig koherens. En hvilken som helst lærer vil ikke komme inn og ha alle disse egenskapene. Dette er særegne ferdigheter som oftest tilhører

dyktige matematikklærere, men som også andre lærere mulig kan besitte. Som matematikklærer bør en derfor etterstrebe å utvikle seg innenfor disse områdene.

Et annet spørsmål en kan stille seg er om det kan tenkes at lærere bør være bekymret for om hele jobben deres kan bli overflødig. Kan for eksempel kunstig intelligens (AI) gjøre at lærere bare behøver å være mindre kvalifiserte voksne mennesker som leder barn i et utforskende miljø? Denne studien antyder noe annet. Av dimensjonene fra MKU-rammeverket som programmering i størst grad fremmer, finner vi «matematikkfaglig opptak og tilbakemelding» og «elevdeltakelse i matematiske resonnement og meningsskapning». AI kan blant annet utfordre lærerrollen ved å gi tilbakemelding på hvor feil i et program ligger og hvordan det kan løses. Dimensjonene stiller likevel krav til menneskelige ferdigheter. Spesielt blir lærere som er gode veiledere, som stiller de riktige spørsmålene basert på hva eleven har av bakgrunnskunnskap, og som har kjennskap til hvordan hver enkelt elev lærer best viktig. Kunnskap om faglig innhold og elever er som nevnt en av komponentene Ball et al. (2008) peker på som viktig lærerkunnskap for god undervisning. For å stille de riktige spørsmålene for læring og utvikling kreves kunnskap om faget. AI kan dermed utfordre deler av lærerrollen på sikt, men vil aldri kunne fjerne det menneskelige aspektet. AI er en del av den teknologiske utviklingen som skjer i dag. Selv om AI ikke kan ta over lærerjobben er det likevel et høyaktuelt tema hvordan lærere og elever kan benytte AI som en del av et læringsmiljø for å fremme enda bedre læring og utvikling.

5.5 Hvordan undervise programmering

Implementeringen av programmering i LK20 gjør at mange av lærerne i denne studien kjenner på stress. Dette er naturlig dersom man kjenner man ikke har den tilstrekkelige kompetansen og samtidig har et bilde av å måtte kunne faginnholdet, som bl.a. Ball et al. (2008) og Mas (2020) forskning tilsier, for at det skal bli kvalitet på matematikkundervisningen. Som diskutert tidligere leder dette mot en endring av lærerrollen, og samtidig kan en tenke annerledes om undervisningsformen en benytter. Som nevnt i teorikapittelet vil en undervisningsmetode med utforskende matematikkundervisning være nyttig for utvikling av programmeringsferdigheter (Nosrati & Wæge, 2015) og problemløsning kan være en enklere vei for å implementere arbeid med programmeringsferdigheter (Bergqvist, 2022).

5.5.1 Utforskende matematikkundervisning

LK20 inneholder kompetansemål for hva elevene skal kunne. Disse er produsert med et stort fokus på at elevene skal utvikle anvendte ferdigheter. Derfor er kompetansemålene formulert

med verb som legger opp til at elevene blant annet skal utforske, simulere, bruke, lage og beregne. Elevene skal gjøre heller enn å bare høre. Cuoco et al. (1996) var av de som ønsket å implementere en matematikkundervisning der elevene skal utvikle relasjonell forståelse i matematikk gjennom å blant annet utforske, lete etter mønstre og eksperimentere. Utviklingen av relasjonell forståelse skjer gjennom utforskende arbeid som inkluderer å se sammenheng mellom ideer, fakta og prosedyrer (Hiebert & Grouws, 2007; Nosrati & Wæge, 2015). Avkastingen som følger kan bli stor, for dersom fokuset i undervisningen blir på å utvikle den relasjonelle forståelsen, vil det i samme øyeblikk bli utviklet instrumentell prosedyrekunnskap (Nosrati & Wæge, 2015). Som nevnt tidligere anmoder Hiebert og Grouws (2007) om undervisning der elevene skal bli gitt en «anledning til å lære». Ut fra lærerintervjuene i denne studien har vi sett at en viktig del av denne typen undervisning er at læreren må utfordres på å gi slipp på kontrollen i klasserommet. Dette innebærer ikke å la klasserommet flyte fritt, men heller å gi elevene rom for å utforske. Læreren behøver ikke å være en som besitter fasiten til enhver tid. En lærer forteller at det faktisk kan være et motiverende virkemiddel å informere elevene om at en som lærer ikke sitter med svaret på oppgaven. Han sier, «*Da synes de det er spennende. At lærerne heller ikke vet. Og at vi er likestilte*». Elevene må da først løse problemet, før de må dekomponere det og bevise overfor medelever og læreren, steg for steg, hvordan de har gått frem i sine løsningsforslag.

Læreverket Campus Inkrement er blitt tatt flittig i bruk i leksjoner om programmering i flere av klasserommene til lærerne i denne studien. Av lærerintervjuene kommer det frem at Campus Inkrement i seg selv ikke naturlig legger godt opp til utforskende undervisning. Dette gjelder spesielt fordi en fort havner i et mønster der lærerne lar elevene se en undervisningsleksjon, for så å la elevene arbeide med oppgaver tilhørende leksjonen. Læreverket velger i stor grad å undervise programmering som et eget emne. Fokuset ligger på å utvikle grunnleggende programmeringsferdigheter og den algoritmiske tankegangen. En lærer forteller i denne studien at undervisningen av programmeringsferdigheter i Campus er veldig programmeringsrettet. I så stor grad at det blir utfordrende å koble programmering til matematikk. Som nevnt i teorikapittelet omtaler Bergqvist (2022) hvor verdifull tiden i klasserommet med elevene er. På grunn av dette er omvendt undervisning praktisk, da det går mindre tid til leksjoner på skolen og mer tid til veiledning, refleksjon og utforskning. Vi har også sett at lærerne i denne studien synes nivået eksamen forventer av elevene er veldig lavt. Bergqvist (2022) foreslår at problemløsningsoppgaver vil være en enklere vei for å implementere arbeid med programmeringsferdigheter. En lærer forklarer at Campus Inkrement har noen off-piste-

oppgaver som ikke er bundet til noe gitt emne, men som er sammensatte, utfordrende oppgaver. Utforskning vil kunne være å undersøke slike off-piste-oppgaver eller andre utfordrende problem som krever at elevene benytter de verktøyene de har og kobler kunnskap om ulike emner i matematikk. For at dette skal kunne fungere best mulig må lærerne gjøre et grundig forarbeid med å finne gode problemløsningsoppgaver. De må se matematikken som ligger i problemene, undersøke og predikere ulike løsningsstrategier elevene kan benytte og forberede ulike innfallsvinkler en kan utfordre elever på å undersøke. Læreren bør alltid vurdere om teknologi kan benyttes for å løse, visualisere eller effektivisere løsningsstrategier. Dette krever tid og må prioriteres tid til. Når lærerne i mindre grad behøver å lage spennende undervisningsopplegg, kan de omstrukturere tiden sin til heller å arbeide med dette.

5.5.2 Visuell matematikk

Så hvordan skal programmering i matematikk undervises? Det unisone ønsket fra lærerne i denne studien er at programmering skal bringe noe fysisk og visuelt inn i matematikken. I matematikkundervisningen har lærerne flere forslag til hvordan det kan legges til rette for ulike måter å praktisere matematikkundervisning med programmering. En av lærerne foreslår tverrfaglig opplegg med programmering inkludert. Det er nærliggende å tenke at dette kan gjøres sammen med naturfag, kunst og håndverk eller musikk, da disse også har programmering inkludert i kompetansemålene. Som vi har sett i teorikapittelet sier forskning (Forsström & Kaufmann, 2018; Hussain et al., 2006) at undervisningsopplegg med Lego Mindstorms kan fremme realfagsrettede ferdigheter. En av lærerne i denne studien trekker frem at elevene bør få utforske gjennom å koble ulike sensorer og motorer. Da kan elevene få fikle og skape ulike outputer som kan ta form på ulikt vis, som f.eks. gjennom lys eller lyd. Fysisk programmering, som Lego Mindstorms, krever at det prioriteres økonomi til å kjøpe inn utstyr. Fra lærerne i denne studien kommer det frem at de opplever at skolene de arbeider på har prioritert å kjøpe inn utstyr, og at det derfor gis gode muligheter for å benytte fysisk programmering i undervisningen.

Et annet forslag fra lærerne i denne studien er å lage større konkurranser som krever programmering. Konkurranser som skjer både internt på skoler mellom klasser, men også på tvers av skoler. Fra lærerne i denne studien kommer det frem at dette kan vekke en større motivasjon hos elevene og at de dermed kan engasjere seg også på fritiden sin. Når elevene opplever å skape noe vil de i enda større grad se relevansen av programmering, og samtidig vil de benytte sine matematiske verktøy på veien. Slike prosjekter vil både kreve tid, god

planlegging og logistikk. Med høy sannsynlighet vil også det økonomiske perspektivet vurderes dersom det er behov for store mengder utstyr.

En lærer forteller som nevnt i delkapittel 5.2.1 at elevene i de fleste tilfeller heller ønsker å skynde seg videre til neste oppgave enn å reflektere over de matematiske sidene ved arbeidet deres. Stenseth et al. (2019) peker på viktigheten av kvalitetssikring sammen med elevene i etterkant av programmering. Dette innebærer å se på utregninger og løsningsstrategier. Vi ser også at Nosrati og Wæge (2015) sier at diskusjon av løsninger og oppmuntring til å undersøke nye strategier er helt nødvendig for godt utbytte i en undersøkende matematikkundervisning. Dersom det vurderes å sette opp konkurranser internt eller på tvers av skoler bør det legges opp til et etterarbeid der elevene må presentere produktene sine. Utregninger, refleksjoner og strategier de har benyttet i sitt arbeid vil være viktig etterarbeid for å hjelpe elevene med å utvikle gode ferdigheter som for eksempel selvrefleksjon, muntlig argumentasjon og dekomponering. Læreren kan i dette arbeidet vurdere om det kan være hensiktsmessig å se på programmer eller roboter som ikke fungerte som ønsket, og stille spørsmål som utfordrer elevene til å analysere og reflektere over prosjektene. Lærere bør i dette arbeidet vektlegge koblingen mellom programmering og matematikk, og må aktivt hjelpe elevene til å trekke dette inn og hjelpe de med å stoppe opp og se matematikken de arbeider med i prosjektene. Det er helt nødvendig at elevene er del av et positivt og utforskende læringsmiljø, slik Mason (2016) omtaler som «conjecturing atmosphere», for at elevene skal få et godt læringsutbytte. I et slikt klasseromsklima vil elevene rette fokuset mot at programmene deres kan forbedres, og ikke at programmene har feil eller mangler. Arbeidet med klasseromsklimaet skjer ikke i en håndvending, men arbeid med programmering kan være med på å fremme et positivt klasseroms- og læringsklima.

6 Avslutning

Denne studien har tatt for seg en todelt problemstilling som bygger opp under teorien om at programmering kan benyttes for å fremme god matematikkundervisning og at lærerens rolle med programmering i skolen er i endring. For å svare på forskningsspørsmålene er det blitt gjort en litteraturgjennomgang av tidligere forskning om programmering i skolen og fem lærere har besvart semistrukturerte spørreskjemaer med kvalitativ vinkling samt deltatt i semistrukturerte intervjuer.

6.1 Svar på forskningsspørsmål

Første forskningsspørsmål tar for seg hvordan lærerne opplever at programmering fremmer god matematikkundervisning. Denne studien undersøker lærernes opplevelse av programmeringens innvirkning på matematikkundervisning gjennom et tilpasset MKU-rammeverk. God matematikkundervisning er mer komplekst enn hva MKU-rammeverket kan fange opp, men det kan likevel si noe om den matematiske kvaliteten av undervisningen. Med utgangspunkt i dimensjonene fra MKU kommer det frem av lærerne i denne studien at programmering kan fremme kvaliteten i matematikkundervisningen på spesielt to områder. Det er i størst grad gjennom dimensjonen *elevdeltakelse i matematiske resonnement og meningsskaping*. Lærerne mener det kognitive engasjementet fremmes blant elevene fordi programmering både er kognitivt krevende og kan benyttes som et verktøy i kognitivt utfordrende oppgaver. Denne studien underbygger med dette at programmering har potensialet til å utvikle blant annet elevenes problemløsningsferdigheter og komponenter matematisk kompetanse består av. Det er likevel ikke gitt at man utvikler disse ferdighetene selv om en driver undervisning med programmering. All utvikling krever nemlig at læreren er en god veileder som hjelper elevene med å se sammenheng mellom programmering og matematikk.

Deretter indikerer studien at dimensjonen *matematikkfaglig opptak og tilbakemelding* fremmes med programmering. Programmering kan gi lærere muligheter til å visualisere matematiske situasjoner for elever, og i tillegg gi lærere mulighet til å få innsikt i hvor elever har kunnskapshull. Dette krever lærerkunnskap som Ball et al. (2008) omtaler som spesialisert fagkunnskap og kunnskap om faglig innhold og elever. Studien peker i retning av at med programmering kan lærere gi gode tilbakemeldinger for å hjelpe elever med sin matematiske utvikling. Det er behov for mer forskning på sammenhengen mellom lærerkunnskap og hvordan programmering fremmer denne dimensjonen for å kunne generalisere.

Det er uenighet blant lærerne i denne studien om programmering fremmer eller holder tilbake kvaliteten på matematikkundervisningen gjennom dimensjonen *feil og unøyaktighet*. Det er ikke funnet eksempler i denne studien på at den *matematiske rikdommen* påvirkes av programmering.

Andre forskningsspørsmål tar for seg hvordan læreres rolle oppfattes med programmering i matematikkundervisning. Denne studien undersøker lærernes rolleforståelse gjennom en konvensjonell innholdsanalyse av semistrukturerte intervjuer. Det er enighet blant lærerne i denne studien om at lærerrollen er i endring, både med programmering, men også på grunn av digitale læreverker som benyttes. Studien støtter opp om forskning som sier at læreren opererer mer som en veileder (Forsström & Kaufmann, 2018; Stenseth et al. 2019) eller som en trener enn som en foreleser.

Endring av lærerrollen vil kreve at læreren slipper opp på kontrollen i undervisningsøkten og lar elevene få utfolde seg. Læreren vil være nødvendig som en autoritet som anerkjenner godt arbeid med støttende tilbakemeldinger, som utfordrer elever til å anvende og forklare sin algoritmiske tenkning og som stiller gode spørsmål som utfordrer eleven til å effektivisere og generalisere programkoder. Denne studien underbygger Stenseth et al. (2019) sin forskning om at viktige læreroppgaver er å hjelpe elever med å benytte matematikk i programmeringsarbeid. Den underbygger også Forsström og Kaufmann (2018) sin forskning om nødvendigheten av å utvikle et positivt læringsmiljø i klasserommet gjennom samarbeid der strategier og ideer deles med mål om å utvikle algoritmisk tenkning og problemløsningsferdigheter. På grunn av engasjementet som kan vekkes blant elever i arbeid med programmering er det mulig at en kan arbeide med utvikling av et positivt læringsmiljø i undervisning gjennom arbeid med programmering. Det er behov for mer forskning på dette.

Digitale læreverker som tar for seg leksjoner gir lærerne verdifull tid til gode forberedelser og til omorganisering av tidsbruken for kvalitet i undervisningen i klasserommet. Læreres tolkning av programmering i LK20 ligger til grunn for undervisningens utforming. Det kommer frem av lærerne i denne studien at de fleste anser programmering som et helt eget emne i matematikk. Dette medfører at det undervises i størst grad, slik Gjøvik og Høyland (2022) definerer det, i programmering der utvikling av programmeringsferdigheter vektlegges. Blant ferdigheter forskning peker på som nødvendig for god undervisning antyder denne studien at lærerne ikke opplever å ha høy selvtillit på egne programmeringsferdigheter, lærerne har lav grad av horisontkunnskap innenfor programmering og de finner det utfordrende å knytte matematikk til programmering. At lærerne ikke kjenner seg trygge på egne ferdigheter kan henge sammen med

at de grunnleggende elementene består av et uleselig lavnivåspråk, og at en derfor ikke logisk kan resonnerer seg frem og tilbake fra kjerneelementene til regler slik en kan med matematikk ved aksiomer og teorem gjennom deduksjon. Denne studien peker dermed i retning av at det med tiden bør undervises med mål om å utvikle algoritmisk tankegang, noe som skjer ved undervisning *med* eller *gjennom* programmering, heller enn *i* programmeringsferdigheter alene. Horisontkunnskap går ut på å ha kunnskap om at ulike basisferdigheter kreves som elementer i et større bilde. Gjennom flere år med undervisning av programmering på ulike trinn utvikler lærerne horisontkunnskapen sin. Lærere som ønsker å gjennomføre matematikkundervisning med programmering på høyt nivå bør etterstrebe å utvikle sin kunnskap om tilknytting mellom programmering og andre emner. Denne studien underbygger Stenseth et al. (2019) sin belysning av at det er en viktig oppgave for lærere å finne oppgaver som har riktig vanskelighetsgrad og som fungerer både i matematikk og programmering. Forskning indikerer at utforskende matematikkundervisning utvikler relasjonell forståelse (Nosrati & Wæge, 2015) og problemløsning kan være en enklere vei for å utvikle programmeringsferdigheter (Bergqvist, 2022). Som del av forberedelsene indikerer denne studien at lærere bør vurdere hvordan problem bør legges frem, hvordan problem kan utbedres og hvilke ulike løsningsstrategier elever kan benytte.

6.2 Avsluttende kommentar

I tillegg til svar på forskningsspørsmålene er det gjort andre interessante funn gjennom denne studien. Jeg vil i dette delkapittelet trekke frem de mest interessante funnene som er gjort og som ikke direkte omhandler studiens problemstilling.

Først og fremst ble min førforståelse (beskrevet i kapittel 3) om at lærere opplever programmering krevende å undervise på grunn av lite erfaring delvis bekreftet. Funn fra denne studien viser at lærerne synes det er krevende å koble programmering til matematikk, men er ikke redde for å undervise i programmering. Læreryrket er svært komplekst, og det kan av og til virke som at det legges til arbeidsoppgaver på lærerne uten at det reduseres andre steder. Er det én ting som skinner gjennom alle resultatene fra denne studien er det at lærerne ønsker å lykkes. Fra tidligere forskning på norske lærere pekes det på egenskaper ved læreren heller enn fagkunnskap som mest nødvendig for effektiv matematikkundervisning (Fauskanger, 2017). I denne studien er lærerne klare på at faglig kunnskap om programmering er nødvendig og må tilegnes gradvis. Lærerne ønsker å lære mer, de ønsker å mestre programmering og de ønsker å hjelpe elevene til å oppnå sine mål. De er pliktoppfyllende og støtter implementeringen av

programmering. På tross av dette kjenner flere av lærerne i denne studien på tilstrekkelighet og lav selvtillit på egne ferdigheter i programmering.

Det var interessant at det gjennom denne studiens analyser blir identifisert profesjonsutvikling blant norske lærere gjennom de samme områdene som Ma (2020) anmoder om og omtaler at kinesiske lærere arbeider. De studerer og diskuterer læremateriell i faggrupper, de lærer av elever og de lærer av å arbeide med fagstoff på fritiden. Profesjonsutvikling er viktig for best mulig kvalitet i skolen. Selv om læreplanen for Kunnskapsløftet 2006 (LK06) er utgått, er ikke nødvendigvis innholdet utdatert. I LK06 stod det følgende i den generelle delen om det arbeidende mennesket, «*Den viktigste av alle pedagogiske oppgaver er å formidle til barn og unge at de er i stadig utvikling, slik at de får tillit til egne evner*» (Kunnskapsdepartementet, 2015). Dette målet bør fortsatt gjelde, også for lærerne.

Denne studien ble innledet med et omskrevet utdrag fra den overordnede delen av LK20 med et fokus på hvordan lærere lærer. Å se at norske lærere er ivrige på å lære og at skolesystemet arbeider profesjonelt med utvikling viser at det i Norge er gode forutsetninger for at fremtidens generasjoner får kvalitet i grunnskoleopplæringen. Som vi har sett i denne studien vil det ifølge Ma (2020) være til stor nytte om lærerne kan arbeide etter en tydeligere læreplan, uten den grad av tolkningsmuligheter LK20 gir. Med færre mulige tolkninger av læreplanen kan lærerne prioritere planleggings- og fagmøtetid på å øke kvaliteten på veiledning gjennom utforskning av problem heller enn at de oppslukes av å vurdere og diskutere hvilke mål elevene skal oppnå.

Lærerne i denne studien ønsker at programmering skal bringe noe fysisk og visuelt inn i matematikk. Forskning (Forsström og Kaufmann, 2018) viser til at programmering kan føre til motivasjon hos elevene og lærerne i denne studien støtter dette. Lærerne foreslår å legge opp til større eller mindre konkurranser og tverrfaglig prosjekter med samarbeidslæring på grunn av programmeringens potensiale til å gjøre matematikk visuelt. Dette gjør at arbeid med programmering kan benyttes i utviklingen av et positivt lærings- og klassemiljø. Samarbeidsoppgaver kan også være hjelpelig for å utvikle anvendte ferdigheter som er høyt ettertraktet i arbeidsmarkedet (OECD, 2019). Målet med programmering skal likevel ikke være å drive programmering for å motivere elever, men for å utvikle problemløsningsferdigheter og algoritmisk tankesett. I likhet med undersøkende matematikkundervisning vil den avsluttende delen i matematikkundervisning med programmering være viktig for å bevisstgjøre elevene på matematiske sider ved arbeidet deres, samt å diskutere ulike løsningsstrategier. Dersom lærerne selv finner det krevende å se matematikk i programmering vil sannsynligvis også elever gjøre

det samme. Denne delen krever derfor gode forberedelser der det legges opp til å stilles gode spørsmål som hjelper elevene å dekomponere og trinnvis forklare regneoperasjoner de har gjort.

6.3 Implikasjon for videre forskning

Temaet som er belyst i denne studien er både aktuelt og i konstant utvikling. På grunn av dette er det behov for kontinuerlig oppdatert forskning innenfor fagfeltet. Det ville være spennende å gå inn i et klasserom og studere praksisen av undervisning med programmering i matematikk. Denne studien har belyst at en viktig oppgave for lærere er å koble matematikk og programmering, og å hjelpe elevene med å se matematikken i arbeidet med programmering. Det ville vært spennende å se hvordan dette kan gjøres ved å observere lærere som mestrer dette. Da kunne det blitt undersøkt hvordan elevinitierte løsningsforslag trekkes frem og hvordan læreren hjelper elevene å se matematikk i deres egne løsninger av arbeid med programmering. Det er også et behov for, og interessant med, mer forskning på elevers motivasjon i arbeid med programmering i matematikk og hvordan en skal vurdere elever i samarbeidssituasjoner med programmering.

Et annet område det ville vært interessant å gå enda dypere inn i er hvordan undervisning endres av digitale læreverker og etter hvert av AI. Denne studien har så vidt berørt hvordan Campus Inkrement kan være med på å endre lærerrollen. Spørsmål som har reist seg er blant annet hvordan lærere bruker digitale læreverker for effektivt læringsutbytte hos elevene og hvordan digitale læreverker kan bidra til at lærere kan forbedre egen undervisning. Det er behov for mer forskning på dette. At AI blir en del av hverdagen, virker også mer og mer uunngåelig. Hvordan kan vi forberede oss på en fremtid med AI i klasserommet, og hvordan kan AI benyttes for å styrke elevers læring og utvikling av anvendte ferdigheter?

7 LITTERATURLISTE

- Archambault, L. M., & Barnett, J. H. (2010). Revisiting technological pedagogical content knowledge: Exploring the TPACK framework. *Computers & Education*, 55(4), 1656–1662.
- Balanskat, A. & Engelhardt, K. (2015). *Computing Our Future: Computer Programming and Coding – Priorities, School Curricula and Initiatives across Europe [Update 2015]*. European Schoolnet.
- Ball, D.L., Thames, M.H. & Phelps, G. (2008). *Content knowledge for teaching: What makes it special? Journal of Teacher Education*, 59(5), s. 389–407. *Mathematics Education*, 19 (3-4), 101–117.
- Berg, T.K. (2021). Analog programmering. *Tangenten – tidsskrift for matematikkundervisning*, 32(3), 42–52.
- Bergem, O. K., Nilsen, T. & Scherer, R. (2016). Undervisningskvalitet i matematikk. I O. K. Bergem, H. Kaarstein & T. Nilsen (Red.), *Vi kan lykkes i realfag: Resultater og analyser fra TIMSS 2015* (s. 120–136). Universitetsforlaget.
- Bergqvist, E. (2022). An inquiry of different interpretations of programming in conjunction with mathematics teaching. *Bringing Nordic mathematics education into the future. Proceedings of Norma 20. The ninth Nordic Conference on Mathematics Education, Oslo, 2021*. SMDF. (Nr. 17). 17–24. http://matematikdidaktik.org/wp-content/uploads/2021/04/NORMA_20_preceedings.pdf
- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). *The nordic approach to introducing computational thinking and programming in compulsory education*. Report prepared for the Nordic@BETT2018 Steering Group. <https://doi.org/10.17471/54007>
- Bryman, A. (2012). *Social research methods* (4. utg.). Oxford university press.
- Casner-Lotto, J., & Barrington, L. (2006). *Are they really ready to work? Employers' perspectives on the basic knowledge and applied skills of new entrants to the 21st century US workforce*. Partnership for 21st Century Skills. 1 Massachusetts Avenue NW Suite 700, Washington, DC 20001.
- Christoffersen, L. og Johannessen, A. (2012). *FORSKNINGSMETODE FOR LÆRERUTDANNINGENE* (1. utg.). Abstrakt forlag AS.
- Crick, T. (2017). Computing education: An overview of research in the field. *Royal Society*. 90.

- Cuoco, A., Goldenberg, E. P., & Mark, J. (1996). Habits of mind: An organizing principle for mathematics curricula. *The Journal of Mathematical Behavior*, 15(4), 375-402.
- Dahlum, S. (2021, 9. mars), Validitet. SNL. <https://snl.no/validitet>
- Dalland, O. (2015). *METODE OG OPPGAVESKRIVNING* (5. utg.). Gyldendal Akademisk.
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39.
- Dolonen, J. A., Kluge, A., Litherland, K., & Mørch, A. I. (2019). *Litteraturgjennomgang av programmering i skolen*. <http://urn.nb.no/URN:NBN:no-79405>
- Dvergsdal, H. (2019, 6. november). *Python – programmeringsspråk*. I Store norske leksikon. Hentet 26. januar 2023 fra [https://snl.no/Python - programmeringsspr%C3%A5k](https://snl.no/Python_-_programmeringsspr%C3%A5k)
- Engelsen, B.U. (2015). *KAN LÆRING PLANLEGGES?: Arbeid med læreplaner – hva, hvordan, hvorfor?* (7. utg.). Gyldendal Akademisk.
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28.
- Fauskanger, J. & Mosvold, R. (2015). En metodisk studie av innholdsanalyse – med analyser av matematikklæreres undervisningskunnskap som eksempel. *Nordic Studies in Mathematics Education*, 20(2), 79–96
- Fauskanger, J., (2016). Matematikklæreres oppfatninger om ingrediensene i god matematikkundervisning. *Acta Didactica Norge*. 10(3) Art. 5.
- Fauskanger, J., (2017). Kunnskap nødvendig for effektiv matematikkundervisning – slik lærere selv ser det. *Norsk Pedagogisk Tidsskrift*. 17(1) 45–56.
- Fauskanger, J., Bjuland, R. & Mosvold, R. (2010). ”Eg kan jo multiplikasjon, men ka ska eg gjørr?” – det utfordrende undervisningsarbeidet i matematikk. I T. Løkensgard Hoel, G. Engvik & B. Hansen (Red.), *Ny som lærer: sjansespill og samspill* (s. 99–114). Tapir Akademisk Forlag.
- Forsström, S. E. & Kaufmann, O. T. (2018). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research* 17(12), 18–32. <https://doi.org/10.26803/ijlter.17.12.2>
- Gjøvik, Ø. & Høyland, J. (2022). *Kloss for kloss: blokkprogrammering for lærere*. Universitetsforlaget.
- Grover, S. & Pea, R. (2013). Computational thinking in K–12: a review of the state of the field. *Educational Research*, 42(1), 38–43.

- Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.
- Harvard University. (u.å.). *Mathematical Quality of Instruction (MQI)*. Hentet 27. februar 2023 fra <https://cepr.harvard.edu/mqi>
- Hellevik, O. (1999). *Forskningsmetode i sosiologi og statsvitenskap*. (6. utg.). Universitetsforlaget.
- Hiebert, J., & Grouws, D. A. (2007). The Effects of Classroom Mathematics Teaching on Students' Learning. In F. Lester (Ed.), *Second Handbook of Research on Mathematics Teaching and Learning* (pp. 371-404). Charlotte, NC: Information Age.
- Hill, H. C., Blunk, M. L., Charalambous, C. Y., Lewis, J. M., Phelps, G. C., Sleep, L. & Ball, D. L. (2008). Mathematical knowledge for teaching and the mathematical quality of instruction: An exploratory study. *Cognition and instruction*, 26(4), 430-511.
- Holme, I. M. og Solvang, B.K. (1996). *Metodevalg og metodebruk*. (3. utg.). Tano.
- Hussain S., Lindh, J. & Shukur, G. (2006). The effect of LEGO Training on Pupils' School Performance in Mathematics, Problem Solving Ability and Attitude: Swedish Data. *Educational Technology & Society*, 9(3), 182–194.
- Imsen, G. (2014). *Elevens verden* (5. utg.). Oslo: Universitetsforlaget.
- Johansen, A.-K. (2020, 11. juli). - *Programmering vil bli utfordrende for lærere*. Forskning.no. <https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold-matematikk/programmering-vil-bli-en-utfordring-for-laerere/1711838>
- Kaufmann, O. T. & Maugesten, M. (2022). "I do not know much about programming, but I think that it is good for mathematics": views of student teachers in Norway on integrating programming into mathematics education. *Bringing Nordic mathematics education into the future. Proceedings of Norma 20. The ninth Nordic Conference on Mathematics Education, Oslo, 2021*. SMDF. (Nr. 17). 17–24.
http://matematikdidaktik.org/wp-content/uploads/2021/04/NORMA_20_preceedings.pdf
- Kaufmann, O. T. & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 52(7), 1029–1048.
- Kaufmann, O. T., Stenseth, B., Berggren, S. A. & Forsström, S. (2022). "Where is my angle?" – students cooperating to make a square. *Bringing Nordic mathematics education into the future. Proceedings of Norma 20. The ninth Nordic Conference on Mathematics*

- Education, Oslo, 2021*. SMDF. (Nr. 17). 121–128. http://matematikdidaktik.org/wp-content/uploads/2021/04/NORMA_20_preceedings.pdf
- Knuth, D. E. (1974). Computer science and its relation to mathematics. *The American Mathematical Monthly*, 81(4), 323–343.
- Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge? *Contemporary Issues in Technology and Teacher Education*, 9(1), 60–70.
- Kunnskapsdepartementet. (2014). *Strategi: Lærerløftet: På lag for kunnskapsskolen*. Kunnskapsdepartementet. https://www.regjeringen.no/globalassets/upload/kd/vedlegg/planer/kd_strategiskole_w eb.pdf
- Kunnskapsdepartementet. (2015, 25. august). *Generell del av læreplanen (UTGÅTT) - Det arbeidende mennesket*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2006. <https://www.udir.no/laring-og-trivsel/lareplanverket/utgatt/generell-del-av-lareplanen-utgatt/det-arbeidende-mennesket/>
- Kunnskapsdepartementet. (2017, 1. september). *Overordnet del – Å lære å lære*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/2.4-a-lare-a-lare/>.
- Kunnskapsdepartementet. (2019). *Læreplan i matematikk 1.-10. trinn (MAT01-05)*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/mat01-05?lang=nob>
- Kvale, S. og Brinkmann, S., (2019). *Det kvalitative forskningsintervju* (3. utg). Gyldendal Akademisk.
- Liljedahl, P., Santos-Trigo, M., Malaspina, U., & Bruder, R. (2016). Problem solving in mathematics education. I G. Kaiser (Red.), *Problem Solving in Mathematics Education* (s. 1-39). Springer International Publishing.
- Lindsø, J. F. (2020, 27. februar). *Hva er programmering?*. NDLA. <https://ndla.no/article/22522>
- Ma, L., (2020). *Knowing and teaching elementary mathematics : teachers' understanding of fundamental mathematics in China and the United States* (3. utg.). Routledge.
- Mason, J. (2016). When Is a Problem...? “When” Is Actually the Problem!. I P. Felmer,, J. Kilpatrick & E. Pekhonon (Red.), *Posing and Solving Mathematical Problems: Advances and New Perspectives* (s. 263-285). Springer International Publishing.

- Maugesten, M., Stigberg, H., & Stigberg, S. (2022). Examining TPACK among 8th–10th grade teachers after introducing a Use-Modify-Create programming approach. . *Bringing Nordic mathematics education into the future. Proceedings of Norma 20. The ninth Nordic Conference on Mathematics Education, Oslo, 2021*. SMDf. (Nr. 17). 169–176. http://matematikdidaktik.org/wp-content/uploads/2021/04/NORMA_20_preceedings.pdf
- Meld. St. 28 (2015–2016). *Fag – Fordypning – Forståelse — En fornyelse av Kunnskapsløftet*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/meld.-st.-28-20152016/id2483955/?ch=1>
- Michaelsen, A. S. (2019). *Det digitale klasserommet: Utnytt mulighetene!* (2. utg.). Cappelen Damm Akademisk.
- NESH. (2019, 10. februar). *Generelle forskningsetiske retningslinjer*. De nasjonale forskningsetiske komiteene. <https://www.forskningsetikk.no/retningslinjer/generelle/>
- NESH. (2021, 16. desember). *Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora*. De nasjonale forskningsetiske komiteene. <https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora/>
- Nosrati, M. & Wæge, K. (2015). Sentrale kjennetegn på god læring og undervisning i matematikk. *Matematikksenteret*. <https://www.matematikksenteret.no/sites/default/files/attachments/product/Oppdatert%20september%202019%20Sentrale%20kjennetegn%20p%C3%A5%20god%20l%C3%A6ring%20og%20undervisning%20i%20matematikk.pdf>
- OECD (2019), *OECD Skills Outlook 2019: Thriving in a Digital World*, OECD Publishing, <https://doi-org.ezproxy.uis.no/10.1787/df80bc12-en>.
- Olsen, R. V. (2013). Undervisning i matematikk. I M. Kjærnsli & R. V. Olsen (Red.), *Fortsatt en vei å gå: norske elevers kompetanse i matematikk, naturfag og lesing i PISA 2012* (s. 121–156). Universitetsforlaget.
- Opsvik, F. & Skorpen, L. B. (2014). *Matematisk kvalitet i undervisning*. Nordic Studies in Pólya, G. (2014). *How to solve it: A new aspect of mathematical method* (Vol. 85). Princeton university press.
- Postholm, M. B. & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm Akademisk.
- Rossen, E. (2019, 6. november). *Programmeringsspråk*. I Store norske leksikon. Hentet 26. januar 2023 fra <http://snl.no/programmeringsspr%C3%A5k>

- Sevik, K. (2016). *Programmering i skolen*. Notat fra Senter for IKT i utdanningen.
https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Shulman, L.S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2), 4–14.
- Silverman, D. (2020). *Interpreting qualitative data* (6. utg.). Sage.
- Stenseth, B., Kaufmann, O. T. & Forsström, S. E. (2019). Programmering og matematikk. *Tangenten – tidsskrift for matematikkundervisning*, 30(2), 7–12.
- Svartdal, F. (2020, 3. april). *Reliabilitet*. SNL. <https://snl.no/reliabilitet>
- Tamborg, A. L., Elicer, R., & Spikol, D. (2022). Programming and Computational Thinking in Mathematics Education: An Integration Towards AI Awareness. *KI-Künstliche Intelligenz*, 36(1), 73–81.
- Taylor, H. D. & Taylor, L. (1993). *George Pólya: Master of discovery 1887-1985*. Dale Seymour Publication.
- Thagaard, T. (2013). *Systematikk og innlevelse: En innføring i kvalitativ metode* (4. utg.). Fagbokforlaget
- Tjora, A. (2017). *Kvalitative forskningsmetoder i praksis* (3. utg.). Gyldendal akademisk
- Utdanningsdirektoratet. (2019a, 13. mars). *Dybdeløring*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/laring-og-trivsel/dybdelaring/>
- Utdanningsdirektoratet. (2019b, 27. mars). *Algoritmisk tenkning*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020.
<https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Waite, J. (2018). Pedagogy in teaching computer science in schools: A literature review. *Royal Society*, 253.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Yin, R. K. (2018). *Case study research and applications: design and methods* (6. utg.). SAGE

VEDLEGG

Godkjennelse fra Sikt

Vedlegg 1: Meldeskjema til Sikt

Vedlegg 2: Bekreftelse fra Sikt

Informasjonsskriv

Vedlegg 3: Informasjonsskriv til lærere

Intervjuguide

Vedlegg 4: Intervjuguide

Transkripsjonsarbeid

Vedlegg 5: Transkripsjonsnøkkel

Vedlegg 6: Transkripsjonsmal

GODKJENNELSE FRA SIKT

Vedlegg 1: Meldeskjema til Sikt

21.03.2023, 09:31

Meldeskjema for behandling av personopplysninger



[Meldeskjema](#) / [Masteroppgave i matematikdidaktikk](#) / Eksport

Meldeskjema

Referansenummer

836232

Hvilke personopplysninger skal du behandle?

- Navn (også ved signatur/samtykke)
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Beskriv hvilke bakgrunnsopplysninger du skal behandle

Arbeidssted og stilling vil bli behandlet. Arbeidsstedet skal anonymiseres i likhet med navnet til informanten.

Prosjektinformasjon

Prosjekttittel

Masteroppgave i matematikdidaktikk

Prosjektbeskrivelse

Formålet med oppgaven er å forske på overgangen i matematikken i skolen med programmering som fokus.

Begrunn hvorfor det er nødvendig å behandle personopplysningene

Personopplysningene jeg vil behandle er adekvate i form av forskningens tematikk og problemstilling, og personopplysningene er nødvendige og begrenset for å sikre at informantene er relevans for forskningen.

Prosjektbeskrivelse

[PROSJEKTBEKRIVELSE_MASTER_SAERSLAND.pdf](#)

Ekstern finansiering

Ikke utfyllt

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Eirik Særsland, eirikts1997@gmail.com, tlf: 41322272

Behandlingsansvar

Behandlingsansvarlig institusjon

Universitetet i Stavanger / Fakultet for utdanningsvitenskap og humaniora / Institutt for grunnskolelærerutdanning, idrett og spesialpedagogikk

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Sean Peter Kåss Martin, sean.martin@uis.no, tlf: 91 122279

Skal behandlingsansvaret deles med andre institusjoner (felles behandlingsansvarlige)?

Nei

Utvalg 1

Beskriv utvalget

Matematikklærere

Beskriv hvordan rekruttering eller trekking av utvalget skjer

Jeg tar kontakt med min siste praksislærer via mobil for å få informasjon om det er interesse for deltakelse til intervju. Hun tar kontakt med sin leder som skal forhøre seg med sine ansatte om de ønsker å delta.

Alder

23 - 67

Personopplysninger for utvalg 1

- Navn (også ved signatur/samtykke)
- Lydopptak av personer
- Bakgrunnsopplysninger som vil kunne identifisere en person

Hvordan samler du inn data fra utvalg 1?

Personlig intervju

Vedlegg

[Intervjuguide Lærer.docx](#)

Grunnlag for å behandle alminnelige kategorier av personopplysninger

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

Informasjon for utvalg 1

Informerer du utvalget om behandlingen av personopplysningene?

Ja

Hvordan?

Skriftlig informasjon (papir eller elektronisk)

Informasjonsskriv

[Informasjonsskriv_lærere.docx](#)

Tredjepersoner

Skal du behandle personopplysninger om tredjepersoner?

Nei

Dokumentasjon

Hvordan dokumenteres samtykkene?

- Manuelt (papir)

Hvordan kan samtykket trekkes tilbake?

Informantene kan trekke tilbake samtykke ved å kontakte undertegnede per SMS eller epost, eller ved direkte kontakt når undertegnede er på skolen.

Hvordan kan de registrerte få innsyn, rettet eller slettet personopplysninger om seg selv?

Informantene i intervjuene får utdelt informasjonsskriv med tydelig informasjon om deres rettigheter. Her står også kontaktmuligheter. Dersom jeg blir kontaktet av informanter som ønsker innsyn oversender jeg datamateriale som kun omhandler den personen. Dette vil mest sannsynlig bli transkripsjoner av intervjuet som er lagret på datamaskin med fiktive navn. Oversikt over kobling mellom fiktive navn og reelle navn er lagret på en ekstern harddisk. Dersom informant vil rette opp eller slette opplysninger har de rett på dette, og jeg vil etterfølge deres ønske.

Totalt antall registrerte i prosjektet

1-99

Tillatelser

Skal du innhente følgende godkjenninger eller tillatelser for prosjektet?

Ikke utfyllt

Behandling

Hvor behandles personopplysningene?

- Mobile enheter tilhørende behandlingsansvarlig institusjon

Hvem behandler/har tilgang til personopplysningene?

- Student (studentprosjekt)

Tilgjengeliggjøres personopplysningene utenfor EU/EØS til en tredjestat eller internasjonal organisasjon?

Nei

Sikkerhet

Oppbevares personopplysningene atskilt fra øvrige data (koblingsnøkkel)?

Ja

Hvilke tekniske og fysiske tiltak sikrer personopplysningene?

- Personopplysningene anonymiseres fortløpende

Varighet

Prosjektperiode

14.11.2022 - 31.12.2023

Hva skjer med dataene ved prosjektslutt?

Data anonymiseres (sletter/omskriver personopplysningene)

Hvilke anonymiseringstiltak vil bli foretatt?

- Lyd- eller bildeopptak slettes
- Koblingsnøkkelen slettes
- Personidentifiserbare opplysninger fjernes, omskrives eller grovkategoriseres

Vil de registrerte kunne identifiseres (direkte eller indirekte) i oppgave/avhandling/øvrige publikasjoner fra prosjektet?

Nei

Tilleggsopplysninger

Vedlegg 2: Bekreftelse fra Sikt

21.03.2023, 09:32

Meldeskjema for behandling av personopplysninger



[Meldeskjema](#) / [Masteroppgave i matematikdidaktikk](#) / Vurdering

Vurdering av behandling av personopplysninger

Referansenummer
836232

Vurderingstype
Standard

Dato
17.11.2022

Prosjekttittel

Masteroppgave i matematikdidaktikk

Behandlingsansvarlig institusjon

Universitetet i Stavanger / Fakultet for utdanningsvitenskap og humaniora / Institutt for grunnskolelærerutdanning, idrett og spesialpedagogikk

Prosjektansvarlig

Sean Peter Kåss Martin

Student

Eirik Særsland

Prosjektperiode

14.11.2022 - 31.12.2023

Kategorier personopplysninger

Alminnelige

Lovlig grunnlag

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

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 31.12.2023.

[Meldeskjema](#)

Kommentar

OM VURDERINGEN

Personverntjenester 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.

Personverntjenester har nå vurdert den planlagte behandlingen av personopplysninger. Vår vurdering er at behandlingen er lovlig, hvis den gjennomføres slik den er beskrevet i meldeskjemaet med dialog og vedlegg.

VIKTIG INFORMASJON TIL DEG

Du må lagre, sende og sikre dataene i tråd med retningslinjene til din institusjon. Dette betyr at du må bruke leverandører for spørreskjema, skylagring, videosamtale o.l. som institusjonen din har avtale med. Vi gir generelle råd rundt dette, men det er institusjonens egne retningslinjer for informasjonssikkerhet som gjelder.

DEL PROSJEKTET MED PROSJEKTANSVARLIG

For studenter er det obligatorisk å dele prosjektet med prosjektansvarlig (veileder). Del ved å trykke på knappen «Del prosjekt» i menylinjen øverst i meldeskjemaet. Prosjektansvarlig bes akseptere invitasjonen innen en uke. Om invitasjonen utløper, må han/hun inviteres på nytt.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til den datoen som er oppgitt i meldeskjemaet.

LOVLIG GRUNNLAG

Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake.

Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER

Personverntjenester vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen

formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles til nye, uforenlige formål

dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet

lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER

Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), og dataportabilitet (art. 20).

Personverntjenester vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER

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).

Ved bruk av databehandler (spørreskjemaleverandør, skylagring eller videosamtale) må behandlingen oppfylle kravene til bruk av databehandler, jf. art 28 og 29. Bruk leverandører som din institusjon har avtale med.

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og/eller rådføre dere med behandlingsansvarlig institusjon.

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. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde: <https://www.nsd.no/personverntjenester/fyll-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema>

Du må vente på svar fra oss før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

INFORMASJONSSKRIV

Vedlegg 3: Informasjonsskriv til lærere

Vil du delta i forskningsprosjektet «Programmering i matematikken»?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å undersøke hvordan lærere har erfart og utøver overgangen med programmering inn i matematikken. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Prosjektet er en masteroppgave og vil foregå i perioden 2022-2023. Målet er å undersøke hvordan ulike lærere opplever å skulle undervise i programmering og hvordan de benytter programmering i undervisningen sin. Prosjektet har et særlig fokus på hvordan lærere mener programmering kan fremme god matematikkundervisning, og koblingen mellom programmeringen og matematikken vil være interessant å sikte seg inn på. Det overordnede målet med prosjektet er å bidra til større forståelse av matematikklæreres praktisering av det nye temaet «programmering». Dette er et forskningsprosjekt som ledes Eirik Særsland, masterstudent ved Universitetet i Stavanger, med tett oppfølging av universitetslektor Sean Peter Kåss Martin ved Institutt for grunnskolelærerutdanning, idrett og spesialpedagogikk. Resultatene av studien vil kunne formidles i forskningsrapporter, tidsskriftartikler, bokkapitler og konferansepaper.

Hvem er ansvarlig for forskningsprosjektet?

Dette er et forskningsprosjekt som ledes av Eirik Særsland, masterstudent ved Universitetet i Stavanger, med tett oppfølging av universitetslektor Sean Peter Kåss Martin ved Institutt for grunnskolelærerutdanning, idrett og spesialpedagogikk.

Hvorfor får du spørsmål om å delta?

Jeg har spurt matematikklærere i universitetets praksisnettverk, hvor jeg selv har erfaring med å ha vært i praksis, om å delta i prosjektet. Utvalget informanter er valgt strategisk fordi jeg har grunn til å tro at samtlige matematikklærere har en erfaring og noe å bidra med inn mot forskningens formål om programmering i matematikken.

Hva innebærer det for deg å delta?

I løpet av de ukene prosjektet forgår vil undertegnede, masterstudent Eirik Særsland, intervju matematikklærere og gjøre lydopptak og ta notater av dette. Intervjuet med matematikklærer vil gjøres etter avtale.

Etter bearbeidelse av datamateriale kan det bli aktuelt å ta kontakt for et oppfølgingsintervju. Lærerne vil bli spurt om de er åpne for å delta på et oppfølgingsintervju med observasjon av en undervisningstime ved en senere anledning (samme skoleår) i intervjuet.

Oppfølgingsintervjuet og observasjon av undervisning vil ikke være påkrevd ved deltakelse i undersøkelsen, men mitt ansvar for deres personvern er det samme i oppfølgingen. Det vil bli gjort lydopptak og tatt notater av oppfølgingsintervju. Det vil bli tatt notater i observasjonen av undervisningsøkten.

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. Dette kan gjøres ved å ta kontakt med masterstudent Eirik Særsland. 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. Opplysningene som blir samlet inn i dette prosjektet vil kun være tilgjengelig for masterstudent Eirik Særsland. Opptakene vil under prosjektperioden lagres på ekstern harddisk som blir forsvarlig lagret. I alle skriftliggjøringer av datamaterialet vil både lærere og skole bli gitt fiktive navn. Deltakerne vil ikke kunne gjenkjennes i publikasjoner.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?

Prosjektet vil etter planen avsluttes 31. desember 2023. Alle lydopptak blir da forsvarlig slettet, og kun anonymiserte tekster vil bli tatt vare på.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke. På oppdrag fra Universitetet i Stavanger har Personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter

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

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

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

- Universitetet i Stavanger ved veileder og lektor Sean Peter Kåss Martin (tlf. 91 12 22 79)
- Masterstudent Eirik Særsland (tlf. 41 32 22 72)

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

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

Med vennlig hilsen

Eirik Særsland
Masterstudent
Institutt for grunnskolelærerutdanning, idrett og spesialpedagogikk

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet [*sett inn tittel*], og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i intervju

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

(Signert av prosjektdeltaker, dato)

INTERVJUGUIDE

Vedlegg 4: Intervjuguide

Intervjuguide

Lærerintervju 1. del

Guide laget av Eirik Særsland

Velkommen m/intro

- Takke for villighet til intervju
- Informert samtykke, anonymitet og taushetsplikt

Bakgrunn

- Hvilke klasser underviser du matematikk i?
- Hvordan vil du beskrive det faglige nivået på elevene dine?

God undervisning

- Hva er god matematikkundervisning for deg?
 - Hvordan arbeider du for å skape gode matematikkundervisninger?
 - Hva er nødvendige/grunnleggende elementer for de gode undervisningsøktene?
 - Hvordan arbeider du for å bygge bro mellom det teknologiske og det matematiske emnet som det undervises i?

Generelle oppfølgingsspørsmål:

- Kan du gi et eksempel?
- Kan du utdype litt?
- På hvilken måte ...?
- Hvis jeg forsto deg rett, så sa du at ...
- Hva legger du i ...?

Programmering i skolen

Introduksjon med henvisning til læreplanen og programmering inn i matematikken.

- Hvordan har det vært å skulle undervise i programmering?
 - Hvordan opplever du dine programmeringsferdigheter?
 - Har du måttet lære deg programmering etter ny læreplan kom? Hvordan?
 - Hvordan opplever du programmeringsferdighetene til andre matematikklærere?
- Hvordan ser du relasjonen mellom matematikk og programmering?
 - Hvordan **implementerer du** programmering i matematikken?
 - Hva tilfører programmering matematikkundervisningen?
 - Hvordan **implementerer du** matematikken i programmering?
 - Hvilke **temaer** i matematikken fremmer programmering?
- Hvis du gjennomfører en **matematikkøkt** som inneholder programmering du vil betegne som god, hva er det som kjennetegner denne økten?
- Hva bringer du selv med deg inn i en god **matematikkøkt** i programmering?
 - Hvilken kunnskap ser du på som nødvendig for å kunne planlegge og gjennomføre en god **matematikkøkt** i programmering?
- **Programmering for å lære programmering eller verktøy for å lære matematikk?**
- Hvor bør skolen være om fem år når det gjelder programmering?
 - Hvordan skal vi komme oss dit?
- Prioriterer ledelsen på arbeidsplassen din å gi deg mulighet til å lære programmering?
 - Hvordan?
 - Hvilke tanker har du om hvordan **f.eks. Udir/kommunen/fylket** kunne lagt bedre til rette for å gi lærerne den nødvendige kompetansehevingen de trenger for å bruke programmering i matematikkundervisningen?
- (Vise kompetansemål) Er det andre kompetansemål som programmering bør knyttes til?
- Sammenheng mellom arbeid i programmering og læring i andre kompetansemål/temaer. For eksempel. Algebra, geometri eller statistikk.

Avslutning

- Er du villig til at jeg kan komme tilbake og få intervju deg en gang til?
- Er du villig til at jeg kan komme tilbake og få observere en eller flere undervisningsøkter av deg?
- I forbindelse med at jeg holder på med et masterprosjekt om programmering i skolen, er det andre ting du tenker er relevant for meg å vite?

TRANSKRIPSJONSARBEID

Vedlegg 5: Transkripsjonsnøkkel

Funksjon	Tegn	Beskrivelse
Overlapp	[tekst] [tekst]	Blir brukt når to personer sier noe samtidig
Overtakelse	tekst~ ~tekst	Indikerer når en person overtar og fortsetter å snakke uten at det er pause imellom
Pause (≥ 1 s)	(ns) der n = antall sekunder Eks. (6s)	Pauser i antall sekunder
Kort pause (≤ 1 s)	(.)	Pauser på under et sekund
Ukjent tekst	(ukjent tekst)	Indikerer når det som blir sagt er helt ugjenkjennelig og blir ikke transkribert
Forsterkning	TEKst	Indikerer at ord eller setninger blir forsterket

Vedlegg 6: Transkripsjonsmal

Nr.	Tid	Hvem	Diskurs	Kommentar
001				