

Implementering av programmering i matematikkfaget

En studie om hvordan ungdomsskolelærere er kommet i gang med implementering av programmering i matematikkfaget, og hvilke utfordringer de har møtt i dette arbeidet.

Tove-Mette Reznig

VEILEDERE

Nils Kristian Hansen
Anders Skarpeteig Fidje

Universitetet i Agder, 2022

Fakultet for Teknologi og realfag
Institutt for matematiske fag

Forord

Dette forordet markerer innspurten på tre års masterutdanning som lærerspesialist i matematikk for 5.-10.trinn. Fagfornyelsen medførte store endringer i matematikkundervisningen. Dette ga meg som 53år gammel matematikklærer en følelse av maktesløshet. Hvordan skulle jeg klare å endre praksis så drastisk? Jeg måtte gjøre noe for å få ny inspirasjon. Resultatet ble masterstudie på UIA.

Dette har vært tre lærerike, men krevende år. Vi har vært en flott gjeng med voksne studenter som har hatt et godt samhold til tross for at store deler av studietiden har vært preget av covid-19. En del samlinger i Kristiansand har blitt erstattet med forelesninger og eksamen på zoom, til vår fortvilelse.

Det er mange jeg må takke for at jeg nå kan levere min masteroppgave. Først og fremst alle lærerne som har brukt tid i sin travle hverdag på å bidra i spørreundersøkelser og intervjuer. Kolleger som har bidratt i ulike prosjektoppgaver underveis i studiet, samt hjelp og støtte i hele prosessen. Skoleledelsen som har vært velvillige og støttende hele veien. Herlige studentkolleger som har støttende og oppmuntrende, det har vært en herlig gjeng.

Gode venninner, kolleger og svigerdatter som har stilt opp som korrekturlesere ved flere av mine innleveringer. Det har vært gull verdt.

Jeg vil også takke veilederne mine Nils Kristian Hansen og Anders Skarpeteig Fidje for god og verdifull veiledning. Dere har utfyllt hverandre og vært til fantastiske god støtte og oppmuntring.

Sist, men ikke minst vil jeg takke alle venner og familie som har støttet meg og hatt tro på meg i denne prosessen. En spesiell takk til min mann som har holdt ut med meg disse tre åra med mange opp og nedturer. Jeg lover at dette markerer slutten på alle studier for mitt vedkommende.

Tusen takk!

Tove-Mette Reznqi

Kristiansand, mai 2022.

Sammendrag

Ved innføring av ny læreplan LK20 er programmering kommet inn som en del av kompetansemålene i matematikk på i alle trinn i ungdomsskolen. Det er derfor interessant å undersøke i hvilken grad og hvordan matematikklærere er kommet i gang med implementering av programmering i matematikkfaget vel et år etter innføringen av ny læreplan. Denne studien handler om i hvilken grad og hvordan matematikklærere på ungdomstrinnet har implementert programmering i matematikkfaget. Den gir eksempler på hvilke pedagogiske muligheter programmering i matematikkfaget lærerne har erfart, og hvilke utfordringer disse lærerne har støtt på i dette arbeidet. Forskningsspørsmålene til denne studien er:

Hvordan har noen matematikklærere i ungdomsskolen implementert programmering i matematikkundervisningen? og

Hvilke utfordringer møter matematikklærere i implementeringen av programmering i matematikkfaget?

Dette er en case-studie, med kombinasjon av kvantitativ og kvalitativ forskning. Den kvantitative delen er en elektronisk spørreundersøkelse sendt til alle som underviser i matematikk i en kommune. Undersøkelsen gir svar på hvilke lærere som har startet med programmering i matematikkfaget, hvordan de har gjort det, og hvorfor noen ikke er kommet i gang med programmering i matematikkfaget. Spørreundersøkelsen er utgangspunkt for intervjuene. Det er foretatt intervju med fire lærere ved hjelp av lydopptak som er transkribert. Intervjuene hadde en semistrukturell form med kombinasjon av forberedte spørsmål og oppfølgingsspørsmål.

Resultatene viser at mange lærere ikke er kommet i gang med programmering i matematikkfaget ennå. Den største utfordringen her er manglende programmeringskompetanse hos lærerne.

De som er kommet i gang med programmering i matematikkfaget, ser en del pedagogiske muligheter: - for mer variert undervisning, mer virkelighetsnære og åpne oppgaver og større rom for tverrfaglighet og kommunikasjon. De største utfordringene er tid, klassestørrelse og ulik kompetanse hos elever og lærere.

Det teoretiske rammeverket handler om TPACK, matematisk kompetanse, pedagogiske muligheter i teknologirike klasserom og programmering i matematikkfaget. Denne teorien er med på å belyse forskningsspørsmålene og drøfte resultatene.

Abstract

With the introduction of a new curriculum (LK20), programming has been introduced as a part of the competence goals for mathematics at all levels of junior secondary school, in Norway. In light of this, it is interesting to investigate to which extent and how mathematics teachers have managed to implement programming into the subject, a good year after introducing the new curriculum.

This study looks into at which degree and how mathematics teachers in lower secondary school have implemented programming into the subject. It gives examples of which pedagogical possibilities the teachers have experienced in the subject of mathematics, and which challenges these teachers have met during this work. The research questions for this study are:

How have some teachers of mathematics in lower secondary school, implemented programming in the teaching of the subject? and

Which challenges do the mathematic teachers meet in implementing programming into the subject?

This is a case-study, and a combination of quantitative and qualitative research. The quantitative part is an electronic questionnaire sent to all mathematic teachers within a commune. The questionnaire answers which teachers have started the programming-work, how they have introduced it, and why some teachers have not started the work yet. The questionnaire is the basis for the interviews. The four interviews have been recorded and later transcribed. The interviews held a semi-structural form, combining prepared questions and follow-up questions.

The result shows that many teachers have not started the programming part of mathematics as a subject. The greatest challenge is lack of programming skills among the teachers.

Those who have started it, see some educational possibilities, more varied teaching, more real-life and open tasks, more room for co-operation between subjects and communication. The biggest challenges, however, are number of pupils in one class, and a great span in skills both with pupils and teachers.

The theoretical framework is about “TPACK”, mathematical skills, pedagogical possibilities in technological equipped classrooms and programming in the subject of mathematics. This theory is here to highlight the research questions and to discuss the results.

Innhold

Forord	3
Sammendrag	5
Abstract	6
1 Innledning.....	9
2 Teori	11
2.1 Læreplanen	11
2.2 TPACK	12
2.3 Matematisk kompetanse	17
2.3.1 Trådmodellen til Kilpatrick, Swafford og Findell	17
2.3.2 Niss og Jensens åttedelte kompetansemodell.....	18
2.4 Pedagogiske muligheter.....	20
2.5 Programmering i matematikkfaget	22
2.5.1 Hvorfor programmering i matematikkfaget?	22
2.5.2 Hvilke programmeringsspråk skal man velge for å programmere i matematikkfaget?	23
2.4.3 UMC	24
3 Metode.....	27
3.1 Datainnsamling	28
3.1.1 Spørreundersøkelsen	28
3.1.2 Intervjuene	29
3.2 Analyse	30
3.2.1 Analyse av spørreundersøkelsen.....	31
3.2.2 Analyse av intervjuene.....	32
3.3 Validitet og reliabilitet.....	33
3.4 Etikk.....	35
4 Resultat.....	37
4.1 Spørreundersøkelsen.....	37
4.1.1 Spørsmål 1: Er du kommet i gang med programmering i matematikkfaget?	37
4.1.2 Spørsmål 2: Dersom du svarte ja på spørsmål 1, kan du si litt om hvordan du gjør det?.....	37
4.1.3 Spørsmål 3: Dersom du svarte nei på spørsmål 1, kan du si litt om årsaken til dette?	38
4.1.4 Spørsmål 4: I hvilken grad føler du at du som lærer har fått nok opplæring i programmering i matematikk, for å kunne undervise i det? (skala fra 1-5, der 1 er dårligst og 5 er best).....	39

4.1.5 Spørsmål 5: Hvilke(t) trinn underviser du i matematikk dette skoleåret?	40
4.1.6 Hvilket kjønn er du? Mann eller kvinne.	40
4.2 Intervjuene	40
4.2.1 Hvordan implementere programmering i matematikkfaget?	41
4.2.2 Hvilke pedagogiske muligheter gir programmering i matematikkfaget?	43
4.2.3 Hvilke utfordringer skaper implementering av programmering i matematikkfaget?	45
5 Drøfting.	47
5.1 Spørreundersøkelsen.....	47
5.2 Intervjuene	48
5.2.1 Hvordan har matematikklærerne implementert programmering i matematikkfaget?	48
5.2.3 Hvilke utfordringer skaper implementering av programmering i matematikkfaget?	51
6 Avslutning	53
6.1. Konklusjon.....	53
6.2 Refleksjon og veien videre	55
Referanser.....	58
Vedlegg1: Kvittering fra NSD	60
Vedlegg 2: Samtykkeskjema.....	63
Vedlegg 3: Intervjuguide.....	65
Vedlegg 4: Transkripsjoner.....	66

1 Innledning

Samfunnet vi lever i i dag blir stadig mer digitalisert. Vi er omgitt av digitale verktøy privat, på skolen og i arbeidslivet. Fordi arbeidslivet nå og i fremtiden blir stadig mer avhengig av teknologisk kompetanse, begrunnes ofte programmering i skolen som en kompetanse for å lære, arbeide og leve i dagens og morgendagens samfunn. Teknologisk kompetanse handler ikke bare om kompetanse til å utvikle morgendagens teknologi og maskiner, men også kompetansen til menneskene som skal anvende dette i sin yrkesutøvelse (Sevik, 2016).

Ludvigsen-utvalget leverte i juni 2015 sin utredning om fremtidens skole som beskriver hvordan skolen må fornyes for å møte fremtidens kompetansebehov. Utvalget hevder at ved å knytte programmering opp mot matematikkfaget, kan man kombinere fagspesifikke ferdigheter med ferdigheter i programmering. De anbefalte derfor å styrke matematikkfaget i skolen (NOU 2015: 8, 2015).

Som lærer må man holde seg oppdatert for at elevene skal få opplæring i de digitale verktøy det er naturlig å bruke i de enkelte fag. Tradisjonelt har man sett på kompetansebehovet hos lærere som todelt; pedagogisk og fagdidaktisk kompetanse. Teknologisk utvikling endrer kravene til lærerens kompetanse. Dagens lærere må ha tre overlappende kompetanser: faglig, pedagogisk og digital, kalt TPACK (Koehler & Mishra, 2009). Flere forskningsinnsatser har vært rettet mot å forberede matematikklærere på å undervise med teknologier de siste tiårene (Niess, 2008). Teknologi åpner for nye designmuligheter for matematikklærerne (Pepin, Gueudet, & Trouche, 2017).

Bruk av teknologi i matematikkfaget har vært et tema i flere tiår både nasjonalt og internasjonalt. Teknologiske verktøy som regneark og graftegner har gradvis blitt innført i nasjonale læreplaner. Med innføring av LK20 er programmering kommet inn som et av kompetansemålene i matematikk på alle trinn fra 5.-10.trinn (Utdanningsdirektoratet, 2020b).

Temaet for denne studien er programmering i matematikkfaget. Dette er et viktig tema å sette seg inn i nå da dette er nytt med den nye læreplanen. LK20 legger i større grad enn tidligere vekt på at undervisningen skal være praktisk, relevant og variert. Programmering i matematikkfaget kan være et av flere verktøy som er med på å innfri dette målet.

For matematikklærere i ungdomsskolen har LK20 direkte innvirkning på arbeidshverdagen. Da programmering inngår som kompetansemål i matematikk på alle trinn på ungdomsskolen, må matematikklærere undervise i det.

Det er viktig at målene for studien står sentralt. Disse måla kan deles opp i personlige, intellektuelle og praktiske mål (Krumsvik, 2015).

Det å forske på i hvilken grad matematikklærere er kommet i gang med implementering av programmering i matematikkfaget og hvordan de har gjort dette i praksis er et personlig mål for denne studien.

Det intellektuelle målet for denne studien er knyttet til teori rundt muligheter implementering av programmering i matematikkfaget kan gi og hvordan utfordringer rundt dette kan løses.

Det praktiske målet er knyttet til hvordan lærere i praksis kan implementere programmering i sin matematikkundervisning.

Spørsmålet er om lærerne er klare for denne utfordringen. Teknologi åpner for nye designmuligheter for matematikklærere. Det er avgjørende for å få til programmering i matematikkfaget at lærerne innehar en kompleks interaksjon av innhold, pedagogikk og teknologi. Min hypotese er at mange lærere opplever at de mangler den nødvendige teknologiske kunnskapen for å sette i gang med programmering i matematikkfaget. At matematikklærere ønsker å lære seg programmering, men opplever at de ikke har den nødvendige kompetansen er et av funnene til Berggren og Jom (2019).

Fagfornyelsen har innlemmet programmering i matematikkfaget. Aktuelle spørsmål her kan være: Hva innebærer det? Hva menes med programmering i matematikkfaget? Og hvordan kan matematikklærere få det til i sin undervisning?

Derfor er det interessant å se nærmere på hvor langt matematikklærere i en middels stor østlandskommune er kommet med å implementere programmering i matematikkfaget.

Forskningsspørsmål i denne studien er:

Hvordan har noen matematikklærere i ungdomsskolen implementert programmering i matematikkundervisningen? Og

Hvilke utfordringer møter matematikklærere i implementeringen av programmering i matematikkfaget?

2 Teori

I denne delen presenteres det teoretiske grunnlaget for denne studien. Den teoretiske rammen rundt denne studien handler om LK20, TPACK, matematisk kompetanse, pedagogiske muligheter i teknologirike matematikk-klasserom og programmering i matematikkfaget.

2.1 Læreplanen

Med LK20 bestemte Utdanningsdirektoratet seg for å integrere programmering i matematikkfaget på lik linje med Sverige og Finland. Denne endringen i læreplanen er årsaken til valg av forskningsspørsmålene i denne studien.

Begrunnelsen for å ta inn programmering i matematikkfaget i skolen er at det ligger mye matematikk i det, fordi en må forstå og håndtere algoritmisk tenkemåte. Det kan være med på å inspirere elever til å interessere seg for teknologiske fag. Mange av dagens elever vil i fremtiden være med på å utvikle teknologi som er viktig for samfunnet. Derfor trenger alle elever en utdanning som gjør dem i stand til å lære og forstå de grunnleggende prinsippene som ligger bak programmering (Kaufmann, Stenseth, & Holone, 2018).

I LK20 er algoritmisk tenking blitt et begrep, og en av hovedgrunnene til at programmering er blitt implementert i matematikkfaget.

«Å tenke algoritmisk er å vurdere hvilke steg som skal til for å løse et problem, og å kunne bruke sin teknologiske kompetanse for å få en datamaskin til å løse (deler av) problemet. I dette ligger også en forståelse av hva slags problemer/oppgaver som kan løses med teknologi og hva som bør overlates til mennesker. Algoritmisk tenkning er den norske oversettelsen av det engelske computational thinking» (Utdanningsdirektoratet, 2019).

Det finnes mange definisjoner på algoritmisk tenking, men de fleste er sammenfallende og noen av nøkkelbegrepene finner vi i figur 1, hentet fra Utdanningsdirektoratets sider (Utdanningsdirektoratet, 2019). Algoritmisk tenking handler om å bryte ned komplekse problemer i mindre, og mer håndgripelige deler som lettere lar seg løse. Det handler om å jobbe seg gjennom problemene steg for steg, prøve og feile og ikke gi seg. I slike prosesser er det en fordel å samarbeide og å dele tanker og løsningsstrategier. Man må være nøyaktig og analytisk, men også skapende og utprøvende, og ikke være redd for å gjøre feil.



Figur 1 modell av den algoritmiske tenkeren hentet fra Utdanningsdirektoratet 2019.

LK20 har følgende kompetansemål i programmering i matematikk for ungdomsskolen:

8.trinn: *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 (Utdanningsdirektoratet, 2020b).*

2.2 TPACK

Innføring av programmering i matematikkfaget krever en utvikling av lærernes TPACK.

For å undersøke lærernes kompetanse med tanke på programmering, vil lærernes utsagn i denne studien ses i lys av begreper i TPACK-modellen.

Koehler og Mishra (2009) beskriver i sin artikkel et rammeverk for teknologisk pedagogisk innholdskunnskap, TPACK for lærere. Dette bygger på Lee Shulmans konstruksjon av pedagogisk innholdskunnskap PCK, men inkluderer teknologikunnskap.

Undervisning er en kompleks praksis som krever en sammenveving av mange typer spesialisert kunnskap. TPACK er en kompleks interaksjon mellom: fagkunnskap, pedagogikk og teknologi. Utviklingen av nyere digitale teknologier utfordrer lærerne på nye måter da disse er ustabile og ugjennomsiktige. Vi kan ikke se hva datamaskinene gjør og det utvikles stadig nye programmer for undervisning (Koehler & Mishra, 2009). Dette er et argument for å

jobbe med programmering i og med at vi da ser hva datamaskinene (eller programmene) faktisk gjør.

Sosiale og konvensjonelle faktorer kompliserer også forholdet mellom undervisning og teknologi. Sosiale og institusjonelle sammenhenger er ofte ikke støttende til lærernes innsats for å integrere teknologi i undervisningen. I tillegg har mange lærere utilstrekkelig erfaring med bruk av digital teknologi for undervisning, og opplever derfor at de ikke tilstrekkelig forberedt på å bruke ny teknologi i klasserommet. Å tilegne seg ny kunnskap kan være utfordrende, spesielt dersom det er en tidsintensiv aktivitet som må passe inn i en ellers travel timeplan. For at lærerne skal ta ny teknologi i bruk i sin undervisning er det viktig med tilstrekkelig opplæring (Koehler & Mishra, 2009).

Figur 2 viser hvordan fagkunnskap, pedagogisk kunnskap og kunnskap om teknologi går i hverandre som et venndiagram.

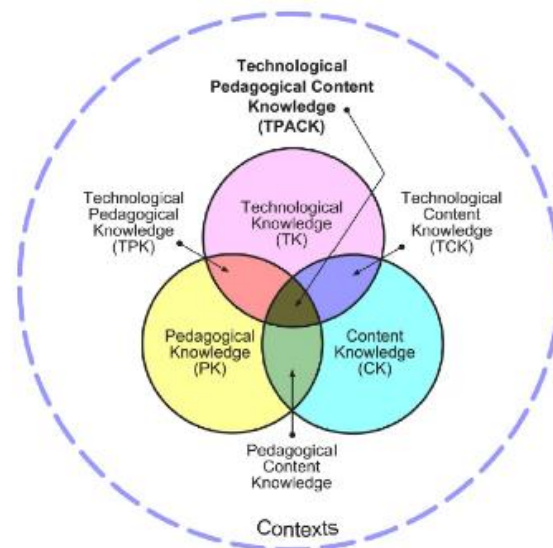
For å undervise er det viktig at lærere har god *fagkunnskap*, at de kan det faget de skal undervise i.

De må ha *pedagogisk kunnskap*, det vil si at de vet hvordan de skal undervise. At de vet hvordan elevene lærer, har god klasseromsledelse, kan planlegge undervisningen og kan vurdere elevaktiviteter. *Pedagogisk innholdskunnskap* (PCK) er i samsvar med Shulmans idé, en forestilling om transformasjon av emnet for undervisning. Dette skjer når læreren tolker emnet, finner flere måter å presentere det på og skreddersyr undervisningsmaterieell til elevenes forkunnskaper (Koehler & Mishra, 2009).

Teknologikunnskap er ingen ensartet kunnskap, da teknologien utvikler seg hele tiden. Men det finnes visse måter å arbeide med teknologi som kan gjelde for alle teknologiverktøy og ressurser. *TK* som brukes i TPACK-rammeverket krever at man forstår informasjonsteknologi godt nok til å anvende det produktivt på jobben og i hverdagen for å gjenkjenne når den kan hjelpe til med å nå et mål for informasjonsbehandling, kommunikasjon eller problemløsning. *TCK* - handler om forståelsen av hvordan *teknologi og fagkunnskap* påvirker og begrenser hverandre. Lærere må kunne mer enn faget de underviser i, de må også ha forståelse for hvordan faget kan endres ved anvendelse av teknologi. De må vite hvilken teknologi som er best egnet i temaene innen sitt fag, og hvordan faget eller emnet kan være med på å endre teknologien (Koehler & Mishra, 2009).

TPK- teknologisk pedagogisk kunnskap handler om hvordan undervisning og læring kan endres når teknologier brukes på bestemte måter. Dette innebærer å kjenne til pedagogiske begrensinger ved en rekke teknologiske verktøy. Man må for eksempel være bevisst på at en rekke populære programvarer ikke er utviklet for pedagogiske mål, men for

forretningsmiljøer, underholdning og kommunikasjon. Lærere må derfor vurdere hvordan man kan tilpasse disse teknologiene til pedagogiske formål (Koehler & Mishra, 2009). *TPACK- kunnskap om teknologi, pedagogikk og fag (innhold)* handler om interaksjoner mellom disse tre kunnskaper, og er ulik kunnskap om hver av de tre kunnskapene individuelt. Det handler om lærerens evne til å navigere fleksibelt mellom fagkunnskap, pedagogikk og teknologi, og den komplekse interaksjonen mellom disse (Koehler & Mishra, 2009).



Figur 2 Venn diagram for TPACK-modellen hentet fra <http://tpack.org/>

Niess (2005) viser til fire sentrale komponenter i TPACK som troen på det som en matematikklærers kunnskap er i samsvar med:

1. *En overordnet forestilling om formålene for å innlemme teknologi i undervisningen i matematikk.* Denne forestillingen er hva læreren vet og tror om matematikkens natur, det som er viktig for elevene å lære, og hvordan teknologi støtter læring av matematikk. Disse fundamentene av lærerens kunnskap og tro på undervisning i matematikk med teknologi tjener som grunnlag for deres avgjørelser om klasseromsundervisning (mål, strategier, oppgaver, læreplan og evaluering av elevens læring).
2. *Kunnskap om elevenes forståelse, tenkning og læring i matematikk med teknologi.* På dette området stoler læreren på og opererer ut ifra kunnskap om hvordan elevene lærer matematikk med teknologi og mener at teknologi er nyttig for å lære matematikk.
3. *Kunnskap om læreplaner og læreplanmaterieell som integrerer teknologi i læring og undervisning i matematikk.* Læreren vurderer ulike tilgjengelige teknologier for å undervise i

ulike emner og hvordan temaene og ideene i et teknologiforbedret miljø med bekymring for hvordan aktivitetene er organisert, og hvordan det kan integreres gjennom hele læreplanen.

4. *Kunnskap om instruksjonsstrategier og representasjoner for undervisning og læring av matematikk med teknologi.* Når det gjelder undervisning og læring, tilpasser læreren matematiske representasjoner med teknologi på flere måter for å oppfylle spesifikke instruksjonsmål og behovene til bredden av elever i klassen (Niess, 2008).

Studien til Niess (2008) viser at lærerne i ulik grad tar i bruk ny teknologi. Dette handler ikke om at de enten har eller ikke har TPACK for undervisning i matematikk med passende teknologier. De er forskjellige i sine handlinger med hensyn til hver av komponentene når de blir konfrontert med om de skal akseptere eller avvise bruken av ulike teknologier i undervisningen i matematikk. Forskjellene deres er en funksjon av deres kunnskap om matematikk, deres kunnskap om teknologiene, og deres kunnskap om undervisning og pedagogikk. Niess trekker videre frem Rogers (1995) sine fem nivåer:

1. *Å anerkjenne* (kunnskap) der lærere er i stand til å bruke teknologiene og gjenkjenne tilpasning av evnene til teknologiene med matematisk innhold.
2. *Aksept* (overtalelse) der lærere danner en gunstig eller ugunstig holdning til å integrere passende teknologi i matematikkundervisningen.
3. *Tilpasning* (beslutning) der lærere deltar i aktiviteter som fører til et valg om å adoptere eller avvise passende teknologi i matematikkundervisningen.
4. *Utforske* (implementering) der lærere aktivt integrerer teknologi i matematikkundervisningen.
5. *Fremgang* (bekreftelse) der lærere vurderer resultatene av beslutningen om å integrere passende teknologi i matematikkundervisning (Niess, 2008).

Niess (2008) konkluderer med at hvis utdanning handler om å forberede innbyggerne på å leve og arbeide i det tjueførste århundret, må lærerne være forberedt på å integrere ny teknologi som læringsverktøy i sin undervisning i matematikk-klasserommet.

Ruthven (2014) retter i sin artikkel et kritisk blikk på måten TPACK-modellen er blitt brukt på. Han mener det ligger en tvetydighet i måten det noen ganger er blitt fokusert på oppmerksomheten på hele systemet med to- og treveis interaksjon mellom komponentene i TPACK-bildet. Andre ganger brukes begrepet til å plukke ut treveis skjæringspunkt i kjernen, som ellers blir referert til som TPCK. For det andre er karakteren i interaksjonen mellom kunnskapsdomenet undervurdert, det speiler de ulike definisjonsstyrkene funnet i gjeldende

bruk av pedagogisk innholdskunnskap (PCK): fra en svak definisjon som ikke krever mer enn en enkel kombinasjon av felles kunnskap om innhold med generisk pedagogisk kunnskap, til en sterkere definisjon som insisterer på at PCK blir understreket av noen særegne innholdsspesifikke pedagogiske resonneringer. For det tredje er det et hierarki implisitt i merkeregulene der innhold er mer grunnleggende enn pedagogikk, og begge disse er mer grunnleggende enn teknologi (Ruthven, 2014).

Til slutt mener Ruthven (2014) at det er en tvetydighet om nivået der det pedagogiske og det teknologiske er unnfanget: mellom et mer konkret nivå der kunnskap tas i forhold til en bestemt pedagogikk eller teknologi, og et mer refleksivt metanivå der disse begrepene er forbeholdt kunnskap om pedagogisk eller teknologiske alternativer.

I denne studien er det interessant å se nærmere på hvordan lærerne opplever at det innbyrdes forholdet mellom innhold (i dette tilfellet matematikk), pedagogikk (undervisning) og teknologi (i dette tilfellet programmering) utspiller seg i implementeringen av programmering i matematikkfaget. Er det matematikken som legger grunnlaget for undervisningen, eller er det programmeringsaktivitetene som styrer hvilke temaer innen matematikk det undervises i?

Berggren og Jom (2019) legger frem resultatene fra en spørreundersøkelse om holdninger til innføring av programmering i matematikkfaget blant norske lærere. Spørreundersøkelsen var anonym og ble foretatt blant 30 barneskolelærere.

Denne undersøkelsen konkluderer med at matematikklærerne er motiverte for innføring av programmering i matematikkfaget, 23 av 30 mente det var bra at programmering ble innført i matematikkfaget, mens 7 svarte vet ikke, det vil si at ingen svarte benektende på dette spørsmålet. Mange mente de hadde for lite kompetanse til å kunne si noe om innføring av programmering i matematikkfaget vil ha noen innvirkning på deres undervisning. 80% av disse lærerne mener elevene vil bli mer motiverte i matematikk ved å innføre programmering i matematikkfaget. 26 av 30 matematikklærerne kunne tenke seg å lære programmering for å kunne undervise i det. De fleste vil lære seg å programmere og hvordan de skal undervise i det. Men 21 av 30 lærere vil gjerne ha ferdige opplegg de kan bruke direkte i klasserommet.

Artikkelforfatterne poengterte også at det er viktig at lærerne får nødvendig opplæring for at Utdanningsdirektoratet skal lykkes med ambisjonene om at programmering skal implementeres i matematikkfaget (Berggren & Jom, 2019).

Ved å få mer opplæring i programmering vil lærerne kunne utvikle sin TPACK med hensyn til TK- teknologikunnskap. Ved å utvikle sin TK kan man utvikle sin TCK, hvordan man kan

knytte programmering til faget matematikk. Ved å utvikle sin kunnskap om programmering, og programmering og matematikk vil man også kunne få ideer til hvordan man kan implementere programmering i matematikkfaget, derfor er disse teoriene viktig for denne studien.

2.3 Matematisk kompetanse

Målet for all matematikkundervisning må være at elevene skal tilegne seg matematisk kompetanse. Derfor må målet med implementering av programmering i matematikkfaget også være at elevene skal tilegne seg matematisk kompetanse. Dette har vært fokus ved utforming av noen av spørsmålene i intervjuguiden. Derfor er teori omkring matematisk kompetanse viktig for denne studien.

Det finnes flere modeller som beskriver matematisk kompetanse. I denne studien blir Kilpatrick's trådmodell for matematisk kompetanse (Kilpatrick, Swafford, & Findell, 2001), og Niss og Jensens åttedelte kompetansemodell (Niss & Jensen, 2002) presentert. Disse to modellene har noen overlappende kompetansebegrep, og noen særegne.

2.3.1 Trådmodellen til Kilpatrick, Swafford og Findell

Programmering i matematikk er nært knyttet til problemløsning. I boken «Adding it up: Helping Children Learn Mathematics» (Kilpatrick et al., 2001) hevder forfatterne at elevenes arbeid med problemløsning vil kunne føre til at de arbeider med alle de fem trådene i sin trådmodell (se figur 3), for derigjennom kunne oppnå matematisk kompetanse. Kilpatrick's modell består av fem tråder som er tett sammenvevd. De er avhengige av hverandre og må ses som en helhet.

Anita Valenta (2016) har oversatt disse kompetansene i sin artikkel på matematikksenterets sider:

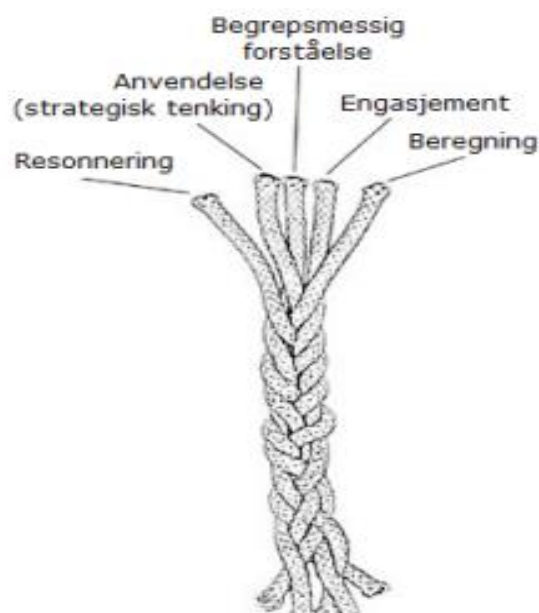
-*Conceptual Understanding* - Begrepsmessig forståelse – av matematiske begreper, operasjoner og relasjoner

-*Proceduan Fluency* - gjennomføring av prosedyrer fleksibelt, effektivt og nøyaktig

-*Strategic competence* - Anvendelse – gjenkjenning og formulering av matematiske problemer, utvikling av løsningsstrategier

-*Adaptiv reasoning* - logisk tenking, refleksjon og argumentasjon.

-*Productive Diposition* - Engasjement - se matematikk som fornuftig og verdifull, ha tro på at det er mulig å bli kompetent i matematikk (Valenta, 2016).



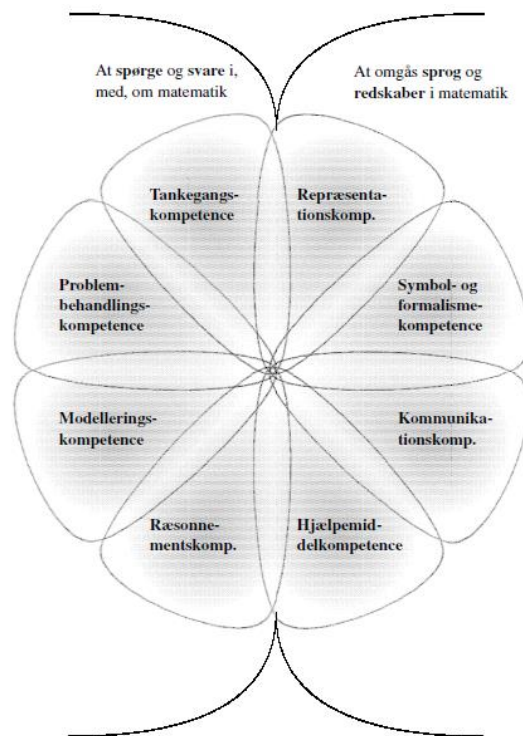
Figur 3 Kilpatrick's trådmodell for matematisk kompetanse hentet fra matematikksenteret.no

I arbeidet med programmering i matematikkfaget er det viktig å ha begrepsmessig forståelse. Både matematiske begreper og spesifikke begreper for programmeringsspråk er viktig for elevene å tilegne seg. Å gjennomføre prosedyrer nøyaktig og effektivt er spesielt viktig når man arbeider med programmering, ellers vil ikke programmet man har laget fungere. Da vil logisk tenking og refleksjon, samt å utvikle løsningsstrategier være en forutsetning. Når man implementerer programmering i matematikkfaget, er det viktig å gjenkjenne matematiske problemer, slik at programmeringen ikke blir noe som står på siden av faget, men en integrert del av det.

For at elevene skal opparbeide seg matematisk kompetanse, er engasjement viktig. Uten at elevene engasjerer seg i faget, er det vanskelig å opparbeide seg en kompetanse i det.

2.3.2 Niss og Jensens åttedelte kompetansemodell

I Danmark ble det i år 2000 nedsatt en arbeidsgruppe utnevnt av utdanningsstyrelsen i samarbeid med naturvitenskaplig utdanningsråd, som skulle se på kompetansebegrepet matematikk og nye ideer for fornyelse av matematikkfaget. Niss og Jensen (2002) presenterer i sin rapport åtte sentrale kompetanser som de deler i to hovedkategorier (se figur 4).



Figur 4 Modell for matematisk kompetanse hentet fra Niss og Jensen 2002, s.45

Venstre side av modellen: å spørre og svare, i, med, om matematikk. Denne hovedkategorien inneholder fire kompetanser: tankegangskompetanse, problemløsningskompetanse, modelleringskompetanse og resonneringskompetanse. Tankegangskompetanse innebærer å være klar over arten av matematiske spørsmål og svar, å forstå og kjenne til matematiske begreper, og å skille mellom matematiske utsagn. I denne studien er det interessant å undersøke om programmering kan være med på å hjelpe elevene til å forstå enkelte matematiske begreper. Problemløsningskompetanse handler om å formulere og løse matematiske problemer. Modelleringskompetanse omhandler modellanalyse og modellbygging. Resonneringskompetanse handler om å forstå matematiske resonneringer, og å forstå og begrunne et bevis (Niss & Jensen, 2002).

Problemløsningskompetanse, tankegangskompetanse og resonneringskompetanse er viktige egenskaper elever må inneha for å jobbe med programmeringsoppgaver (Husain, Kamal, Ibrahim, Huddin, & Alim, 2017). Derfor er disse ideene til Niss og Jensen relevante for denne studien.

Høyre side av modellen: å omgås språk og redskaper i matematikk har følgende fire delkompetanser: representasjonskompetanse, symbol- og formalismekompetanse, kommunikasjonskompetanse og hjelpemiddelkompetanse. Representasjonskompetanse er å

forstå og bruke ulike matematiske representasjoner. Symbol- og formalismekompetanse betyr å avkode, behandle og bruke ulike symboler og formler. Kommunikasjonskompetanse innebærer å kunne forstå og uttrykke seg både skriftlig, muntlig og visuelt (Niss & Jensen, 2002).

Hjelpemiddelkompetanse vil si å ha kjennskap til ulike matematiske og tekniske hjelpemidler man kan bruke i matematikken. Det kan blant annet være tabeller og linjal, men også kalkulatorer og datamaskiner (Niss & Jensen, 2002).

Representasjonskompetanse, symbol og formalismekompetanse og kommunikasjonskompetanse er nødvendige egenskaper for å jobbe med programmeringsspråk som for eksempel Scratch (Husain et al., 2017). Derfor er disse ideene relevante for denne studien som handler om implementering av programmering i matematikkfaget.

2.4 Pedagogiske muligheter

Integreringen av digital teknologi konfronterer lærere, pedagoger og forskere med mange spørsmål: Hva er potensialet i IKT for læring og undervisning? Hvilke faktorer er avgjørende for å få det til å fungere i matematiske klasserom? Drijvers (2013) tar for seg seks tilfeller av ledende studier i feltet for å finne avgjørende suksessfaktorer.

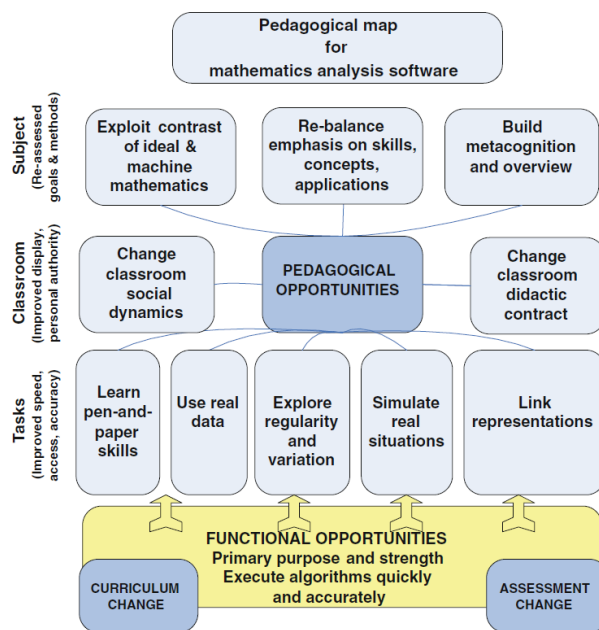
Hvilke suksessfaktorer matematikklærere ser ved implementering av programmering i matematikkfaget, er et av spørsmålene som blir stilt i denne studien. Begrunnelsen for valg av dette spørsmålet, var fokus på positive erfaringer hos informantene. Det er da interessant å se på lærernes utsagn om pedagogiske muligheter i lys av begrepene i det pedagogiske kartet til Pierce & Stacey (2010).

I flere tiår har flere interessenter fremhevet potensialet i digitale teknologier for matematikkutdanningen. Drijvers (2013) konkluderer med at det er tre avgjørende faktorer: det første gjelder design, det vil si utforming av den digitale teknologien som er involvert, utforming av tilsvarende oppgaver og aktiviteter, og utforming av leksjoner. I den senere tid har man fått et mer realistisk syn på relasjonen mellom digital teknologi, elevenes tenking, og papir-og-blyant arbeid. Den andre faktoren er lærerens rolle, som ikke er ubetydelig for at elevene skal lykkes.

Her er det nødvendig med en prosess med faglig utvikling som inkluderer lærerens TPACK, se kapittel 2.2. Det pedagogiske kartet (figur 5) bidrar til å forstå hva som er annerledes i

undervisning med teknologi, og hvordan lærere engasjerer seg i en prosess med faglig utvikling. Den tredje faktoren gjelder den pedagogiske konteksten, som inkluderer matematisk praksis og elementer i det pedagogiske kartet designet av Pierce og Stacy (2010), (figur 5), (Drijvers, 2013).

Robyn Pierce & Kaye Stacey (2010) introduserte ideen om et pedagogisk kart i sin artikkel: «*Mapping Pedagogical Opportunities Provided by Mathematics Analysis Software*». Dette pedagogiske kartet ser på hvilke pedagogiske muligheter som ligger i det digitale klasserommet.



Figur 5 Pierce & Staceys pedagogiske kart (Pierce & Stacey 2010).

Selv om denne modellen er bygd opp omkring matematiske analyseprogramvarer som for eksempel Geogebra, kan de samme pedagogiske momentene spille en sentral rolle ved innføring av programmering i matematikklasserommet. I LK20 er programmering nå blitt en del av kompetansemålene i matematikk fra 5.-10. trinn. Det vil kunne føre til endringer i klasserommets sosiale dynamikk ved at elever som tidligere ikke var så interessert i matematikk får interesse gjennom programmeringsaktiviteter. Implementering av programmering kan føre til endring av didaktikk ved at man går fra å løse matematiske problemer med penn og papir til å lage programkoder som kan løse problemene for oss. Dette åpner muligheter for at elevene får mer kontroll over sin egen læring, at de blir mer delaktige i sin egen læringsprosess. Det åpner også opp for større grad av dybdelæring og mer samarbeid mellom elever, dersom lærer tillater det (Pierce & Stacey, 2010). Med LK20 er

kommunikasjon et av kjerneelementene i matematikk, og det er sentralt at elevene skal samtale om matematiske problemer og resonnementer i matematikklasserommet.

Det kan også føre til endring av emner, og oppgavene vil kunne endre seg noe (Pierce & Stacey, 2010). For eksempel kan man få mer avanserte oppgaver som følge av at programmet utfører beregningene for elevene.

2.5 Programmering i matematikkfaget

2.5.1 Hvorfor programmering i matematikkfaget?

Temaet i denne studien er programmering i matematikkfaget, og hvordan lærere er kommet i gang med dette i sin undervisning. Derfor er det viktig for denne studien å se nærmere på hvorfor man skal programmere i matematikkfaget.

Programmering ble tatt i bruk i skolen allerede på slutten av sekstitallet da Paperts programmeringsspråk LOGO ble lansert i 1967. Målet var at elevene skulle bli flinkere i problemløsning. På åttitallet ble det gjort flere forsøk med programmering i skolen. Dette ble en aktivitet for spesielt interesserte. En av grunnene til dette var at skolene hadde begrenset med utstyr, og at lærere flest ikke hadde den nødvendige kompetansen (Kaufmann et al., 2018).

I LK06 ble digital kompetanse en del av de grunnleggende ferdighetene, og programmering kom igjen på dagsordenen. Et argument for å innføre programmering i matematikkfaget i Norge, er «algoritmisk tenkemåte». Kaufmann mfl (2018) viser til flere studier som tyder på at programmering fører til økt motivasjon hos elevene for å lære matematikk, at det fører til økt fremgang for elever, og at elevenes problemløsningsferdigheter og tankegangskompetanse øker.

Et annet argument for innføring av programmering i matematikkfaget er at programmering er med på å utvikle elevenes logiske tankegang og evne til problemløsning og samarbeid (Kaufmann et al., 2018).

Gjennom programmering vil elevene møte matematikken på en annerledes måte enn gjennom arbeid i læreboka. I læreboka blir elevene presentert for regler og sammenhenger, for så å anvende disse for å løse oppgaver. Gjennom programmering må elevene selv sette seg inn i og beskrive reglene og sammenhengene korrekt dersom programmet skal virke, på denne måten får elevene større begrepsforståelse (Kaufmann et al., 2018).

2.5.2 Hvilke programmeringsspråk skal man velge for å programmere i matematikkfaget?

I oppstarten av implementering av programmering i matematikkfaget, er det naturlig å se nærmere på ulike programmeringsspråk, og se litt nærmere på et blokkprogrammeringsspråk og et tekstbasert programmeringsspråk. Her går jeg litt nærmere inn på hva Scratch og Python er, da det er de to mest brukte programmeringsspråkene i dag (Haraldsrud, Sveinsson, & Løvold, 2020). Det vil derfor være naturlig å se på lærernes erfaringer med undervisning i programmering ved hjelp av disse programmeringsspråkene.

Scratch er et blokkprogrammeringsspråk som er velegnet til introduksjon av programmering eller koding for barn og ungdom. Tanken er at språket skal være et steg inn i programmeringens verden for å kunne lære seg hvordan tankegangen bak programmering er, uten å måtte sette seg inn i syntaksen til et tyngre tekstbasert språk. Mange av elevene som kommer fra barneskolen til ungdomsskolen, vil ha brukt blokkprogrammering fra før, og lærer på ungdomsskolen vil ha nytte av å sette seg inn i dette for å kunne lette overgangen fra blokk til tekst. Det vil også være nyttig å bruke blokkprogrammering side om side med tekstbasert språk (Haraldsrud et al., 2020).

Fra og med femte klasse nevnes programmering i læreplanene. For mange vil blokkprogrammering være en enklere vei inn i programmeringskonseptene. Her er det lettere å lage mer visuelle programmer, som for eksempel animasjoner av dyr og mennesker, lyd og bilder. Dette kan være en motiverende faktor (Haraldsrud et al., 2020).

En studie av Fengfeng Ke (2014) konkluderer med at ungdomsskoleelevers erfaringsdrevne spilldesignprosesser i Scratch bidro til å aktivere elevenes refleksjon over hverdagslige matematiske opplevelser (Ke, 2014).

Ideen er at språket skal være et steg inn i programmeringens verden for å kunne lære hvordan tankegangen bak programmering er, uten å måtte sette seg inn i syntaksen til et tyngre tekstbasert språk. For en del elever som kommer fra barneskolen til ungdomsskolen vil kun blokkprogrammering være kjent, og lærere på ungdomsskolen bør sette seg inn i dette for å lette overgangen fra blokk til tekst. For mange elever, og for mange av målene i ungdomstrinnet, vil det også være fint å bruke blokkprogrammering side om side med tekstbasert språk. Selv om elever vil kjenne til konstruksjoner som vilkår, løkker og funksjoner fra Scratch eller andre blokkbaserte språk vil antakelig overgangen være stor for mange (Haraldsrud et al., 2020).

Den største utfordringen i overgangen fra blokk til tekstbasert programmering er å skrive riktig. Med blokkprogrammering er det lite vi kan gjøre feil, da blokker som ikke hører sammen heller ikke vil henge sammen. Det er ingen ord, parenteser, kolon man må huske eller sette på rett plass, eller biblioteker som må importeres for å få tilgang til funksjoner (Haraldsrud et al., 2020).

Når man går over til tekstprogrammering, vil mye av hjernekapasiteten gå med til å skrive riktig syntaks, det vil si at det er mindre hjernekapasitet igjen til selve matematikken. Det er derfor viktig å begynne med enkel matematikk i begynnelsen, for at ikke elevene skal miste motet og blir negativt innstilt til programmering (Haraldsrud et al., 2020).

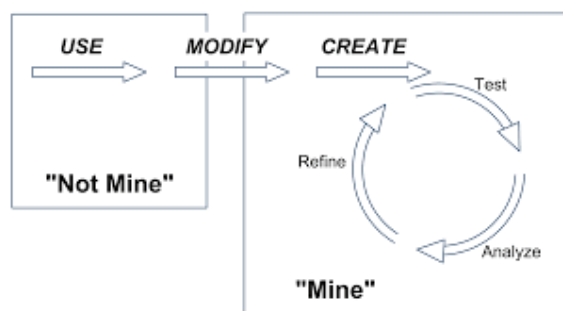
Når elevene er blitt kjent med tekstprogrammeringen og fått utforsket litt, kan man begynne med mer avansert matematikk. En måte å gjøre overgangen lettere på er å «oversette» oppgaver de har løst i Scratch til Python. Det er ofte stor forskjell på hvor raskt elevene lærer seg Python-språket, det kan derfor være lurt å la elevene jobbe sammen to og to eller tre og tre med oppgavene (Haraldsrud et al., 2020).

Det finnes mange alternative programmeringsspråk, men det viser seg at Python er det mest brukte programmeringsspråket. Noen grunner for det er: det er gratis, det kan kjøres på ulike plattformer: Windows, Mac og Linux, det har et kraftig språk for beregning, det får programmereren til å lage programkode med god visuell struktur, det er populært og utbredt, og det er svært mye materiell fritt tilgjengelig på nettet (Haraldsrud et al., 2020).

2.4.3 UMC

Fokus i denne studien er implementering av programmering i matematikkfaget, noe som er nytt med innføringen av LK20 skoleåret 2020/2021. Det finnes mange pedagogiske muligheter når man skal introdusere nye temaer i et fag.

Use-Modify-Create - (UMC) ble utviklet av Irene Lee og Fred Martin og konseptualisert i 2011, etter å ha blitt sammenliknet med den produktive integrasjonen av *CT- «Computational thinking»* som på norsk er oversatt til algebraisk tenking av Utdanningsdirektoratet (Utdanningsdirektoratet, 2019). Siden den gang er UMC-modellen blitt fremmet som et middel til å hjelpe elever og lærere med å komme i gang med å implementere programmering i matematikklasserommet (Martin, Lee, Nicholas, Sentance, & Lao, 2020).



Figur 6: UMC-modell hentet fra Lee mfl 2011

Irene Lee og Fred Martin så i sitt arbeid med ungdom og CT-computational thinking, på ungdomsskole- og videregående nivå, behovet for konkrete tiltak for å støtte innlæringen. De utviklet derfor en tretrinnsmodell for å lette innlæringen både for elever og lærere. Denne modellen går ut på å gi elevene et ferdig program de skal prøve ut, *Use*.

Når de har jobbet med dette programmet en stund, skal de endre noen av komponentene i programmet, *Modify*. I denne fasen er det viktig å gi elevene rom for kreativitet. Her er det nødvendig at elevene har en forståelse for abstraksjon og automatisering som finnes i programmet. Gjennom en rekke endringer utvikles nye ferdigheter og forståelser. På denne måten går elevene fra å bruke et program andre har utviklet til gradvis å utvikle sitt eget program. Innenfor *Create*-stadiet vil tre viktige aspekter ved algoritmisk tenking være tilstede: abstraksjon, automatisering og analyse (Lee et al., 2011).

Marianne Maugesten, Henrik Stigberg og Susanne Koch Stigberg ved Høyskolen i Østfold, har gjennomført et prosjekt der ungdomsskolelærere har prøvd ut programmering i matematikk og delt erfaringer i etterkant. Resultatet av dette prosjektet er presentert i Maugesten et al. (2014).

Høyskolen i Østfold prøvde i sitt prosjekt ut arbeidsmåten Use - Modify – Create (UMC) både i blokkprogrammering og tekstprogrammering, noe lærerne ga gode tilbakemeldinger på. Denne metoden er brukt i USA for å støtte elevens utvikling av algoritmisk tenking (Lee et al., 2011). Prosjektlederne ved Høyskolen i Østfold har gjennomført et prosjekt der ni ungdomsskolelærere har deltatt på fem helgesamlinger der de har gitt dem input med oppgaver de skulle prøve ut i egne klasser, med påfølgende fokusintervjuer der lærerne diskuterte erfaringer. Målet var å undersøke hvordan programmering kan støtte matematikken, og hvordan man kan gi et tilbud til matematikklærere med liten eller ingen

erfaring med programmering. Matematikklærerne i dette prosjektet fikk input fra prosjektlederne med ferdige opplegg de skulle prøve ut i sine klasser. I etterkant ble det foretatt fokusgruppeintervju der lærerne diskuterte erfaringer med utprøvingen i egne klasser. Prosjektlederne oppgir tre grunner for å prøve ut UMC i programmering i matematikkfaget: 1. man kan bruke ferdiglagde koder, som tydelig viser sammenhenger mellom matematikk og programmering, 2. det sparer tid for lærerne og 3. det hjelper lærere som ikke har mye programmeringserfaring, med å komme i gang (Maugesten, Stigberg, & Stigberg, 2021).

Metoden UMC- Use – Modify - Create fungerer slik: USE - innebærer at man i undervisningssituasjonen introduserer en ferdig kode elevene kan prøve ut. MODIFY- i denne fasen skal elevene prøve å forstå koden og endre noen av forutsetningene. Her er det viktig at elevene ikke føler at de ødelegger noe, men kjenner på utforsker glede og utforskertrang. CREATE - I denne fasen er det meningen at elevene kan gjøre større endringer i kodene, og eventuelt lage en helt ny kode.

Resultatet av dette prosjektet viser at ved å benytte UMC har lærere og elever lettere igangsatt samtaler og diskusjoner om kodene og resultatene de observerte både i Scratch og Python. Lærerne mente at programmeringsaktiviteten høynet kvaliteten på den matematiske samtalen. Dette er avhengig av lærerens rolle, ved at matematiske begreper kobles til muligheter i kodene, og at man stiller de rette spørsmåla. Flere lærere trakk også frem at denne arbeidsmåten ga gode muligheter for å tilpasse opplæringen fordi kodene fungerte som rike oppgaver. Alle kunne bruke dem, og de ga muligheter til utvidelse ut ifra elevenes forutsetninger. De mente også at UMC var spesielt godt egnet til tekstprogrammering (Maugesten et al., 2021).

Siden dette prosjektet ved Høyskolen i Østfold viste at UMC-modellen kan være med på å lette elever og læreres arbeid med innføring av programmering i matematikkfaget, kunne det være interessant å se på om lærerne i denne studien har brukt disse ideene i sin praksis. Det ble derfor innlemmet som et av de utfyllende spørsmålene under intervjuene.

3 Metode

Dette kapitlet handler om forskningsdesign og forskningsmetoder brukt for å belyse forskningsspørsmålene i denne studien.

Fokuset for denne studien er i hvilken grad ungdomsskolelærerne i en middels stor østlandskommune er kommet i gang med å implementere programmering i matematikkfaget og hvordan de gjør det. For å få et innblikk i hvordan dette gjøres i praksis, kunne man valgt ulike forskningsdesign: observasjon av undervisning, intervju av elever, intervju av lærere eller spørreundersøkelse. Valget i denne studien ble en kombinasjon av spørreundersøkelse og dybdeintervju. Intervju som forskningsdesign er en effektiv måte å få innblikk i hvordan lærere tenker rundt sin egen praksis. Her kan man få mye informasjon på kort tid. Dersom man skulle brukt observasjon, ville det tatt lenger tid og man ville kun fått innblikk i hvordan lærerne utfører sin praksis, men man ville ikke fått informasjon om hvordan lærere tenker rundt egen praksis.

Hovedpoenget ved bruk av spørreundersøkelse i forkant var muligheten til å undersøke hvilke informanter det kunne være interessant å velge ut til dybdeintervju. Samtidig ga den en del tilleggsinformasjon som var interessant for denne studien.

Dette er en case-studie, med kombinasjon av kvantitativ og kvalitativ forskning, også kalt «blandede metoder» («mixed methods») (Kvale & Brinkmann 2015). Denne type kombinert studie egner seg godt til å studere fenomenene slik de forekommer i den virkelige verden på et gitt tidspunkt, og den egner seg for bruk av ulike kilder og ulike former for datainnsamling, der kvantitative og kvalitative data med fordel kombineres (Fuglseth & Skogen, 2012).

Spørreundersøkelsen tar for seg graden av implementering, og intervjuene omhandler hvordan lærerne jobber med å implementere programmering i matematikkfaget.

Bruk av spørreskjema som datainnsamlingsmetode er som regel en *survey*-design (Holand, 2012). Det vil si en undersøkelse som tar sikte på å trekke ut spesifikke data fra en bestemt gruppe mennesker, på et gitt tidspunkt. I denne studien fra matematikklærere i en østlandskommune, høsten 2021. Det er mange ting å ta hensyn til når man skal lage et slikt spørreskjema. Spørsmålene bør være endimensjonale, de bør ikke være ledende, de bør være korte og konsise og de bør være forståelige og forstås likt av alle. Når det gjelder rekkefølgen på spørsmålene, bør man innlede og avslutte med enkle spørsmål, og ha de viktigste og vanskeligste spørsmålene i hoveddelen (Holand, 2012). I denne studien utgjør spørsmålskjema en liten del av studien og inneholder få spørsmål, i utgangspunktet ment for å

plukke ut intervjukandidater. Men disse kriteriene er likevel tatt hensyn til ved utformingen av spørreskjemaet.

Intervju er en av de viktigste metodene innen kvalitativ forskning (Brekke & Tiller, 2013). Det er flere måter å gjøre det på. Det finnes ulike typer intervjuer, i denne studien er det valgt et semistrukturert intervju. I slike intervjuer har forskeren bestemt enkelte tema på forhånd, men samtidig er det rom for å følge opp det informanten kommer med. Det er altså ikke en helt åpen samtale mellom to eller flere personer, men det er heller ikke en muntlig versjon av et spørreskjema. Ved å bruke en slik intervjuform kan man lettere få fram alle synspunktene til informanten, samtidig som man passer på å spørre om de temaene som interesserer forskeren. Innen kvalitativ metode er det semistrukturerte forskningsintervjuet det vanligste (Krumsvik, 2015). Det sentrale ved denne intervjuformen er å innhente informantens opplevelser av temaet der og da. Det kvalitative forskningsintervjuet søker å forstå verden sett fra informantens side, og å få frem betydningen av folks erfaringer og avdekke deres opplevelser av verden. Informanten bidrar i forskningsintervjuet gjennom å skape mening og forståelse om et bestemt emne (Kvale & Brinkmann, 2015). I dette tilfelle ved å bidra med sine erfaringer med implementering av programmering i matematikkundervisningen.

3.1 Datainnsamling

3.1.1 Spørreundersøkelsen

Spørreundersøkelsen i denne studien er basert på en elektronisk spørreundersøkelse rettet mot alle lærere som underviste i matematikk i denne kommunen på det tidspunktet undersøkelsen ble sendt ut, oktober 2021.

Spørreundersøkelsen ble sendt ut på kommunens intranett i matematikkteamet. Her har alle matematikklærerne i kommunen tilgang, og de får også varsel når det blir lagt ut nye innlegg der. I forkant ble det sendt mail med informasjon til alle rektorer i kommunen om at det ville komme en slik forespørsel til matematikklærerne. Her ble det lagt ved en samtykkeerklæring, se vedlegg 2, litt informasjon om prosjektet og en forespørsel om rektorene kunne oppfordre sine matematikklærere til å bidra til denne undersøkelsen.

Innlegget i matematikkteamet ble sendt flere ganger, og det ble også sendt melding til avdelingslederne på samtlige skoler i kommunen med oppfordring om å minne matematikklærerne om undersøkelsen. I alle henvendelser ble det også presisert at dette var en kort undersøkelse som bare ville ta et par minutter, slik at det ikke skulle ta for mye av deres dyrebare tid å svare på denne spørreundersøkelsen.

Holand beskriver en spørreundersøkelse av denne typen som en survey-forskning, som betyr oversikt eller overblikk. Survey-forskning er en samlebetegnelse på kvantitativ forskningsdesign hvor datainnsamlingen foregår ved bruk av strukturerte spørreskjema. Datainnsamlingen vil gi et her og nå bilde av situasjonen for disse informantene. Ifølge Holand må vi regne med frafall i en survey-undersøkelse og et frafall på 50% er akseptabelt (Holand, 2012).

Responser på spørreundersøkelsen i denne studien endte på 35%, som da er noe under det Holand (2012) anser som akseptabelt. Valget om å bruke resultatene fra spørreundersøkelsen i denne studien likevel begrunnes med at den gir noen viktige og interessante resultater med henblikk på forskningsspørsmålene. Den gir svar på i hvilken grad lærere har startet med programmering i matematikkfaget, til dels hvordan de har gjort det, hvorfor de eventuelt ikke er kommet i gang med programmering i matematikkfaget, og i hvilken grad matematikklærerne føler de har fått god nok opplæring for å undervise i programmering i matematikk.

3.1.2 Intervjuene

Det er flere måter å gjennomføre og registrere intervjuer på med henblikk på senere dokumentasjon: lydopptak, videoopptak og notatskriving, der lydopptak er den vanligste (Kvale & Brinkmann, 2015). Valg av lydopptak fører til at man er friere i intervjusituasjonen, da man slipper å notere underveis. Dette gir intervjuet et mer naturlig preg, man får best mulig kontakt med intervjuobjektet i samtalen, og det sikrer at alt som blir sagt i intervjuene blir registrert. For å kunne foreta lydopptak eller videoopptak, der deltakerne kan identifiseres, må man søke NSD, se vedlegg 1.

Det er foretatt intervju med fire ungdomsskolelærere som har startet arbeidet med implementering av programmering i matematikkfaget. Informantene er plukket ut fra resultatene av spørreundersøkelsen. Kriteriene for utvelgelse av intervjuobjekter er at de skulle være i gang med programmering i matematikkfaget, at de underviser i matematikk på ungdomsskoletrinnet, at de representerte ulike ungdomsskoler i denne kommunen og at begge kjønn var representert.

Dato, tid og sted for gjennomføring av intervjuene ble avtalt via teams. Informasjonsskriv og samtykkeerklæring ble sendt informantene via e-post sammen med de faste spørsmålene i intervjuguiden, vedlegg 3. Informasjonsskrivet og samtykkeerklæringen ble utformet med hensyn til malen fra NSD og inneholder blant annet tema for oppgaven, problemstilling,

rettighetene til informantene, prosjektets slutt og hvordan prosjektets data skal håndteres, se vedlegg 2.

Datainnsamlingen ble gjennomført i perioden 3. desember 2021 til 10. januar 2022. De tre første intervjuene ble gjennomført med en ukes mellomrom, mens det siste ble gjennomført etter nyttår.

De to første intervjuene ble gjennomført fysisk, på informantenes egen skoler. Informantene hadde på forhånd reservert grupperom der man ikke skulle bli forstyrret. Det første intervjuet varte i 16 minutter og det andre varte i 22 minutter. De to siste intervjuene ble foretatt over teams som følge av Covid- restriksjoner. Ved det tredje intervjuet befant informant og forsker seg på hjemmekontor. Dette intervjuet varte i 15 minutter. I det fjerde intervjuet befant informant og forsker seg på et uforstyrret grupperom på hver sin arbeidsplass. Dette intervjuet varte i 21 minutter. I alle intervjusituasjonene ble det gjort lydopptak. Intervjuene ble foretatt på steder hvor man ikke ble forstyrret av bakgrunnsstøy, og det ble brukt ekstern mikrofon ved de fysiske intervjuene, som sto midt mellom informant og intervjuer for at lyden skulle bli så klar og tydelig som mulig, og dermed enklere å transkribere. I intervjuene over teams ble kun pc-mikrofon brukt, da begge parter naturlig vil sitte omtrent like nær sin egen pc, og lyden dermed blir like hørbar fra begge.

I etterkant av hvert intervju ble lydopptakene transkribert. Transkripsjonene er anonymisert og lydopptakene vil bli slettet når prosjektet er ferdig vurdert i juni 2022.

3.2 Analyse

Denne studien hadde en intensjon om å bruke en blanding av deduktiv og induktiv tilnærming, det vil si at noe teori blir valgt ut i forkant med utgangspunkt i spørsmålene i spørreundersøkelsen og intervjuene, og noe teori er valgt ut i etterkant basert på funnene i denne studien. I etterkant er det nok mest riktig å si at denne studien baserer seg på en deduktiv metode. Det vil si at det alt vesentlig er tatt utgangspunkt i teori for å undersøke om praksisfeltet stemmer med teorien.

Årsaken til dette er at spørsmålene i spørreundersøkelsen og i intervjuguiden er basert på teorien. Spørsmålene i spørreundersøkelsen er laget med formål om å finne intervjukandidater til dybdeintervjuene. Tanken bak var implementeringen av programmering i matematikkfaget i LK20. Med en hypotese om at flere lærere føler de har for lite opplæring i programmering, ble TPACK en sentral teori å trekke frem.

Spørsmålene i intervjuguiden er formulert med tanke på matematisk kompetanse, programmering i matematikkfaget, det pedagogiske kartet og TPACK. Til tross for relativt åpne spørsmål og informantenes mulighet til å snakke fritt om temaet, kom det ikke frem momenter som krevde drøfting opp mot annen litteratur.

Ved oppstart av denne studien dukket det opp en artikkel i tangenten (tidsskrift for matematikkundervisning), med overskrift: «Programmering på ungdomsskoletrinnet». Siden dette var tema for denne studien, falt valget på å undersøke om informantene mine hadde brukt UMC som metode for å introdusere programmering i matematikkfaget. Dette ble da et naturlig oppfølgingsspørsmål i intervjuene, og teori rundt dette ble naturlig å ta med i denne studien.

3.2.1 Analyse av spørreundersøkelsen

Spørreundersøkelsen inneholder fire spørsmål med ulike graderinger, og et (to, men man skal bare svare på et av dem, enten spørsmål 2 eller 3) oppfølgingsspørsmål, hvor informantene kan svare litt mer utfyllende.

Spørsmål 1: Er du kommet i gang med programmering i matematikkfaget?

Et ja og nei spørsmål, som er ment for den første utsilingen av mulige informanter.

Spørsmål 2: Dersom du svarte ja på forrige spørsmål, kan du si litt om hvordan du gjør det?

Resultatene fra dette spørsmålet ga litt mer informasjon om de ulike informantene og var ment som en hjelp i utvelgelsen av intervjukandidater.

Spørsmål 3: Dersom du svarte nei på spørsmål 1, kan du si litt om årsaken til det?

Siden programmering er blitt en del av kompetansemålene i matematikk fra høsten 2020, er det interessant for denne studien å finne årsaken til at noen lærere ikke er kommet i gang med programmering i sin matematikkundervisning.

Spørsmål 4: I hvilken grad føler du at du som lærer har fått god nok opplæring i programmering i matematikk, for å kunne undervise i dette? (skala fra 1-5 der 1 er dårligst og 5 er best).

I forkant av studien lå det en hypotese om at en del lærere ikke følte seg kompetente til å implementere programmering i matematikkfaget. Det var derfor interessant å finne ut i hvilken grad de følte de hadde fått noen opplæring i programmering i matematikk.

Spørsmål 5: Hvilket trinn underviser du i matematikk dette skoleåret? Fra 1. til 10.trinn.

Dette spørsmålet ble stilt med utgangspunkt i ønsket om å intervju matematikklærere på ungdomstrinnet.

Spørsmål 6: Hvilket kjønn er du? Mann eller kvinne. Bakgrunnen for spørsmålet var et ønske om å forta intervju med informanter av begge kjønn.

Disse spørsmålene er i utgangspunktet gitt for å finne intervjukandidater. Se kriterier over i kapittel 3.1.2. Målet var å finne ut hvordan ungdomsskolelærere har implementert programmering i matematikkfaget.

3.2.2 Analyse av intervjuene

Intervjuene startet formelt med presentasjon og informasjon om informantens rettigheter, og opplysning om at det ville bli gjort lydopptak av intervjuet. Da jeg startet opptaket, var vi ferdig med den formelle presentasjonen slik at vi gikk rett på spørsmåla. Dette for å opprettholde anonymiteten.

Informantene jobber ved fire ulike ungdomsskoler i samme kommune. De har noe ulik erfaring med programmering i matematikkfaget. Første informant har bakgrunn fra videregående skole og jobber første år på ungdomsskolen, med 9.trinn nå. De tre andre informantene har kun erfaring fra ungdomsskolen. Informant to og tre har jobbet med programmering i matematikk et par år. Informant fire har svært lite erfaring med å undervise i programmering i matematikk, men ønsket å bidra til undersøkelsen da han har en del tanker om hvilke pedagogiske muligheter det kan gi og hvilke utfordringer lærere står overfor i denne prosessen. Denne informasjonen fikk jeg gjennom de to innledende spørsmåla i intervjuguiden se vedlegg 3, som handler om hvor lenge de har undervist i matematikk, og hvilket trinn de underviser i matematikk dette skoleåret.

Intervjuene hadde en semistrukturert form, det vil si at det var seks faste spørsmål gitt til alle informantene, disse ble tilsendt på forhånd slik at de kunne forberede seg, se intervjuguiden vedlegg 3. Spørsmålene skulle gi svar på hvordan lærerne hadde implementert programmering i matematikkfaget, og hvilke fordeler og utfordringer de hadde erfart i dette arbeidet. I tillegg kom en del oppfølgingsspørsmål underveis, der det var naturlig for å utdype enkelte svar.

Neste fase av analyse av intervjuene handler om transkribering. Det vil si å gjøre om tale til tekst. Intervjuene ble tatt opp ved hjelp av taleopptak på pc, som ble lagret i OneDrive, i kommunens nettverk. Transkripsjonene ble utført fortløpende rett i etterkant av hvert intervju, for å ha fullt fokus på hvert av intervjuene. Transkripsjonene ble skrevet i Word ved hjelp av tabeller. Tabellene har tre kolonner, der 1.kolonne angir sitatnummer. 2.kolonne angir hvem som har kommet med sitatet, intervjuer angitt ved I, de fire informantene angis som L1, L2,

L3 og L4. I den tredje kolonnen er de enkelte sitatene gjengitt. Denne måten å organisere transkripsjonen på gir en god oversikt som er lett å referere til i resultatdelen.

I resultatdelen og drøftingen er svarene delt inn i tre ulike kategorier:

Hvordan implementere programmering i matematikkfaget? Under denne kategorien knytter det seg resultater fra spørsmål 1. Hvordan jobber du med programmering i matematikkfaget? Og spørsmål 2. I hvilke temaer i matematikken kan elevene ha nytte av programmering?

Hvilke pedagogiske muligheter gir programmering i matematikkfaget? Dette avsnittet er knyttet til spørsmål 3, 4 og 5. Hvordan er programmering med på å gjøre matematikkfaget mer spennende for elevene? Hvordan ser elevene sammenhengen mellom programmeringen og matematikkfaget? Og Hvilke fordeler ser du med å bruke programmering i matematikkfaget?

Hvilke utfordringer skaper implementering i matematikkfaget for lærerne? Denne delen er knyttet til spørsmål 6 fra intervjuguiden: Hvilke utfordringer opplever du ved å implementere programmering i matematikkfaget?

Denne inndelingen henger også sammen med hvilken teori funnene i de ulike spørsmålene drøftes opp mot.

Målet med denne studien var å finne ut hvordan ungdomsskolelærere har implementert programmering i matematikkfaget. Samtidig var det interessant å finne ut hvilke pedagogiske muligheter disse lærerne så ved å implementere programmering i matematikkfaget, og hvilke utfordringer de har identifisert i dette arbeidet. Dette er årsaken til denne tredelingen av funnene i resultat og drøftingsdelen. Som forsker hadde jeg dette som bakteppe da jeg laget intervjuguiden og under selve intervjuet.

3.3 Validitet og reliabilitet

I forskningsarbeid er validitet og reliabilitet sentralt. Validitet handler om gyldighet. I samfunnsvitenskapene dreier det seg om hvorvidt en metode er egnet til å undersøke det den skal undersøke (Kvale & Brinkmann, 2015). I dette tilfelle hvor målet var å undersøke: *hvordan matematikklærere på ungdomstrinnet har implementert programmering i matematikkfaget*, var det å intervju matematikklærere et naturlig valg. Spørreundersøkelsen var et ledd i utvelgelsen av informanter, men er også med på å underbygge informasjonen fra informantene i intervjuene spesielt når det kommer til den delen som handler om mangel på programmeringskompetanse blant matematikklærere.

Reliabilitet har med forskningsresultatenes konsistens og troverdighet å gjøre (Kvale &

Brinkmann, 2015), om hvorvidt et resultat kan reproduseres på et annet tidspunkt av andre forskere. Da denne studien representerer et lite utvalg læreres virkelighetsoppfatning på et gitt tidspunkt, er dette umulig å gi et sikkert svar på. Spørreundersøkelsen innehar 42 av 210 mulige informanter (35%), og intervjuene representerer fire informanters erfaringer med programmering i matematikkfaget.

Informantene i spørreundersøkelsen representerer matematikklærere fra 1.-10. trinn i en middels stor østlandskommune. Informantene i intervjuene jobber på fire ulike ungdomsskoler i den samme kommunen. Dette utvalget er ikke stort nok til å kunne si noe generelt om hvordan ungdomsskolelærere har implementert programmering i matematikkfaget, men gir en pekepinn om hvordan det kan gjøres, og hvordan noen matematikklærere i ungdomsskolen har gjort det.

Både i spørreundersøkelsen og i intervjuene ble det stilt samme åpne spørsmål til alle informantene. Spørreundersøkelsen ble utført digitalt, hvor informantene kunne gjennomføre denne i fred og ro på valgfri arena.

Den semistrukturerte formen til intervjuene besto av noen faste spørsmål, som informantene fikk tilsendt i forkant, og oppfølgingsspørsmål underveis for å utdype eller presisere informantens svar. At alle informantene fikk tilsendt de samme spørsmålene på forhånd er med på å styrke konsistensen i studien. Oppfølgingsspørsmål underveis er stilt for å utdype og presisere svarene, det er med på å styrke troverdigheten fordi man sikrer at informant og forsker har oppfattet meningsinnholdet i svarene likt.

Opptakene av intervjuene ble lyttet til flere ganger, for å transformere den muntlige samtalen til skrift. Det ble også lyttet igjennom samtidig med gjennomlesing av det ferdig transkriberte materialet. Dette er en metode som er med på å styrke validiteten til transkripsjonen (Kvale & Brinkmann 2015). Transkripsjonene er også gjort tilgjengelig for informantene, slik at de hadde mulighet for å komme med kommentarer dersom det er noe som er feiltolket, eller misforstått under transkriberingen. Ingen av informantene hadde noen kommentarer eller innvendinger med hensyn til transkripsjonene. Intervjuguide vedlegg 3 og transkripsjoner vedlegg 4, er også lagt inn i denne oppgaven, for å styrke troverdigheten.

Siden denne studien handler om hvorvidt matematikklærere i ungdomsskolen har implementert programmering i matematikkfaget, noe som er nytt i LK20, er det sannsynlig at man ville fått et annet resultat dersom man hadde foretatt den samme undersøkelsen med de samme lærerne på et senere tidspunkt.

3.4 Etikk

Innen forskning er etikk sentralt, både for å beskytte forskningsobjekter og for å sikre forskningens validitet og reliabilitet. Den etiske planleggingen startet med søknad hos NSD. Godkjenningbrev ligger som vedlegg 1. I spørreundersøkelsen hadde informantene mulighet til å svare anonymt. Undersøkelsen inneholdt ingen personlige spørsmål, kun spørsmål som gikk på yrkesutøvelse. Informasjonsskriv og samtykkeerklæring utformet hos NSD ble sendt ut sammen med spørreskjemaet. Dette ble også sendt på mail til informantene i intervjuene sammen med de planlagte spørsmålene. Hverken de planlagte eller de spontane spørsmålene var av en personlig karakter.

Dette er i tråd med et av Kvale og Brinkmanns syv forskningsstadier (Kvale & Brinkmann, 2015), der man i planleggingsfasen skal innhente samtykke fra informantene. De trekker også frem viktigheten av at informanten skal føle seg vel i *intervjusituasjonen*. I dette tilfellet kunne alle de fire informantene selv velge sted for intervju. To av intervjuene ble gjennomført på informantens egen arbeidsplass, og to av intervjuene ble foretatt over teams, der informanten L3 befant seg på hjemmekontor og L4 oppholdt seg på egen arbeidsplass. Informantene er matematikklærere i ungdomsskolen på lik linje med forsker, de er derfor å regne som jevnbyrdige parter. På denne måten er informantenes trygghet i situasjonen ivaretatt.

Transkriberingen fra lydopptak til skrift er utført ordrett, uten noen form for tolkning i analysearbeidet, men det er lagt vekt på en så nøyaktig og presis gjengivelse som mulig. I forbindelse med lydopptakene ble det heller ikke nevnt navn på personer eller arbeidssted. Dette for å sikre en så stor grad av anonymitet som mulig, Transkripsjonene er også anonymisert. I resultat- og drøftingsdel blir det ikke brukt navn i forbindelse med sitater, kun omtalt som L1, L2, L3 og L4.

4 Resultat

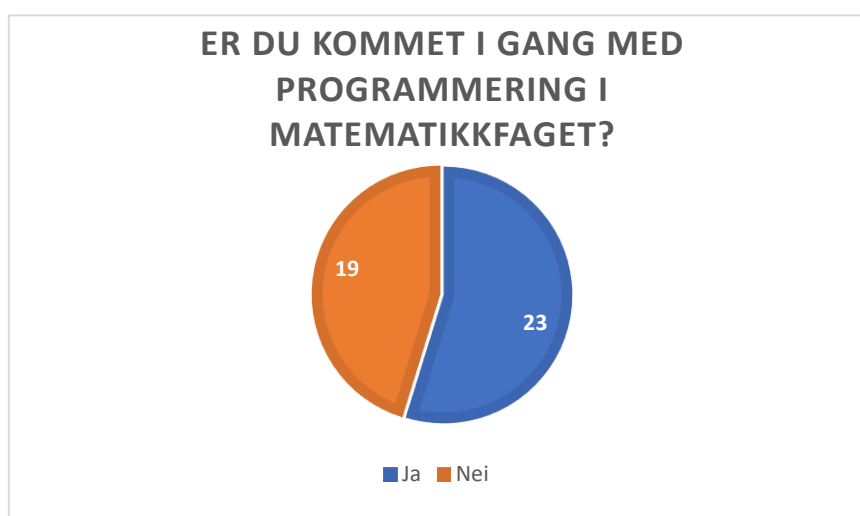
I dette kapitlet presenteres resultatene denne studien har fremskaffet. Delkapittel 4.1 inneholder resultatene fra spørreundersøkelsen. I 4.2 presenteres resultatene fra de fire intervjuene.

4.1 Spørreundersøkelsen

Resultatene fra spørreundersøkelsen baserer seg på innkomne svar fra 42 matematikklærere.

4.1.1 Spørsmål 1: Er du kommet i gang med programmering i matematikkfaget?

Undersøkelsen viser at 54% av informantene hadde kommet i gang med programmering i matematikkfaget, høsten 2022.



Figur 7: Resultat fra spørsmål 1: Er du kommet i gang med programmering i matematikkfaget?

4.1.2 Spørsmål 2: Dersom du svarte ja på spørsmål 1, kan du si litt om hvordan du gjør det?

Resultatene viser at de som er kommet i gang med programmering bruker ulike programmeringsspråk som: Scratch, Python, Bit Bot, Micro Bit og Edison roboter.

Undersøkelsen viser også at de fleste som er kommet i gang med programmering i matematikkfaget støtter seg på ferdige opplegg på en læringsplattform eller en nettside som Kodeklubben, Campus Inkrement, Kikora, Salaby Og Code.org

Her er noen av informantenes svar:

«Vi har jobbet litt med MicroBit og BitBot. Elevene har programmert BitBot slik at de kjører en bestemt strekning, svinger osv. Vi har også gjort noen enkle oppgaver i Campus i forbindelse med Mattemaraton»

«Bruker campus inkrement og edison roboter Bruker også ExCeL for å vise at programmering ikke bare er kode»

«Vi hadde litt grunnleggende Python opplæring i fjor på våren (10 trinn), men har ikke startet med dette enda (har 8 trinn i år) Viste noen grunnleggende funksjoner og brukte "tutorial" tilknyttet dette, hentet opplegg/ oppgave på kidsakoder.no»

«Jeg har vesentlig brukt programmeringsopplegg i Salaby, Campus og kikora,» (flere svarer dette).

«Laget opplegg som starter helt enkelt med blokk-programmering, og som vi videre skal utvikle til grunnleggende tekst-programmering. På VGS vil de i hovedsak møte Python som språk, noe vi da har fokus på.»

«Ved hjelp av code.org og andre nettsteder.»

«Enkel programmering i scratch. Feks interaktive bilder og animasjonsfilmer med interaktivite»

«Lager små programmer som skal gjøre ulike matematiske operasjoner, med ilagte småfeil, som elevene skal utvikle og forbedre»

4.1.3 Spørsmål 3: Dersom du svarte nei på spørsmål 1, kan du si litt om årsaken til dette? Av nitten som svarer at ikke er kommet i gang med programmering i matematikkfaget, begrunner ti informanter dette med at de ikke har nok kunnskap selv. De andre svarer at har det på årsplanen først til våren, og/eller at de underviste etter forrige lærerplan i skoleåret 2020/2021.

Her er noen disse resultatene:

«Jeg er forholdsvis ny til programmering men studerer dette på deltid, føler jeg enda ikke har fått oversikt over hvordan man implementerer programmering sammen med matematikktema som lærerplanen legger opp til.»

«Jeg har dessverre ikke nok kunnskap rundt programmering, så syns det er vanskelig og skulle starte opp med dette.»

«Kan ikke nok selv... har så vidt prøvd litt med elevene, men vil ikke si jeg er kommet i gang.»

«Fordi programmering er satt som siste mål i årsplanen. Det tar også tid å sette seg inn i, og blir derfor vanskelig å slenge inn her og der på andre temaer, selv om det ville vært det beste.»

«Ønsker å utforske, prøve ut og bli mer trygg selv først.»

«Kompetansemålet som omhandler programmering har ikke vært på årsplanen så langt. Mangler uansett ressurser, kunnskap og informasjon om hvordan dette bør gjennomføres.»

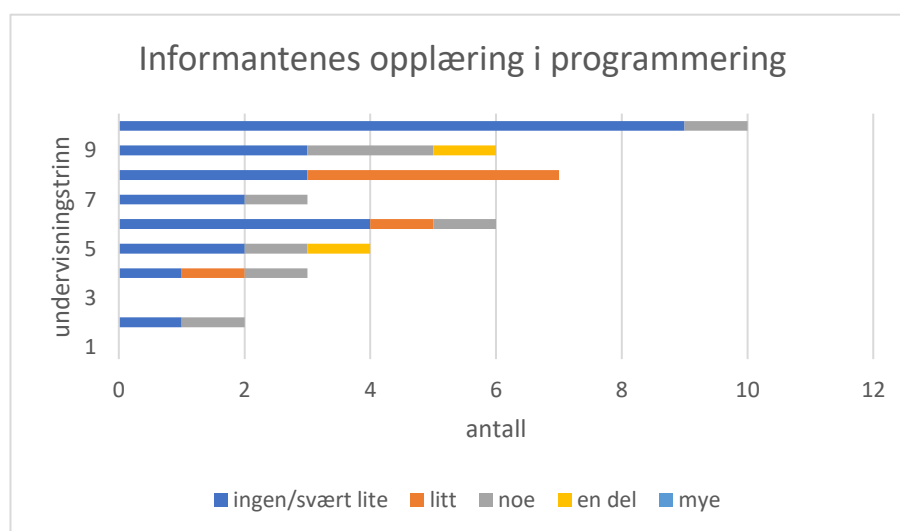
4.1.4 Spørsmål 4: I hvilken grad føler du at du som lærer har fått nok opplæring i programmering i matematikk, for å kunne undervise i det? (skala fra 1-5, der 1 er dårligst og 5 er best)

Resultatet fra dette spørsmålet er presentert i tabellen under:

I hvilken grad føler du at du som lærer har fått god nok opplæring i programmering i matematikk, for å kunne undervise i dette? (skala fra 1-5 der 1 er dårligst og 5 er best)				
1 ingen eller svært lite opplæring	2 litt opplæring	3 noe opplæring	4 endel opplæring	5 mye opplæring
24 57%	6 14%	9 21%	3 7%	0 0%

Her ser man at 71% av matematikklærerne svarer at de har fått ingen, lite eller litt opplæring i programmering i matematikk.

Figur 9 viser at 55% av informantene underviser på ungdomstrinnet, og når man går inn i detaljene i denne statistikken, kan man lese av at de fleste av disse svarte 1 eller 2 på spørsmålet fra tabellen over. Se resultater i Figur 8 under:



Figur 8 Informantenes følte opplæringsnivå i programmering, fordelt på hvilket trinn de underviser i matematikk

Denne figuren viser at 65% av informantene fra ungdomsskoletrinnene svarer at de har fått ingen eller svært lite opplæring i programmering.

4.1.5 Spørsmål 5: Hvilke(t) trinn underviser du i matematikk dette skoleåret?

Begrunnelsen for dette spørsmålet er ønsket om å finne informanter på ungdomstrinnet til dybdeintervju.

Informantene fordeler seg på følgende undervisningstrinn:



Figur 9 Resultat av spørsmål 5 i spørreundersøkelsen: Hvilket trinn underviser du i matematikk i år?

Vi ser her at 23 av 42 av informantene jobber med matematikk på ungdomstrinnet, det vil si at de utgjør en betydelig majoritet blant informantene. Dette er relevant fordi denne studien er mest interessert i resultatene fra ungdomsskolelærere, og hvordan de har implementert programmering i matematikkfaget.

4.1.6 Hvilket kjønn er du? Mann eller kvinne.

Det siste spørsmålet dreide seg om hvilket kjønn informantene representerte, her var det 24 kvinner og 18 menn. Begrunnelsen for dette spørsmålet var ønsket om å bruke intervjuobjekter av begge kjønn.

4.2 Intervjuene

I dette delkapitlet presenteres resultatene fra de fire intervjuene som er gjennomført i denne studien. Resultatene er delt inn i tre kategorier: hvordan informantene har implementert programmering i matematikkfaget, hvilke pedagogiske muligheter dette har gitt, og hvilke utfordringer informantene har stått overfor, se begrunnelse for denne inndelingen i kapittel 3.2.2 analyse av intervjuene.

4.2.1 Hvordan implementere programmering i matematikkfaget?

Informant L2 valgte å starte med å lage koder med penn og papir og fikk elevene til å lage fortellinger der de skulle styre en annen elev til å utføre en handling, og hvor elevene i etterkant ble forklart at det er slik en kode er bygd opp. Elevene laget også en instruksjon til læreren hvor han utførte nøyaktig det instruksjonen sa, på den måten fikk de erfare hvor viktig detaljene i en kode er: «...for å være litt vrang, så gjorde jeg den akkurat sånn som det stod. Så hvis det for eksempel står der at man skulle løfte opp et egg. Så tok jeg på en måte ikke nødvendigvis tak i egget. fordi de ikke hadde skrevet at jeg skulle ta tak i det, så jeg tenker sånn da og hadde hånden min over for eksempel egget, men løftet ikke opp...» (L2-6).

Deretter introduserte han blokkprogrammeringsspråket Scratch. Et program L3 og L4 også startet opp med i sin undervisning i programmering i matematikk på ungdomsskolen. De har startet med å la elevene gjøre seg kjent med programmet og finne ut hvilken funksjon de ulike blokkene har.

«...fikk prøve seg rundt i programmet. Hva er det på en måte de ulike blokkene gjør...» (L2-8).

L3 valgte å starte med blokkprogrammering i forbindelse med introduksjon av algebra og variabelbegrepet: «...starter vi enkelt med blokkprogrammering med Scratch. Som en del av algebra, forståelsen for algebra at man kan sette inn ulike verdier... at jeg gjør sammen med de veldig sånn skritt for skritt..» (L3-6).

Når de har gått videre, har informant L2, L3 og L4 brukt ferdige oppgaver som de har laget selv eller funnet på nettet. Her har de gått frem trinn for trinn, og de påpeker at det er viktig å gå sakte frem for å få med seg alle. Lærerens instruksjoner blir også trukket frem som en viktig faktor.

«Jeg fant en oppskrift, for eksempel på sånn påskespill, ...på en måte sånn gjør dette gjør dette, gjør dette, så vi på en måte gjorde gradvis. ..og da fikk du på en måte etterhvert det samme spillet som meg, og noen måte valgte jo på en måte å prøve nesten å kopiere det spillet, men da ville ha sin egen handling på det. Noen valgte jo da mer å prøve å gå nye veier da, men kanskje med noen av de samme funksjonene» (L2-12).

«...I starten modellerer, at at jeg gjør sammen med de veldig sånn skritt for skritt...» (L3-6).

«...Scratch da hovedsakelig ja, og da er jo de da noen ferdig ferdige opplegg eller ferdig oppskrifter om elevene følger og setter sammen...» (L4-10).

Noen av informantene påpeker at det er stor forskjell i elevgruppene, hvor mye de kan fra før. De mener derfor det er viktig å bruke tid på å være nøyaktig i introduksjonen.

«...de som på en måte har virkelig kjørt på med programmering i barneskolen, og de som så vidt vet hva det er.» (L2-18)

«...for de var jo veldig nybegynnere disse elevene...har jeg liksom hatt tilpasset opplæring...noen jobbet for seg selv..» (L3-6)

«Store elevgrupper. Da må jeg på forskjellige nivå og jobbe forskjellig» (L3-30).

Likevel mener L2 at det er viktig å la elevene utforske og prøve og feile litt.

«fikk beskjed om å prøve å klikke seg rundt og se og gjøre feil da prøve og feile litt» (L2-8).

L3 forteller at hun har brukt enkelte elever som har kunnet en del fra før til å være hjelpelærere. Dette for at de ikke skal kjede seg, og at de skal føle mestring.

«... fikk i oppgave å være hjelpelærer, mens vi mens jeg hadde undervisning på tavla, løp de mellom å hjelpe til hverandre» (L3- 6).

Alle informantene opplever viktigheten av å bruke åpne oppgaver, slik at alle kan klare noe og at det er utviklingsmuligheter for alle, fordi elevene kommer til ungdomsskolen med veldig ulik erfaring med programmering.

«... med tanke på at oppgavene kan være ganske åpne..» (L2-20).

«...så da fikk de en relativt åpen oppgave..» (L3-6).

«...at inngangsterskelen må være såpass lav at alle sammen faktisk skjønner det for den programmeringa der det er vanskelig når du ikke har vært der før, og da kan man ikke gå ut og si at, det er bare gjøre sånn og sånn..» (L4-16).

Det er viktig både å finne oppgaver/opplegg som er enkle nok til at alle kan føle mestring, samtidig som elever med mye erfaring eller som mestrer dette raskt har nok å strekke seg etter.

«for de som på en måte synes at blokkprogrammeringen blir kjedelig, fordi at det er for lett, så på en måte utfordrer de på en måte å sette seg litt inn i helt grunnleggende python» (L2-16)

Informantene trekker frem algebra som et tema som er spesielt egnet for programmering. De trekker da spesielt frem dette med å forstå variabelbegrepet og algoritmer, at programmering kan være en hjelp til å forstå hva det egentlig innebærer. Geometri er et tema som blir trukket frem da det her også i stor grad er snakk om formler og variabler. L1 og L2 mener det vil kunne brukes innen alle temaer i matematikken.

«Det kan jo i stor grad brukes i alle temaer» (L1-12). «Vi bruker jo det ganske mye mer på en måte for å forstå hva for eksempel algoritmer gjør da, at det er på en måte en sånn forklaring

på at detaljer er viktig. Sånn rent temamessig, så kan man jo bruke det innen de fleste tenker jeg» (L2-14).

L4 trekker også frem tallregning, det at elevene kan lage seg egne kalkulatorer i ulike temaer innen tallregning, og at man kan bruke det som en oppgavegenerator. Han mener også dette er en enkel kode som de fleste elever kan lære seg. «*.. tallregning....egentlig rett og slett til tørrtrening på å lage oppgaver. Så hvis man skal trene på multiplikasjon da tosifra, ensifra..... å bare ha en kode for det som bare kan trykke ny oppgave, ny oppgave så generere den nye oppgaver i det parameterne du selv har satt., ... de lærer å sette opp en ,... oppgavegenerator..» (L4-20).*

Alle informantene trekker frem at det er viktig at elevene blir introdusert for tekstprogrammering i ungdomskolen, da det er dette programmeringsspråket de skal jobbe med i videregående skole. L2-L4 har introdusert dette gradvis, etter å ha jobbet grundig med blokkprogrammering først.

«Lærte vi dem helt basic Python, veldig basic som kommando.» (L3-6).

L1 har ikke gått veien om blokkprogrammering, men gått rett på tekstprogrammering. Han startet da med helt elementære ting som hvordan sette opp et regnestykke i Python. Han mener selv at det er litt kjedelig, men at det er viktig å få på plass helt grunnleggende ting først.

«Da har vi fokusert på, nå først å lære hvordan du setter opp et mattestykke i pyton. Det er jo ganske kjedelig måte. Det er litt sånn programmering for programmerings del. Men det må være i bunn, for at du skal klarer å gjøre noe morsomt videre» (L1-6).

4.2.2 Hvilke pedagogiske muligheter gir programmering i matematikkfaget?

I dette delkapitlet vil jeg trekke frem hvilke pedagogiske muligheter mine informanter ser ved å implementere programmering i matematikkfaget.

Alle fire informanter viser til at det er med på å skape variasjon i matematikkundervisningen. «*det et ekstra verktøy å bruke. Det er med på å variere undervisningen» (L1-26).*

“...at oppgavene kan være ganske åpne. Du kan få ganske mange resultater og, det åpner opp for at de må reflektere på en helt annen måte også...” (L2-20).

“..variasjon er jo en ting da. Og det er jo, her handler det ikke bare om de som er interessert i og liker pc, men det å også gjøre forskjellige ting, det er motiverende, men bruddet på undervisning..” (L4-22)

At det gir rom for åpne oppgaver, som gjør at alle kan føle mestring. De trekker også frem at man kan treffe flere elevgrupper. At elever som vanligvis ikke viser så stor interesse for matematikk synes at dette med programmering er spennende. Ellers svake elever kan føle mestring, som igjen gir motivasjon.

«Har sett noen elever nå og også tidligere som våkner opp når de sitter og jobber på maskinen. De ser klarer å se at det er sånn alt av apper og alt av dataspill fungerer på kode» (L1-14).

«..merket at de hadde lyst til å gjerne ha matematikk mer liksom mer motivasjonen og mestringsfølelsen kom fra, og den smitter jo litt over og så videre i faget..» (L2-26).

L3 nevner at det er inn i tiden og gjør at flere ser nytten av å kunne programmere enn de for eksempel ser i å løse likninger med penn og papir.

«det er jo veldig inn, og det er jo veldig fremtidsriktad å å få en grunnleggende forståelse før programmering». «.,jeg håper at det skal, hvis man da for implementert programmeringen her er at man skal få litt mer sånn forståelse føre at matematikken er et viktig fag da i særlig fremtiden..» (L3-14).

En annen faktor tre av informantene trekker frem er tverrfaglighet, at matematikk kan få en sterkere rolle inn i tverrfaglige prosjekter. Alle disse tre trekker frem prosjekter der de har jobbet sammen med naturfag og samfunnsfag. Der matematikken har ansvaret for programmet, kan andre fag ha ansvar for fortellingen.

«vi hadde et tverrfaglig tema med matte og naturfag, så matematikken tok jo dette med algoritmer og oppbygning, syntaks..» (L2-10)

«åpner jo for mye da altså tverrfaglig ... trekke programmering opp mot det og på en måte lage en fortelling i form av det eller forklare noe faglige få programmet til å regne på et eller annet...Mens matematikk som tar selve programmet, og kan noe annet ta selve handlingen» (L2-32)

Informantene mener programmeringsoppgaver i matematikk kan gi mer nærhet til virkeligheten, at oppgavene blir mer realistiske enn oppgaver i en matematikkbok. Elevene lever digitale i sin hverdag med spill, telefoner, Ipad og andre digitale verktøy. Det er derfor mer motiverende for dem å prøve å forstå hvordan programmene i disse verktøyene er bygd opp enn mange av de teoretiske oppgavene de er vant til å få servert i en matematikkbok.

«Det er at du kan motivere en helt spesifikk elevgruppe som ellers kan komme med ganske, men med ganske lav motivasjon til matematikk da» (L3-22).

«faktum at det her er jo, jo mer virkelighetsnært da enn å regne oppgaver på campus eller å regne i boka» (L4-22)

Dybdelæring er også en faktor som blir trukket frem av flere informanter. De ser at elevene gjennom programmeringsaktiviteter får en annen og større dybdeforståelse i matematikk generelt.

«det er mest denne dybdeforståelsen å tvinge dem til å bearbeide. Det er jo også en ekstrem øvelse i å være logisk og konkret, og konsis» (L1-22).

«kan jo få økt forståelse av for eksempel areal av trekanteder ved å sitte å regne oppgaver eller ved å lage program som løser det. For da må du også faktisk teste at det funker for rare trekanteder også» (L1-26).

«tror at det liksom gir en dypere forståelse, særlig for det er at med det abstrakte språket da, men den algebraen» (L3-32).

Samarbeid og gode matematiske samtaler mellom elevene er trukket frem som positive opplevelser ved å jobbe med programmering i matematikkfaget.

«ganske mange fine samtaler på tvers av kjønn som gjorde at de kunne komme med ideer til hverandre» (L2-22).

4.2.3 Hvilke utfordringer skaper implementering av programmering i matematikkfaget?

I dette delkapitlet ønsker jeg å trekke frem hvilke utfordringer mine informanter ser ved å implementere programmering i matematikkfaget. Informanter trekker frem manglende kompetanse og manglende opplæring som den største utfordringen ved implementering av programmering i matematikkfaget. De føler de har fått lite og til dels sprikende opplæring i hvordan de skal undervise i dette i sine klasser.

«føler den største utfordringen er jo på en måte, og hva skal vi lære selv. Jeg føler jo på en måte mye av det første jeg hadde den, og så var det jo mer enn at jeg leste meg opp og elevene kommer rett etter. Og at det etter hvert ble noen elevene som gikk forbi meg» (L2-38).

«utfordrende at det er både mangel på opplæring og spesifikasjoner da på hva, hva de egentlig vil at de skal sitte igjen med når de går ut av ungdomsskolen?» (L4-39).

Ellers er tid en faktor som trekkes frem. Det at det tar tid å lære elevene å programmere. De fleste synes det er en tidstyv, men noen har tro på at det kan lønne seg på sikt. Det vil si at den tiden vi investerer nå vil både elever og lærere ta igjen etter hvert som de blir flinkere og kommer lenger i programmeringsprosessen.

«jeg lurar på om det kan stjele tid noen ganger, for det tar tid å sette seg inn i

programmeringen» (L1-24).

«Dette er en ting til som tar plass....da er det fort enda et tema man må innom» (L2-38).

«det tar mye tid. Absolutt. Men jeg tror, jeg tror ikke, at det er en tidstyv på den måten, at jeg tror at det liksom gir en dypere forståelse» (L3-32).

Ellers trekker L3 frem at størrelsen på klassen er en utfordring.

«Det er utfordrende å ha en stor elevgruppe. Da må jeg på forskjellige nivå og jobbe på forskjellige. Noen tar det veldig fort» (L3-30).

Alle informantene synes spriket i kunnskapen elevene har i programmering når de begynner på ungdomsskolen er utfordrende. Noen elever har ikke jobbet noe særlig med programmering før ungdomsskolen, mens andre har jobbet masse med det tidligere. Det skaper en ekstra utfordring med hensyn til hvordan man skal legge opp undervisningen.

Et siste moment informantene trekker frem som utfordrer implementeringen av programmering i matematikkfaget, er utformingen av kompetansemålene i LK20. Hva er det Utdanningsdirektoratet egentlig vil at elevene skal sitte igjen med av kunnskap om programmering i matematikk når de er ferdig med grunnskolen? Samtlige av mine informanter mener målene er for vage og sier at de ønsker seg mer spesifikke og konkrete mål. Hvilke program skal de kunne? Det finnes mange ulike programmeringsspråk å velge blant, og det er ikke nødvendigvis slik at de ulike programmeringsspråkene kommuniserer med hverandre.

«fint å se hva slags type oppgaver som kan komme på eksamen blant annet. Også hva de vil møte på videregående» (L2-44).

«med tanke på programmeringsspråk, skulle vi lære Java, skal vi gjøre Scratch eller Python. Hvilke programmer er det de bruker mest?» (L2-46).

«Det jeg sliter med å se hvor vi egentlig vil med programmeringen. Hva er hva er målet bak det? Bortsett fra at elevene skal kode, hvilket program skal de kode i nøyaktig? Hva skal de kode altså? Jeg føler, at det her er bare. Nå skal elevene kunne programmering, ikke hva de skal kunne programmere.» (L4-34).

5 Drøfting.

I dette kapitlet drøftes resultatene fra spørreundersøkelsen og intervjuene opp mot teoretisk rammeverk og tidligere forskning.

5.1 Spørreundersøkelsen

Spørreundersøkelsen avdekker at det kun er 54% av informantene som er kommet i gang med programmering i matematikkfaget. 71% prosent av informantene svarer at de har fått lite eller ingen opplæring i å undervise i programmering (de har svart 1 eller 2, på en skala fra 1-5, der 1 er lavest og 5 er høyest grad av opplæring). Det vil si at også lærerne som er kommet i gang med programmering i matematikkfaget mener de har fått liten eller ingen opplæring i dette selv. Dette indikerer at disse lærerne ikke har fått nok opplæring i den delen av sin TPAC som omhandler programmering innenfor teknologidelen.

Få matematikklærere har lært matematikk ved hjelp av databasert tankeverktøy, eller gjennomført sin lærerutdanning med tanke på å integrere teknologi i undervisningen i matematikk (Niess, 2008). Denne artikkelen er fra 2008 og for en del lærere i dag stemmer ikke det helt. For de fleste stemmer det likevel hvis vi ser på programmering spesielt. Lærerne svarer i spørreundersøkelsen at de mangler nødvendige kunnskap om programmering for å kunne ta dette i bruk i sin matematikkundervisning.

TPAC handler om interaksjonen mellom teknologikunnskap, i dette tilfellet programmering, fagkunnskap, i dette tilfellet matematikk og pedagogikk, som handler om hvordan man skal undervise i dette. En av informantene i spørreundersøkelsen sier direkte i sitt svar at han mangler informasjon og kunnskap om hvordan han skal jobbe med kompetansemåla i matematikk som omhandler programmering.

De fleste som ikke er kommet i gang med programmering begrunner det med at de ikke kan nok om programmering selv og derfor ikke føler seg klare til å undervise i det. De mangler teknologikunnskap (TK i TPACK-modellen, se figur 2, i kapittel 2.2) som omhandler programmering. Det er ikke mulig å vite hvilke pedagogiske muligheter man har med programmering, når man ikke kan noe, eller har lite kjennskap til hva programmering går ut på, da har lærerne mangler i sin TPK-teknologisk og pedagogisk kunnskap (Koehler & Mishra, 2009). Når lærerne mangler kunnskapen om programmering, blir det vanskelig å vite hvilke muligheter programmering kan gi i de enkelte temaer i matematikkfaget. Det vil si at de har mangler i sin TCK-teknologi og fagkunnskap.

Undersøkelsen viser likevel at lærerne ikke mangler vilje til å lære seg programmering. De ønsker å utforske og lære seg å programmere, de er bare ikke kommet dit ennå. Dette samsvarer med funnene i Berggren og Jom sin spørreundersøkelse om læreres holdninger til innføring av programmering i matematikkfaget (Berggren & Jom, 2019). Der 26 av 30 lærere kunne tenke seg å lære seg å programmere for å kunne undervise i det.

Blant lærerne som har startet med programmering i matematikkfaget i denne studien svarer et flertall at de bruker ferdige opplegg fra læringsressurser på nettet. Det kan være en grei inngang når man ikke har så mye erfaring og kunnskap med programmering selv. Mange av disse oppleggene legger også opp til en fremgangsmåte der man bruker UMC modellen. Dette kan gjøre innlæringen lettere både for elever og lærere (Maugesten et al., 2021).

5.2 Intervjuene

Drøftingen av intervjuene er delt inn i de samme tre delkapitlene som resultatene.

5.2.1 Hvordan har matematikklærerne implementert programmering i matematikkfaget?

De fire informanter har valgt litt ulike strategier for å implementere programmering i matematikkfaget. L1 har startet opp direkte med tekstprogrammeringsverktøyet Python. De tre andre informantene har valgt å starte med blokkprogrammeringsprogrammet Scratch. Dette gir et mer visuelt programmeringsuttrykk, som også lager lyd, noe som kan være en motiverende faktor for unge elever ved introduksjon av programmering i matematikkfaget (Haraldsrud et al., 2020).

Informant L2, L3 og L4 hadde ubevisst (de hadde ikke hørt om metoden før) benyttet en introduksjon til programmeringen som kalles UMC.

Denne metoden er utprøvd på ni lærere som deltok på et kurs ved Høgskolen i Østfold (Maugesten et al., 2021). Disse lærerne påpeker at elevene og de selv lettere igangsetter samtaler og diskusjoner om kodene og resultatene de observerte i Scratch og Python.

En av informantene i denne studien observerte gode matematiske samtaler om på tvers av kjønn, hvor de fikk inspirasjon av hverandre under arbeidet med blokkprogrammeringen i matematikktimen (L2-22). Her er et av kjerneelementene i LK20 sentralt, representasjoner og kommunikasjon:

Representasjoner i matematikk er måter å uttrykke matematiske begreper, sammenhenger og problemer på. Representasjoner kan være konkrete, kontekstuelle, visuelle, verbale og

symbolske. Kommunikasjon i matematikk handler om at elevene bruker matematisk språk i samtaler, argumentasjon og resonnementer. (Utdanningsdirektoratet, 2020a).

Både lærer i prosjektet i Østfold (Maugesten et al., 2021), og informanter i denne studien mente at metoden ga gode muligheter for tilpasset opplæring fordi kodene fungerte som åpne og rike oppgaver. Se resultater kapittel 4.2.2.

De svakeste elevene jobber lenger med den første fasen mens elever som følte seg trygge på programmering kunne bruke kreativiteten sin på å lage nye programmer. En av informantene i denne studien, L3, har for eksempel brukt oppgaver hentet fra Kodeklubben (L3-8). Informant L2 hadde funnet flere slike opplegg på nettet, blant annet et påskespill (L2-12). I Berggren og Jom sin spørreundersøkelse (2019), svarte 21 av 30 lærere at de ønsket ferdige opplegg til bruk i klasserommet. Dette tyder på at både informanter fra denne studien og fra annen forskning føler det trygt å kunne lene seg på opplegg som ligger tilgjengelig på nett, for å slippe å lage alt fra bunnen selv.

Å bruke et blokkprogrammeringsspråk til introduksjon av programmering i matematikk kan være med på å gi elevene en mestringsfølelse, da det er lite som kan gå veldig galt, og det er eventuelt enkelt å rett opp i slik at programmet gjør de operasjonene man har tenkt (Haraldsrud et al., 2020). Dette kan være med på at flere elever føler mestring og dermed blir motiverte til programmering i matematikk, og at de gjennom programmeringen ser nytten med matematikk (L2-26 og L3-14).

En av informantene i denne studien har ikke brukt blokkprogrammering i sin introduksjon til programmering i matematikkfaget, men gått rett på tekstprogrammering i Python. Han begynte med helt enkle ting som å sette opp regnestykker for at de skulle lære seg tekstprogrammet Python (L1-6). Han sier selv at det er en ganske kjedelig måte å lære på, men at man må lære noe grunnleggende om programmet for å kunne gjøre morsommere ting etter hvert. Dette blir en motsatt fremgangsmåte at det UMC-modellen følger. Her begynner man med et ferdig program som fungerer korrekt med en gang, som elevene skal teste ut, før man begynner å plukke det fra hverandre og endre på de ulike kommandoene for å se hva som skjer, før du til slutt lager ditt eget program som. Informant L1 i denne studien har en mer instrumentell tilnærming. Men han presiserte at elevene får utforske og eksperimentere etter hvert når de forstår hvordan programmet fungerer (L1-26).

De andre informantene påpekte at det er viktig å gradvis gå over til tekstprogrammering i løpet av ungdomsskolen da det er det som venter elevene på videregående (L2-20, L3-6 og

L4-16). Ved overgangen til tekstprogrammering er det viktig å legge matematikken på et lavt nivå for å få med seg alle, da mye av hjernekapasiteten til elevene nå vil fokusere på selve tekstprogrammeringsspråket, for ikke å skrive feil i kodene (Haraldrud et al., 2020). Så kan man heller la elever som er tar dette raskt få lov til å eksperimentere på egen hånd.

Ruthven (2014) nevner i sin artikkel at det er et implisitt hierarki mellom innhold (fag), pedagogikk og teknologi. Der faget kommer først, så pedagogikken og teknologien til slutt. Hovedinntrykket er at informantene i denne studien har latt matematikken, og de enkelte temaene innen matematikk bestemme når man skal bruke programmering i undervisningen. Programmeringsaktiviteten har nok vært med å styre pedagogikken, altså hvordan undervisningen er lagt opp, men matematikken har vært det styrende i aktiviteten.

5.2.2 Hvilke pedagogiske muligheter gir programmering i matematikkfaget?

Resultatene i denne studien viser at lærerne mener det ligger pedagogiske muligheter i å implementere programmering i matematikkfaget. Fagfornyelsen LK20 har ført til endrede oppgaver, spesielt trakk informantene frem muligheten til å bruke ekte situasjoner til simulering og utforskning. Endring i læreplan kan føre til funksjonelle muligheter (Pierce & Stacey, 2010). L2 har også brukt programmering med penn og papir, ved at elevene har laget «fortellinger» hvor de har programmert hverandre til å utføre bestemte handlinger. «*Learn pen and paper skills*» (Pierce & Stacey, 2010).

Programmering har også ført til endringer i didaktikk, ved at disse oppgavene krever andre aktiviteter enn når man jobber med tradisjonelle oppgaver i boka. Se L4-22 i resultatet. Dette gir lærere en ny didaktisk utfordring, *change classroom didactic contract* (Pierce & Stacey, 2010) som krever at man utvikler sin TPACK (Koehler & Mishra, 2009). Fordi lærere har ulik kompetanse og ulik interesse for programmering, vil graden av og formen på undervisningen her være ulik. Det viser også funnene som er kommet frem i intervjuene. Derfor kan UMC være et godt hjelpemiddel for både elever og lærere til å komme i gang med programmering, da terskelen for å komme i gang blir lavere (Maugesten et al., 2021).

Både denne studien og prosjektet ved Høgskolen i Østfold viser til at programmering gir mulighet for tilpasset opplæring fordi kodene fungerer som åpne oppgaver (Maugesten et al., 2021). Se også referanser i resultatene 4.2.2. L2-20, L3-6 og L4-16. Denne typen oppgaver åpner også for gode matematiske samtaler både elever imellom og mellom lærer og elev (Maugesten et al., 2021). Se referanse under resultater 4.2.2, L2- 22. Begrepsmessig forståelse av matematiske begreper, operasjoner og relasjoner er en av de matematiske kompetanser i

Kilpatrick.s modell (2001). Kommunikasjonskompetanse, representasjonskompetanse og symbol- og formalismekompetanse, er blant kompetansebegrepene i matematikk hos Niss og Jensen (2002). Denne samhandlingen kan også gi rom for dypere forståelse ved at elevene lærer av hverandre. Det kan føre til økt matematisk kompetanse ved at elevene kan få en bedre begrepsmessig forståelse og utvikle gode løsningsstrategier effektivt og nøyaktig. At de lærer logisk tenking, refleksjon og argumentasjon. Og at de får økt engasjement fordi de ser matematikk som fornuftig og verdifull og har tro på at det er mulig å bli kompetent i matematikk (Valenta, 2016).

Dette stemmer med funn fra denne studien. Flere informanter har observert at flere blir engasjert i matematikk gjennom programmering. De ser at elever som ikke viste interesse for matematikk tidligere er interessert i programmering, og gjennom programmeringen får de også interesse for matematikk (L1-14 og L2-26). Spesielt der man trekker inn spilleelementet (L3-24). Informantene i denne studien trekker også frem økt dybdeforståelse og begrepsforståelse i forbindelse med programmering i matematikkfaget (L1-22 og 26 og L3-32). I arbeidet med ny læreplan har dybdelæring fått en sentral plass under prinsipper for skolens praksis i overordnet del (Utdanningsdirektoratet, 2020c).

Programmering kan være med på å: *cange classroom sosial dynamics* (Pierce & Stacey, 2010), ved at elever som tidligere var svake i matematikk, nå føler mestring og engasjement L1-42, L2-20, L3-22 og L4-26. Engasjement er en viktig faktor for matematisk kompetanse (Kilpatrick et al., 2001).

Til sist mener informantene at programmering i matematikk fører til større variasjon i undervisning og oppgaver (L1 26). Informantene trekker spesielt frem muligheten for å bruke mer virkelighetsnære oppgaver (L2-6, L2-22 og L3-22). *Use real date og Simulate real situation* er en av de mulighetene for endringer i oppgaver Pierce Stacey (2010) ser i teknologirike rom i sitt pedagogiske kart.

5.2.3 Hvilke utfordringer skaper implementering av programmering i matematikkfaget?

Med implementering av programmering i matematikkfaget ligger det også noen utfordringer. Informantene mener de som lærere både mangler tilstrekkelig opplæring i programmering (L2-38 og L4-39), og veiledning om hvilken kompetanse i programmering elevene skal sitte igjen med etter endt grunnskole (L2- 42 og 46).

TPACK ble introdusert for å rette oppmerksomheten mot måten nye teknologiske ressurser er med på å omforme pedagogisk kunnskap, innholdskunnskap og pedagogisk innhold og

kunnskap. TPACK- beskriver den kunnskapen lærere trenger for undervisning med og om teknologi i sine tildelte fagområder som for eksempel matematikk (Niess, 2008). Niess (2008) påpeker at det er viktig at lærere får støtte til faglig utvikling av sin TPACK.

Andre utfordringer som spesielt blir fremhevet er at kompetansemålene er lite spesifikke. Det er vanskelig å vite hvilke programvare man skal velge og hva det forventes at elevene skal kunne ved endt grunnskole. Se under resultater 4.2.3 sitat: L2-44 og 46 og L4-34.

Kompetansemålene (Utdanningsdirektoratet, 2020b) fra ungdomsskolen er følgende:

8.trinn: *utforske hvordan algoritmer kan skapes, testes og forbedres ved hjelp av programmering.* Her skal altså elevene lage en fremgangsmåte (algoritme) og teste og forbedre den ved hjelp av programmering. Det blir lite spesifikt mener informantene i denne studien.

9.trinn: *simulere utfall i tilfeldige forsøk og beregne sannsynligheten for at noe skal inntreffe, ved å bruke programmering.* Her skal elevene etterlikne (simulere) ulike situasjoner, og kunne regne ut sannsynlighet for at noe skal inntreffe ved hjelp av programmering.

10.trinn: *utforske matematiske egenskaper og sammenhenger ved å bruke programmering.* Her skal elevene utforske matematiske egenskaper (det som er spesielt for matematikkfaget), og sammenhengene deres ved bruk av matematikk. Det mine informanter lurer på da er hvordan de som lærere skal forberede sine elever mot eksamen. Hva er det forventet at de skal kunne når de kommer dit?

Andre utfordringer informantene trekker frem i denne studien er store forskjeller i kunnskapsnivå hos elevene. Elevene kommer til ungdomsskolen med variert bakgrunn når det kommer til programmering. LK 20 gjelder fra august 2020, dermed må man anta at dette vil jevne seg ut om noen år. Programmering i matematikk ligger i kompetansemåla fra 5.-10.trinn. De som nå går i 9.trinn har ikke fått undervisning etter LK20, før de begynte på ungdomsskolen. Om noen år når det starter elever på ungdomsskolen som skal ha fått opplæring i programmering på barneskolen fra 5.trinn, det vil si høsten 2024, vil forhåpentligvis situasjonen være annerledes. Men det er også avhengig av at lærerne får den nødvendige kompetanse i å undervise i programmering i matematikk.

6 Avslutning

Avslutningsvis vil jeg komme med en konklusjon på hvordan resultatene viser relevante funn som belyser forskningsspørsmålene. Til slutt retter jeg et kritisk blikk på egen studie, og ser på veien videre både med hensyn til videre studier på dette temaet og med tanke på egen praksis.

6.1. Konklusjon

Denne studien har hatt som mål å belyse hvordan matematikklærere i ungdomsskolen er kommet i gang med å implementere programmering i matematikkfaget. Samtidig som det har vært interessant å se på hvorfor noen matematikklærere ikke er kommet i gang med å implementere programmering i matematikkfaget. Bakgrunnen for dette målet er LK20 og innføring av programmering i matematikkfaget fra skoleåret 2020/2021.

Jeg har valgt å avgrense studien til følgende forskningsspørsmål:

1. Hvordan har matematikklærere implementert programmering i matematikkundervisningen? og

2. Hvilke utfordringer møter matematikklærere i implementeringen av programmering i matematikkfaget?

Den nye læreplanen legger større vekt enn tidligere på at undervisningen skal være praktisk, relevant og variert. Resultater fra denne studien viser at programmering i matematikkfaget kan være med å bidra til nettopp dette. Informantene refererer til at programmering er med på å variere undervisningen, og at man i større grad kan bruke virkelighetsnære oppgaver, noe som gjør dem relevante og praktiske.

Informantene i denne studien fremhever at programmering kan brukes i mange temaer innen matematikk, og at det er velegnet til dybdelæring og tverrfaglig arbeid, som er fokusområder i overordna del i LK20 (Utdanningsdirektoratet, 2020c).

Flere av mine informanter mente at programmering er med på å gi mer motivasjon i matematikk og at en del elever får større grad av mestring og derigjennom mer engasjement, som er en av faktorene i Kilpatricks trådmodell for matematisk kompetanse. Gjennom programmeringsarbeidet har mine informanter identifisert: Begrepsmessig forståelse – av matematiske begreper, operasjoner og relasjoner, og da spesielt med tanke på den abstrakte matematikken. Beregning – gjennomføring av prosedyrer fleksibelt, effektivt og nøyaktig. Anvendelse – gjenkjenning og formulering av matematiske problemer, og utvikling av

løsningsstrategier. Alle disse tråder i Kilpatrick's trådmodell (Valenta, 2016) og (Kilpatrick et al., 2001).

Lærerne refererer også til en del av kompetansebegrepene Niss og Jensen (2002) har i sin modell. Spesielt problemløsningskompetanse og modelleringskompetanse vil man lettere kunne utvikle gjennom programmering i matematikk mener informantene. Symbol og formalismekompetanse er sentral. Sitater fra mine informanter og resultatet fra prosjektet ved Høgskolen i Østfold (Maugesten et al., 2021) viser at den matematiske kommunikasjonskompetansen kan øke ved å implementere programmering i matematikkfaget.

På grunn av stor variasjon i elevenes forkunnskaper fremhever informantene at det er viktig å bruke god tid på innlæringen, slik at man får med seg alle elevene. Da kan alle føle mestring. I den forbindelse er det viktig å bruke rike oppgaver som har lav inngangsterskel og stor takhøyde. Rike oppgaver er med på å gi alle elever mestring, samtidig som det er nok utfordringer til alle. De vektlegger også at det er viktig at elever som har jobbet med programmering i barneskolen også får noe å strekke seg etter.

Tidsaspektet er også noe som trekkes frem i forholdet mellom tidsbruk og nytte, med hensyn til å løse matematiske problem. Informantene viser til at man i innlæringsfasen nødvendigvis bruker mye tid, men at man på sikt vil tjene denne tiden inn når elevene har lært seg programmering godt, ved at elevene da kan løse avanserte matematiske oppgaver raskt og effektivt.

Det å benytte seg av metoden UMC-use, modify, create, ser ut til å være en suksessfaktor hos flere av mine informanter. Den gir en hjelp både til lærere og elever som går inn i dette med lite forkunnskap.

Resultatene fra spørreundersøkelsen og intervjuene i denne studien viser at et flertall av informantene opplever at de har fått for dårlig opplæring i programmering i matematikkfaget til å kunne implementere det i sin matematikkundervisning. Dette ser de på som en av de største utfordringene i dette arbeidet. Stofftrensel og tidsaspektet er også viktige faktorer som er kommet frem i denne studien. I tillegg er det stort sprik i elevenes bakgrunnskunnskap, altså i hvilken grad elevene har fått noen undervisning i programmering før de begynner på ungdomsskolen.

Til tross for noen utfordringer, er mange av informantene i denne studien kommet godt i gang med programmering i matematikkfaget og har en positiv holdning til at dette på sikt skal gå bra. Samtidig viser spørreundersøkelsen at 46% av informantene ikke er kommet i gang med programmering i matematikkfaget, til tross for at dette ble en del av kompetansemåla i matematikk fra og med forrige skoleår (2020/2021). Lærerne oppgir ulike årsaker til dette. Et flertall viser til at de selv mangler kompetanse. Men det er også andre årsaker, som at de ikke er kommet til programmering ennå i sin årsplan, og at de underviste etter gammel årsplan forrige skoleår. Dette er et resultat som er representativt for disse lærerne her og nå. Det er sannsynlig å tenke seg at resultatet vil være annerledes på et senere tidspunkt.

Konklusjonen blir altså at jeg har fått svar på hvordan noen matematikklærere har implementert programmering i matematikkfaget, og hvilke utfordringer noen matematikklærere har møtt i sitt arbeid med implementering av programmering i matematikkfaget.

6.2 Refleksjon og veien videre

Siden programmering i matematikkfaget er forholdsvis nytt, det ble innlemmet i kompetansemålene i matematikk i LK20 som trådte i kraft fra skoleåret 2020/2021, er dette et aktuelt tema å belyse. Men siden dette er et såpass nytt element i læreplanen, vil situasjonene rundt læreres implementering av programmering i matematikkfaget være i stadig endring de nærmeste årene. Denne studien er et her og nå bilde, som sier noe om hvordan situasjonen var i en middels stor østlandskommune høsten 2021. Undervisning og læring er en dynamisk aktivitet, som endrer seg over tid, det vil si at bildet vil se annerledes ut etter hvert som tiden går, og de største endringene når det kommer til implementering av programmering i matematikkfaget vil vi nok se de første årene. Det vil derfor være interessant å foreta en tilsvarende studie igjen om noen år. Forhåpentligvis vil resultatet da se annerledes ut enn i dag.

I arbeidet med denne studien er det gått bort mye tid til venting, og i ettertid ser man at tiden kunne vært utnyttet bedre. Det tok fem uker å få godkjenning fra NSD. Og det tok lang tid å få inn data fra spørreundersøkelsen. Denne tiden kunne med fordel vært utnyttet bedre. Aktuell litteratur kunne vært samlet inn og intervjuene kunne startet tidligere. Man kan jo også stille spørsmål i etterkant om denne spørreundersøkelsen var nødvendig for denne studien, da det både tok lang tid å få inn data, og det totalt sett ble liten deltakelse. Hovedmålet med spørreundersøkelsen var å finne deltakere til dybdeintervju, men de kunne

man funnet ved å kontakte ungdomsskolene i den aktuelle kommunen. Resultatene fra spørreundersøkelsen har gitt en del tilleggsinformasjon som er interessant for denne studien, og det er derfor relevant å bruke den her.

Tidspunkt for intervjuene kunne vært fremskyndet, men spørsmålet da er jo om man hadde fått de samme resultatene. For noen av informantene ville resultatet blitt det samme, da de i stor grad refererte til arbeid med programmering i matematikkfaget forrige skoleår.

Utgangspunktet for studien var ønsket om en mest mulig induktiv forskning, altså at resultatene skulle styre teorigrunnlaget. I ettertid ser jeg at det i stor grad er blitt en deduktiv studie. Spørsmålene i spørreundersøkelsen og i intervjuguiden er i stor grad styrt av forhåndsutvalgt teori. Selv om spørsmålene var relativt åpne, ga ikke resultatene grunnlag for å endre på valg av teori i særlig grad.

I oppstarten av prosjektet hadde jeg også planer om å observere undervisningssituasjoner hos noen av informantene, noe som kunne gitt et bredere grunnlag for denne studien. Det ble det ikke tid til. En av grunnene til det var at flere av informantene hadde planer om å bruke programmering i sin matematikkundervisning på slutten av skoleåret.

I tiden fremover er det viktig å få på plass tilstrekkelig opplæring av lærere i programmering i matematikkfaget, slik at alle kan føle seg klare for å denne oppgaven. Hvor vidt resultatene fra denne studien er representative for andre kommuner i landet, og eventuelt i andre land, er umulig si noe sikkert om. Men hovedinntrykket fra samtaler med lærere fra andre steder av landet er at situasjonen med hensyn til lærernes egenfølte kompetanse i programmering i matematikk er den samme der.

Denne studien har også hatt et personlig mål: hvordan jeg kan endre min praksis for å oppfylle målene i den nye læreplanen. Læreplan i matematikk, LK20, har som sine kjerneelementer: utforsking og problemløsning, modellering og anvendelser, resonnering og argumentasjon, representasjon og kommunikasjon, abstraksjon og generalisering, alle begreper som kan relateres til programmering i matematikk. Flere av disse kjerneelementene har vært utfordrende å jobbe med for meg som matematikklærer i ungdomsskolen.

Arbeidet med denne studien har gitt meg troen på at jeg kan få til å implementere programmering i matematikkfaget, til tross for manglende opplæring i det. Jeg er blitt positivt overrasket over informanternes positive resultater med hensyn til elevenes mestringsfølelse og økte matematiske kompetanse. Det at elevene lettere starter matematiske samtaler og skaper

en dypere forståelse for matematikk gjennom programmering skal jeg ta med meg videre i mitt arbeid som matematikklærer på ungdomstrinnet.

Referanser

- Berggren, S., & Jom, P. (2019). Fagartikkel: Lærerne er positive til programmering - men mangler kunnskap. *19:45*. <https://www.utdanningsnytt.no/fagartikkel/fagartikkel-laererne-er-positive-til-programmering---men-mangler-kunnskap/220753>
- Drijvers, P. (2013). Digital technology in mathematics education: Why it works (or doesn't). *PNA* (8(1)), 485-501.
- Fuglseth, K., & Skogen, K. (Eds.). (2012). *Masteroppgaven i pedagogikk og spesialpedagogikk design og metoder* (3.opplag ed.). Oslo: J.W.Cappelen Forlag as.
- Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. Oslo: Universitetsforlaget.
- Holand, A. (2012). Survey-forskning. In K. Fuglseth & K. Skogen (Eds.), *Masteroppgaven i pedagogikk og spesialpedagogikk* (pp. 41-51). Oslo: Cappelen Akademiske Forlag.
- Husain, H., Kamal, N., Ibrahim, M. F., Huddin, A. B., & Alim, A. A. (2017). Engendering problem solving skills and mathematical knowledge via programming. *Journal of Engineering Science and Technology*,(12), 1-11.
- Kaufmann, O. T., Stenseth, B., & Holone, H. (2018). Programmering i matematikkundervisningen. In A. Norstein & F. O. Haara (Eds.), *Matematikkundervisning i en digital verden* (pp. 73-96). Oslo: Cappelen Damm Akademiske.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73,, 26. doi:10.1016 /j.compedu.2013.12.010
- Kilpatrick, J., Swafford, J., & Findell, B. (Eds.). (2001). *Adding it up: Helping Children Learn Mathematics*. Washington DC: National Academies Press.
- Koehler, M. J., & Mishra, P. (2009). What Is Technological Pedagogical Content Knowledge (TPACK)? *Contemporary Issues in Technology and Teacher Education*,(9(1)), 60-70.
- Krumsvik, R. J. (2015). *Forskningsdesign og kvalitativ metode* (2.opplag Ed.). Bergen: Fagbokforlaget.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., . . . Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, 2(1), 32-37. doi:10.1145/1929887.1929902
- Matin, F., Lee, I., Nicholas, L., Sentance, S., & Lao, N. (2020). Extending and Evaluating the Use-Modifying-Create Progression for Engaging Youth in Computational Thinking. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 807-808. doi:10.1145/3328778.3366971
- Maugesten, M., Stigberg, H., & Stigberg, S. K. (2021). Programmering på ungdomstrinnet. *tangenten, tidsskrift for matematikkundervisning*, 3, 2-7.

- Niess, M. L. (2008). Mathematics Teachers Developing Technology, Pedagogy and Content Knowledge (TPACK). *Association for the Advancement of Computing in Education (AACE)*.
- Niss, M., & Jensen, T. H. (Eds.). (2002). *Kompetencer og matematikl ring*. K benhavn: Undervisningsministeriet, Danmark.
- NOU 2015: 8. (2015). *Fremtidens skole — Fornyelse av fag og kompetanser*. Retrieved from <https://www.regjeringen.no/contentassets/da148fec8c4a4ab88daa8b677a700292/no/pdfs/nou201520150008000dddpdfs.pdf>
- Pepin, B., Gueudet, G., & Trouche, L. (2017). Refining teacher design capacity: Mathematics teachers' interactions with digital curriculum resources. *ZDM Mathematics Education*(49), 799-812. doi:10.1007/s11858-017-0870-8
- Pierce, R., & Stacey, K. (2010). Mapping Pedagogical Opportunities Provided by Mathematics Analysis Software. *Int J Comput Math Learning*(2010) 15:1-20, 20. doi:10.1007/s10758-010-9158-6
- Ruthven, K. (2014). Frameworks for Analysing the Expertise That Underpins Successful Integration of Digital Technologies into Everyday Teaching Practice. In A. Clark-Wilson, O. Robutti, & N. Sinclair (Eds.), *The Mathematics Teacher in the Digital Era* (Vol. 2, pp. 373-393): Springer.
- Sevik, K. (2016). Programmering i skolen. Retrieved from https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Utdanningsdirektoratet. (2019). *Algoritmisk tenking*. Retrieved from <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2020a). *Kjerneelementer*. (MAT01-05). Retrieved from <https://www.udir.no/lk20/mat01-05/om-faget/kjerneelementer?lang=nob>
- Utdanningsdirektoratet. (2020b). *L replan i matematikk fellesfag 1.-10. trinn*. Retrieved from <https://data.udir.no/k106/v201906/laereplaner-lk20/MAT01-05.pdf>
- Utdanningsdirektoratet. (2020c). *Overordnet del - verdier og prinsipp for grunnp lringa*. Retrieved from <https://www.udir.no/lk20/overordnet-del/?kode=nor01-06&lang=nob>
- Valenta, A. (2016). Kognitive krav i matematikkoppgaver. *Matematikkenteret*, 1-15.

Vedlegg1: Kvittering fra NSD



Vurdering

Referansenummer

760671

Prosjekttittel

Masteroppgave i matematikdidaktikk

Behandlingsansvarlig institusjon

Universitetet i Agder / Fakultet for teknologi og realfag / Institutt for matematiske fag

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Nils Kristian Hansen, nilskh@uia.no, tlf: +4738141523

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Tove-Mette Reznici, tove-mette.reznici@sandefjord.kommune.no, tlf: 92221552

Prosjektperiode

13.09.2021 - 17.06.2022

Vurdering (1)

08.10.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 08.10.2021, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 17.06.2022.

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.

TAUSHETSPLIKT

Infomantene i prosjektet har taushetsplikt. Datainnsamlingen må gjennomføres uten at det fremkommer

opplysninger som kan identifisere elever.

PERSONVERNPRINSIPPER

NSD 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).

NSD 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

NSD 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).

Spørreskjemaleverandøren vil være databehandler i prosjektet. NSD legger til grunn at behandlingen oppfyller kravene til bruk av databehandler, jf. art 28 og 29.

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 NSD 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/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema>

Du må vente på svar fra NSD før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

Vedlegg 2: Samtykkeskjema

Vil du delta i forskningsprosjektet

Programmering i matematikk?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å *finne ut hvordan matematikklærere i kommunen som er kommet i gang med programmering i matematikkundervisningen*. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

Jeg holder på med en master i matematikk, og ønsker å finne ut hvordan matematikklærere er kommet i gang med programmering.

Forskningsspørsmål: «Hvordan har matematikklærere implementert programmering i matematikkundervisningen?»

Hvem er ansvarlig for forskningsprosjektet?

Universitetet i Agder(UIA) er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?

Jeg ønsker å finne ut hvordan matematikklærerne i kommunen som er kommet i gang med programmering i matematikkundervisningen

Hva innebærer det for deg å delta?

- *«Hvis du velger å delta i prosjektet, innebærer det at du deltar i et dybdeintervju om din praksis med hensyn til implementering av programmering i matematikkfaget. Dette vil ta deg 15-30min.*

Det er frivillig å delta

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

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- *Det er kun jeg som student og min veileder ved UIA Nils Kristian Hansen som vil ha tilgang til disse dataene.*
- *Navnet vil jeg kun bruke til å plukke ut intervjuobjekter og vil ikke bli lagret.*

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er **juli 2022**.

Dine rettigheter

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

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra UIA har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?

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

- UIA ved *Nils Kristian Hansen*

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

- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Med vennlig hilsen

Nils Kristian Hansen
veileder

Tove-Mette Reznqi
Student

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)

Vedlegg 3: Intervjuguide

Intervjuguide:

Denne intervjuguiden sees kun av intervjuer under intervjuet.

Dette er et notat for å hjelpe intervjueren med å huske spørsmål og for å gi struktur under selve intervjuet. Før intervjuet Presentere seg selv og målet med studien.

- Masterstudie om “Hvordan lærere har implementert programmering i matematikkfaget?»

Informasjon om opptaket: - Ønsker å ta opp intervjuet digitalt slik at jeg kan transkribere det og få med alle nyanser i intervjuet. Opptaket er kun tilgjengelig for meg, veileder og sensor.

Informantene blir anonymisert ved transkripsjonen. Opptaket slettes lokalt i november, men blir brent på CD, for at sensor kan ha tilgang til dette under sensuren. Intervjutype:

Semistrukturert intervju: Kommer til å stille spørsmål rundt valgte temaer og det kommer evt. oppfølgings spørsmål basert på deltakernes svar.

Lurer du på noe? HUSK: lytte, tillate stillhet, gi feedback, smile Hjelpespørsmål: Kan du utdype det? Ville du sagt at.... Mener du da at.... Hva tenker du om det...

Innledende sp.m til alle informanter:
1. Hvor mange år har du undervist i matematikk?
2. Hvilke(t) trinn jobber du på i år?
Faste spørsmål:
1. Hvordan jobber du med programmering i matematikkfaget?
2. I hvilke temaer i matematikken kan elevene ha nytte av programmering?
3. Hvordan er programmering med på å gjøre matematikkfaget mer spennende for elevene?
4. Hvordan ser elevene sammenhengen mellom programmeringen og matematikkfaget?
5. Hvilke fordeler ser du med å bruke programmering i matematikkfaget?
6. Hvilke utfordringer opplever du ved å implementere programmering i matematikkfaget?

Vedlegg 4: Transkripsjoner

Intervju 1:		
nr		sitat
1	I	Eee Da lurer jeg på hvor lenge du har undervist i matematikk?
2	L1	Jeg er vel inne i mitt tredje år nå, to fulle år, også hadde jeg et år hvor jeg var litt inne som vikar, litt sånn spredt, så med vikartimer så, to, to og et halvt år da ja.
3	I	Hvilket trinn jobber du på nå?
4	L1	Nå jobber jeg på niende trinn
5	I	Hvordan jobber du med programmering i matematikkfaget?
6	L1	Det har vi nettopp startet med i niende trinn. Da har vi fokusert på, nå først å lære hvordan du setter opp et mattestykke i pyton. Det er jo ganske kjedelig måte. Det er litt sånn programmering for programmerings del. Men det må være i bunn, for at du skal klarer å gjøre noe morsomt videre. Da må du skjønne hvordan ganging og pluss og minus, potens og deling skal settes opp i pyton.
7	I	Men har du jobbet med andre programmer tidligere?
8	L1	Jeg jobbet med det på videregående, litt programmering der. Der var det også litt samme innfallsvinkel, startet med litt sånn basic ting, som dette som gjør matte. Dette er en variabel. Sånn her kan du be om å få verdien til variabelen, og så kommer jo helt til setninger der i løpet av et år. Vi skal komme litt lenger nå når vi starter med det tidligere, ikke sant.
9	I	Men det er vesentlig python du har brukt eller?
10	L1	Ja, det er, det er gratis.
11	I	Smart, hvilke temaer i matta kan elevene ha nytte av programmereren tenker du?
12	L1	Det kan jo i stor grad brukes i alle temaer. Nå er det jo plangeometri som vi har jobba med, utenom programmering i det siste, og det er egentlig, det er egentlig veldig fin ting å ha i forkant og, for da har du mye formler. Og formler er jo python eller programmering helt gull å skulle bruke til da. For da kan du sette opp, da har du formelen for arealet av en sirkel, så trenger du bare å bytte ut radiusen underveis hvis du skal regne en ny sirkel og så har du oppsettet klart. Der har jo også ting standard variabler du bruker, lengde, bredde, L, B, r for radius, d for diameter, h for høyde, g for grunnlinjen.
13	I	Ja, i begrunnelsen til udir for å putte programmering i matematikkfaget, så er jo en av begrunnelsen det at tenker at det skal gjøre matematikk faget mer spennende for elevene. Og hvordan tenker du at det kan stemme?
14	L1	Tja for noen vil det definitivt stemme. Har sett noen elever nå og også tidligere som våkner opp når de sitter og jobber på maskinen. De ser klarer å se at det er sånn alt av apper og alt av dataspill fungerer på kode. At du kan legge opp ting til å jobbe for deg. Hvis du legger det opp på en smart måte, er det jo mange som synes er spennende. En del av de som allerede sliter med å se nytten av matematikk, de er veldig på. Ja, hvorfor i all verden skal vi lære dette her også? Ja sånn som de sier om algebra og ligninger, og da.
15	I	Så du tenker at det er litt delt?
16	L1	Det er, det er nok litt delt. Jeg tror fort altså det er noen kanskje som ikke er så interessert i matematikken, eller som kanskje kobler seg litt på at det blir litt annerledes.
17	I	Tenker du at det er noe forskjell på gutter og jenter når det gjelder dette?
18	L1	Har observert på videregående. Når jeg hadde dette der, så merker jeg litt forskjell. Det kan også være at det var to forskjellige linjer som så det. Jeg kan si at den, der de fleste guttene gikk, det var medier og kommunikasjonslinja, så det kan nok

		også spilt mer inn der. Ja så blir den der. Ja, det er mer interesse, enn kjønn så klart hvis det er flere, det er flere gutter som er petes interessert, kanskje så da vil du bli en fordeling deretter.
19	I	Men hvordan tenker du at elevene kan se sammenheng mellom programmering og matematikkfaget?
20	L1	Nei, det bruker jo programmeringen til å gjøre matte i stor grad nå det, det er jo en veldig fancy kalkulator på mange måter, og det er kalkulator du kan legge opp, legge opp mye selv. Så den kan jo for de sterkeste så tenker jeg at det her kan bidra til dybdeforståelsen. At, at de må faktisk, bearbeide matten, og bryte den ned for å lage kode, at de ser at det er nytte. Og så for kanskje de mindre sterke, så er det et ekstra hjelpemiddel, da kan du gjøre unna det der en gang, også kan du få regnet ut en del regnestykker uten å få noen andre dumme feil som du kan få på en vanlig kalkulator.
21	I	Ja, så da har du egentlig svart litt på neste spørsmål også. Hvilke fordeler du ser man bruk av programmeringen i matematikkfaget? Men er det noen andre ting du tenker på i forhold til fordelene?
22	L1	Nei, skal du se, det er, det er mest denne dybdeforståelsen å tvinge dem til å bearbeide. Det er jo også en ekstrem øvelse i å være logisk og konkret, og konsis. Har jo åpnet med å vise dem en sånn video. En som ber om å få eksakte instruksjoner for å lage en brødiskive med peanøttsmør og syltetøy, hvor han gjør akkurat det som står på arket å få inn at sånn her er det python funker, og eller programmering funker også, at du må være veldig presis i språket ditt. Du må sette opp ting i riktig rekkefølge, du kan ikke løse ting i så stor grad på innfallsmetoden, at det bare regner litt her og litt der, den må faktisk ha veldig presis rekkefølge på hva den skal gjøre?
23	I	Ja da er det jo på en måte litt inne på neste spørsmål også, hvilke utfordringer du ser ved å implementere programmering i matematikkfaget. Men det kan jo være andre utfordringer også kanskje?
24	L1	Ja, ja, det er jo for mange, så er det jo krevende, det at den må være så ekstremt konsis og jeg vet ikke, det føles kanskje verre å få en sånn stor rosa syntax error, kontra få vite at svaret du regner ut var feil. Det føles ganske ekstremt når det står liksom masse ting du ikke forstår og type error og syntaks error og unexpectet prentesis, og ikke måte på, Og så kan det jo da fort, jeg lurer på om det kan stjele tid noen ganger, for det tar tid å sette seg inn i programmeringen, at den tiden kanskje hadde vært mer nyttig noen ganger å bruke på. Å forstå matten mer med penn og papir. Ja og så er det jo på samme måte som noen kobler seg veldig på når du drar fram datamaskinen, så er det jo noen som bare kobler seg helt ut.
25	I	Så det jeg tenker da det er at du ser liksom både nytteverdi og, og utfordringer hva du tenker du veier tyngst eller hva tenker du i forhold til det, og at vi nå skal, skal jo bruke det. Tenker du at det har den nytteverdien som kanskje ligger er som en begrunnelse fra udir da?
26	L1	Ja om ikke annet, så er det et ekstra verktøy å bruke. Det er med på å variere undervisningen. For mange så tror jeg det, for alt du kan si om programmering kan du nesten også si om å regne med penn og papir og liksom nytte kontra utfordring, så for variasjonens del så er det jo fint. Bare det at du får litt andre måter å jobbe på i mattefaget, og så er det jo i bunn og grunn den samme forståelsen du trenger for å få sett opp systemet riktig ja, du kan jo få økt forståelse av for eksempel areal av trekanten ved å sitte å regne oppgaver eller ved å lage program som løser det. For da må du også faktisk teste at det funker for rare trekanten også.

27	I	Det er jo som du sier at man må jo bryte det ned i veldig i de..., hvis ikke man på en måte gjør det helt riktig, så blir det jo veldig sånn.... Da blir det fort feil, ja, så det blir veldig tydelig da. Eller så er det noe du tenker at du ville utdype eller noe du tenker i forhold til programmering som ikke har vært inne på?
28	L1	Ja, det er jo, jeg har funnet en sånn morsom... , det er mye gøy man kan finne på med programmering da, jeg har jo et program som demonstrerer proteinsyntesen. Hvor du kan skriver inne en streng av, av RNA enke da AB nei, AU GC og så kan man oversette det til hvilke, hvilke aminosyrer som bygges opp og så fikset jeg den litt så den oversatte fra en RNA-frekvens til en setning, så reverserte jeg den, så den kunne gå fra en setning til en RNA-sekvens, og det var en veldig fin måte å vise mutasjon på. Da skrev jeg først dette er en lesbar og forståelig setning, kjørte den fikk ut AUG blablabla så kopierte jeg den til den motsatte koden også fjernet jeg den første bokstaven, siden den går på tre og tre i DNA sekvensen, så ble alt bare forskjøvet så det gikk fra, først kjørte jeg det uten noen endringer, også fikk den ut dette er en lesbar og forståelig setning, også fjernet jeg den første bokstaven, og da ble det svursj.... etterpå
29	I	Da skjønnte de at det var en mutasjon?
30	L1	Bare for å gjøre det ekstra forståelig, så la jeg inn for jeg hadde flere tripletter enn det er bokstaver, den hadde både stor og liten bokstav skilte den på, jeg la også inn spesialtegn som utropstegn, #, @, så det ble enda mere merkelig setning ut av det.
31	I	Men det høres jo ut som du er godt i gang med programmering da både i matte og naturfag. Så det er jo kjempebra.
32	L1	Så er det jo en liten engelskbit i det også det som er fint med python er jo at det er mye brukt rundt om også, og er veldig standard måte ting er satt opp på, noen som gjør at de lett kan google seg frem til noe andre har gjort, da må du fort lese litt på engelsk også. Da ligger det gjerne på github som går på engelsk.
33	I	Da blir det kanskje litt tverrfaglig også?
34	L1	Ja da kan du få det ganske tverrfaglig i programmeringa.
35	I	Det er jo genialt i forhold til den nye læreplanen. Hvor det legges opp til mye tverrfaglighet.
36	L1	Også er jo syntaks error en god mulighet til å snakke om syntaks, som er i alle språkfag
37	I	Ja det er jo ting ikke jeg hadde tenkt på i alle fall, men det er klart at det krever at du har litegrann engelsk kunnskaper også når du starter opp med programmering. Ja, Nei, hvis ikke du har noe mer enn du tenker, så lurer på om vi bare skal ..
38	L1	Jeg kan jo bare, også litt inne i engelsk biten nå, for der har du noen sånne kommandoer som er engelske ord input, og då har du for som bruker som brukes så den engelske four, in range, et par sånn ord som hvis du vet hva de betyr, if els. selvfølgelig. Nå må du skjønne hva disse ordene betyr for å skjønne hvis du skjønner hva de ordene betyr, så hjelper det deg å kode litt raskere og.
39	I	Men tenker du at det er noen elever som, at de føler det blir liksom veldig overveldende at de liksom..
40	L1	Det kan nok fort bli det? Jeg prøver å holde meg igjen. Men, men jeg syns jo dette er veldig moro å holde på med, også skal jeg bare vise dem no kult. Da kan det bli litt mye for noen.
41	I	Da detter de litt av? men tenker du at det er liksom får med deg sterke og svake elever at det er en god miks?

42	L1	Ja, det tror jeg det er. Ja, de sterkt får du jo med uansett på en måte, men jeg tror det er. Ja, de som responderer bedre på programmeringen enn typisk penn og papir, så er det nok en god miks. Det er noen ja, altså noen kanskje de svakere som ser at her kan jeg få mye hjelp til matten, den kan gjøre matte for meg. Mens de som kanskje synes det er tungt å sitte å sette opp de samme regnestykkene om og om igjen, og er redd for å få sånne småfeil som blander pluss og minus. Hvis du har satt opp koden riktig, så vil du aldri få den problematikken der da, om du bare viser dem at du kan jo nevne for dem. Dere kan bruke den koden på terminprøven så. Ja, så er det en motivasjon i det også. Det gir en motivasjon.
43	I	Så man kan faktisk fange opp en del svake elever da, som kan tenne på dette?
44	L1	Jeg tror det
45	I	De har kanskje datainteressen?
46	L1	Ja kanskje datainteressen er der og man kan jo vise dem, ja du kan bruke denne koden på terminprøven så er det en motivasjon i det også.
47	I	Det er klart det gir en motivasjon ja. Ja, nei, men det så spennende ut så det. Det er jo veldig, veldig bra at du er godt i gang

Intervju 2:		
nr		sitat
1	I	Ok da begynner jeg med å spørre hvor mange år du har undervist i matematikk.
2	L2	I fem år.
3	I	hvilket trinn her du jobber på
4	L2	Nå er det niende
5	I	Så bra. Hvordan er det du jobber med programmering i matematikkfaget?
6	L2	<p>Og akkurat nå så er det jo ganske nytt i det faget, så det er jo på en måte, noe vi må på en måte, hva skal jeg si, bane vei inn i, og prøve oss frem i, i ganske stor grad. For ingen av oss har noe særlig kompetanse i det sånn formelt. Nei, og da blir det fort til at man søker mye på nettet, og prøver å finne opplegg som jeg tenker at elevene kan starte med. Og så er det jo vanskelig og må tenke, hva er det barna kommer med fra barneskolen, det er på en måte som både har vært, hva kan de, kan alle det, og det er jo. Det å måtte komme seg inn i den veien at alle sammen har muligheten til å få vært med da, for det er litt kjipt hvis vi begynne for høyt og elevene ikke forstår hva programmering egentlig er. Det sånn vi på en måte begynt, og vi starter egentlig opp med at elevene skrev koder for hånd.</p> <p>Skrev på en måte en fortelling som om det skulle styre en annen elev. Og da videre forklart, at det her er jo sånn en kode er bygd opp må starte en spesiell rekkefølge for at du skal få det til å skje i den rekkefølgen du ønsker å gå videre på. En måte at jeg som lærer skulle bli styrt av en av kodene til elevene, fordi at da måtte jeg gjennomføre en handling. Og for å være litt vrang, så gjorde jeg den akkurat sånn som det stod. Så hvis det for eksempel står der at man skulle løfte opp et egg. Så tok jeg på en måte ikke nødvendigvis tak i egget. Jeg løfta det bare nettopp fordi at vi skal ikke skrevet at jeg skulle ta tak i det, så jeg tenker sånn da og hadde hånden min over for eksempel egget, men løftet ikke opp. Altså de skulle på en måte fått meg til å knuse egg i en bolle. Så de måtte liksom lære på den måten at jeg måtte de være veldig detaljert i koden sin og en mer detaljert kode gjøre at jet er mer presis. Vi starter jo der før vi gikk over til blokkprogrammering da.</p>
7	I	Ja så bra det ja når dere startet da med blokk, blokkprogrammering, lot du elevene bare begynne for seg selv, eller hadde du begynte som et ferdig opplegg?
8	L2	Da starter jeg med at de på en måte fikk prøve seg rundt i programmet. Hva er det på en måte de ulike blokkene gjør. Jeg prøver å sette sammen noe, og da merker man ganske fort at noen hadde gjort det her før, og noen lurte på hva de knappene var til å de på en måte å prøve å fikse på en måte, fikk beskjed om å prøve å klikke seg rundt og se og gjøre feil da prøve og feile litt. Og da fikk de på en måte etter hvert muligheten til å sette sammen ulike sekvenser, noen mer avanserte enn andre. Og, og så hadde vi på en måte en gjennomgang av programmet da sånn totalt sett etter jeg hadde fått prøvd litt sånn at vi er på en måte helt klare på at det var jo de ulike knappene gjør og hvordan de setter inn bakgrunner og helt sånn grunnleggende oppbygning av programmet da.
9	I	Ja, men jeg tenker litt mer sånn når du liksom skulle bruke det til en matematikkoppgave for eksempel. Når på en måte hadde fått lukter litt på det da, og de skulle løse en oppgave. Hvordan har du lagt opp? Da fikk de også da lov å prøve og feile seg fram eller? Fikk dem først en ferdig, altså de
10	L2	På en måte fikk jo altså, hvis du eksempler på en måte på hva de kan oppnå og liksom på en måte for å da hente jeg på en måte eksempler fra nettet, og på en

		<p>måte visst at så avansert kan det være. Her skal ikke vi nødvendigvis komme, men her kan dere se potensialet programmet fordi at for eksempel scratch. Kan på en måte virke veldig enkelt, når de først begynner, men når du på en måte ser hva andre har greid å bygge opp i det enkle programmet, men ganske avanserte programmer. Så på en måte ser du på en måte elevene, at her kan du faktisk nå ganske mye lenger. Og da visste jeg på en måte, det her er en måte. Kan du det her, så er det på en måte super i det, men det er ikke her jeg forventer at dere skal komme, men noen av dere har potensial og så på en måte visst jeg på en måte en helt enkel kode som jeg hadde laget som gjorde bare enkle handlinger og første oppgaven vi hadde innen programmering var jo en måte å prøve å lage et slags spill da. Eller en slags handling, for vi hadde et tverrfaglig tema med matte og naturfag, så matematikken tok jo dette med algoritmer og oppbygning, syntaks, og så er det på en måte naturfagen som på en måte selve oppgaven handlingen. For de skulle da programmere en naturfaglig handling da. For eksempel, forklarer fotosyntesen.</p>
11	I	<p>Men fordi jeg har vært borti en eller to. Det jeg har prøvd da det er etter en metode som kalles. På kalles for umc, altså use, modify og create, og så at man på en måte får en kode og så eller et opplegg og i koding da eller med scratch for eksempel og så at man altså at elevene skal, først testet den koden som en har laget, og så at de kan endre på den og så at de lager noe fra scratch liksom</p>
12	L2	<p>Det er noe vi også har funnet. Jeg fant en oppskrift, for eksempel på sånn påske spill, for det her er noe vi begynte med påske, og da fikk vi på en måte sånn gjør dette gjør dette, gjør dette, så vi på en måte gjorde gradvis. Litt sånn ulike timer, og da fikk du på en måte, etterhvert det samme spillet som meg, og noen måte valgte jo på en måte å prøve nesten å kopiere det spillet, men da ville ha sin egen handling på det. Noen måte valgte jo da mer å prøve å gå nye veier da, men kanskje med noen av de samme funksjonene.</p>
13	I	<p>Ja da blir jo det noe av den metoden. Det var for det jeg var litt interessert i om du hadde prøvd det. Hvilke temaer i matta tenker du at elevene kan ha god nytte av programmering?</p>
14	L2	<p>Vi bruker jo det ganske mye mer på en måte forstå hva for eksempel algoritmer gjør da at det er på en måte en sånn forklarelse på at detaljer er viktig ved sånn rent temamessig, så kan man jo bruke til å innen de fleste tenker jeg. Så man kan jo lage statistikk kalkulatorer ved bruk av den mann som spesifikt går på at taster inn dette. Så får du gjennomsnittet, i geometri kan man jo lage ulike figurer. Så det er på en måte ganske stort og ligninger til og med sånn sett.</p>
15	I	<p>Så bruker du ulike type programmering og så både blokk og tekst</p>
16	L2	<p>jeg er ikke helt kommet til tekst med dem, og da har vi på en måte prøvd å legge til niende klasse, at åttende klasse begynner på blokkprogrammering på en måte forstått prinsippene rundt programmering, og så på en måte for de som på en måte synes at blokkprogrammeringen blir kjedelig. Fordi at det er for lett, så på en måte utfordrer de på en måte å sette seg litt inn i helt grunnleggende python på en måte, og for å på en måte få den lille gulrot at vi har strekke seg litt videre da, fordi at de kanskje har hatt blokkprogrammering helt siden femte klasse da.</p>
17	I	<p>Ja dette, for det er sikkert ulike nivåer blant elevene når de kommer hit?</p>
18	L2	<p>Vi vi merker de som på en måte har hatt noen som på en måte har vel virkelig kjørt på med programmering i barneskolen, og så og de som så vidt vet hva det er.</p>

19	I	Det er jo kanskje den store utfordringen eller en av de. En av begrunnelsene for. For at programmering ble lagt til matematikkfaget og er jo at det tenkte kanskje at vi matematikk faget skal bli mer spennende da for elevene, hva? Hvordan tenker du at programmering kan være mer spennende eller gjøre matte og mer spennende?
20	L2	Altså, det er jo litt noe annerledes kan man jo si. Det er ikke den standard her, en oppgave å løse den, og så får du et svar. Jeg, jeg synes jo det med programmering er ganske interessant selv med tanke på at oppgavene kan være ganske åpne. Du kan få ganske mange resultater og, det åpner opp for at de må reflektere på en helt annen måte også, og at de også må bli vant til dette her med å prøve og feile. Altså. Det er ikke sånn at koden din virker perfekt første gang vi merker jo det veldig fort at elevene sleit med å på en måte være prøve seg fram. Da endrer litt her endelig der litt sånn som du kanskje må gjøre matematikken hvis du får feil svar da at du må resonnerer deg fram til at det tallet er ganske stort og litt sånn som at det var feil i koden min. Og så er det jo det at vi ser på guttene for eksempel lyste spesielt opp, men også at jentene blir også ganske interessert i hva det her er for noe. Så de kaster seg over en veldig åpen og rik oppgave for eksempel. Dette med alle skulle lage et spill da.
21	I	Tenker du at det er noe stor forskjell på gutter og jenter? Hvordan eller hvor interessert de er og å lære seg programmering?
22	L2	Mitt inntrykk er at guttene er litt mer på, når vi altså komme over til tekstprogrammeringa. Di er spør ofte om det, når skal vi dit og og. Mens når det kom til blokkprogrammering så synes jeg at begge kjønn kastet seg over det ganske likt selvfølgelig litt ulike temaer på det de lager, men de på en måte fant hva de hadde lyst til å fokusere på og tok inspirasjon fra hverandre. Og det var ganske mange fine samtaler på tvers av kjønn som gjorde at de kunne komme med ideer til hverandre.
23	I	Tenker du at det er noe forskjell på sterke og svake elever sånn hvis du tenker sånn matematikk generelt? Hvordan de ligger sånn til vanlig i matematikk, om det er en forskjell på hvordan de griper an det med programmering.
24	L2	Jeg føler jo at de kaster seg også mer på at de på en måte kanskje merker at her er det lav inngang da. Her kan jeg også få til noe, og så merker jeg jo på en måte at de som på en måte kanskje sliter i matematikk, har vi faktisk fikk til programmering på en helt annen måte. En de som på en måte kanskje klarer seg veldig bra der, så jeg synes jo på en måte at du får på en måte med de på en enklere måte litt mer naturlig.
25	I	at du får løfte, løfte litt i bunn.
26	L2	Ja, en måte merket at de hadde lyst til å gjerne ha matematikk mer liksom mer motivasjonen og mestringsfølelsen kom fra, og den smitter jo litt over og så videre i faget. At nå fikk jeg til det i matte da kanskje jeg kan få til mer.
27	I	Ja det har jo det er jo positivt hvis man kan få dratt med flere, for det er jo synes jeg i hvert fall vært en utfordring at mange har liksom dratt ned en rullegardin før de begynner på ungdomsskolen, og at den ofte er vanskelig å få opp igjen. Absolutt at jeg kan vekke interesse, så er jo det. Så helt klart positivt. Så tenker du at de klarer å se sammenhengen mellom programmering og matematikk
28	L2	Tror det er litt vanskeligere. Jeg føler på en måte at de. Tar og skiller det som på en måte eget tema, som på en måte ikke nødvendigvis at vi følger den naturlige matematikken, men på en måte at den var nå har vi på en måte mulighet til å gjøre det her litt litt sånn som nesten skulle sette en film i timen

		på en måte som kanskje ikke har noen mening med faget som kanskje de ser, men at de på en måte føler at det er sånn avbrekk fra matematikken da. Så det er vel mer den sånn som man sier at man nesten lurer matematikken inn, selv om på en måte de ikke kanskje forstår at det er matematikk de har.
29	I	Ja, ikke sant? Og da blir det kanskje gjennom fordelene da ved å bruke programmering i mattefaget.
30	L2	Ja, du får på en måte, så kan du jo måtte etterhvert kanskje jobbe mer med å få bruke programmering inn i matematikken på en helt annen måte, og da snakke om, hva er det du husker du i programmering. Det var på en måte en algoritme, dette her er også en algoritme, på en måte bruker begrepene som på en måte vil prøve, og på en måte bruke ganske greit med. Innføring er litt mer sånn syntaks oppsett i. Programmeringen, mens algoritmen er jo mer oppskriften da, som du kan bruke.
31	I	Så ja, er det andre fordeler du ser og bruker programmering i mattefaget?
32	L2	Jeg synes at du åpner jo for mye da altså tverrfaglig på tvers. Altså. Det er mye enklere å på en måte samarbeide med prosjekter med andre, fordi at hvis de har om de har et tema i samfunnet, så kan man på en måte på en måte trekke programmering opp mot det og på en måte lage en fortelling i form av det eller forklare noe faglige få programmet til å regne på et eller annet, så det er jo mye enklere å få til tverrfaglig opplegg på tvers, men det er kanskje vanskelig å samarbeide med for mange fag, da det kan du for eksempel tenke på at du kunst og håndverk enn kun å jobbe med et eller annet grafisk knytta til det å laget med bakgrunnen for eksempel. Mens matematikk som tar selve programmet, og kan noe annet ta selve handlingen.
33	I	Og det tenker jeg er fint ting nå som da er vi skal jobbe mer tverrfaglig og mer temabasert da så er det jo flott hvis vi kan få det til. Men utfordringen er kanskje selve opplæringa.
34	L2	Vi har jo heldigvis fått noen kurs her og der, men vi føler på en måte at vi tolker kompetansemålene på en måte, og at de som på en måte kurser oss tolke det på en litt annen måte. At de får det til å se veldig enkelt ut mens jeg føler kompetansemålet i programmering kan være så mye mer. Så. Det kan jo være at det er vanskelig å på en måte å si hva de forventer på eksamen.
35	I	Ja, det kan jo bli spennende
36	L2	når du ser på en måte kursene sier at du kan programmere en lego robot, og så er det egentlig, kompetansemålet for 10. trinn. Mens du kan lage en eller en terning i niende klasse med sannsynlighet, og så har du gjort det kompetansemålene inntrykk jeg jeg føler det blir for enkelt. Det er jo på en måte den hadde vært litt bedre med. Klare føringer på dette er maks nivå. Vi tenker ungdomsskolen.
37	I	Hvilke utfordringer ser du ved å implementere programmering i matematikkfaget?
38	L2	Dette er en ting til som tar plass. Man føler at da er det fort enda et tema man må innom. Og så er det jo det her med, hvordan skal man få den naturlig inn og hvilken rekkefølge skal komme i og så videre. Og det er sant at jeg på en måte føler den største utfordringen er jo på en måte, og hva skal vi lære selv. Jeg føler jo på en måte mye av det første jeg hadde den, og så var det jo mer enn at jeg leste meg opp og elevene kommer rett etter. Og at det etter hvert ble noen elevene som gikk forbi meg, men jeg tenkte å si på en måte ganske åpent til eleven at dette er ikke noe jeg kan skikkelig godt, så jeg lærer samtidig som dere, og det tror jeg også elevene synes var egentlig ganske greit, fordi vi måtte

		jo på en måte gå sammen om å løse problemet med koden. På en måte som var en utfordring ble også kanskje noe positivt da?
39	I	Og det kan jo faktisk være altså at de føler en ekstra mestring også ved at vi legger oss flate og sier at dette er ikke noe vi kan godt heller.
40	L2	Og at de ser at jeg faktisk også må sette meg ned å jobbe ja, ikke sant? At jeg gir meg ikke selv om jeg møter på et problem.
41	I	Ja for det tenker jeg kanskje er en av de største utfordringene i mattefaget, spesielt da det er at de. Ikke er så veldig utholdende at de skal gjerne ha svar og sånn og gir så det er liksom, men sånn er det, Mer tid, men tenker du at, tenker du at det kan være det at det etter hvert at det kan være litt sånn at man kan at den tida kan på en måte hentes inn også da? altså etter hvert som du blir god på det liksom
42	L2	Ja. Jeg tenker jo på en måte det at når kompetansemål litt lenger ned i skolen blir gjennomgått. Sånn klassevis nå at nå skal vi på en måte begynne ganske tidlig med det, at da slipper jo kanskje jeg begynner med blokkprogrammering og kommer i åttende klasse. Selv om det kanskje for noen elever vil kanskje være greit å jobbe med blokkprogrammering, så kan vi på en måte være litt derfor mer trygg på at dette her, og at de har da litt mer kan gå over til tekstprogrammering på en måte jobbe med det på. For der har du litt andre alternativer på programmering i det. Så jeg føler på en måte at man kan hente det inn på den måten, men jeg tror det er jo en del år til, og da er det jo også litt sånn spennende om det hva som kommer i mellomtida. Men på den tida så har vi også erfart litt med hva som forventes
43	I	Nei, fordi at det har. Det er vel kanskje en av de største eller det som har vært mest motstand da mot, og at det programmering skal bli en del av matematikkfaget. For det her var jo en diskusjon før det ble implementert i kompetansemålene var jo at om det skulle være et eget fag. Eller om det skulle inn i de eksisterende fagene. Og da er det jo da var jo en av en av poenget var jo at det. At det vill tar tid da fra, og det ser man jo også. Jeg, jeg tenker bare når man startet med geogebra for eksempel, så var jo det. Det tok jo mye tid i begynnelsen, men du kan si etter hvert som man lærer seg å bruke det, så går det jo mye kjappere. Og enn å tegne de her gamle grafene på millimeter papir, innså det.
44	L2	Ja, jeg vil jo tro at vi kan måtte etterhvert få flere temaer inn i programmering, at vi kan på en måte litt sånn som med geogebra mot konstruksjonen kom inn der og slapp på en måte all den passeren, selv om jeg også savner den passeren, så det er som regel en liten time eller 2 hvor vi sitter og jobber med passer bare for å at de skal se hvor sirkler og sånt kommer da noe. Geogebra hjelper ganske mye da, og da tenker jeg også programmering kan komme inn på en positiv måte at vi etterhvert ser at der kan du gjøre ganske mange oppgaver. Og det tenker jeg også at fint å se hva slags type oppgaver som kan komme på eksamen blant annet. Også hva de vil møte på videregående, ikke minst så vi tenkte å prøve å holde oss oppdatert på hva som kommer de neste årene i bøkene der.
45	I	Ja for det er jo her er det jo, vil det jo bli en viktig sånn greie i forhold til hvilke forventninger videregående har etter hvert, og sånn sett håper jeg ikke de har så store forventninger sånn foreløpig. Men etter hvert, så kan du jo sikkert. Eller da. Det er jo viktig med et samarbeid sånn som vi har jo hatt samarbeid med videregående her i Sandefjord, og det er er jo på en måte vært bra, tenker jeg i

		forhold til å kunne, kunne vite litt om, og møte hverandre litt da på hva som forventes, hva tenker de er viktig at elevene kommer med..
46	L2	ja, hvis det er aller minste, hvis jeg måtte ønske det, og hvertfall det med tanke på programmeringsspråk, skulle vi lære java, skal vi gjøre scratch eller python. Hvilke programmer er det de bruker mest?
47	I	Ettersom jeg har skjønt litt eller bare fra sånn som jeg har forstått det, så. Er det vel bruker de vel mye python på videregående.
48	L2	Ja det er det bøkene også legger opp til.
49	I	Ja. Det kan kanskje ha noe med at det er et gratis program og
50	L2	og ganske enkelt språk å lære seg hadde som er alternativet. Tror det er mye hjelp der.
51	I	Ellers så er det noe du tenker at du sitter inni med som tenker. Jeg kunne vært nyttig å å nevne i forhold til det med programmering i matematikkfaget er noe du føler ikke har fått frem?
52	L2	Ikke som jeg kommer på nei
53	I	Nei, da får jeg bare si takk for hjelpa så.

Intervju 3:		
nr		Sitat
1	I	Da tror jeg det skal være klart, skal vi se? Ok så da begynner jeg med å spørre deg om hvor mange år du har undervist i matematikk.
2	L3	I matematikk har jeg undervist oi. I en trodde vel. Jeg var ferdig i 2011. Det blir 12 år. 11 kanskje ja ja 11 år. Ja da er vi 12. år nå ja.
3	I	Hvilke trinn at du jobber på nå?
4	L3	Akkurat nå har jeg på 8.trinn ja.
5	I	Og hvordan er det du jobber med programmering i matematikkfaget? Hvordan har du startet?
6	L3	Nei, jeg starter opp altså, men den forrige runde rund vi hadde, så starter vi enkelt med blokkprogrammering med scratch. Som en del av algebra, forståelsen for algebra at man kan sette inn ulike verdier og så, men fortsatt bruke den samme strukturelle oppbygningen da. Og så har vi jobbet både. Både med med scratch, mest scratch for de var jo veldig nybegynnere disse elevene. Og har du har prøvd å å, I starten modellerer, at, at jeg gjør sammen med de veldig sånn skritt for skritt. Og så har vi bygd opp att etter hvert, så har de fått en oppgave med noen tydelige kriterier på, og jeg ønsker at alle skal være med. Hvordan man skal bygge opp. Og så da har jeg liksom hatt tilpasset opplæring med at jeg har hatt det samtidig. Noen har jo da jobbet for seg selv, ikke sant? Mens andre da fulgt min undervisning i timen da jeg har gjort det litt sånn sammen. Og når de var ferdig med niende klasse, så fikk de da bare hjemmeskole, så da fikk de en relativt åpen oppgave da knyttet til naturfag, men altså i naturfag faget der han skulle forklare et. Ja, men det handler med økonomer, verdensrommet, forskning i verdensrommet, altså for eksempel reise til mars eller det med sol og og natt og dag og måneder. Ja, det er litt sånne ulike kunne velge litt ulik da. Da fikk de både lage sin problemstilling og lage en animasjon fra fra visualisere den her. Da de kommer i 10. Klasse, så på vårparten 10. Klasse fikk vi beskjed fra videregående at de ønsker veldig gjerne ønsket tekstprogrammering i python. Så da. Lærte vi dem helt basic Python, veldig basic som kommando. Og der var det jo flere som da falt litt av, og vi hadde også delvis mye mindre tid da, men det det er benyttet vi oss av medlem, medlemmer, elever som, som er gode og som har gode tips programmering som da blir liksom som medlærere. Som fikk i oppgave å være hjelpelærer, mens vi mens jeg hadde undervisning på tavla, løp de mellom å hjelpe til hverandre.
7	I	Ja. Så bra jeg. Jeg har ikke drevet med programmering, men jeg. Har gjort et opplegg da i fjor, og da brukte jeg en metode som kaller use-modify-create som går, og det så jeg bare lurte på om du kjenner deg litt igjen i det, for det hørtes nesten sånn ut i forhold til når dere drev med blokkprogrammering da. Det handler jo om at man har et ferdig opplegg på en måte som de tester ut, elevene. Og så at de endrer på noen, noen blokker da for å endre litt på programmet, og så at de kan lage noe fra fra scratch da.
8	L3	Det er det. Det er akkurat sånn. Det tenker jeg. Det er akkurat sånn jeg har gjort. Jeg har liksom. Starter med at vi skal gjennomføre noe helt spesielt sammen, og så med noen variabler som de kan velge eller variere på over. Og da har jeg brukt ofte at det utgangspunktet i ferdig opplegg for kodeklubben som ligger digitalt. Ja, så der er de veldig fine. Kommer nesten ferdig og kan skrive ut bruksanvisning til.
9	I	Supert er det hvilke temaer i matematikken tenker du at elevene kan ha nytte av programmering?

10	L3	Ligningen, for eksempel litt sånn algebra at de litt basic forståelse for den her abstrakte matematikken at det at man bytter ut tall med bokstaver eller symboler med da. Spesifikk informasjon det jeg tenker jeg for å få den grunnleggende forståelse av matematikk, så jeg så er det et godt hjelpemiddel da. Men så brukte noe i sånn når man skal lære figur tall ikke så vellykket, men vi prøvde oss.
11	I	Ja så ja er det andre andre temaer? Du tenker at det kan være nyttig? Har du prøvd å nå i geometri, for eksempel ikke programmering eller ikke?
12	L3	Har ikke prøvd, men jeg ser jo nytten av det. Kanskje jeg ser at kunst og håndverk? Hvordan brukte for å lage mønster der. Og da henger du på en måte litt sammen med matematikken det her nå. Skjønner, vinkler og retningsbeskrivelser og i menneskefigurer.
13	I	Hvordan er programmering med på å gjøre matematikken mer spennende for elevene? Det er jo liksom på en måte en av de begrunnelsene fra udir og for å putte programmeringen inn i matematikkfaget var jo at det skulle gjøre det mer engasjerende.
14	L3	Ja altså. Det tror jeg er fordi at det er jo veldig inn, og det er jo veldig fremtidsriktad å å få en grunnleggende forståelse før programmering altså. Mer og mer av vår verden blir digital. Og for å få litt mer forståelse med hva som ligger bak da, så så tror jeg at det er med på at man må koble det sammen matematikk da for at ofte så min erfaring er jo at du kan bruke et ord som likeverd da i samfunnsfag, og da skjønner du hva det betyr eller Norsk. Men når du sier at han vil likeverdig i matematikken. Så er det sånn at man snakker, da skjønner de ikke konseptet av ordet da? Altså nå skjønner ikke at det betyr at han skal være like mye. Så jeg håper at det skal, hvis man da for implementert programmeringen her er at man skal få litt mer sånn forståelse føre at matematikken er et viktig fag da i særlig fremtiden, eller?
15	I	Ja sånn at ja at du at de ser på en måte mer enn nytten av matten liksom?
16	L3	Ja, og den blir mer brukervennlig at det ikke handler om å løse oppgaver bak. Veldig vanlig spørsmål fra elever er jo: hva skal jeg med det her til hva bruker vi det til i framtida, så ja.
17	I	Men tenker du at det også da med på å få de til å se sammenhengen mellom, at de klarer å se sammenhengen mellom programmering og matematikk? Eller tenker du at de liksom tenker at det nå gjør vi programmering at ikke det har noe med matte å gjøre?
18	L3	Nei, jeg håper jo at han skal komme dit at han skulle skjønne at når jeg logiske tenkning er matematikk, liksom at det er basert på matematikk og algebra, og at det henger sammen da, at det liksom er bygd opp på samme system og struktur.
19	I	Ja det er jo. Ja, det er jo det som er målet. Kanskje at hvis du får det vi jobber mot?
20	L3	Ja at det skal bli litt forståelse.
21	I	Ja, ja, hvilke fordeler ser på da ved å bruke programmering i mattefaget?
22	L3	Det er at du kan motivere en helt spesifikk elevgruppe som ellers kan komme med ganske, men med ganske lav motivasjon til matematikk da. Nå stigmatiserer jeg lite, men jeg ser en ganske stor sånn guttegruppe som, synes matematikk, matematikk i python, men tenker at programmering det er kult. Det har de lyst til, ikke sant?
23	I	Så bra
24	L3	kan man kan man få fram det her, knyttet det samme, kanskje med spillelementer, for da for de er ofte veldig opptatt av gaming og få han til å skjønne at her er det også matematikk. Så tenker jeg at da, da tror jeg, at vi kan

		øke deres forståelse over litt gleden med matematikken også at jeg får litt forståelse for faget da, håper jeg da..
25	I	Ja, for det er jo gjerne sånn at de sterkere de får man med uansett
26	L3	Ja, ikke sant? De er jo med på det meste, og alle disse plikttoppfillende. Det gjør jo også det de skal uten å sette spørsmålsteget ved det.
27	I	Nei, så hvis at man kan få engasjert i de kanskje ellers svake elevene da, så er jo det supert,
28	L3	og da tror jeg det er viktig det her med at ikke bruke, ikke går for fort fram, ikke gjøre for mye hver gang. Du må liksom oppleve mestring nesten hver time. Ikke sant at han får det til at det er små delmål. Hele tiden da som ellers i matematikken blir sånn.
29	I	Ja så hvilke utfordringer er det du ser da i forhold til programmering i matematikkfaget?
30	L3	Det er at det, er det. Det er utfordrende å ha en stor elevgruppe. Da må jeg på forskjellige nivå og jobbe på forskjellige. Noen tar det veldig fort og som du sier, ikke sant? De som skjønner det. Fort kan føle at det blir litt kjedelig da, eller at det går litt. Og må liksom ha nok utfordringer for alle, og kommer jo de som driver med det her, sikkert en del på fritida og sånn så må man også kunne utfordre de da. Til å se mening?
31	I	Ja. Men tenker du at det at utbytte holdt jeg på å si da, i forhold til hva de lærer, tenker jeg liksom matematikk da, at, at utbyttet er like stort som i forhold til den tiden det tar da eller føler at jeg føler det stjeler noe tid, eller? Tenker du at du utbytte er såpass bra at det er verdt det?
32	L3	Ja, jeg tenker at det er fremfor alt en ting som er viktig. Det er at vi må gi dem nok forkunnskaper da for at du opplever at du mestrer det. Og så må man ha litt is i magen og gi det tid, men det tar mye tid. Absolutt. Men jeg tror, jeg tror ikke, at det er en tidstyv på den måten, at jeg tror at det liksom gir en dypere forståelse, særlig for det er at med det abstrakte språket da, men den algebraen. At her er det noe som betyr noe annet, ikke sant det? At rammen er fast. Vi må bare fortelle hva som skal være inne, at vi kan vise dem til dette variabel. Man kan legge inn i ulike variabler at rammen er her, men når du legger inn ulike variabler, så får du ulike resultater da, og det er matematikk liksom.
33	I	Så bra. Ja, jeg da har jeg på en måte vært gjennom de spørsmål jeg tenkte, er det noe mer? Du tenker at det kunne vært interessant for meg å vite at det er noe annet du tenker som er spennende i forhold til det å bruke programmering i matematikkfaget?
34	L3	Jeg tenker jo at det her veldig spennende man kommer til den her av dybdelæring og dermed tverrfaglighet, at at det er vanskelig å jobbe med programmering konsekvent med på kun faget matematikk og må nesten bygge andre opp sammen med fag for å få noe meningsfullt innhold da altså man kan lære seg. Programmering rent matematisk å lære seg å bygge opp strukturen i, i en kode sekvens, men hvis det skal være noen mening, så må det nesten fylle på med nå faglig fra et annet område. Tenker jeg.
35	I	Ja så bra. nei men da tror jeg bare sier tusen takk ja..

Intervju 4:		
Nr.		Sitat
1	I	Da er mitt første spørsmål: hvor mange år du har undervist i matematikk?
2	L4	Da, tenker du med karakteransvar?
3	I	ja eller egentlig kan det være generelt.
4	L4	Ja for de 2 første årene, så hadde jeg aldri karakteransvaret i fagene, jeg var med som spelærer og student. Men 4 år da med enten faglærer eller kontaktlærer med matematikk. ja. Ja 6 dersom vi skal ha hele.
5	I	Ja OK. og da, hvilket trinn er det du har matte på i år?
6		Niende trinn.
7	I	Og når du har jobbet med programmering i matematikkfaget, så lurer jeg litt på hvordan du starter opp.
8	L4	Ja og der er jo liksom akilleshælen med en gang jeg da fordi. Jeg har ikke jobbet så mye med programmering.
9	I	Nei. Men du har gjort litt?
10	L4	Det er jo først nå i fagfornyelsen at det er på plan sånn ordentlig. Men. I fjor når jeg var på 8. trinn, så bestemte vi oss for å puche det til niende var jo det som var på planen mener jeg å huske. Og nå som jeg er på niende, så har vi ikke kommet dit enda, for de har satt det til våren. Så har jo ikke jobbet noe organisert ordentlig systematisk med det. Nei, men av det jeg har gjort da, for jeg har jo vært med på hatt noen kodedager og litt sånn og litt spredte opplegg her, og da har det jo vært scratch da hovedsakelig ja, og da er jo de da noen ferdig ferdige opplegg eller ferdig oppskrifter om elevene følger og setter sammen. Og det det stort sett det hele egentlig, av det har jobba med programmering. Det er ikke mye, men sånn er det bare.
11	I	Men jeg tenker på nå, det du sier at de har fått et ferdig opplegg, men har de fått noe anledning da til å videreutvikle det opplegget for å da på en måte lage sitt eget?
12	L4	Ikke i stor grad, og de som var ordentlig motiverte til det og som likte det de satte i gang også pusle med dette var jo kan huske da så. Flesteparten sa seg liksom ferdig når de var ferdig. Flesteparten rakk ikke å bli ferdig engang.
13	I	Fordi at. Nei, fordi det er jo et sånt opplegg eller en sånn undervisningsmetode da som heter use-modify and create som går på liksom at du først får en ferdig et ferdig program og så som du skal på en måte teste ut da, og så skal du endre på noen av kodene sånn at den. Ja gjør det litt annerledes, for eksempel endre på noen variabler og sånt, og så at man til slutt kanskje lager sitt eget program. Ja til slutt da.
14	L4	Den beskrivelsen kjenner meg veldig igjen i, men det jo. For det meste til uttrykk på det vi lærere har jobbet med. Vi har litt rundt og opp til programmeringen. Ja, har ikke vært kommet dit hvor jeg i hvert fall har testet det ut i noen særlig grad på eleven enda.
15	I	Neida da. Men hvilke? Ja, der har du sikkert ikke noe sånn suksessfaktor i hvilken hva du tenker kan gjøres for at det skal bli en suksess da. Så.
16	L4	Nei, men jeg har jo noen tanker rundt det da. Det er jo at inngangsterskelen må være såpass lav at alle sammen faktisk skjønner det for den programmeringa der det er vanskelig når du ikke har vært der før, og da kan man ikke gå ut og si at, det er bare gjøre sånn og sånn det her sier seg selv liksom må ta ekstremt grundig.

17	I	Ja da og det er det programmering også på en måte er da at man må være grundig i oppskriften også for at det skal fungere. Sånn som, sånn som forventer for at programmet skal fungere
18	L4	Eller så, så ble det jo også veldig det opplegget, det som vi holder på med i vi i fjor. Det som omgjøringsmaskin. Det tenker jeg veldig fin introaktivitet for elever som datamaskinen er egentlig skokk dum om at det her må vi være pinlig nøyaktig på de kommandoene og det vi skriver inn for at det faktisk skal bli, som vi tenker at det er kan være en fin introktivitet på. Det må være å tegne tegne etter oppskrift. Da, hvis husker ikke helt da
19	I	Men hvilke temaer tenker du at elevene kan ha nytte av programmering da innenfor matten.
20	L4	Umiddelbart når jeg så spørsmålet, så skrev jeg ned 3, 3 temaer ja og algebra først, og da spesielt det forståelsen med variabler. Også geometri, geometri er også med mye variabler med lengde og bredde og radius og alt sånn der, alt det kan jo plottes inn sin egen kode for å regne ut og areal på aktuelle ting. Og tallregning, og det er jo noe jeg har puslet med selv, egentlig rett og slett til tørrtrening på å lage oppgaver. Så hvis man skal trene på multiplikasjon da tosifra, ensifra. Så er det veldig slitsomt, å måtte finne nye oppgaver hele tiden. Det er jo mye enklere å bare ha en kode for det som bare kan trykke ny oppgave, ny oppgave så generere den nye oppgaver i det parameterne du selv har satt. Så det tenker jeg, nå som jeg har lært å gjøre det. Det er noe jeg kan ta med på åttende trinn, hvor det temaet er aktuelt da de lærer å sette opp en sånn god for den er ikke noe spesielt vanskelig. Den er jo ganske grunnleggende da. At og å bruke det som rett og slett og oppgaver generator da. Så det er sikkert mulig å koble programmering på andre temaer også. Men det var de 3 som traff meg med en gang? Jeg tenker at snevre vi inn på noe spesielt ennå så det ikke alt sammen.
21	I	Ja men hvordan tenker du at programmering kan gjøre matte faget mer spennende for elevene da?
22	L4	..variasjon er jo en ting da. Og det er jo, her handler det ikke bare om de som er interessert i og liker pc, men det å også gjøre forskjellige ting, det er motiverende, men bruddet på undervisning også. Ja og så kombinert med faktum at det her er jo jo mer virkelighetsnært da enn å regne oppgaver på campus eller å regne i boka. De er jo digitale vesener så det holder alle nettsider de er i på, og hvert spille i bruk telefonen hver app, alt det er det en kode bak og det å forstå grunnleggende åssen dette med koder fungerer. Det kan jo kanskje trigger nysgjerrigheten da og å utforske altså. Og og den biten. Den har jo sett av det lille jeg har gjort oppsummering med eleven at. De som blir tent på det her. De begynner jo å pusle med egne programmer og skrive koder, så som selv jeg ikke forstår.
23	I	Ja, men tenker du at det er noe forskjell på på sterke og svake elever her?
24	L4	Ja, det tror jeg. Tror det er mer i. Det er jo matematisk tankegang som ligger i bunnen, da der, logisk resonnering. Da tror fortsatt de sterke matematikk elevene vil ha fordel der altså. Men, tror du at uten at jeg har noe empiri eller erfaringer å bygge det på? Det er bare antagelser,
25	I	Men tenker du at tenker du at det er kanskje noen svake elever som man kan som kan tenne på det her som synes det er spennende da som kanskje ikke tenner så på andre ting innenfor matte faget
26	L4	Ja det tror jeg, og kanskje spesielt elever som ikke nødvendigvis er, i mangel på et bedre ord, som ikke nødvendigvis er kognitivt svake, men underytere i stedet

		kanskje finner litt mer mening og litt mer motivasjon i å kode å jobbe i et program snarere enn å jobbe i boka. Men de som er digitalt interesserte. De tror jeg vi tenner på det uansett om de har anlegg for matematikk eller ikke..
27	I	Og da har vi jo liksom litt inn i neste spørsmål, egentlig, hvordan tror du elevene ser sammenhengen mellom matematikk og programmering tror du de ser liksom at, tror du du tenker at, tror du de ser matte da i programmeringen eller tenker du at de. At de ser på det som en, om noe eget da som å se en film, for eksempel i et annet fag.
28	L4	Skjønner Når jeg så spørsmålene, så skrev jeg en vanskelig som lærer å svare på det spørsmålet hadde vel vært mer hensiktsmessig og stille elever, men. Den utfordringen å få programmeringen til å henge sammen med hva matematikken, at det her skal de se at det hadde faktisk er en er en sammenheng. Den faller jo på læreren som så mange andre praktiske aktiviteter også må knytte ja. Praxis til teori og da det som ligger banda, det er jo lærerens programmeringskompetanse, som jeg mener er den absolutt aller største utfordringen i i skolen nå, og jeg er på ingen måte god i programmering i det hele tatt. Jeg kan bitte lite grann, men jeg likevel så er det et hav foran det mange andre lærere kan. Så er det det. Eller noen få som kan mer enn meg føler jeg.
29	I	Ja da for det som sagt så er jo. Eller som du sier, så er jo kanskje det noe av utfordringen da med at man, at opplæringen har vært litt skranten for å si det mildt.
30	L4	Det kommer på syvende spørsmål, kan ikke mer om med det der
31	I	Vi kan ta det da. Jeg bare tenkte du kunne, men kan godt ta og se på noen fordeler først. Hvilke fordeler du ser med programmering i mattefaget?
32	L4	Det er motivasjon hos elever, det faktum at, nå lærer de noe som faktisk som faktisk blir brukt. for vi kan gjøre det argumentet gang på gang at det er ligninger får du bruk for i virkeligheten, men det er vel snarere den matematiske forståelsen vi er ute etter. Det er ikke ofte vi trenger å løse en likning. Men koding og programmering blir det bare mer og mer av, mer blir digitalt, og. Ja, de fleste unge kommer til å være borti det før eller siden, uansett tror jeg.
33	I	Ja. Det er nok riktig det. Men ja utfordringene da bortsett fra manglende kompetanse hos lærere. Ja
34	L4	det jeg sliter med å se hvor vi egentlig vil med programmeringen. Hva er hva er målet bak det? Bortsett fra at elevene skal kode, hvilket program skal de kode i nøyaktig? Hva skal de kode altså? Jeg føler, at det her er bare. Nå skal elevene kunne programmering, ikke hva de skal kunne programmere det er eneste stor tåke skal vi se se og det.
35	I	Det var faktisk noe jeg. Altså, det er jo ikke noe føringer fra udir da om hvilke, hvilke programmer eller noen ting i forhold til, for eksempel når det når det gjelder graftegner da, så er det og excel, så er det litt mere sånn. Det er litt mer håndfast det som har stått i læreplanene føler jeg i hvert fall, enn når det gjelder programmering da der er det bare liksom på en måte kastet inn at de skal programmere, og så er det veldig lite, håndfast som du sier,
36	L4	Det virker nesten litt hastverk utenpå det, lite gjennomtenkt. Nå skal elevene lære programmering. Vi må bare ha det med på planen. Vi vet ikke helt hva de skal. Vi skal lære programmering, så så må lærere bare gjøre denne jobben. Ja er derfor kan da. Om det er scratch de skal jobbe i om det er java, html koder. Ja jeg sist var på et kurs i forbindelse med, jobben min her nå hvor vi lærte litt i

		python og på H-nett programmer jeg aldri har hørt om før. Men det er jo sikkert ørten andre programmer også som vi brukte forskjellige steder, og det er jo en utfordring for det. Det er jo ikke likt det er ikke sånn at det du lærer på, i det programmet kan overføres til det andre. Det er jo forskjellige kodespråk, forskjellig grensesnitt som må tilpasses. så her. Det må rett og slett være mer spesiell, altså mer tilspisses. Mer standardisert da
37	I	Ja helt enig
38	L4	. et knippe programmer som faktisk dette må du lære, og det skal elevene kunne som det faktisk er mulig for lærere og lære det uten å måtte ta 30 studiepoeng det
39	I	nei fordi at ja, det er jo noe av det som kanskje mangler da, som er litt sånn utfordrende at det er både mangel på opplæring og spesifikasjoner da på hva, hva de egentlig vil at de skal sitte igjen med når de går ut av ungdomsskolen?
40	L4	Ja jeg har jo lært å kode litt i det der programmet som snakker om nå tony i python kode. Jeg føler meg fortsatt ikke et snev og klokere på. Hva skal vi faktisk lære eleven. Hva skal de sitte igjen med? Det er veldig vanskelig, og det hjelper lite å ha kurs og lære om det ene eller andre programmet. Hvis vi fortsatt ikke kan bestilling på elevene, skal sitte igjen med.
41	I	Det er jeg helt enig. Men men det får jo håpe at vi får noe mere. Blir litt klokere da når at det er når du kommer eksamen etter hvert. I det her. Sånn sånn at man vet litt mer hva man skal sitte igjen med. Det er litt viktig, tenker jeg. Yes er det andre tanker du har rundt programmering.
42	L4	Ikke som jeg kommer på nå umiddelbart. Jeg følte jeg fikk ventilert ganske hvis det der.
43	I	Ja, ja, men det er jo litt sånn jeg tenker det er og er en ganske sånn felles eller og gjengs oppfatning. Det da at det er, at du ikke vet hvor vi skal hen. Og at det er frustrerende. For det er ne viktig greie i forhold til det. Det å vite hva vi hva vi skal. Så så du er ikke alene om å sitte med den frustrasjonen.
44	L4	Nei det, det vet jeg forsåvidt det. Så jeg tror det er en av årsakene til. At jeg og vi teamet jeg har jobbet på også har valgt å utsette dette her for det, kompetansen er ikke der og mål med hva vi skal gjøre med det er ikke der,
45	I	ikke sant? Og hadde man hatt noen konkrete mål, så hadde det jo vært lettere å skaffe seg kompetansen også. Så det er jo litt. Det henger jo veldig sammen alt det der. nei da men jeg er fornøyd jeg med det det du er kommet med, så jeg kan jo bruke det, det er nyttig informasjon så det så da tror jeg bare tar og så stopper dikteringen