



UiT Noregs arktiske universitet

Fakultet for naturvitskap og teknologi

Programmering si rolle i matematikkfaget

Frå eit elevperspektiv

Olaug Grut Andersen

Masteroppgåve i matematikk ved lektorutdanninga trinn 8-13, MAT-3907, Juni 2022

Forord

Denne masteroppgåva markere slutten på lektorstudiet og mine fem første år i Tromsø. Desse fem åra på studiet har vore fantastiske takka vere folka. Takk lektor realfag 2017 kullet for ei fantastisk studietid, med gode samtalar, festar, turar og ikkje minst vennskap for livet.

Eg må få takke IT-klassen min for at de ville vera informantar i studia mi, og vist interesse i korleis det går med oppgåva undervegs i året. Oppgåva hadde ikkje komme i mål utan hjelp frå rettleiaren min Jan N. Roksvold, takk for gode idear og ikkje minst hjelp til å struktur i tankar og oppgåva.

Ikkje minst takk til mine foreldre, mori og faren. Takk for all hjelp opp igjennom skuleåra, men ikkje mist for alle timane med øving på å lese og rettskriving. Var sikkert ikkje alltid like lett eller gøy med ei dotter med «kronisk maur i rumpa» og dysleksi. Men utan utfordringane med dysleksi hadde eg sikkert ikkje fått min kjærleik for matematikk. Takk mori og faren for korrekturlesing og gode tips på omstrukturering av håplause setningar, når eg sjølv har fått hjerneteppe på denne oppgåva.

Tromsø, mai 2022
Olaug Grut Andersen

Samandrag

Hausten 2020 vart den nye kunnskapsløftet(LK20) innført i skulen. I den nye læreplanen i matematikk, kom algoritmisk tenking og programmering inn. Det er blitt skriva og forska ein del på programmering i skulen, men lite om det frå eitt elevperspektiv. Difor ville eg høyre kva elevane synst. Problemstillinga i denne undersøkinga blei til *Elevar sine tankar og kjensler kring programmering i matematikkfaget*.

For å undersøkje denne problemstillinga har eg samla inn datamaterialet gjennom resultat frå prøvane informantane hadde i haust, spørjeskjema etter kvar prøvane og intervju med tre elevar. Intervjua gav mest datamateriale og har fått størst plass oppgåva. Det vart gjennomført ei induktiv koding av dei transkriberte intervjua og svara frå spørjeundersøkinga. Klassen som er brukt for datainnsamling er den klassen eg underviser i 1T. Eg har difor også observert elevane mens dei jobbar, hjelpt dei med matematikkoppgåver og har såleis blitt kjent med dei. Det gjere at eg har tolka utsegna deira på ein måte som eg ikkje kunne gjort utan denne relasjonen.

Denne undersøkinga implisere at programmering krev noko anna av elevane enn den vanlege instrumentale matematikken som dei er vand med. Det gjere truleg at det kan vera vanskelegare å forstå programmering for elevane, og difor kan gjera at dei har dårlegare motivasjon for programmeringa samanlikna med den tradisjonelle matematikken. Elevar har eitt inntrykk av programmering er koding, noko som kan gjere det truleg tørt og utfordrande for elevane. For å imøtekomme denne utfordringa er det nok viktig å la elevane få nok tid til å bli komfortabel med grunnprinsipp i programmering og gje dei meningsfulle oppgåver der fokuset er på meir enn sjølve kodinga. Korleis ein skal løyse oppgåva og problemet burde vera like viktig som kva som står i koden.

Innholdsliste

1	Innleiing	1
1.1	Bakgrunn for val av oppgåva.....	2
1.2	Hensikt og føremål	2
1.3	Problemstilling og forskningsspørsmål	3
1.3.1	Forskingsspørsmål.....	3
1.4	Omgrep tydnad	3
1.5	Oppgåvas struktur.....	4
2	Teori	5
2.1	Programmering	6
2.1.1	Matematikk i programmering.....	7
2.2	Programmering i skulen.....	9
2.2.1	Kvifor programmering i den norske skulen?.....	10
2.2.2	Læreplanen 2020 (LK20)	10
2.3	Algoritmiske tenking	11
2.4	Læring.....	13
2.5	Motivasjon i programmering	13
2.5.1	Motivasjon i matematikk.....	14
2.6	Tidigare forskning om programmering i skuleverket.....	16
2.6.1	Forsking.....	16
2.6.2	Relevante masteroppgåver	18
3	Metode.....	21
3.1	Val av metode.....	21
3.2	Intervju og spørjeskjema som metode	22
3.2.1	Spørjeskjema etter prøvar.....	22
3.2.2	Intervju	23

3.3	Val av informantar.....	24
3.4	Min rolle som lærar og forskar	25
3.5	Etiske omsyn.....	25
3.6	Rehabiliteten og validitet i oppgåva	26
3.7	Utføring av analyse.....	27
4	Analyse.....	31
4.1	<i>Elevar sine kjensler om programmering</i>	32
4.1.1	Grad av meistring på oppgåver bestemmer kva kjensler elevane får om programmering	32
4.1.2	Tradisjonell matematikk er betre enn programmering	34
4.2	<i>Elevar sine tankar kring programmering si rolle i matematikk</i>	34
4.2.1	Seint start med programmering	34
4.2.2	For lite tid til programmering	35
4.2.3	Programmering er koding.....	36
4.2.4	Programmering er matematikk fordi det brukast matematiske funksjonar	37
4.2.5	Alt kan løysast ved programmering	37
4.2.6	Sjølvmotsetningar i elevar sine meiningar om programmering	38
5	Diskusjon.....	39
5.1	Grad av meistring på oppgåver bestemmer kva kjensler elevane får om programmering	39
5.2	Tradisjonell matematikk er betre enn programmering	42
5.3	Seint start med programmering	44
5.4	For lite tid til programmering	45
5.5	Programmering er koding.....	48
5.6	Programmering er matematikk fordi det brukast matematiske funksjonar	50
5.7	Alt kan løysast med programmering	52
5.8	Sjølvmotsetningar i elevar sine meiningar om programmering	54
6	Konklusjon	57

6.1	Vegen vidare.....	59
7	Referanselise	61
8	Vedlegg	65
8.1	Samtykkeskjema.....	65
8.2	Intervjuguide.....	68
8.3	Utforsningsoppgåver timen før intervju	69
8.4	Meldeskjema frå NSD	70

1 Innleiing

I det kapitalistiske samfunnet er det ikkje er nokre lover som regulera dei økonomiske ulikskapen som blir til. Med eit uregulert økonomisk samfunn, er det blitt store økonomiske forskjellar. Det har vore fleire diskusjonar på kor mykje skal me motverke noko som berre er «blitt slik». I åra framover kjem samfunnet meir og meir til å bli avhengig av digitale løysingar og samfunnet vil bli mykje meir digitalt. Det digitale samfunnet kan skape store ulikheitar på same måte som det økonomiske samfunnet. Difor er det viktig at me som samfunn ikkje sit på side linja og let det «berre bli slik» (Haraldsrud m. fl, 2020).

Hausten 2020 vart den nye kunnskapsløftet(LK20) innført i skulen. For at skulen skal vera relevant og framtidsretta må kompetansen elevane lærer i dag, kunne bli brukt på nye og ukjente område (Utdanningsdirektoratet t , 2021a). Fokuset i LK20 er på å lære for å lære. Då det skal danne grunnlag for elevane å lære gjennom heile livet. Dette fordi skulen kan ikkje lære elevane alt det dei treng i framtida.

Område som er vektlagt for å stimulere læring for å lære i den nye læreplanen (LK 20) er; djupnelæring, aktiv i eiga læring, kritisk tenking, algoritimisk tenking og programmering. Det skal også fokuserast på tverrfaglege tema og eit verdiløft. Verdier som skal fremja ei betre framtid for menneskje, samfunnet og kloden (Utdanningsdirektoratet , 2021a). Det digitale samfunnet som me er på veg inn i kan skape store ulikheitar, noko som ikkje skapar eit betre samfunn for alle. Ein viktig føresetnad for å kunne ta del i digitaliseringa er ei grunnleggjande forståing for programmering . Då kan alle vera med i diskusjonane rundt korleis det digitale samfunnet skal utvikle seg. (Haraldsrud m. fl, 2020)

Arbeidet med utviklinga av den nye læreplanen(LK20) starta hausten 2017. Då blei arbeidet starta med å utvikle kjerneelementa til kvart enkelt fag (Utdanningsdirektoratet, 2021b). Kjerneelementa er det viktigaste elevane skal lære innafor kvart emne. I matematikk er det fem kjerneelementa:

- utforsking og problemløysing
- modellering og anvendingar
- resonnering og argumentasjon
- representasjon og kommunikasjon
- abstraksjon og generalisering

Det er rundt desse kjerneelementa at kompetansemål er utvikla frå (Kunnskapsdepartementet, 2021b).

Algoritimisk tenking og programmering er ein sentral kompetanse for framtida. Det er nytt tema i læreplanen og skal inn i fleire fag (Utdanningsdirektoratet, 2021a). Hovud satsinga på algoritmiske tenkinga og spesielt programmeringa er lagt til realfaga og då spesielt matematikkfaget. Algoritimisk tenking er lagt inn under kjerneelementet utforsking og problemløysing.

1.1 Bakgrunn for val av oppgåva

Programmering i skulen og matematikkfaget er «nytt igjen». Teknologien er annleis og meir tilgjengeleg for alle, enn den var sist gong det skulle inn i skulen (M87). Det er skrive og forska lite på programmering i skulen. Då programmering er «relativt nytt» i skulesamanheng, og ulike land innføre programmeringa på forskjellige måtar i skulen. I Noreg er det innført som ein del av matematikkfaget.

Den meste av forskinga som eg har funne både internasjonalt og nasjonalt handlar om korleis elevane lærar programmering. Ofte er fokuset på kva som er viktig å huske på når ein skal ha programmeringsundervising. Det er forska på korleis elevane arbeidar med programmering i timar, eller der lærarar er brukt som informantar. Det er skrive nokre masteroppgåver som undersøker kva elevane tenkjer om programmering, programmering i matematikkfaget eller som valfag. Desse masteroppgåvene har ungdomsskuleelevar som informantar.

1.2 Hensikt og føremål

Hensikta med denne oppgåva er å bidra til meir forskning der eleven sjølv får vera i sentrum. Med det meiner eg kva elevane sjølv oppfattar og meiner om programmering. Sidan det er skrive og forska lite på kva eldre elevar meiner om programmering, vil denne masteroppgåva forhåpentlegvis kunne medverka til å tetta nokre kunnskapshol på dette område. Når me som lærarar har kunnskap om kva oppfatningar elevane har om eit tema, kan vi lettare forstå og hjelpe dei. Då ein utviklar seg mykje gjennom åra på skulen er det også viktig å vite kva dei eldre elevane meiner, og ikkje berre kva ungdomsskuleelevane meiner.

Føremålet med undersøkinga er altså finne ut kva vidaregåande elevar tenkjer om at programmering er blitt ein del av matematikkfaget.

1.3 Problemstilling og forskingsspørsmål

For å kunne undersøkje dette var det danna ei problemstilling :

Elevar sine tankar og kjensler kring programmering i matematikkfaget

For å belyse denne problemstillinga er det utforma to forskingsspørsmål.

1.3.1 Forskingsspørsmål

1. Kva er elevar sine kjensler knytt til programmering i matematikkfaget
2. Kva er elevar sine tankar kring programmering si rolle i matematikkfaget.

1.4 Omgrep tydnad

At me forstår tydnaden av eit omgrep er avgjerande for vår oppfatning av kva omgrepet inneber. Difor vil eg definere nokre sentrale omgrep og korleis dei blir brukt i denne oppgåva.

Programmering: Å programmere vil seie at ein bryt ned eitt problem til enkle handlingar/kommandoar. Desse kommandoane er instruks på korleis ein kan løyse problemet. Ved digital programmering blir instruksane gitt til ei datamaskin som utfører dei. Det blir gjort ved at kommandoane blir koda til eitt språk datamaskina kan forstå. Ved analog programmering er det inga datamaskin som utfører kommandoane, men det kan vere eitt menneskje som får ei detaljert oppskrift på korleis oppgåva skal utførast, eller løysast, utan at personen skal måtte finne ut dette sjølv.

Koding: Å skrive ein kode, på eit språk som datamaskina forstår. Dette er ein del av det å programmere. I denne oppgåva er blir det skilt mellom programmering og koding, då programmering er meir enn berre koding og gjev eitt anna læringsutbytte.

Kjensler: Kjensler er ein spontan reaksjon som kjem på ei sanseoppleving. Det kan vera meir samansette sinnsreaksjonar, då ofte kalla emosjonar. Dersom det er meir einsretta og situasjonsbestemte reaksjon kallar ein det ofte affekt (Jansen, 2019). Me kan i svært liten grad styre kjenslene våre, det me kan styre er reaksjonen på desse kjenslene.

Tankar: Ein tenkjer ein tanke. Når ein tenkjer, arbeider ein seg igjennom ei mental sanseoppleving, kjensler eller noko ein hugsar, eller noko heilt anna. Resultatet av å tenkje kan ofte vera at ein løyser eit problem, eller kanskje trekkjer ein ei slutning ut frå tidlegare

erfaringar og opplevingar (Tenking, u.å.). I denne oppgåva er tanke brukt om kva meiningane baserer seg på.

Meiningar: Ei meining kan vera ein eller fleire tankar, eller ein fornuftig oppfatning (Teigen & Alnes, 2022). Ei meining er ofte prega av kjenslene ein fekk når tanken vart til. I denne oppgåva er meining brukt når elevane er sikre i sin sak og har enda opp med ein «konklusjon» om noko.

Matematikkfaget : I denne oppgåva vil matematikkfaget bli brukt om den matematikken elevane lærer i IT, men også den matematikken dei har lærd i grunnskulen.

1.5 Oppgåvas struktur

I kapitel 2 vil det teoretiske rammeverket som gjev grunnlaget for drøftinga av empiri bli presentert. Kapitlet vil også ta føre seg ~~det~~ noko av grunnjavnaden for at programmering er blitt ein del av matematikkfaget.

I kapitel 3 vil val av metode og tilnærminga til den bli gjort greie for. Studia er ei kvalitativ studie gjennomført med elevar som eg underviser. Difor vil min rolle og etiske omsyn bli vektlagt i kapitlet.

Korleis rådata i denne oppgåva vart gått gjennom vil bli presentert i innleiinga til kapitel 4, analyse-kapitlet. Vidare vil analyse-kapitlet gjere greie for kva funna i denne studia er. Det vil bli brukt utsegn og samandrag av intervjuar som støttar opp om funna.

Moglege årsaka til funna, vil bli diskutert i kapitel 5, funna vil også bli sett i lys av tidlegare forskning. I kapitel 5 vil det og bli vurdert kva ein som lærar bør ta med seg frå det spesifikke funnet, og kva relevans det har for forskingsspørsmålet. Før det i kapitel 6 vil bli avslutta med kva konklusjonar dette gjev og det impliserer. Kapitel 6 vil bli avslutta med å presentere ein veg vidare.

2 Teori

Då undersøkinga er ei kvalitativ undersøking som er analysert induktivt er det ikkje noko rammeverk for analysen. I dette teorikapitelet vil eg difor presentere tidlegare forskning og teori som vil bli brukt for å diskutera funna frå analysen.

Sidan oppgåva undersøker kva elevane meiner om programmering i matematikkfaget handlar første del av teori-kapitelet om programmering. Der vil eg også sjå på kva matematikk som er i programmering. Då programmering har sitt utspring frå matematikk, er det viktig å vite om elevane må kunne mykje matematikk for å greie å programmere.

Korleis er det eigentleg tenkt at programmering skal inn i den norske skulen og spesielt matematikkfaget, er relevant då eg har undersøkt kva elevane tenkjer om programmering si rolle i matematikkfaget. Ser og meiner elevane det same som det som har bestemt eller har dei ei heilt anna oppfatninga av situasjonen.

Programmering er knytt til matematikkfaget igjennom algoritmisk tenking, som er viktig i kjerneelementet utforskning og problemløysing. I det tredje del -kapitelet vil ta føre seg kva algoritmisk tenking er og korleis det er knytt til programmering i matematikkfaget. Det vil også bli belyst korleis utdanningsdirektoratet meiner at den algoritmiske tenkjar jobbar.

Kjensler og motivasjon heng saman. Kva kjensler elevane har om programmering i matematikkfaget, kan påverke motivasjonen deira. Difor er det naturleg å sjå på kva som påverke motivasjonen til elevane. Då det ikkje er skrive så mykje om motivasjon i programmering, har eg valt å ta med noko av det som er skrive om motivasjon i matematikk. Dette er fordi programmering er blitt ein del av matematikkfaget og difor burde det vera relevant det som er skrive om motivasjon i matematikkfaget.

Sist i dette kapitelet vil det bli belyst kva forskinga om programmering i matematikkfaget seier. Mykje av forskinga som er gjort i dei Noreg og Sverige er det ikkje frå eitt elevperspektiv, difor er det tatt med nokre relevante masteroppgåver til slutt.

2.1 Programmering

Programmering er eit ord dei fleste har høyrte om, og truleg også brukt sjølv. Eg trur likevel mange ikkje veit om alle moglegheitene som ligg i omgrepet programmering. Dette delkapitelet har som hensikt i gje ei lita innføring i av kva programmering er.

Programmering er å utvikle nye program og eller vedlikehald av eit programmet som utfører noko. Å utvikle nye program innebere å designe, skrive, teste og feilsøke. Den som skriv programmet gjev instruksjonar til korleis programmet skal oppføre seg og kva det skal utføre, blir kalla ein programmerar («Programmering», 2019).

Sanne et al. (2016) gav ut ein rapport på mandat frå Utdanningsforbundet der dei såg på kva teknologi kompetanse elevar treng i framtida, dei definerer programmering litt annleis.

«Programmering vil sei å bryte eit gitt problem ned i eit sett av kommandoar og så få ei datamaskin til å utføre desse kommandoane» (Sanne et al.,2016). Denne definisjonen er litt mangelfull, då den legg til grunne at ein må ha ei datamaskin til å utføre kommandoane, den utelet då analog programmering. Eg har valt å ta med denne definisjonen då gruppa utarbeida rapporten på mandat frå utdanningsdirektoratet og gav noko av grunnlaget for dei endringane me har fått med digitale teknologien i skulen.

Programmering handlar som oftast om å gje ei datamaskin ei rekkje instruksjonar som maskina kan forstå. Det blir gjort gjennom programmeringsspråk. Lågnivåspråk er det som prosessoren i datamaskina kan lese. Det kan vere maskinkode som er dei binære talla, 0 og 1 eller assemblerkode. Høgnivåspråk som kan mykje enklare lesast av oss menneskje, dette kan vere språk som Python og C. Høgnivåspråk blir ofte gjennom ein kompilator oversett til lågnivåspråk slik at datamaskina lese koden (Programmeringsspråk u.å.).

Blokkprogrammeringsspråk er også høgnivåspråk, det er for eksempel Stretch, i blokkprogrammering skriv ein ikkje ein kode i tekst, det er ferdig «blokke» med kodar som set saman. Alt dette er digital programmering, og er ofte det som blir meint når ein brukar omgrepet programmering. Men me har og analog programmering, som er enkelt sagt programmering utan datamaskin. Ved analog programmering løyser ein eitt problem og lagar ei framgangsmåten utan digitale hjelpemiddel. Dette kan førekjem på mange forskjellige måtar, og liknar litt på algoritmisk tenking.

Litt som det meste ein gjere, er det metodar som er meir effektiv enn andre og difor finst det anbefalte arbeidsmåtar. Dette gjeld når ein skal programmere. Før ein kan begynne å kode eit

program må problemet som skal løyse, analyserast. I analysen lagar ein kravspesifikasjon, som er alle oppgåvene som må gjerast for å løyse problemet. Neste steg er å bryte problemet ned i mindre deler. Deretter løyser ein kvart av desse del-problema, og til slutt skriv koden til kvar enkelt av del-problema. Når dette er gjort, sett ein alle del-programma i lag slik at programmet kan køyre (Rossen, 2019).

2.1.1 Matematikk i programmering

Dei elektriske datamaskinene som er forgjengarane til dei datamaskinene me har i dag, kom etter krigen. Og med det den digital programmering slik me kjenner det i dag. Dei første programmerarane var flinke matematikkar. I denne undersøkinga er programmering sin rolle i matematikkfaget sentralt, altså korleis ein kan bruke programmering for å bli betre i matematikk. Det er likevel relevant å sjå på kva matematikk ein må kunne for at ein skal kunne programmere. Er det nødvendig å vera ein flink matematikkar for å kunne programmere, og kva fordelar har det å vera god i matematikk når ein skal programmere. Det har vore vanskeleg å finne gode skriftlege kjelder som tek for seg kva matematikk ein må kunne for å programmere. Det er mange artiklar som omhandle kva matematikk ein burde kunne for å bli ein programmerer. Det er ein forskjell på kva ein burde kunne og kva ein må kunne av matematikk for å kunne greier å programmere. Det er mykje matematikk som gjere jobben med å programmere lettare, fordi for å løyse problemet bør ein kunne matematikken, men matematikken er ikkje nødvendig for å kunne programmere. Eg har ikkje funne nokre som er skriva om kva matematikk ein må kunne for å programmere, berre kva som er lurt å kunne. Difor er det eit samandrag av samtalar med min rettleiar om kva matematikk ein må kunne for å programmere.

Det er relativ enkel matematikk i programmering, litt slik som sjakk, enkle reglar, men det kan vera ekstremt komplisert å spela, likevel. Ved programmering må ein kunne talsystem, ti-talsystemet vårt, men sidan datamaskina brukar to-tall-systemet er det lurt å kunne det og. Vidare må ein kunne bruke dei matematiske rekne operatorane; addisjon, subtraksjon, multiplikasjon, divisjon. Ved divisjon er det også nødvendig å kunne dividere å få ein rest. I multiplikasjon ligg også potens- og rot-rekning. Det er også veldig viktig at ein forstår variablar og korleis ein reknar med det, sjølv om variablar i matematikk og programmering er litt ulik. Det siste ein må kunne av matematikk når ein skal programmere er matematisk logikk eller symbolsk logikk. Det er nok truleg her det blir vanskeleg.

For å programmere må ein kunne relativ enkel matematikk, men likevel er det mykje matematikk som blir nytta i dataprogram. Programmering har gjort det mogleg for oss å rekne med meir avansert matematikk, og i symbiosen av matematikk og programmering har me fått ekstremt mange nye digitale hjelpemiddel og moglegheita. Det er nødvendig å vera god til å resonnera for å vera god til å programmere, og ein kan vera god til resonnering utan å vera god i matematikk. Difor kan det vere ein fordel å vera god i matematikk, men det er ikkje nødvendig.

Elevar tileignar seg tidleg i grunnskulen den matematikken dei treng for å programmere. Difor kunne ein tenkja seg at det ikkje vera eit problem for elevar å programmere. Likevel ser ein at studentar som skal lære seg programmering, slit med matematikken. Matematikk dei burde ha lærd tidlegare skulegonge hindrar dei i frå å bli gode programmerarar (Pacheo et al.,2008). Ei undersøking frå USA gjort på universitet-studentar syner at studentane ikkje meistra programmering på grunn av dårleg problemløysing evner. Det er spesielt matematisk metodar og gode metodar for å løysa eit problem som gjere det utfordrande for studentane (Pacheo et al.,2008).

Pacheo et al.(2008) finn i undersøkinga si at studentar som har gode karakterar i matematikk frå grunnskulen og kalkulus presterer betre i programmeringskursa på universitet. Om det er den matematiske logikken eller «vanleg» problemløysing som gjere det utfordrande for studentane å meistre programmeringskursa konkludere Pacheo et. al.(2008) ikkje med.

Programmering kan brukast til å løyse mykje matematikk. Dette fordi datamaskinene er mykje meir effektive til å løyse utrekningar enn menneskje er. Datamaskinen vil aldri gjere dei same reknefeila som oss menneskjer og utfører rekneoperasjonane ekstremt mykje raskare. Ein del matematiske utrekningar er ikkje hensikt messig å løyse med programmering, då det å lage programmet tek lenger tid enn å løyse det sjølv.

Kan alt av matematikk løysast ved programmering? Det spørst om ein meiner at ein kan bruke programmering til ved bevisløysing. «*Computer-assisted proof*» er matematiske bevis som delvis eller heilt blir løyst ved hjelp av programmering. Ved hjelp av *computer assisted proof* er det blitt bevist fleire matematiske resultat som me menneskjer ikkje har klart å løyse sjølv. *Computer assisted proof* kan delelast opp i to steg. Steg 1 er det same som å løyse matematiske bevis på vanleg måte, ein argumenter for at eit program har vise eigenskapar. Steg 2 er ikkje eit bevis på den måten den tradisjonelle matematikken definerer bevis. Fordi i

steg 2 sjekkar ein berre om programmet køyrde, og dersom beviset stemmer frå ein beskjeden «Ja, det fungera». Det kan skje mange feil som gjere at ein får riktig, utan at det er det, og me menneskjer greier ofte ikkje å dobbelt sjekke det som er gjort fordi bevisa er så lange (Zwick, 2002).

Fleire fagpersonar meiner at matematiske bevis som blir utlede (i mangel på eit nynorsk ord med same tydnad) ved hjelp av programmering, ikkje er verkelege matematiske bevis då det er så mange logiske slutningar som ikkje kan bli verifisert av menneskjer. Bevisa kan ikkje bli bevist av menneskjer då bevisa er for lange. Det betyr at matematikkar må vike frå logiske slutningar til å stole på datamaskina. Datamaskiner som kan bli påverka av feil i kodar og hardware (Gómez-Serrano, 2019).

2.2 Programmering i skulen

Sjølv om det for mange nok er nytt at programmering skal inn i den norske grunnskulen, er det ikkje første gong at programmering blir skriva inn i læreplanen. I mønsterplanen for grunnskulen frå 1987 (M87) skulle programmering inn gjennom datalære. Datalære var eit tverrfagleg tema, som skulle brukast på tvers av fag. Programmering skulle knytast til matematikken. Der skulle det ta utgangspunkt i algoritmeomgrepet, men vart også nært knyta til emnet problemløysing. Sjølv om det var detaljerte beskrivinga om korleis programmering skulle nyttast i mønsterplanen vart det ikkje ein suksess. Det kan vere mange grunnar til, men tilgangen på læremiddel og den digitale teknologien var heilt annleis då enn no (Bueie, 2019).

Gjøvik og Torkildsen(2019) skreiv ein artikkel om algoritmisk tenking og programmering medan det vart jobba med LK2020, der dei diskuterte korleis programmering på ein god måte kunne innarbeidast i matematikk . Dersom programmering skal bli implementert i matematikkfaget må undervisinga vera meir enn å implementere allereie eksisterande algoritmar. Fokuset må vera problemløysing og utvikle metodar og algoritmar for å løyse mange forskjellige problem. Det er ikkje viktig at det blir brukt digitale verktøy for dette. Fokuset må ikkje bli på programmeringsspråk og syntaks, då det blir utvikla og forandra heile tida. Målet må vera læring av matematikk med programmering, ikkje berre å lære koding (Gjøvik og Torkildsen, 2019).

Programmering kan utfordra elevane i kritisk tenking og resonnering gjennom ulike problemstillingar. Elevane får gjennom programmering moglegheita til å bruke sin kreativitet

og fantasi til å skapa noko, ved å bruke ein ide og omsetta den til ei handling. Undervising i programmering handlar om å gi elevane gode hjelpemiddel for å løyse problem eller oppgåver. Ein informatikkar som skal løyse eitt problem brukar problemløysingsmetode som blir kalla algoritmisk tenking. Ein slik måte å arbeide på kan også brukast i mange andre fag og samanhengar (Selvik et al.,2016).

2.2.1 Kvifor programmering i den norske skulen?

I 2016 vart ei arbeidsgruppe oppnemnt av Utdanningsdirektoratet der dei utarbeida ein rapport om kva kompetanse elevar i framtida treng om teknologi og teknologirelaterte emnar. Barn og ungdom som går på skule i dag er framtida. Det er dei som skal utvikle og sette sitt preg på morgondagen og samfunn i framtida. Elevane skal ikkje berre fungere, men og vera med å utvikle samfunnet. Det vil krevja meir av elevane enn at dei berre kan bruke no tidas digitale teknologi (Sanne et al.,2016).

For å møte denne utfordringa må elevane forstå dei grunnleggjande prinsippa for den digitale teknologien. Det vil gje elevane moglegheita til å bruke sin kreativitet og kunnskap til å utvikle teknologien for framtida. Den teknologi som er til det best for seg sjølv, men også for samfunnet. For å ha kunnskapen og moglegheita til å gjere dette må elevane kunne og forstå dei grunnleggjande digitale-teknologiske prinsippa. I dette ligg også tanken om at elevane må kunne og programmere (Sanne et al.,2016). Sanne et al.(2016) anbefalte at det skulle opprettast eit nytt fag som skulle ha dette ansvaret.

2.2.2 Læreplanen 2020 (LK20)

Om ein ser igjennom læreplanane til fellesfaga i grunnskulen og vidaregåande skule, er det berre i matematikk og naturfag programmering er nemnt med ord i kompetansemål, grunnleggjande ferdigheitar eller i kjerneelement (Kunnskapsdepartementet, 2019). Digitale verktøy er nemnt i alle læreplanane til alle fag, men ikkje kva digitale verktøy, og føremålet med bruk av dei digitale verktøya er nemnt. Frå dette kan det tolkast som at det ikkje er snakk om programmering, men det å kunne bruke forskjellige program som hjelpemiddel.

Læreplanen i matematikk skal leggja til rette for at elevane skal bli gode problemløysarar og greie å sjå samanhengar mellom matematikkfaget sine kunnskapsområde og andre fag. Evna til å sjå slike samanhengar legg til rett for forståing og djupnelæring i faget. Algoritmisk tenking er meir vektlagt og gjort meir synleg fordi det er ein problemløysingstrategi. Eit godt verktøy for å få ein betre matematisk forståing kan vera å bruke programmering for å løyse og

utforsk problem. Det er fleire av kompetansemål der det er eigna å bruke programmering (Utdanningsdirektoratet, 2020).

2.3 Algoritmiske tenking

I læreplanen(LK20) i matematikk er algoritmisk tenking peikt på som ein viktig del i prosessen med å løyse problem. Elevane må kunne vurdere om dei skal løyse problema analogt eller med digitale hjelpemiddel. Digitale hjelpemiddel og ferdigheit er blant anna å bruke programmering til å utforske og løyse matematiske problem (Kunnskapsdepartementet, 2019). Det kan tolkast som at programmering skal bli brukt som eit hjelpemiddel i algoritmisk tenking i matematikken.

Algoritmisk tenking har inga klar definisjon på norsk. Korleis ein oversett omgrepet til engelsk er også ulikt, og med det ulik definisjon på engelsk også. Difor vil eg sjå på dei to orda omgrepet består. Sidan det er Kunnskapsdepartementet som er forfattar av læreplanen vil deira definisjon av algoritmisk tenking bli vektlagt til slutt.

I matematikk er algoritme : *«Algoritme er i matematikk og databehandling en fullstendig og nøyaktig beskrivelse av fremgangsmåten for løsning av en beregningsoppgave eller annen oppgave.»* (Hovde & Grønmo, 2020). Eit eksempel på kva ein algoritme kan vere er ei oppskrift med ein framgangsmåte, som ei oppskrift på ei kake.

«Tenking er å bearbeide mentalt sanseopplevelser, ting man husker, forestillinger, følelser, verbale symboler, med mer. Bearbeidingen kan resultere i at man trekker en slutning, forstår, gjenkjenner, løser et problem»(Tenkning, u.å.). Tenking er noko alle menneskjer gjere. Det at me kan arbeide med sanseintrykka våre gjere oss i stand til å løyse matematiske oppgåver. Sidan me kan endre korleis me arbeidar med sanseintrykka våre, kan me også ved å øve og arbeide med matematikk bli betre til å løyse matematiske problem.

Algoritmisk tenking er då noko med ein spesifikk framgangsmåte og å tekne. Det er ei open og upresis formulering av omgrepet. Omgrepet blir definert litt ulikt, men det er mykje likt.

«Tankeprosessen knyttet til å formulere et problem og dets løsning(er) på en slik måte at det kan løses av en informasjonsprosesser (menneske eller maskin)» (Algoritmisk tenking, u.å.).

Å kunne løyse eitt (generelt) problem ved å programmering blir kalla algoritmisk tenking (Sanne et al.,2016). «Algoritmisk tenking er ein problemløysingsmetode»

(Utdanningsdirektoratet, 2019). Dette er berre nokre av måtane å definere omgrepet på.

Gjøvik og Torkildsen (2019) trek fram at det er mange ulike definisjonar av omgrepet. Dei

meiner ei opprydding i omgrepet algoritmisk tenking og andre omgrep som liknar er lurt for lærarar som skal bruke den algoritmiske tenkjaren og programmering i skulen. I denne vil algoritmisk tenking bli brukt som ei blanding av beskrivingane ovanfor.

Algoritmisk tenking innebere å bryte ned eit problemet i mindre del-problem. Det er fordi del-problema er enklare å løyse. Deretter lager eleven ein algoritme, også kjent som framgangsmåte for å finne ei løysing på problemet. Når ein har ei løysing, kan denne ofte generaliserast og brukast til å løyse andre problem (Utdanningsdirektoratet, 2019).

Ein elev som meistra den algoritmiske tenkjaren er; kreativ, har ei vilje til å skape og finne nye og alternative løysingar. Eleven er strukturert og kan analysere sitt eige arbeid. Den algoritmiske tenkaren tør å gjere feil, og greier å sjå på det som ein del av prosessen. Det kan vere krevjande for elevane å halda ut når dei gjere feil og ikkje får det til. Det er viktig at elevane får samarbeid, fordi få gode idear oppstår i vakuum, ved samarbeid oppnår ein eit tankemylder (Utdanningsdirektoratet, 2019).

I denne oppgåva vil omgrepet algoritmisk tenking beskrive ein problemløysingsmåte. Der ein får eitt problem som ein løysa på ein strukturert måte, ved å bryte det ned til mindre problem. Ein løyser del problema og får så leis løysinga på heile problemet. Løysinga er ein nøyaktig framgangsmåte på problemet, men ein kan ofte greie å generalisere denne løysinga til å gjelda for fleire liknande problem.

«Et naturlig miljø for å implementere algoritmisk tenking er via programmering» (Gjøvik, Torkildsen, 2019). Det er viktig å huske på at koding og programmering ikkje er det same. Dette sjølv om omgrepa av og til blir brukt som om dei har same omgrepsinnhald (Gjøvik, Torkildsen, 2019). Koding er ein del av programmering, der ein reformulere løysinga til problemet til eit programmeringsspråk som maskina kan lese. Dersom ein berre kodar får ein ikkje trena den algoritmiske tenkjaren.

Ut frå det eg har sett kva lærebøkene og ekstra pakkane i programmering byr på, er det viktig at læraren veit at det er ein stor forskjell på læringsutbytte av koding og programmering. Spesielt dersom ein skal programmere for å trene den algoritmiske tenkjaren. Det er viktig at det ikkje blir slik at elevane berre kodar, og at læraren då tenkjer no oppfyller eg eit kompetansemål. Mi erfaring frå praksis og vikarjobbing tilseie at dersom det er mogleg for elevane skriv mange av dei berre av koden og tek minste motstand veg for å «løyse» oppgåva i boka. Dersom ein som lærar ikkje er oppmerksom på dette, endar elevane opp med å berre

kode og ikkje få det tiltenkte læringsutbytte av aktiviteten og difor får ikkje trena den algoritmiske tenkjaren.

2.4 Læring

To av funna i denne undersøkinga er knytt til læring, då når ein bør begynne å lære noko og kor mykje tid ein treng. Sidan eg ikkje har funne noko forskning som omhandlar når det er ideelt for elevane å starte å lære programmering eller kor mykje dei tid treng til det. Vel eg og skrive kort om kva læring er og kva som fører til læring.

Læring kjem av tidlegare erfaringar, som fører til langvarige endringar i åtferd og opplevingar. For å lære ferdigheit meiner ein at det er repetisjon eller å øve regelmessig på ferdigheita som avgjere om ein får det til. Menneskjer lærar mykje gjennom verbalt eller skrive materiale. Dette sjølv om forskning syner at me menneskjer kan lære utan å vere bevisst det (Svartdal, 2020).

Det er skrive og forska mykje på korleis me menneskjer lærer, og opp igjennom åra. Testprøvar og læring over tid er synt svær effektivt for å betre læring og hukommelsen. Å kombinere desse, på den mest effektive metoden blir kalla *Successive relearning* - læring over tid. Dette skjer igjennom å ha «test»-prøvar, der elevane må finne fram kunnskapen frå langtidshukomminga, altså repetisjon og studere/lese det ein skal igjennom fordelt over fleire dagar. Altså litt over ein lenger periode, enn alt på ein gong (Rawson et.al., 2013).

Kapasiteten til korttidsminne kan trenast, men di eldre mi blir di dårlegare blir evna vår til å utnytte den. Korttidsminne behandlar det me blir oppmerksame på, for at arbeidshukommelsen skal kunne arbeide igjennom informasjonen og overføre den til langtidsmiynet vår, må oppmerksamheita vår vera på (Busch, 1992).

2.5 Motivasjon i programmering

Det er skrive nokre forskingsartiklar om motivasjon i programmering i Noreg. Det er også skrive nokre masteroppgåver der noko av diskusjonen har nemnt programmering og motivasjon. Nedanfor er eit samandrag av dei to artiklane som eg meiner er relevante for problemstillinga i denne oppgåva.

Eliassen et al. publiserte i 2022 artikkelen « *Motivasjon for bregningsorientert biologi og sammenhengen med matematikk R2 frå videregående opplæring* » der dei har undersøkt motivasjonen til studentar før og etter innføring av eit R2-krav. Eliassen et al.(2022) finn at den største motivasjonen for faget berekningsorientert biologi var nytte verdien, ytre motivasjon, mens generelt for studieprogrammet var det indre motivasjon. Studentane med R2 matematikk haust 2018 (før innføring om R2-kravet) hadde opplevd større meistring i faget enn dei andre medstudentane, dei rapporterte og at dei hadde hatt mindre utfordringar knyta til programmeringa. Dette er naturleg då R2 har komponentar i faget som har vist seg å vera ein fordel når studentar måtar programmering for første gang. Foreløpige tall frå 2020 syner at karakteren i R2 er lågare og dei spekulere i om det kan spele inn på motivasjonen, då dei vil ha eit lågare ferdigheitsnivå.

Berg (2021) har undersøkt korleis elevar og lærarar opplever å bruke analog programmering i matematikkundervisninga. I studia er det intervjuet ein lærar og den sine elevar (8.klasse) har svart på eit spørjeskjema. Læraren meiner noko av det viktigaste utbytte av den analoge programmeringundervisningsøkta gav var motivasjon for programmering. Det trur hen kan vera for di analog programmering ikkje stiller like store krav til dei tekniske ferdigheitene til elevane og difor vil fleire elevar meistra oppgåvene. Læraren trekk og fram at fleire av elevane truleg fekk ein betre forståing av oppgåva ved den analoge programmeringa då dei sjølve måtte finne ut kva som skjedde.

2.5.1 Motivasjon i matematikk

Sidan programmering er blitt ein del av matematikkfaget i den norske skulen, vil det bli brukt litteratur og teoriar frå matematikk som utdjupar og forklar kva motivasjon i matematikk er og utspelar seg. Eg vil prøve å gje eksemplar som gjere det relevant for programmering. Wæge og Nosrati gav i 2018 ut ei bok « *Motivasjon i matematikk* », nedanfor er eitt samandrag av viktige tema i lys av problemstillinga i denne oppgåva. Dei trek fram at det ikkje er skrive eller forska så mykje på motivasjon i matematikkfaget.

Motivasjon er ikkje noko som ein kan observere direkte hos elevane. Det me kan observere og undersøkje er elevane sine handlingar, kva dei tenkjer og føler. Dette er faktorar som motivasjon kan gje utslag i. I matematikkfaget er motivasjonen til elevane heilt avgjerande for korleis dei gjere dei i faget. Motivasjonen spelar inn på alle vala elevane tek i matematikkfaget, frå kva oppgåve dei vel å gjere og engasjerer seg i, til om dei følgjer og deltek i matematikdiskusjonar i klasserommet (Wæge & Nosrati, 2018).

Motivasjon blir ofte delt inn i ytre og indre motivasjon. Indre motivasjon er ei indre glede som ein har og får av å gjere ei handling. Eleven har indre motivasjon dersom det er kjekt, spennande og interessant å jobbe med matematikk eller programmering. Dersom eleven arbeide med matematikk og programmering for å oppnå eitt resultat, enten satt av seg sjølv eller av omgjevnaden er det ytre motivasjon. Ytre og indre motivasjon opptrer ofte ikkje kvar for seg, det opptrer ofte i hop for eleven i klasserommet når den skal lære og jobbe med matematikk og programmering (Wæge & Nosrati, 2018).

Indre motivasjon er viktig for at elevane skal ha eit uthald i å løyse matematikkoppgåver. Indre motivasjon vil også ofte gje elevane meir sjølvtrillit og kreativitet i matematikkfaget. Dette gjer at elevane som har indre motivasjon i matematikkfaget evnar å nytte fleire og forskjellige problemløysingsstrategiar når dei jobbar med matematikkoppgåver eller programmeringsoppgåver enn dei elevane som har ytre motivasjon (Wæge & Nosrati, 2018).

Dei høgare opp i utdanningssystemet ein kjem, dei mindre verkelegheitsnært blir undervisinga i matematikk. Oppgåvene og problema elevane får i matematikken er dei første åra i grunnskulen knyta til ting og opplevingar dei lett kan kjenne igjen. Når elevane blir eldre og matematikk blir vanskligare dei mindre blir oppgåvene knyta til verkelegheita. Dette kan vere noko av grunnen til at den indre motivasjonen til elevane i matematikk forsvinn når dei blir eldre. For at den indre motivasjonen ikkje skal forsvinne i matematikk treng elevane å sjå relevansen av oppgåvene og opplegget dei har (Wæge & Nosrati, 2018).

Motivasjon blir påverka av fleire ulike faktorar. Faktorar knytt direkte til eleven som; tidlegare erfaringar, verdiar og behovet eleven har. Menneskje sine tre grunnleggjande behov, spelar også inn på motivasjonen. Det er kompetanse, autonomi og tilhøyrslø (Wæge og Nosrati, 2018).

I matematikkfaget handlar kompetanse om, at elevane kjenner på ei utvikling av forståing og evnene sine. Det er viktig at elevane får kjenne på meistring i matematikkfaget, når dei jobbar med oppgåver og undervisningsopplegg. Meistringsfølelse i matematikkfaget kan komme av meir enn å kunne løyse ei oppgåve «riktig», sjølv om dette også er viktig. Meistringsfølelse i matematikkfaget kan oppstå ved å stille spørsmål, argumentera, resonnera, finne og forklare forskjellige måtar å løyse ei oppgåve på eller å forstå matematiske omgrep (Wæge og Nosrati, 2018).

Undersøkjande matematikk eller utforskande matematikk skil seg frå tradisjonell matematikkundervising. Enkelt sagt skal elevane skal undersøkje eit problem som er presentert av læraren. Etter at elevane har jobba med problemet reflektere dei saman med medelevane og lærarane. Under refleksjonen tar dei opp forskjellige løysingsstrategiar. (Wæge & Nosrati, 2015). Undersøkjande matematikkøker kan minne om korleis ein jobbar når ein programmerar. Difor er det relevant å sjå på korleis motivasjonen til elevane blir påverka av øker med utforskande matematikkundervising. Motivasjonen til elevane kan variera frå aktivitet til aktivitet, det avhenger av elevane sin følelse av kompetanse og autonomi ved aktiviteten. Oppgåver som gjev elevane meir motivasjon er opne, har ei praktisk vinkling eller problemløysingsoppgåver. Der det gjev elevane moglegheit til å utvikle sine egne strategiar, gjere noko praktisk og er ein moglegheit for samarbeid. Elevane som kjenner at matematikkoppgåvene er passe utfordrande vil oppleve meir glede i matematikkfaget (Wæge, 2007).

2.6 Tidigare forskning om programmering i skuleverket

Eg finn ikkje at det er skriva så noko som er fagfelleurdert om kva elevar tenkjer om programmering si rolle i matematikkfaget. Noko av det elevane meiner i denne undersøkinga kan vera kjent gjennom anna forskning. Difor har eg valt å ta med forskning som er skriva om moglegheita og utfordringar ved programmering i matematikk, sjølv om det ikkje er frå eitt elevperspektiv. Eg har valt å ha fokus på undersøkingar frå Noreg og Sverige, då det er forskjell på korleis programmering er innført i skulesystema i forskjellige vestlege land.

2.6.1 Forsking

For at programmering skal bli ein vellykka del av matematikkfaget set det store krav til læraren. Hen må greie å få elevane til å sjå programmering som eitt verktøy for problemløysing innan matematikk. Det er ein reel moglegheit for at programmeringa blir for teknisk og elevane ikkje skjønner samanhengen til matematikk (Kaufmann & Stenseth, 2021).

Fleire studiar gjennomført dei siste åra syner at det er potensial ved å ha programmering som ein del av matematikkfaget. Forsström & Kaufmann(2018) har analysert fleire av desse undersøkingane. Resultatet er at det er umogleg å seie kva læringsutbytte kvar enkelt elev får av programmering i matematikk. Det er fordi læringsutbytte avhenger av kva val eleven tek undervegs i problemløysinga. Fleire av dei analyserte undersøkingane syner auka motivasjon og at det er eit positivt bidrag for å få betre resultatet i matematikken, men det er ingen automatikk i dette. Elevar slit ofte med å sjå læringsutbytte og potensialet programmering har

i matematikk. Dette sjølv om programmering har mogelegheita til å gjere matematikk meir praktisk og realistisk (Forsström & Kaufmann, 2018).

Eitt av argumenta for å innføre programmering var at det skulle vera eit verktøy for å bli betre problemløysarar. Det er ikkje ein automatikk i at elevane blir betre problemløysarar berre fordi dei programmerer. Elevane sine svake evner til å reflektere og argumenter kan ofte hindre elevane i å løyse problemet. Dette set store krav til at læraren greier å rettleie elevane på ein riktig måte, men også gje elevane problem som dei kan evne å løyse. Dette fordi når ein skal programmere er ein avhengig av å kunne generalisere løysinga på problemet, slik at ein kan programmere ein generalisert kode (Kaufmann & Sthenset, 2021).

I Sverige er programmering også innført i matematikkfaget, men der er innført i emnet algebra. Så i motsetnad til i Noreg der programmering er innført i matematikk litt meir generelt er det i Sverige bestemt kva emne det skal brukast i. Kilhamn et al. gav i 2021 ut ein rapport om korleis prosessen med å implementere programmering i matematikk har gått. Dei har undersøkt på kva læraren har lært bort til elevane, ikkje kva elevane kan.

70% av undervisinga som Klihamn et al.(2021) undersøkte hadde berre fokus på programmering eller så vart programmeringa berre blei brukt for å repetere matematikk elevane allereie kan. Med bakgrunn i at matematikk berre er rammene for programmeringundervisinga i kategori 2 og i kategori 1, spør Kilhamn et al.(2021) om det er riktig at programmering skal vera ein del av matematikkfaget. Kvifor skal det vere programmering i matematikk, når det fører til at matematikken forsvinn heilt eller nesten heilt frå matematikkundervisinga. (Klihamn et al.,2021)

Kilhamn et al.(2021) tolkar funna frå undersøkinga på to måtar. I den første tolking trekk dei parallellar til innføringa av statestikk i matematikkfaget. Kanskje programmering blir ei eiga «grein» av matematikk, og om ei stund er det utvikla nye metodar og spesielle tema som er inkludert i matematikken. Programmering i matematikk går frå å bruke matematikk til å utvikle programmering til at programmering er ein eiga «del» av matematikken.

Den andre tolkinga til Kilhamn et al.(2021) er å anerkjenne at programmering framleis er nytt både for elev og lærar. Lærarane og elevane må først få kjennskap til programmeringsen sin kva og korleis, før det kan brukast og utnyttast. Når både elev og lærarar kan meir og er trygge på programmering, har programmering moglegheit for å bli eitt verktøy for utforsking, problemløysing og matematiske utrekningar. Programmering blei raskt innført i matematikk

og algebra, det har ført til usikkerheit og stress for mange lærarar. Mange lærarar er «reine» matematikklærarar og har ikkje programmeringskunnskap. Det har heller ikkje vore nok tid til å gjev alle lærarar denne kunnskapen. Det kan har resultert i at mange lærarar slit med å ha meningsfulle oppgåver og aktivitetar som trekk gode band mellom matematikk og programmering.

2.6.2 Relevante masteroppgåver

Etter det vart klart korleis endringar den nye læreplanen (LK20) la opp til i matematikkfaget, og er blitt tatt i bruk, er det blitt skrive fleire masteroppgåver om programmering i matematikkfaget. Mange av desse er mindre relevant for denne oppgåva, men det er tre oppgåver som har interessante funn og tolkingar. Sidan dette er masteroppgåver er ikkje forskinga og resultatet fagfelle vurdert, men det er likevel relevant å ha med, då eg ikkje har funne så mykje fagfelleurdert forsking spesifikt med mitt tema.

Hermansen(2021) har undersøkt korleis førestillingar elevar kan ha om programmering i matematikk. Denne problemstillinga er relativ lik mi, men han har gjort undersøkinga på ungdomsskulen med elevar som har blitt introdusert til programmering gjennom super:bit på Nord Norskvitensenter. Fleire av elevane har berre erfaringane frå programmering kunne frå dette prosjektet, og difor berre førestillingar derifrå. Elevane er yngre enn informantane i denne oppgåva, dei har og fått introdusert programmering gjennom eit prosjekt som er litt meir praktisk enn mine informantar. Elevar knyt programmering i matematikk til bruk av datamaskin, blokkprogrammering, automasjon, matematikk og omgrep frå algoritmisk tenking. Hermansen(2021) finn at dei fleste av elevane har førestillingar som forbind programmering med matematikk, då det er varierende og kva. Ikkje alle elevane evnar å kople variablar-blokker (variablar i blokkprogrammering) mot matematikk. Sjølv om elevane tydleg er klar over at dei brukar matematikk når dei programmere, ikkje greier å snu det, å bruke programmering i ein tydleg matematisk kontekst. Det kan komme av at elevane har forskjellige førestillingar om kva som skal til for å lykkast med programmering i matematikk, men alle har likevel ei førestilling om at dei lykkast på sin måte. Hermansen(2021) peikar på at resultatet i oppgåva syner at elevane ikkje har jobba med eller reflektert meir om programmering i matematikk enn det dei gjorde i super:bit prosjektet, noko som truleg førte til at dei fleste av elevane laga ein førestillingar om det og ikkje meir generelt om temaet.

Nyutstumo(2021) har undersøkt korleis elevar som har brukt PRIMM-modellen til å lære seg programmering gjennom problemløysing i matematikken. I studia blir elevane guida

gjennom forskjellige programmeringskonsept steg for steg, ved hjelp av PRIMM-modellen, elevane meistarar også å utforske matematiske konsept undervegs også. Nyutstumo(2021) si studie tydar at programmering som blir brukt for å utforske problem er eit godt verktøy for å utvikla ein matematisk forståing. Då studia er så liten, vil han ikkje konkludere med om det egentleg er slik.

Verstad skreiv ei masteroppgåve om valfaget programmering kunne hjelpe, koplast saman med matematikkundervisninga i 2017. Dette var før den nye læreplanen (LK20) var ferdig utarbeida, så programmering var ikkje ein del av matematikkfaget. I valfaget programmering skulle elevane jobbe problemløysande gjennom ein algoritmisk tankegang. Verstad(2017) observerte elevane og intervjuar dei, spesielt intervjuar er intressante. Utsagn som « *Matematikk hjelper i programmeringen, matematikk i programmeringen hjelper ikke i matematikken*», «*I matematikken forholder man seg som oftest til en metode, mens man i programmering må forholde seg til flere*», «*...ville sikker matte og programmering slått seg sammen*» (Verstad, 2017), syner at elevane ser koplinga mellom programmering og matematikk. Fleire av elevane synst det er vanskeleg å sjå programmering skal hjelpe i matematikken, men ikkje omvendt. Verstad(2017) i peikar på at dei elevane som evner å sjå koplinga mellom matematikk og programmering, ser på matematikken som dei brukar er grunnleggjande matematikk. Nokre av informantane meiner at det er kjekkare arbeidsmåtar når ein arbeidar med matematikk i programmering enn den tradisjonelle matematikkundervisninga.

3 Metode

«Hensikt med forskning er å frambringe gyldig og troverdig kunnskap om virkeligheten.» (Jacobsen, 2016. s15). For å evne det må forskaren ha ein strategi for korleis det skal gjerast. «Denne strategien er metode» (Jacobsen, 2016. s15)

I dette kapitlet vil eg gjere greie for val av metodar i denne studia. Eg vil beskrive korleis eg har jobba med henta inn data og korleis vegen var frå rådata til funn vart til. Det har vore litt fram og tilbake undervegs i oppgåva med innhenting av data og korleis analysere det på best mogleg måte. Dette vil bli beskrive så godt som mogleg, samstundes som eg vil grunngje mine val for å gjera det på denne måten.

3.1 Val av metode

Korleis metode ein vel, er avhengig av kva kunnskap om verkelegheita ein vil syne fram. I denne undersøkinga er ynsket å finne ut kva elevane meiner om at programmering er blitt ein del av matematikkfaget. Den beste måten å finne ut kva elevar meiner om noko er å spørje dei. Dette kan ein gjere ved å enten spørje dei munnleg eller skriftleg.

Føremålet med denne undersøkinga er å undersøkje kva elevane tenkjer og føler om programmering si rolle i matematikkfaget. Det er vanskeleg å forklare kva ein meiner gjennom tall, og når verkelegheita er for kompleks å redusere til tall bør ein forske kvalitativ (Jacobsen, 2016). Ei casestudie er avgrensa av enten tid eller rom, der fokuset er på individ eller grupper eller liknande. Det som gjere ei undersøking til ein casestudie er at den «unike» konteksten spelar ei rolle (Postholm & Jacobsen, 2018., s.63). Elevane i denne undersøkinga er avgrensa til ei 1T-klasse, som lærer programmering for første gong på vidaregåande som ein del av matematikkfaget. Det er dette som gjere denne gruppa og individa unike. Dette vil også bli gått meir innpå i del-kapitlet *Val av informantar*. Kunnskapen frå denne studia vil derfor vera avgrensa til denne spesielle konteksten og vera «lokal kunnskap» (Postholm & Jacobsen, 2018).

Denne undersøkinga er ei kvalitative casestudie, der meste av datamaterialet er innhenta gjennom det fenomenologisk intervju. I det fenomenologisk intervju kan også det vere med observasjonar og dokument som datakjelde i tillegg til transkriberte intervju (Postholm & Jacobsen, 2018). For å undersøkje kva elevane meiner er det i denne casestudiet blitt nytta

både intervju og spørjeskjema. Det første eg gjorde var å gi elevane eit spørjeskjema etter matteprøvene. Då kunne elevane skrive ned deira meiningar og kjensler basert på erfaringane dei gjorde seg når dei løyste prøven. Problemet med å be informantar å skrive ned noko, er at det ofte ikkje er så utførleg som når dei snakkar (Postholm & Jacobsen, 2018). Dette var også eit problem for meg i gjennomføringa av undersøkinga, då svara til elevane på spørsmåla var ofte både korte og mangelfulle, og til dels ikkje svara på. For å få meir utfyllande svar som kunne gje grundigare og betre forståing valte eg å intervju nokre elevar. Det meste av datamaterialet i analysen bygger på desse intervju. I tillegg til det transkriberte intervjuet, observerte eg elevane igjennom skuleåret og brukte svara på spørjeskjema og resultat frå prøvene til elevane som bakteppe for diskusjonen. Helst ville eg brukt svara elevane leverte på programmeringsoppgåvene som vart gjeve på prøvene. Sidan dei fleste av elevane leverte blankt på programmeringsoppgåvene i matematikkprøvene, gav det for lite datamateriale til å kunne analysere å få eitt bra resultat. Difor gjekk eg vekk frå dette og valte ha det med som eitt bakteppe for å få eit betre bilde av elevane som vart intervju.

3.2 Intervju og spørjeskjema som metode

Ein vil oppnå ein balanse mellom nærheit og avstand som forskar. Nærheit for å forstå, men avstand for å kunne få eitt perspektiv. Ein avstand kan ein ofte oppnå ved spørjeskjema, då det er lettare og distansera seg, mens i intervju eller observasjonar må ein sette seg inn i andre situasjon (Postholm & Jacobsen, 2018).

Eg opplevde å få lite datamateriale og difor ein dårlegare forståing av kva elevane meinte av spørjeundersøkinga. Undersøkinga gav ein peikepinn, men var for dårleg til å finne nokre funn. Difor valte eg å intervju nokre elevar for å få ein djupare forståing og om spørjeundersøkinga og intervju peika i same retning. Kvale & Brinkmann(2015) sjekkar ein elev sin utsegn i eit intervju igjennom å lage spørjeskjema som mange fleire elevar svarde på. På ein så måte kunne dei teste generaliteten til svara til den intervju eleven (Kvale & Brinkmann, 2015).

3.2.1 Spørjeskjema etter prøvar

Tanken med å gje elevar spørjeskjema etter matematikkprøvene var at dei skulle ha frisk i minne kva dei hadde gjort og kvifor, og difor kunne gje betre svar. Mange av svara på spørjeskjema var korte, eitt ord eller ufullstendige setningar. Ein av grunnane til dette var at mange av elevane ikkje løyste programmeringsoppgåvene på matematikkprøven. Det var ulike grunnar til det; nokre fekk dårleg tid, men fleste har sagt at dei ikkje klarte å løyse

oppgåva fordi dei ikkje forstod oppgåvene. Det var flest som synst det å finne ei løysing på problemet var vanskelegast, og kom seg ikkje til skriving av koden.

Difor måtte eg finne ein anna måte og finne ut kva elevane tenkte. Valet falt på intervju, og eg vil bruke mest tid og plass på å grunngje dette.

3.2.2 Intervju

Intervjuaren sine praktiske ferdigheitar og personlege vurderingar formar intervjuet. Det er igjennom å intervju nokon ein lære seg intervjuferdigheiter. I eit intervju skapar eg som intervjuar og eleven som informant kunnskap saman. Dette skjer igjennom ein erkjenning-prosess som er sosial og intersubjektiv (Kvale & Brinkmann, 2015).

Det fenomenologiske intervjuet er det viktig å få svar på kva erfare eleven når hen er bevist og i korleis samanheng er fenomenet erfart (Postholm & Jacobsen, 2018). Det datamaterialet ein sit att med etter eit slik type intervju gjev første-ordensforklaring, altså eleven sin eigen beskriving av erfaringa (Postholm & Jacobsen, 2018).

Sidan eg ikkje har gjennomført intervju av informantar og elevar før, og sidan eg ville samanlikna og sjå svara til elevane i samanheng blei det utforma ein intervjuguide.

Intervjuguiden (sjå vedlegg 2) inneheld dei fleste spørsmåla som eg spurde elevane. Dersom eg var usikker på kva eleven meinte valte eg og stille oppfølging spørsmål. Ved å legge opp til eit semistrukturert intervju, hadde eg spørsmål som eg skulle stille, men også moglegheita til å stille dei oppfølgingsspørsmåla eg trengte for å forstå eleven best mogleg. Dette gav meg som ny forskar og intervjuar ein trygghet og struktur i arbeidet. Denne kombinasjonen av openheit og struktur gjev intervjuet ein klar struktur, men og gjev moglegheita til å kunne gå i djupna på uventa svar (Gleiss & Sæther, 2021).

Intervjuguiden vart utforma for å hjelpe meg til å stille dei «riktige» spørsmåla til elevane. Målet med intervjuet var å finne ut kva elevane tenkte om programmering si rolle i matematikkfaget. Det var likevel viktig å «bil kjent» med eleven, fordi tidlegare erfaringar og mål vil påverke kva me meiner. Dei første spørsmåla i intervjuguiden hadde som hensikt å finne ut om eleven hadde programmert før (tidlegare erfaringar) og kva eleven skulle vidare, då i kor stor grad dei tenkte at dei trengte matematikk og programmering (mål). Under utforminga av intervjuguiden var eg redd for at elevane skulle ha for «lite» å sei på kvart spørsmål eller ikkje forstå spørsmåla. Difor vart det forholdsvis mange spørsmål som kunne hjelpe meg i intervjuet om det var vanskeleg å få eleven i snakk. Sett i ettertid gjorde nok

dette at eg vart veldig fokusert på alle spørsmåla og kanskje ikkje greidde å stille dei gode utfyllingsspørsmåla.

3.3 Val av informantar

Hensikta i undersøkinga er å undersøkje kva elevar tenke om programmerings si rolle i matematikkfaget. Sidan det er forska litt på kva ungdomsskuleelevar meiner, men ikkje vidaregåandeelevar meiner var det interessant å bruke vidaregåandeelevar som informantar. Eldre elevar kan ofte også vera meir reflekterte. Ein god intervjuinformant er ærleg, samarbeidsvillig og evnar å halde seg til spørsmåla, men eit ideell intervjuobjekt finst ikkje (Kvale & Brinkmann, 2015). Sidan eg er lærar på vidaregåande samstundes med eg skriv masteren var det praktisk å bruke mine elevar som informantar. Alle elevane var gamle nok til å samtykke til å vera med i forskingsprosjektet, dette gjorde det lett å innhente samtykke.

I fenomenologisk intervju studie bør det vera meir enn tre informantar, nokon seier 3 til 10 og andre 5- 25, som helst er heterogene (Postholm & Jacobsen, 2018). Om det berre er tre informantar bør dei ha dei same erfaringane og valt ut med bakgrunn i dei same kriterium (Postholm & Jacobsen, 2018). I intervju som det meste av datamaterialet analysen byggjer på er det berre tre informantar. Desse informantane har ikkje programmert før, og har alle lært programmering i det same klasserommet.

Alle elevane i 1T-klassen min fekk utdelt samtykke skjema til å vera med i studia tidleg i haust. Fleire av elevane bytta i løpet av hausten matematikk, til 1P. Mange valte å gjere dette etter første vurdering. Då ingen av elevane trakk tilbake samtykket sitt, er svara dei gav på spørjeskjema etter prøven fortsett ein del av undersøkinga.

Elevane som vart intervjuar var dei elevane som synst det var ok å bli intervjuar. Intervjuar vart gjort i ein periode då mange elevar var borte anten med idrett, eller på grunn av korona. Sidan klassen allereie er ganske liten, og nokon ikkje hadde lyst å bli intervjuar, var det desse tre elevane som vart intervjuar. Talet på elevar som vart intervjuar, var ca. så mange elevar som planen på forhånd la opp til, då det frå nyare undersøkingar viser seg at det er betre å ha færre intervju og konsentrere tida rundt førebuing og analyse av intervjuar (Kvale & Brinkmann, 2015). Sidan eg vurderte desse tre elevane til å vera reflekterte, litt forskjellige og på fagleg ulikt nivå, gjorde det at eg ikkje valte å vente til fleire elevar var tilbake på skulen for å kunne hatt moglegheit til å fått intervjuar fleire.

3.4 Min rolle som lærar og forskar

Når eg er lærar og skal forske på den same elevgruppa vil eg ha ein nærheit til informantane, som ikkje gjere det mogleg å ikkje påverke resultatet. Eit konstruktivistisk tilnærming innebere at ein vedkjenner seg at ein som forskar ikkje kan unngå å påverke resultatet. Forskaren har også ein nærheit og moglegheit til å setta seg inn i informantane sin livssituasjon og liknande (Postholm & Jacobsen, 2018).

Min rolle andsynes elevane kan beskrivast som to-delt. Den eine rolla er at eg er læraren deira. I denne rolla er det mykje ansvar og mange oppgåver. Eg skal rettleie og hjelpe dei til å lære matematikk, vurdere dei, men også vera der som ein vaksen som bryr seg og ser dei. Den andre rolla er intervjuar og forskar. Når eg intervjuar elevane gjekk eg inn i anna rolle, sjølv om det nødvendigvis ikkje var tydeleg for elevane. Eg skulle no forske finne ut av kva dei tenkjer.

Det er eit ulikt maktforhold mellom lærar og elev, i favør av meg som lærar. Forholdet mellom meg og elevane var nok ikkje annleis i den korte stunda eg gjennomførte intervjuet. Eit forskingsintervju er heller ikkje ein heilt fri og open dialog mellom to likestilte partar. Då Intervjuaren styrer samtalen og stiller spørsmåla i ein einvegsdialog. Etter intervjuet fortolkar og analyserer forskaren kva som blei sagt og framstiller dette i ein rapport (Kvale & Brinkmann, 2015).

Eg er sikker på at det var tydeleg for elevane som blei intervjuar at det me snakka om skulle brukast i mi masteroppgåve, og at eg skulle analysere samtalen vår. Forskjellen mellom meg som lærar og forskar frå eit elev perspektiv var truleg ikkje så stor. Sidan eg var den same, og eg viste kven dei er, kva dei meiner og korleis dei jobbar. Difor trengte dei ikkje å gjere seg til på nokon måte, sidan me allereie var godt kjent med kvarandre. Men sidan eg er læraren deira, kunne elevane ha frykta at det dei sa i intervjuet kom tilbake til dei i timane, eller på vurderingane deira. Eg trur at sidan dette var eitt intervju om elevane sine tankar om programmering, så hadde dette truleg ikkje så stor betydning for kva dei svara.

3.5 Ethiske omsyn

Når ein forskar på samfunnet bør det tene menneskje og vitskapen. Det betyr at eg som forskar må vera respektfull overfor eleven samstundes som at eg ynskjer å få intervjuet så grundig som mogleg. Igjennom heile intervjuet vil det oppstå etiske problem, kor mykje kan

eg presse elevane til å fortelja om privatliv sitt, for at eg kan få meir ut av forskinga mi (Kvale & Brinkmann, 2015).

Eg er læraren deira, og har blitt kjent med eleven og veit truleg mykje meir om dei enn ein vanleg forskar ville gjort. Ein lærar vil normalt vera ein person som elevane har tillit til. Elevane stolar nok difor truleg mykje meir på meg enn på mange andre. Dette gjev meg større moglegheiter i intervjuet, men dermed også og eit særskilt ansvar. Ein moglegheit til å få unike og ærlege svar. Med det kjem eit ansvar om å i vare ta elevane og passe på at eg ikkje utnyttar eller framstiller dei feil. Eg må vise meg verdig den tilliten dei har vist meg, så omsynet til deira beste bør alltid gå først. Dette gjer seg nok i større grad gjeldande sidan eg er læraren deira. Kjensla av å bli utnytta av læraren sin vil vera verre enn å skulle bli utnytta av ein tilfeldig forskar.

Elevane som er med i denne undersøkinga har alle skrive under på eitt samtykkeskjema. Sidan dei er elevar på ein skule og det er i den samanheng dei er blitt valt til informantar, er eit skriftleg samtykke lurt, for å beskytte dei men også meg som forskar (Kvale & Brinkmann, 2015). I denne undersøkinga var det elevane sjølv som gav samtykke, dette er fordi alle elevane var over 15 år og det ikkje vart samla inn nokre personopplysingar. Alle elevane i 1T-klassen gav samtykke til å delta i undersøkinga med innsamling av dei forskjellige rådata. Elevane vart mint på at det var frivillig om dei ville svare på spørsmål etter prøven, og alle elevane som vart intervjuet vart spurdt på nytt munnleg rett før intervjuet.

3.6 Rehabiliteren og validitet i oppgåva

Rehabiliteren i ei undersøking handlar om at ein annan forskar skal kunne gjera den same undersøkinga og få det same svaret (Kvale & Brinkmann, 2015). Som nemnt tidlegare har eg her hatt både rollen som elevane sin lærar og forskaren som skal undersøka tankane deira kring programmering. Det er vanskeleg å vita i kva grad denne rolleblandinga kan ha påverka undersøkinga. Dermed er det også vanskeleg å kunne vurdere om ein annan forskar kunne ha stilt dei same spørsmåla til dei same elevane, og fått dei same svara.

Det var eit maktforhold-forskjell mellom meg og elevane, fordi eg er læraren deira. Dersom leiande spørsmål blir spurdt nokon som er lett påverkelege må ein vurdere validiteten til svara (Kvale & Brinkmann, 2015). Elevane i denne undersøkinga vart nok i ulik grad påverka av meg. Slik eg opplevde intervjuet trur eg spesielt Signe og Liv Hilde kan bli påverka av ledande spørsmål. Liv Hilde spurde fleire gonger om det ho sa var riktig, og Signe blir av meg

oppfatta som oppteken av å vera flink og er difor ute etter å ei få «godkjenning» av meg som lærar.

Ei kvalitative undersøking er berre så god som det datamaterialet ein greier å samla inn, det materielt kjem alltid frå ei kjelde. Informantane si gyldigheit vil difor påverke gyldigheita til datamaterialet i undersøkinga. Når elevane svarar på spørsmål frå forskaren vil svara ofte komma som ein respons på spørsmålet. Difor får eg som forskar berre svar på det eg spør etter (Jacobsen, 2016). Hadde eg formulert spørsmåla til elevane annleis, eller stilt andre spørsmål kunne eg fått andre svar. Då eg hadde fleire like spørsmål som bygde på kvarandre håpar eg å ha minska sjansen for å fått andre svar ved å ha stilt spørsmåla annleis.

For å ta omsyn til det at eg som forskar og lærar kan ha med egne meiningar inn i forskinga, har eg diskutert moglege årsaker til dei forskjellige funna. Denne diskusjonen utfordrar meg sjølv som forskar til å sjå funnet frå forskjellige vinklar, og vil styrka den indre validiteten av forskinga. Når eg avsluttar kvart funn med å sjå det opp mot tidlegare forskning sjekkar eg validiteten av undersøkinga mot andre undersøkingar.

Sidan dette er ei lita case-studie vil den gje kunnskap om korleis elevane tenkjer i denne klassen, på denne skulen. Om denne studia blei gjort om nokre år på det same klassetrinnet, kan det henda at resultatet hadde blitt noko heilt anna. Om nokre år vil elevane som har følgd den nye læreplanen (LK20), ha meir forkunnskap i programmering når dei startar på vidaregåande, enn elevane i min klasse hadde.

3.7 Utføring av analyse

Svara elevane gav på spørjeskjema vart samla i eit eige dokument. Eg trengte ikkje å gjera noko analysearbeid for å forstå at det var for lite datamateriale. Dette saman med kor få av elevane som hadde gjort programmeringsoppgåvene på prøven, gjorde at eg og rettleiar meinte eg burde intervju nokre av elevane.

Etter at intervju var gjennomført, transkriberte eg lydfilene med ein gong. Dette tok tid og var litt utfordrande, blant anna fordi det var nokon ord som forsvann eller ikkje gav meining, men mest fordi eg aldri ha gjort det før. Difor hugsa eg mykje meir kva eleven frå det første intervjuet hadde sagt og korleis den oppførte seg enn det eleven i det siste intervjuet eg transkriberte. Når elevane namngav tidlegare skular, eller andre ting som kunne svekke anonymiteten deira, valte eg å ta bort namnet, eller skifte ut det som kunne ha svekka anonymiteten deira.

Dette var første gang eg analysere noko datamateriale av denne størrelsen. Igjennom transkripsjon og fleire gjennom-høyringar av intervjuet vart eg godt kjent med materialet. Sett tilbake var det den enkle delen av analysen. Korleis eg skulle kode datamaterialet synst eg var mykje meir utfordrande. Frå professorar, med-studentar og likande viste eg at eg måtte kode datamaterialet. Koding gjere det enklare å få ein oversikt over datamaterialet, ved å forenkle det og setta ord/ setningar på kva utsegna til elevane seier noko om (Gleiss & Sæther, 2021).

Etter ein del søk i Google og lest i bøker om emnet prøvde eg meg først på deduktiv koding då tanken min var at då veit eg kva eg skal sjå etter i det transkriberte intervjuet. Deduktiv koding blir brukt når ein har ein generell teori ein vil testa og samanlikna med datamaterialet ein har samla inn (Kvale & Brinkmann, 2015). I mi undersøkinga var det ikkje ein teori som eg vil teste ut, og det vart då eigentleg heilt feil nytte metode for deduktiv koding.

Ved ein induktiv analyse, intervjuar eller observerer ein eit tal tilfelle, og prøver utifrå det å kunne seie noko generelt om emnet, eller problemstillinga (Kvale & Brinkmann, 2015). Eg hadde samla inn ein del datamateriale som handla om kva elevar meiner om programmering i matematikkfaget, og ville prøve å sei noko generelt om dette. Difor kom eg til at det var meir rett å bruke induktiv metode for å analysere det innsamla materialet. I starten opplevde eg at det var vanskeleg å sette seg inn i og å bruke denne metoden .

Den induktive analysen vart gjennomført i NVivo. Det å bruke eit dataprogram i analyse kan gjere det litt lettare, då ein slepp å «klyppe og lime» mange sider, kan lett skrive notata og leite etter nøkkelord (Kvale & Brinkmann, 2015). Både dei transkriberte intervju og svara på spørjeskjema vart koda saman i NVivo. Nesten alt av datamaterialet vart til kodar, det kunne vere heile utsegna eller delar. Kodane var kortare eller meir beskrivande for kva elevane hadde sagt, dette slik at det vart lettare og kategorisere kodane og eg kunne sjå eit mønster. Når ein kode er målet å finne koda og kategoriar som beskriv og fange det ein forske fult ut (Kvale & Brinkmann, 2015). Sidan det meste av datamaterialet vart koda, og deretter kategorisert var det ein del av kodane og kategoriane som ikkje var aktuelle til å svare på problemstillinga .

Etter at eg var ferdig med å kode intervju og spørjeskjema las eg igjennom dei fleire gonger for å lage meg eit indre bilde og for å få oversikt. Så starta eg å samle kodane som handla om det same i ein kategori. Slik heldt eg på til alle kodane var lagt i kategoriar. Det var då sju kategoriar. Nokre av kategoriane var meir beskrivande og presise enn andre. Etter at alle

kategoriane var laga, vart det klart at nokre av kodane passa fleire stadar. Difor vart alle kodane gjennomgått på nytt og sjekka om dei passa i fleire kategoriar. Nokre av kategoriane innehald relativt mange kodar, som gjorde kategorien uoversiktleg. Difor vart det laga nokre underkategoriar, slik at det vart lettare å få ein oversikt over kva kategorien fortalte. Det vart tydleg at fleire av dei sju kategoriane som vart laga i starten kunne samlast i ein kategori med underkategoriar. Difor enda eg til slutt opp med 4 hovudkategoriar.

Ein kategori handla om kven eleven er og kva mål den hadde, denne kategorien gav ingen funn. Resten av kategoriane gav alle funn, to av kategoriane gav fleire funn. Ein kategori tankar og meiningar gav berre eitt funn. Det var truleg fordi kategorien var veldig open, då kodane var relevant for oppgåva, men ikkje høyrde heime i dei andre kategoriane vart det ein open kategori. Koda frå den kategorien hjalp til å underbygge andre funn frå andre kategoriar, då frå ei an vinkling.

I arbeidet med å lage kategoriane vart det tydleg at mykje av det datamaterialet eg hadde handla om kjenslene elevane hadde knytt til programmering i matematikkfaget. Det vart naturleg å lage eitt nytt forskingsspørsmål som undersøkte kjensler, sidan kjensler og tankar er nært knytt.

4 Analyse

For å kunne svare på dei to forskingsspørsmåla har eg samla inn data frå ein 1T-klasse. Datamaterialet som er samla inn er resultat frå prøvar, spørjeskjema etter kvar prøve, intervju med tre elevar samt observasjonar gjort i matematikktimar. I dette kapitelet vil eg leggje fram kva funn som blei gjort igjennom analyse av datamaterialet. For å beskrive prosessen frå rådata til funn vil eg bruke funnet *meistringsnivået til elevane på ei oppgåve bestemmer kva kjensle dei får til programmering*, som eksempel. Før funna blir presentert enkeltvis ordna etter forskingsspørsmål vil eg presentere elevane som blei intervjuet. 1T-klassen som er brukt for datainnsamling er den klassen eg underviser i 1T. Eg har observert elevane mens dei jobbar, hjelpt dei med matematikkoppgåver og har såleis blitt kjent med dei. Dette gjere at eg tolkar utsegna deira på ein måte eg ikkje kunne gjort utan å ha fått denne relasjonen til dei.

Analysen blei gjennomført ved at intervjuet var transkriberte, deretter var dei lest igjennom fleire gonger, før intervjuet blei koda i NVivo. Det vart gjennomført ei induktiv koding, der eg koda det i teksten som vekte interesse med tanke på forskingsspørsmåla, men også om det var noko anna som fatta interesse. Dette vart gjort for alle intervjuet og også for spørjeskjema. Så blei kodane sortert i kategoriar som beskreiv og eller samla kodane. Nokre av desse kategoriane var store og det blei utarbeida underkategoriar for å få meir meining og oversikt. Det blei då klart at mykje av datamaterialet handla om kjenslene til elevane kring programmering, difor vart det laga eitt nytt forskingsspørsmål som tok for seg kjenslene til elevane. Til slutt vart kategoriane analysert mot forskingsspørsmåla. Nokre av kategoriane gav funn, mens andre ikkje gav funn som var relevante for forskingsspørsmåla.

Når kodane skulle bli kategorisert var det tydeleg at kjensler måtte bli ein eigen kategori. Det var mange kodar som omhandla kva elevane kjente på. Sidan dette vart ein stor kategori oppretta eg tre underkategoriar som representerte høvesvis; positive, negative og blanda kjensler. Det vart då tydeleg at det måtte opprettast eit forskingsspørsmål som undersøkte kva kjensler elevane har til programmering. I vidare analyse av dei tre underkategoriane viste det seg at dei fleste kodane i dei forskjellige underkategoriane var «ulike sider av same sak». Dette leia til det første funnet som er at; *meistringsnivået til elevane på ei oppgåve bestemmer kva kjensle dei får til programmering*.

I dette kapitelet vil kvart funn bli presentert i del-kapitelet til tilhøyrande forskingsspørsmålet, med eksempel og sitat frå elevane. Det vil bli gjort greie for kvifor kvart av funna er aktuelle for forskingsspørsmåla. Dei to første funna er knytt til det første forskingsspørsmålet og del-kapitelet *Kva er elever sine kjensler knyt til programmering i matematikkfaget*. Funna som vil bli presentert i del-kapitelet *Kva er elever sine kjensler knyt til programmering i matematikkfaget* er;

1. Grad av meistring på oppgåver bestemmer kva kjensler elevane får om programmering
2. Tradisjonell matematikk er betre enn programmering

Funnet «sein start med programmering» kan ligge under begge forskingsspørsmåla. Det vil vera naturleg og diskutera funnet saman og i lys av funnet «for lite tid til programmering», difor ligg det under forskingsspørsmålet *Kva er elevar sine tankar kring programmering si rolle i matematikkfaget*.

Resten av funna er knytt til det siste forskingsspørsmålet og andre del-kapitelet – nemleg; *Kva er elevar sine tankar kring programmering si rolle i matematikkfaget*. Funna som presentere til dette forskingsspørsmålet er;

3. Seint start med programmering
4. For lite tid til programmering
5. Programmering er koding
6. Programmering er matematikk fordi det brukast matematiske funksjonar
7. Alt kan løysast ved programmering
8. Sjølvmotsetningar i elevar sine meiningar om programmering

4.1 Elevar sine kjensler om programmering

4.1.1 Grad av meistring på oppgåver bestemmer kva kjensler elevane får om programmering

Det første funn er at «utfallet» av programmeringsoppgåva eleven jobbar med bestemmer kva kjensle elevane får om programmering, som igjen påverkar motivasjonen. Enkelt sagt; dersom elevane får til ei oppgåve, synst dei programmering er artig. Dersom dei ikkje får til ei oppgåve, likar dei ikkje programmering. Dersom elevane får til fleire oppgåver, er dette truleg bygd på ei forståing av programmering. Kva forståing av programmering elevane har påverkar motivasjonen til elevane. Meistringskjensla er ein av faktorane som påverkar motivasjonen (Wæge & Nosrati, 2018). Kjensler til programmering basert på grad av

meistring, og motivasjon som blir påverka av forståing knytt til programmering, er difor nært knytt til kvarandre.

Signe er eit veldig godt eksempel på kjensler til programmering kjem frå meistring av programmeringsoppgåver. Dersom ho ikkje får til oppgåva hatar ho programmering og angre på at ho valte 1 T, men dersom ho meistrar det synst ho programmering er artig.

Tore sin kommentar om eigne kjensler til programmering er:

«Så eg ikkje heng heilt med i matematikken, nei programmeringa, men elles så hadde det ikkje vært for det, så trur eg hadde likt det.»

Tore uttrykkjer ikkje direkte at han misliker programmering fordi han ikkje meistrar det. Men han heng ikkje heilt med på programmeringa, så det er truleg at han ikkje kjenner på meistringskjensle knytt til programmeringa. Sidan Tore ikkje meistrar programmering, har han ei negativ kjensle til programmering. At han er ikkje likar programmering, seier han kvar gang det blir programmering i timane. Difor tolkar eg det slik; at kva han opplev av meistring legg føring på kva han synst om programmering.

Dei fleste av elevane som har bidreg i denne studia, både intervjuet og spørjeskjema, har negative kjensler til programmering. Dei seier også at dei synst programmering er utfordrande, og slit med å løyse oppgåvene dei får. Dersom dei får til programmeringa, seier elevane at programmering er artig eller gøy.

Det varierer mellom elevane er i kor stor grad dei let seg påverke av dei negative kjenslene dei får når dei ikkje meistrar programmeringsoppgåvene. Liv Hilde er den einaste av elevane som ikkje let seg påverke av å ikkje meistre oppgåva. Ho prøver igjen, og let ikkje dei negative kjenslene få ta over, dette går igjen når ho arbeidar med tradisjonell matematikk og. Signe og Tore let seg påverke av dei negative kjenslene, begge seier at når dei ikkje får til oppgåvene går det utover motivasjonen deira. For begge er motivasjonen for programmering låg, eller at den blir låg med ein gong dei ikkje får programmeringa til .

«Fordi eg føle at det er mange ting eg er god til, men så kommer eg til sånn typ si programmering, så kan det dra meg ned skikkeleg for det er sånne små ting som eg gjere feil liksom» -Signe

Signe let seg påverke mykje av kjenslene sine, og når ho då ikkje får til programmeringa, går det med ein gang utover motivasjonen hennar.

Kor mykje elevane forstår av programmering påverkar kor god motivasjonen deira til programmering er. Liv Hilde og Tore har utsegn som eg tolkar slik at forståinga og motivasjonen deira har utvikla seg saman gjennom året. Liv Hilde har fått betre motivasjon til å programmere dess meir ho har lærd og då di meir ho forstår.

«Sånn no er det betre for no forstår eg jo kva eg gjere»- Liv Hilde

På den andre sida av spekteret er Tore som har fått lite forståing av programmeringa, og har då heller ikkje fått noko meir motivasjon for å lære programmering.

4.1.2 Tradisjonell matematikk er betre enn programmering

Elevane i denne studia likar den tradisjonelle matematikken betre enn programmering.

«Eg synst kanskje sånn tradisjonell er meir, eller eg har meir motivasjon for å gjere det, men det er jo fordi programmering er så ukjent» - Liv Hilde

Liv Hilde som er den som oppfattast mest positiv til programmering, vil også heller rekne matematikk enn å programmere.

«Nei, eeeh eg synst det gjere matematikken vanskeligare enn det det allereie er»- Tore
Motivasjonen til Tore i matematikkfaget blir dårlegare når programmering er ein del av faget. Dette tolkar eg som han likar den tradisjonelle matematikken betre enn programmering. Signe seier det same som Tore, at motivasjonen for å jobbe med matematikk blir dårlegare når programmering er med.

På spørjeskjema elevane svarde på, var det ein elev som svarde at det er kjekt å gjere noko anna enn den vanleg matematikken. Dette må ikkje vera motstridane med funnet om at elevane generelt likar tradisjonell matematikk betre enn programmering.

4.2 Elevar sine tankar kring programmering si rolle i matematikk

4.2.1 Seint start med programmering

Elevane meiner dei skulle begynt med programmering tidlegare i skuleløpet. Alle elevane uttrykte ordrett ein gong i løpet av intervjuet at dei synst dei starta for seint med programmering. Dei meiner at det ikkje har fått lært det grunnleggjande ved programmering før det blir forventa at dei skal kunne det. Signe meiner om dei hadde begynt opplæringa før ville ho hatt moglegheita til *«å ha den basisen»* med *«det elektroniske»* som ho kallar det, på plass til ho skulle bruke og bli vurdert i emnet.

«..., føle me har blitt slengt inn i det» - Tore.

«Ja sånn at man kanskje begynte med det på ungdomskulen, sånn at man hadde til det man til vidaregåande.....» - Liv Hilde.

Eg tolkar det slik at elevane sit med ei kjensle av at dei har blitt kasta inn i noko nytt og ukjent, utan å ha fått opplæringa i det dei skulle hatt. Dei blir vurdert i eitt pensum som tilseier at dei skulle hatt programmering i fleire år, dette synst spesielt Signe er utfordrande.

Dette funnet passar inn under begge forskingsspørsmåla, men er valt plassert under elevar sine tankar kring programmering når alle meiner at ein burde starta før med det. Dette vil hengje saman med funnet *«for lite tid til programmeringa»* i den etterfølgjande diskusjonen

4.2.2 For lite tid til programmering

Elevane synst det er for lite tid til å lære programmering i matematikkfaget, og det er nok anna pensum dei skal igjennom. I løpet av ei vanleg skuleveke med fem skuletimar matematikk er målet at elevane skal ha komme igjennom 2 til 3 del-kapitel i boka. Dette avhenger av kor lange og kor mykje nytt det er i eitt del- kapitelet. På ungdomsskulen meiner elevane at det var betre tid og at det hadde vore fint å begynt med programmering då, når det ikkje betydde så mykje.

«For på vidaregåande er det jo veldig fullstappa, mens på ungdomskulen er det litt meir slakt.» - Liv Hilde

På starten av skuleåret fekk elevane to veker med introduksjon av programmering. Vidare i skuleåret har det vore programmering når læreboka har lagt opp til det, noko som ikkje har vore ofte.

«Eg synst ikkje det er nok tid til at vi skal lære..... At det blir ikkje så veldig mykje sån jobb, vi får berre litt sånne korte gjennomgang, så må du huske på ein del sjølv.....» - Signe

For Signe er dei felles gjennom gangane for korte, ho får heller ikkje nok tid til å jobbe med det etterpå slik at ho lærer det. Ho skulle gjerne hatt ei heil økt etter økta for å jobbe med det nye, spesielt i programmering, men også tradisjonell matematikk.

4.2.3 Programmering er koding

Elevane meiner programmering er koding. For å kunne diskutere kva elevane meiner om programmerings si rolle i matematikkfaget, er det nødvendig å vite kva dei meiner programmering er. Difor er dette funnet tatt med i analysen.

Alle elevane fekk spørsmål om å definere kva programmering er, svara deira varierer litt i ordlyd. Men eg har erfart at Signe si forståing av kva programmering er, er ganske representativ for mange av elevane;

«Koding, eller skrive inn ehhe si skrive inn masse rare formlar eller variablar elle verdier eller sånt også få eit produkt av det»- Signe

For Signe er det å programmera kunn det å skrive koden inn i Pythonisa (programmet elevane brukar for å kode) som er programmering. Ho har ein litt meir snever definisjon av programmering enn kva nokon av dei andre medelevane svarar på det same spørsmålet, men oppfatninga tolkar eg til å vera lik.

Tore svarar at programmering er å bruke eit program for å løyse ei oppgåve, der ein set inn noko og får eit produkt. Dette tolkar eg til at han meiner at programmering er å bruke Pythonisa (programmet elevane brukar for å kode), som er å skrive ein kode, altså koding som Signe seier.

Liv Hilde sin definisjon av programmering med ord, og hennar faktiske oppfatning av kva programmering er, synest å kunne vera noko motstridane. tolkar eg som ulike. Ho uttaler at *«programmering er eit hjelpemiddel for å løyse problem fortare»*.

Men i intervjuet svarar ho slik på kva utfordringar det er med å programmere:

Liv Hilde: Sånn først var det liksom å skrive inn alle di her formlane, kodane og det man måtte ha, men no er det meir sånn å liksom finne, sånn som på den oppgåvene i sted, å finne den der formelen også skrive den.

Meg: Det var oppgåve 2?

Liv Hilde: Ja, så det var eigentleg ikkje noko med programmeringa.

Her rettar ho på seg sjølv, til at programmering ikkje har noko med å løyse eitt problem, men at programmering er å skrive inn det som skal inn i Pythonisa, altså koding. Difor tolkar eg at også Liv Hilde har ei oppfatninga av at programmering berre er å kode.

Frå spørjeskjemaet svarer elevane at deira utfordringar knytt til programmering handlar om å skrive kode, eller om koden er riktig slik at koden fungerer. Dette gjev eitt grunnlag for å sei at dei fleste elvane i denne 1T-klassen har ei oppfatning av at programmering berre er å kode.

4.2.4 Programmering er matematikk fordi det brukast matematiske funksjonar

Elevane synst programmering er matematikk. Programmering er matematikk fordi programmering nyttar tall og formlar, «matematiske funksjonar».

Signe seier at det er basisferdigheitene frå matematikk som blir brukt i programmering, og at det er gjennom programmering at ho vil få bruk for matematikken ho lærer. Liv Hilde meiner programmering er matematikk fordi du kan bruke programmering for å løyse matematikkoppgåver, og at ein brukar tall og formlar som også blir brukt i matematikk. Tore er usikker på om han synst programmering er matematikk. Han meiner det er matematikk fordi ein brukar matematiske funksjonar, men det er også så mykje nytt at det er eit nytt fag.

Elevane har ikkje tenkt over om det er naturleg at programmering er blitt ein del av matematikkfaget, sjølv om det er matematikk i programmering. Dei er einige at det er meir naturleg med programmering som ein del av matematikkfaget enn i andre fag som for eksempel norsk eller samfunnsfag. Elevane er blanda i kjenslene sine om programmering bør vera ein del av matematikkfaget, men eg tolkar Signe og Tore som at dei truleg hadde satt pris på om det var eitt eige fag.

4.2.5 Alt kan løysast ved programmering

Elevane meiner det gjev meinig å bruke programmering til det meste i matematikkfaget.

Elevane fekk spørsmål om når det gav meinig og når det ikkje gav meinig å programmere. Svara på desse to spørsmåla gjere at eg tolkar elevane slik at dei meiner det meste greitt kan løysast med programmering i matematikken.

«Eigentleg alle oppgåva. Eg kan jo få, aaååøø eller ikkje alle oppgåva men di fleste oppgåvene kan eg løyse med programmering.» - Tore

På spørsmål om når det ikkje gav meinig å programmere, meiner jentene at dei ikkje har noko gode eksempel for det meste er naturleg å programmere. Dette gjere at eg tolkar elevane til å meine at det meste dei skal gjere i matematikkfaget er naturleg å programmere.

Elevane vart utfordra på å gje spesifikke eksempel til oppgåve der det er nyttig å bruke programmering. Då drog alle fram oppgåvene dei hadde jobba med i timen før. Liv Hilde og Signe greidde å generalisere kva type oppgåver det var; oppgåver der det var rekne operasjonar som skulle gjentakast

4.2.6 Sjølvmotsetningar i elevar sine meiningar om programmering

Elevane har fleire sjølvmotseiingar om programmering, og programmering i matematikkfaget. Basert på dei tidlegare funna i oppgåva og ein generell oppfatning av kva elevane seier undervegs i intervjuet er det tydeleg at dei seier mot seg sjølv om kva programmering er. Dette har fått eit eige funn, då problemstillinga handlar om elevane sine tankar om programmering i matematikkfaget, då eg vil bruke det i diskusjonen om dei eigentleg har forstått kva programmering er, og difor uttaler seg som dei gjere i programmering si rolle i matematikkfaget. Nedafor er det eksempel på desse sjølvmotseiingane:

Tore meiner at dei fleste oppgåver kan vera lure å programmere, men så lurar han på kvifor han ikkje kan bruke eit halv minutt lenger på å løyse ei av oppgåvene frå utforskingstimen for hand.

Signe meiner det gjev meining å løyse førstegradsligningar ved programmering, men ikkje andre og tredjegrads. Det er berre andregradsligningar me har løyst ved programmering.

5 Diskusjon

I dette kapitlet skal eg diskutera funna frå analyse. Funna vil bli diskutert kvar for seg i rekkefølgjene dei vart presentert i analysen. Først vil eg sjå på moglege årsaker, før funnet blir knyt opp mot tidlegare forskning og teori. Kvar enkelt funn vil bli diskutert på den måten. Diskusjonen om kvart funn blir avslutta med kva det implisere og korleis det er relevant for forskingsspørsmåla. Nokre av funna vil sjåast i samanheng, då dei går litt inn i kvarandre.

5.1 Grad av meistring på oppgåver bestemmer kva kjensler elevane får om programmering

Måloppnåing på programmeringsoppgåva eleven jobbar med bestemmer kva kjensler eleven får om programmering, som igjen påverkar motivasjonen. Det mange grunnar for dette og det er nok samansett. Kjensler og motivasjon heng ofte saman. Ein del av dette funnet er at motivasjonen til elevane blir påverka av grad av forståing. I denne studia hadde eleven med mest negative kjensler også låg motivasjon for programmering. Tore sa tydeleg at han ikkje forstår programmering. Kor høg motivasjon elevane har for å jobbe med programmering blir påverka av kor mykje dei forstår.

Ein av grunnane til at grad av meistring på programmeringsoppgåver påverkar kjenslene til elevane kan vera at når elevane skriv kode blir det veldig tydeleg om dei får det til eller ikkje. Dersom ein er opptatt av å vera flink eller at ein trur ein er god og kan mykje, vil kva ein får til sette denne oppfatninga på prøve. Det kan enten forsterke eleven si eiga oppfatning om at ein er god, noko som truleg vil gje ei god kjensle, eller det kan sette eleven sitt sjølvbilde på prøve. Å ikkje få det til er ikkje ei god kjensle og vil truleg føre til at ein ikkje får ei god kjensle knytt programmering. Dette ser ein teikn til hos Signe. Ho er den av elevane som uttrykte tydelegast at dersom ho greidde programmeringsoppgåver så likar ho det, men om ho ikkje greidde dei då hata ho programmering. Ho uttaler også at ho meiner ho burde vera god til «slikt» og legg eit press på seg sjølv om å få det til.

Det å få til noko er kjekt, det gjev ei god kjensle. Det er difor ikkje så rart at elevane i denne studia blir positive når dei beherskar og greier å løyse programmeringsoppgåvene. Kva ein føler når ein ikkje greier det er nok litt meir varierende, ein kan bli irritert, likegyldig, motivert. Det kan avhenger av kva forventingar ein har når ein begynner å løyse programmeringsoppgåva, og kva motivasjon elevane har. Liv Hilde er den av elevane som har

høgast motivasjon, ho seier også at ho forstår programmering betre no enn før. Liv Hilde har ein forventning om å måtte arbeider med programmering og matematikken for å få det til. Når det er ei programmeringsoppgåve ho ikkje får til arbeider ho til det går opp, fordi ho har ein forventning om at det krevst. Ho vil da til slutt få ei positiv kjensle, og difor blir ho kanskje positiv til programmering.

Liv Hilde kjenner at ho har forstått meir og meir av programmeringa utover i skuleåret, då er det ikkje så rart at ho har blitt meir motivert. Ho ser at det arbeidet ho har lagt ned gjev eit positiv resultat. Ho har greidd å opparbeide seg ein større forståing av programmeringa. Dette kan vere sjølvforsterkande og vil truleg føre til ho er meir positiv til programmeringa og har då meir motivasjon til å arbeide vidare med programmeringa.

Uansett om ein er opptatt av å forstå eller prestere bra vil det å ikkje få til programmeringsoppgåve etter programmeringsoppgåve vera utfordrande, både for kjensla kring programmering og motivasjonen. Tore føler på at han ikkje får til nok eller kan programmering. Dette har han kjent på det gjennom heile året. Difor er det veldig forståleg og naturleg at han ikkje likar programmering. Det skal mykje viljestyrke for å ikkje blir umotivert og negativ når ein kjenner på eit konstant nederlag.

Som nemnt i teori-kapitelet er det ikkje skrive så mykje om programmering i matematikk og spesielt ikkje om kva som påverkar kjenslene til programmering i matematikk. Det som eg har funne vil eg prøve etter beste evne å gjera så aktuelt som mogleg

Når elevar opplever å mestre programmering gjev det elevane motivasjon til å jobbe med programmering. Det er viktig at programmeringa er tilpassa til elevane sitt nivå, slik at dei har moglegheit til å mestre oppgåvene. Fleire elevar opplevde meistring ved analog programmering, og difor fekk betre motivasjon til programmering (Berg, 2021). Berg (2021) støttar opp om at det er ein korrelasjon mellom meistring av programmering og motivasjon i programmering. Det som er verdt å merke seg frå Berg (2021) er at det er ein læraren som er intervjuet og den sitt syn på analog programmering.

Kompetanse er ein av faktorane som spelar inn på motivasjon (Wæge & Nosrati, 2018). Kompetanse i matematikkfaget handlar for elevane om at dei utviklar forståinga og evnene sine. Liv Hilde har hatt ei utvikling av forståinga i programmering, og ho kjenner på ein høgare motivasjon. Det kan tyde på at forskning for motivasjon i matematikk kan overførast til motivasjon i programmering i matematikkfaget.

Meistringkjensle kjem av meir enn berre å få til oppgåver i matematikkfaget, det kan komme av mellom anna av å stilla spørsmål, resonnera og forstå matematiske omgrep (Wæge & Nosrati, 2018). Det er vanskeleg å sei om elevane i denne studia opplever meistringkjensle i programmering som kjem av desse alternativa. Det er lite truleg fordi elevane meiner programmering er koding. Difor er det truleg at dei ikkje resonnerer og stiller spørsmål når dei programmerer og får ei meistringkjensle. Dersom elevane resonnerer i samband med programmeringsoppgåvene er det lite truleg at dei knyter det til programmering, og difor er meistringkjensle dei har i programmering knytt til å meistre kodinga i oppgåvene.

Kva val eleven tek undervegs i problemløysinga og programmeringa spelar inn på kva læringsutbytte eleven får av programmeringa (Kaufmann & Forsström, 2018). Sjølv om fleire elevar har fått ein betre motivasjon av programmering er det inga garanti for det. Fleire elevar slit med å sjå kva læringsutbytte programmering har i matematikk (Kaufmann & Forsström, 2018). Mangel på motivasjon kan komme av at elevane ikkje ser læringsutbytte av programmeringa. Det kan kanskje sjåast i samband med at elevane ikkje får til oppgåvene, og då kan det vera vanskeleg å sjå vitsen med oppgåva.

Den indre motivasjonen til elevane i matematikk blir ofte mindre di eldre dei blir, ein grunn til dette er at matematikken ofte blir lenger frå verkelegheita og vanskelegare. Den indre motivasjonen gjere elevane meir uthalde til å løyse matematikkoppgåver (Wæge & Nosrati, 2018). Ein av grunnane til at Tore og delvis Signe difor slit med motivasjonen kan vera at den indre motivasjonen er låg som vidare gjer at dei har mindre uthald i programmeringa. Det kan komme av dei ikkje ser relevansen for programmering. Det er truleg at gjeld for Tore som seier han ikkje kjem til å jobbe med noko der han kjem til å få bruk for programmering. Han ser dermed ikkje ein grunn for å lære seg programmering.

Som lærar bør ein vere merksam på å gje elevane ei meistringkjensle knytt til programmering, dette for at elevane skal få positive kjensler knytt til programmering. Det kan vera gjennom at dei får til programmeringsoppgåver, eller at elevane blir bevist på at programmering er meir enn koding. Da kan dei gjennom resonnering og ved å stille spørsmål få noko meistringkjensle knytt til programmering. Ei god forståing av programmering gjere at elevane blir meir motivert for programmering, som kan igjen kan føre til meir meistring og forståing. Dette stiller store krav til ein som lærar, at ein evnar å gje tilpassa oppgåver eller problem, men også å rettleie elevane til å meistre oppgåver.

Dette funnet gjev ein mogleg forklaring på kvifor elevane kjenner på dei kjenslene dei har om programmering.

5.2 Tradisjonell matematikk er betre enn programmering

Elevane i denne studia tykkjer den vanlege matematikken er kjekkare enn programmering. At elevane likar den tradisjonelle matematikken og har meir motivasjon for den er ikkje overraskande, då dei har velt 1T som er vanskelegare matematikk enn dei må ha. Ein vel ofte ikkje det vanskelegaste alternativet dersom ein ikkje likar det og finn det spennande. At elevane likar den tradisjonelle matematikken betyr ikkje det nødvendigvis at dei likar det betre enn programmering, men funnet viser at elevane gjere det. Det betyr ikkje at elevane ikkje likar programmering, men at den tradisjonelle matematikken er kjekkast. Kva kan vera grunnane til det? Eitt enkelt svar er at dei verkeleg likar den tradisjonelle matematikken, og at det skal mykje til for at dei likar noko betre.

Matematikkfaget i grunnskulen har ofte vore kritisert for å vera instrumentelt. Det skil seg frå korleis ein matematikkar jobbar, men også korleis ein jobbar når ein programmerar. Elevane har truleg gjennom heile grunnskulen vore vand til at det har vore ein riktig måte å løyse matematikkoppgåvene på, ei oppskrift dei kan følgje, og ein fasit. Dette er ikkje tilfelle når ein programmere, då er det mange måtar å løyse oppgåva på. Det som avgjere om det er «riktig» er om ein greier å løyse problemet. Programmering er ein heilt ny måte å jobbe på, ein måte som dei kanskje ikkje forbind med matematikk og kan vere det motsette av grunnen til at eleven likar matematikk. Spesielt om eleven har ein tanke om at «matematikk har reglar for korleis det skal løysast og det er eitt svar».

Programmering skal inn i matematikkfaget, og kan såleis sjåast som eitt emne innafor matematikk. Ein kan såleis sei at den tradisjonelle matematikken inneheld mange emne, mellom dei programmering. Elevane samanliknar eit store emne som inneheld mange mindre med eitt emne, det er då større von for at elevane har eitt eller fleire emnar som dei likar betre i den tradisjonelle matematikken enn programmering. Det er også enklare å finne noko innafor den tradisjonelle matematikken ein likar.

Programmering er nytt for elevane. Elevane har nok store kunnskapshull og ein smal forståing av kva programmering er. Dersom det er tilfelle veit ikkje elevane alle moglegheitene når det kjem til programmering. Dei ser berre litt av alt dei kan gjere i og med programmering. Om

elevane hadde skjønt meir av kva moglegheiter programmering gjev, hadde kanskje elevane likt programmering betre.

Ein anna moglegheit sett i lys av at programmering er nytt for elevane, er frykta for det ukjente. Når noko er nytt og ukjent er det ofte litt skummelt. Ein blir ofte også mindre nyfiken når ein blir eldre. Elevane må difor tørre utfordre seg på nok heilt nytt, som dei mest truleg ikkje meistrar med ein gong. Det krev mykje av elevane, då dei må tørre å feile, ikkje vera best og ha nullpeiling på korleis det skal gjerast. Dette er skummelt og difor er det ikkje rart om det spelar inn på kva dei synst om programmering, og at dei difor likar det dei kjenner, den tradisjonelle matematikken betre.

Eliassen et al.(2022) gav ut ein artikkel som ser på korrelasjonen mellom kunnskapen i R2, matematikk og motivasjonen i eit berekningsorientert programmering, eit fag med programmering for biologi. Dei finn at studentane meiner det største motivasjonen for å jobbe med faget var nytten av faget. Studentane hadde indre motivasjon for biologistudiet. Elevane i denne studia har valt IT-matematikken, men ikkje programmering. Dersom dei ikkje ser nytta av programmering kan det i lys av Eliassen et al.(2022) si forskning vera noko av grunnen til at dei likar den tradisjonell matematikk betre og dei har meir motivasjon for å jobbe med den enn programmering.

Undersøkjande matematikk som er tilpassa nivået til elevane og oppgåver som dei opplev som meningsfulle kan auke motivasjonen til elevane (Wæge, 2007).

Programmeringsundervising burde enkelt kunne leggjast opp på denne måten. Sidan funna i denne undersøkinga tyder på at den tradisjonelle matematikken er betre og alle elevane har meir motivasjon for den tradisjonelle matematikken kann det tyde på at programmeringsundervisinga ikkje er lagt opp på ein så måte.

Signe synst programmering kan øydeleggja mykje av matematikken dersom ho ikkje får det til. Når elevar jobbar med matematikk eller programmering for å oppnå eit resultat er det ytre motivasjon, då er elevane mindre uthaldne og meir påverkelege for å gje opp (Wæge & Nosrati, 2018). Programmering er vanskeleg for Signe og ho er opptatt av resultat i matematikkfaget, men også vise for seg sjølv at «dette» kan ho. Det indikere at ho har mykje ytre motivasjon. Då er det ikkje så rart at ho likar den tradisjonelle matematikken betre enn programmering, sida det ikkje trugar resultatet hennar i matematikkfaget på lik måte.

Dette tyder på at ein som matematikklærer må ein vera ekstra påpasseleg med korleis ein lagar programmeringsopplegget og legg opp undervisinga, då elevane likare det dårlegare enn den tradisjonelle matematikken. Det gjev implikasjonar på at det er i programmeringa ein burde legge ned ein ekstra innsats for å skape motivasjon hos elevane, då dei ikkje har mykje av det sjølv, og programmering kan verke som eit ork for elevane.

For forskingsspørsmålet gjev det implikasjonar på at elevane sine kjensler kring programmering i matematikk, ikkje er positive om ein samanliknar dei med kjensler for tradisjonelle matematikken.

5.3 Seint start med programmering

Elevane meiner dei har starta med programmering for seint i skuleløpet. Dei var einige om at det hadde vore fint å begynt med programmering på ungdomsskulen. Då hadde dei hatt moglegheita til å lærd seg det grunnleggjande før det betydde noko og matematikken var vanskeleg.

Det er ikkje rart at elevane meiner at dei starta seint med programmering. Det er jo det dei har gjort. Den gamle læreplanen (LK06) elevane har følgd til i år har ikkje hatt noko fokus på programmering eller problemløysing. Når dei starta på vidaregåande begynte dei å følgje LK20. LK20 legg opp til at elevane har programmert og utfordra den algoritmiske tenkaren sidan dei begynte på skulen. Allereie på barneskulen skal elevar bli introdusert for programmering (LK20) , mens elevane i denne undersøkinga ikkje vart det. Elevane skal likevel få opplæring og vurdert i kompetansemål som legg til grunne at dei har lært programmering over fleire år.

Det å begynne å lære noko «heilt» nytt på vidaregåande er seint. Igjennom den algoritmiske tenkjaren kjem programmering inn i matematikkfaget. Programmering er difor ein ny måte å arbeide og tenkje på for elevane, det er og ei ny måte å uttrykkje seg på. Ein kan nesten påstå som Peder sa det, programmering er som eit heilt nytt fag.

Dette funnet må bli sett i samanheng med funnet om at elevane synst det er lite tid til programmering. Det vil bli diskutert og framstilt tydleg i del-kapitlet.

Læringsutbytte til elevane i programmering, avhengige mykje av læraren, og i programmering stiller det store krav. Det kan tenkjast at for elevane som startar med programmering på

vidaregåande er meir avhengige av læraren og hen sitt opplegg enn yngre elevar, då dei ofte er mindre nyfikne dess eldre dei blir (Wæge & Nosrati ,2018).

Korttidsminne vårt som me treng for å starte ein læreprosess blir redusert di eldre ein blir. Dersom elevane slit med å kople korttidsminne mot arbeidsminne vil det påverke deira evna til å lære (Busch, 1992). Det kan vera at elevane opplevde at det var lettare å lære noko når dei var yngre, og no krevst det meir av dei. Dette kan vera ein av faktorane til at elevane synest at det er seint å starte med programmering på vidaregåande.

Eg har i denne studia slitt med å finne relevant litteratur til som støttar dette funne. Det verkar ikkje som det skreve noko mykje litteratur om når ein burde starte å lære programmering. Sidan programmering er på full fart inn i grunnskulen i dei fleste vestlege landa, kan ein tolke det som at det er ein oppfatning om at ein burde lære programmering mykje tidlegare enn vidaregåande. Elevane sine utsegn i denne studia støttar opp om denne oppfatninga.

Som lærar dei neste åra bør ein ha eit fokus på at elevane ikkje har programmert like lenge som kompetansemåla (LK20) legg opp til det. Det bør takast omsyn til i planlegginga og gjennomføringa undervisinga og kanskje til og med vurderinga av elevane. Dette fordi og elevane sjølve kjenner på at dei har starta seint, noko som kan skape eit unødvendig press på dei.

Elevane i denne undersøkinga har starta for seint med programmering, nok dei kjenner på. Når ein har starta sein med noko, er det ofte at ein ikkje har hatt god nok tid til å reflektere og analysere. Difor er det truleg at elevane ikkje har reflekter nok over programmering sin rolle i matematikkfaget.

5.4 For lite tid til programmering

Elevane i denne studia synst det har vore lite tid til programmering. Det er nok fleire grunn til og burde absolutt sjåast i samanheng med at dei meiner at dei starta seint med programmering. Det vil bli gjort til i slutten av diskusjonen til dette funnet, først vil eg diskutere andre moglegheita.

Det å starte på vidaregåande er ei stor omvelting for mange, det er ny skule med mange nye medelevar og lærar. Dette i seg sjølv kan krevja mykje frå elevane. Dei fleste elevane som vel å ta IT har lært matematikk lett, og ofte vore nokre av dei beste i klassen sin. Dei er truleg vand til at læraren og resten av klassen bruker lenger tid enn dei treng for å forstå tema og

rekne dei oppgåvene som er forventa gjort. Ein av grunnane til at det er stort hop å gå frå ungdomsskule matematikk til 1T, er at det i 1T går mykje raskare fram enn på ungdomsskulen. Elevane er ofte vand til å ha mykje meir tid til å gjera det som er forventa, dette kan vera ein grunn til at elevane synst det er lite tid.

Elevane bruker mykje tid på idretten sin, mange er og vekke ein del frå skulen på grunn av idretten. Dei brukar nok meir tid på idrett no enn dei gjorde på ungdomsskulen. Har elevane nok tid til å jobbe med skule og då spesifikt matematikk og programmering dersom dei ikkje rekk alt i timane? Matematikk krevst ofte ein del av elevane og mange elevar jobbe med faget heime. Programmering i matematikken har nok førd til at elevane må på rekne meir tid til å jobbe med programmering heime for å komme igjennom alt. Det er godt mogleg at elevane ikkje har denne tida sida dei brukar mykje tid på idrett og difor føler på at det er lite tid.

Elevane i denne undersøkinga har følgd den gamle læreplanen (LK06) til dei begynte på vidaregåande. Noko som betyr at dei har følgd dei kompetansemåla tilhøyrande den gamle læreplanen (LK06). Dette gjere at elevane ikkje har følgd kompetansemåla som ligg til grunne for dei kompetansemåla dei skal bli vurdert i no i år , i 1T. Elevane har «holl» i kva dei skal kunne, då spesielt i emna på programmering og algortimisk tenking. Det kan også vera andre kunnskapshull i matematikken som må tettast. Det tek tid å tette kunnskapshull, kor lang tid det tek avhengige av kor mykje som må tettast. Dette kan vera grunnar til at elevane treng meir tid enn det som er «berekna». Elevane vil då føle at dei er bakpå og treng tid for å ta igjen det dei skulle ha lærd før.

Det er mykje nytt på lita tid for elevane når dei startar med programmering i 1T. Er det for lite tid og då føler elevane på eitt tidspress som gjere at dei føler på eitt stress knytt til programmering. Dette stresse gjere at dei allereie før dei har starte på programmeringa og gjeve programmering ein sjanse, føler på negative kjensler . Kan det tenkjast at elevane hadde vore vennlegare innstilt og positive om dei ikkje hadde følt programmering kan øydeleggja for dei i matematikken, og at det tek tid som dei kunne brukt på tradisjonell matematikk. Elevar treng tid på å prosessere programmering. Denne prosesseringa legg grunnlaget for å få ei djup forståing. Dersom det har vore for liten tid til programmeringa kan det ha påverka elevanes moglegheit til å sette seg nok inn i programmeringa til å kunne like og forstå det.

Elevane synst det å skrive ein kode er utfordrande. Det er ikkje rart då ingen av dei har skrive ein kode før, på nok som bli kalla eit programmeringsspråk. Elevane må lære å bruke eitt nytt

språk, noko som kan vera ein stor barriere i seg sjølv. Eit språk der alt som skjer eller skal utførast av datamaskina må «inn med teskei». Elevane må forklare datamaskina kva dei gjere stegvis og kva som må skje først som sist. Det er nytt og truleg utfordrande for mange. Det er ikkje uvanleg at elevar slit med eller gløymer å vise utrekning til eit vanleg reknestykke. Når dei skal programmere må dei «vise» heile tankerekka si. Difor kan det opplevast som lite tid, då dei må gjer oppgåva grundigare og utfyllande, og samstundes bli klar over kva dei tenkjer.

Oppgåver som er gode programmeringsoppgåver er ganske forskjellig frå mange av dei tradisjonelle matematikkoppgåvene elevane er vand til. Elevane kan ha ein instrumental forståing av matematikk og då løyser matematikkoppgåvene etter ei fast oppskrift. Ei programmeringsoppgåve kan bli løyst på mange måtar og ofte utan noko bestemt oppskrift det er lurt å nytte seg av. Elevane må no bestemme seg kva strategi dei går for, når ein ikkje er vand til det, tek det ofte lenger tid. Det er også ein ny måte å jobbe på, når nok er nytt treng ein ofte lenger tid for å utføre det.

Kor mykje tid det er tenkt til eller kor stor plass programmering skal ha i IT er basert på at elevane har hatt programmering før. Elevane i denne undersøkinga må også lære programmering på den tida det er tenk at elevane skal bruke til programmering for å utforske matematikk. Difor må funnet om at elevane synst det er lite tid og at dei starta sein med programmering sjåast i samanheng. Det som er sikkert er at elevane skulle utifrå kompetansemåla som dei bli vurdert i, starta med programmering før og begynt læringsprosessen mykje tidlegare. Dersom elevane hadde starta med programmering tidlegare hadde dei også hatt meir tid til å lære det.

Læring skjer effektivt når elevane får repetert ein ting fleire gonger. Når ein repetert nok rett etter ein har «gløymt» det, slik at ein får utfordra langtidshukommelsen og det blir gjort fleire gonger, lærer elevane noko (Rawson et.al., 2013). Det elevane lærer på denne måten vil sitte i lang tid. Det betyr at det må vere nok tid i matematikkfaget til at elevane kan lære programmering og i det dei gløymer det repetere det på nytt. Det må då vera tid til fleire repetisjonsøker. Det betyr at ein treng ein del tid for å repetere, noko som det truleg ikkje har vore i matematikkundervisinga til desse elevane.

Å arbeide med tankar over tid kan resultere i at ein mellom anna løyse problem. Ein kan bli betre til å løyse matematiske problem ved å øve og arbeide med matematiske oppgåver og problem («Tenking», u.å.). Kor lang tid elevane brukar på å arbeide seg gjennom tankar

varierer, men dersom elevane ikkje har nok tid til det, kan det resultere i at dei ikkje greier å løyse programmeringsoppgåvene. Sidan elevane uttrykkjer at det har hatt lite tid, kan det tenkjast at dei ikkje har fått arbeida med tankane sine, noko som vil spele inn på kor godt dei får til programmeringa.

Sanner m. fl (2016) anbefale at det burde opprettast eitt eige teknologi fag, der fokuset var programmering og teknologi. Det kan tolkast som at dei meinte at for at elevane skulle få nok tid til å lære programmering trengtest det eitt nytt fag. Eit nytt fag der programmering ikkje var eitt «bi-fokus», men hovudfokuset. Dersom dette er riktig tolking søtar det opp om elevanes si meining om at det er lite tid til programmering i 1T.

Det kan tyde på at dersom elevane hadde hatt meir tid til å lære programmering kunne dei kanskje fått eitt anna inntrykk av programmering. Elevane treng tid til å lære seg programmering, då det er masse nytt og nye tenkje måtar. Det er truleg ikkje nok tid i 1T til at dei skal rekke å «ta igjen» det dei manglar. Programmering vart også anbefalt innført som eitt nytt fag, det kan tolkast som at programmering treng større plass og meir tid enn det har fått i matematikkfaget.

Dette gjere at om ein spør elevar som har starta tidlegare med programmering vil ein kunne få andre svar. Vidare betyr det for forskingsspørsmålet at elevane truleg ikkje har hatt god nok tid til å reflektere rundt programmering. Difor er det truleg at elevane ikkje har reflekter nok over programmering sin rolle i matematikkfaget.

5.5 Programmering er koding

Elevane meiner at programmering er det same som koding. Programmering er mykje meir enn berre koding. Programmering er å utvikla nye program eller vedlikehald av eksisterande program («Programmering», 2019). Den prosessen er mykje meir enn berre koding.

Det er nok fleire grunnar til at elevane meiner programmering er koding. Ein grunn kan vera korleis læraren snakkar om programmering. Når elevane har løyst programmeringsoppgåver har dei alltid tatt opp programmet (Pythonista) som dei bruker for å skrive koden. Elevane kan ha fått oppfatninga om at programmering er koding frå samanhengen mellom at når læraren seier «no skal me programmere, eller no skal me jobbe med programmering» og at Pythonista blir opna. Elevane ha fått ei forståing av at programmering er når dei skriv koden inn i Pythonista, ikkje korleis dei kjem fram til kva dei skal skrive.

Dersom læraren ikkje ha vore bevist i å forklare dei kva programmering og leggje vekt på heile at heile prosessen med å lage eit program er programmering, kan det vere ein grunn til at elevane tenkjer på programmering som berre koding. Ofte når det er snakk om å programmere og om programmering i media og samfunnet blir det vist til at dette er folk som skriv kodar slik at tekniske ting funkar. Fokuset er på at dei skriv koden, ikkje alt arbeidet som ligg i det å finne ut kva dei skal skrive.

Elevane i denne undersøkinga begynte med tekstprogrammering med ein gong. Dei har aldri fått eit bevist forhold til at programmering er meir enn berre å kode. Programmering har ikkje vorte bevist knyta til den algoritmiske tenkjaren for elevane. Dette og at elevane har hatt liten tid til å lære programmering, kan ha ført til at dei har fått ei snever forståing av programmering.

Hermansen (2021) undersøkte kva førestillingar elevane kan ha om programmering i matematikk, hanes informantar gjekk på ungdomsskulen og var med i eitt prosjekt med super:bit. Kva førestillingar elevane hadde om programmering varierte, men i resultatet i undersøkinga si peikar Hermansen på at elevane ikkje har reflektert rundt programmering meir enn det prosjektet elevane gjorde. Det førde truleg til at elevane laga førestillingar om programmering knytt til det temaet og ikkje generelt. Dette kan vera noko av det same elevane i denne undersøkinga gjorde. Elevane i min 1T -klasse har truleg ikkje fått brukt nok tid til å reflektere rundt kva programmering er, dei har mest brukt programmering når dei har koda. Difor kan det vera at deira tankar om programmering at det er koding.

Mykje tyde på at elevane i denne 1T-klassen har lært mest koding når dei har jobba med programmering. Det er ikkje blitt nok fokusert på problemløysing og utvikling av algoritmar. Noko Gjøvik og Torkildsen(2019) framhevar som viktig når elevar skal lære programmering på skulen. Dei meiner målet for elevane bør vera læring av matematikk igjennom programmering. Ikkje berre at elevane kodar sidan det er framtidsretta, då syntaks og språk forandre seg.

Det er ikkje det same å programmere og kode. Læringsutbytte av berre å kode er nok heilt anna enn å programmere, då koding berre er ein liten del av programmering. Om elevane berre kodar får dei ikkje trenna den algoritmiske tenkjaren (Gjøvik & Torkildsen, 2019). Elevane meiner at programmering er koding og ikkje noko meir. Det betyr ikkje nødvendigvis at dei ikkje har eit læringsutbytte av opplegget. Elevane kan få trenna den

algoritmiske tenkjaren utan å vera bevist det, då dei har ei så snever oppfatning av programmering. Liv Hilde peikar at det er prosessen fram til å finne kva ho skal skrive inn, altså formelen eller liknande som er vanskeleg. Dette kan tolkast som at det er problemløysinga som er utfordrande. Den algoritmiske tenkjaren er nok ikkje satt i ein effektiv prosess, så ho har nok ikkje eitt fullt utbytte frå aktivitet, men ho får forhåpentlegvis trene problemløysaren og den algoritmiske tenkjaren i seg.

Dette kan gjere at elevane ikkje har fullt potensial for å sjå kva programmering kan bidra med i matematikkfaget, og kva rolle det har der. Sidan dei berre ser eit ufullstendig bilde av programmering, vil dei heller ikkje greie å sjå eit fullstendig bilde av rolla programmering kan ha i matematikkfaget. Om elevane hadde hatt eit større bilde på kva programmering er, ville dei då svara annleis i intervjuet, og ville funna til forskingsspørsmålet *elevane sine tankar kring programmering si rolle i matematikk* vore annleis? Og ikkje minst hadde det fått konsekvensar for konklusjonen ?

Elevane si oppfatning av at programmering er koding er ikkje uventa. Særleg dersom læraren ikkje har greidd å rettleie godt nok på heile prosessen, og fokuset i timane har vore på korleis dei skal skrive koden. Det er ikkje dermed sagt at dei berre kodar og ikkje utfordrar sin algoritmiske tenkjar når dei jobbar med programmeringsoppgåver. Læraren har truleg ikkje fått elevane til å reflektere nok rundt kva programmering er og korleis prosessen er.

5.6 Programmering er matematikk fordi det brukast matematiske funksjonar

Elevane i denne undersøkinga meiner programmering er matematikk fordi det blir brukt til å arbeide med matematiske funksjonar. Kva legg elevane i dette og kvifor meiner dei det?

Matematiske funksjonar kan vera mykje, men elevane sa i same andedrag som matematiske funksjonar, formlar og tall. Det eit spørsmål om elevane skil mellom den matematikken dei brukar når dei løyser programmeringsoppgåva og den matematikken som dei må bruke for å programmere. Det er to forskjellige spørsmål, kva matematikk er det egentleg i programmering og kva matematikk kan løysast ved programmering. Det er vanskeleg å sei kva elevane egentleg meiner, om dei sjølv ser at det er to forskjellige spørsmål, eller om at det er to sider av same sak.

Matematikken ein må kunne bruke for å programmere er ganske enkel, dette er matematikk som burde vere grei for elevane å bruke. Sidan elevane truleg kan denne matematikken burde dei greie å uttrykkje kva matematikk det er. Difor kan det tolkast som at elevane ikkje ser at programmering er bygd på ganske enkel matematikk. Det er ikkje så rart då elevane har aldri fått ei innføring i kva og korleis programmering eigentleg fungera, dei har berre brukt programmering. Difor er det ikkje uventa om dei ikkje har tenkt over og ser kva matematikk som eigentleg trengs i programmering.

Elevane har brukt programmering til å løyse forskjellige oppgåver i matematikkfaget, problemløysingsoppgåver og meir reine matematikk oppgåver som andregrads funksjonar. Elevane kan då lett binde matematikken i programmering til kva type oppgåver dei bruke programmering til, altså kva matematikk som kan løysast ved programmering. Dersom dette er tilfellet er det ikkje rart at dei meiner programmering er matematikk. Som Tore seier, han brukar programmering i matematikk, og der har det ein nytte, men ikkje i for eksempel norsk. Dette tyder på at elevane i denne undersøkinga forbind programmering og kva programmering kan bli brukt til med dei områda dei brukar programmering på .

Er dette eigentleg negativt? Ein av grunnane til at programmering skal inn i matematikk er for å bruke programmering til å lære og få ein betre forståing av matematikk. Kva matematikken det er programmering er ikkje viktig, men korleis ein brukar matematikken er viktig for å løyse problema ein programmere.

Hermansen (2021) ser at elevane sine tankar om programmering i matematikk, er knyt til kva elevane har brukt programmering til. Dette er spesielt om elevane ikkje har fått reflektert rundt programmering i matematikken. Elevane i denne undersøkinga har brukt programmering til å løyse forskjellige matematiske oppgåver. Dei har heller ikkje hatt mykje tid til å reflektere over kva programmering er. Det er då venta at om elevane forbind programmering med funksjonar og formlar fordi det er det dei har brukt programmering til.

Dersom det programmeringa blir for teknisk er det ein moglegheit for at elevane ikkje skjønner samanhengen mellom programmering og matematikk. Programmering skal vera eit verktøy for å bli betre problemløysarar innan matematikk (Kaufmann & Stenseth, 2021). Sidan elevane i denne undersøkinga ser at det er ein samanheng mellom programmering og matematikk, tydar det på at programmeringa ikkje er blitt alt for teknisk.

Programmering kan brukast som eit verktøy for å auke forståinga i matematikk ved å løyse og utforske problem. Det å sjå samanhengar mellom matematikk og andre fag legg til rette for djupnelæring og forståing (Kunnskapsdepartementet, 2020). Elevane skal bruke programmering i matematikk som eit verktøy for å løyse problem, då det skal gjev dei ei djupar forståing av matematikken. Difor er det kanskje ikkje så viktig at elevane ikkje ser forskjellen på kva matematikk som blir brukt i programmering, og kva matematikk ein kan løyse ved hjelp av programmering.

Verstad(2017) undersøkte kva elevar som hadde programmering som valfag meinte om programmering og matematikk. Det var varierende kva elevane såg og meinte, nokre av elevane meinte at matematikk hjelp i programmering, men programmering hjelp ikkje i matematikk. I Verstad(2017) studia er det fleire av elevane som synst det er vanskeleg å sjå at programmering skal vera til hjelp i matematikken. Dette er litt avvikande frå kva elevane i mi studie meiner, dei meiner programmering er matematikken. Elevane i mi undersøking har då brukt programmering veldig lite i matematikk, og generelt. Det kan tenkjast at elevane i Verstad(2017) si studie har meir erfaring, så dei har møtt på meir motstand ved programmering enn elevane i denne undersøkinga.

Det verkar som elevane forbind programmering med det dei bruker det til. Elevane har brukt programmering i matematikk og til å løyse forskjellige matematikk oppgåver. Dei forbind då programmering med matematiske funksjonar og meiner difor programmering er matematikk. Det er litt vanskeleg å sei om elevane meiner dette fordi dei ser at programmering nyttar seg av matematikk, eller om det er fordi dei har brukt programmering i matematikkfaget. I det store bilde er det kanskje ikkje så viktig, då programmering skal bli brukt for å få ein betre matematisk forståing.

5.7 Alt kan løysast med programmering

Elevane meiner at det er naturleg å bruke programmering til å løyse det meste i matematikk. Det er jo ikkje direkte feil, ein kan programmere dei fleste av matematikkoppgåvene og matematiske problema elevane møter på. Men er det naturleg å skulle programmere heile tida når dei skal jobbe med matematikk?

Kva problem er det eigentleg lurt å bruke programmering til? Når ein ser det frå eit matematisk perspektiv er det utrekningar som er «enkle», men tar lang tid å gjera sjølv. Dette kan vera for å teste hypotesar, sjå etter motargument eller å bygja modellar. Elevane som blei

intervjua fekk spørsmål om å gje konkrete eksempel på når det gav mening å løyse oppgåver med programmering. Dette greidde dei alle, men jentene greidde å generalisere svaret sitt. Dette bør ein nok sjå i samanheng med at i timen før hadde elevane jobba med oppgåver som det var lurt å programmere for å løyse. Elevane hadde då desse oppgåvene friskt i minne og hadde nok ein oppleving at det var enklare å løyse oppgåvene ved programmering.

Signe er den som utelukka nokre tema eller oppgåver som det ikkje gjev mening å programmere, dette gjere ho etter spørsmål om ho verkeleg meiner at alt me har jobba med i matematikken i år kunne me ha programmert. Ho meiner at det gjev mening å programmere førstegradslikningar, men ikkje andregrads eller likningar av høgare grad. Eg får eitt inntrykk av at ho trekk litt ting ut frå lause lufta. Elevane i denne klassen har programmert andregradslikningar, men ikkje førstegrads. Det kan tenkjast at ho svarer det fordi spørsmålet ho får kan verke ledane og ho føler ho berre må sei noko.

Kan det vera slik at elevane tenkjer at alt kan løysast ved programmering fordi dei brukar produkt som er programmert for å løyse matematiske oppgåver. Mobilen og kalkulatoren er produkt som er programmert, og dei bruker desse flittig i matematikk og elles. Mange av matematikkoppgåvene i 1T blir mykje lettare ved bruk av kalkulator. Difor tenkje elevane at det meste blir jo lettare med produkt som er programmert difor er det naturleg å løyse det meste med programmering. Altså at dei tenkjer indirekte bruke av programmering.

Elevane er vant til matematikk som er instrumental. Resonnering, logisk slutningar og problemløysing som har større fokus i den nye læreplanen (LK20), er matematikk som ikkje er så enkelt å løyse med programmering. Elevane har ikkje mykje trening i desse emna. Det kan difor vere at elevane meiner all matematikk kan løysast med programmering.

Mange matematiske utrekningar kan løysast ved hjelp av programmering, men alt av matematikk kan ikkje løysast ved programmering. Matematiske bevis greier ikkje datamaskiner å løyse aleine, det er dei logiske slutningane datamaskina ikkje klarer (Gómez-Serrano, 2019). Det meste av matematikken elevane i denne studia har vore borti er instrumell matematikk. Dette er då matematikk som ein kan bruke programmering til å løyse. Difor er det godt mogleg at elevane ikkje tenkjer på matematiske bevis, resonnering og logisk tenking når dei tenkjer på matematikk. Det gjere at elevane i denne undersøkinga meiner at all matematikk kan løysast ved programmering.

Verstad(2017) fann i sin master at elevane synst matematikk er nyttig når dei skal programmere, men programmering ikkje ei hjelp i matematikken. Elevane i hannes undersøking meinte også at i matematikk er det ein måte for å løyse oppgåvene, men i programmering må ein forhalda seg til fleire måtar å løyse oppgåver på. Elevane i denne undersøkinga er også einige i at det er fleire måtar å løyse programmeringsoppgåver på enn matematikkoppgåver. Elevane i denne undersøkinga meiner at programmering kan løyse det meste av matematikk, og er eitt hjelpemiddel. Er det fordi elevane i Verstad(2017) si studie hadde valt programmering som valfag og så på det dei gjorde som programmering, mens elevane i mi studie har matematikk og ser på det dei gjere som matematikk, at det er ein forskjell? Elevane i begge studiane er relativt unge og truleg ikkje reflektert mykje over kva og kvifor dei lærer ting. Difor kan det som avgjere kva elevane tenkjer på, når dei høyrer matematikk eller programmering er avhengig av kva fag dei har brukt sin kunnskap i.

Funnet gjev eitt inntrykk av at elevane ikkje ser programmering sine avgrensingar eller matematikk sin kompleksitet. Ein heilt anna moglegheit er at elevane har tenkt at sidan dei brukar produkt som er programmert til det meste i kvardagen og kan alt løysast ved programmering. Som lærar bør ein med fordel vera tydlegare på moglegheitene, men og avgrensingane som ligg i programmering.

Kva elevar meiner programmering kan brukast til, har mykje å seie for kva rolle dei meiner programmering bør ha i matematikkfaget.

5.8 Sjølvmotsetningar i elevar sine meningar om programmering

Elevane som vart intervju i denne undersøkinga hadde fleire sjølvmotseiingar. Dette kom ikkje fram i spørjeundersøkinga som fleire av elevane svarde på, men sidan elevane som vart intervju representere alle nivå i kunnskap kan det tenkjast at det er representativt for dei fleste elevane i denne klassen.

Det at elevane har utsegn som seier mot seg sjølv er interessant av fleire grunnar, og det er nok fleire grunnar til at det er slik. Desse motseiingane gjere at ein bør stille spørsmålet, men der er ikkje sikkert at det gjere validiteten dårleg. Elevane uttaler sjølv at dei kan lite programmering, synst det er vanskeleg og har hatt lite tid til å lære programmering. Dette kan bety at elevane har så store kunnskapshull at dei ikkje ser at det er motseiingar i det dei seier.

Det er ein måte å tolke Signe, når ho blandar inn tema og omgrep som me nesten ikkje har brukt i programmering.

Når elevar har store kunnskapshull er det ofte vanskeleg å sjå samanhengen i kunnskapen ein innehar og då er det vanskeleg å sjå at det ein meiner og seier er motstridande. Det kan verke som elevane i denne undersøkinga har ein oppstykkka kunnskap om matematikk og programmering, som gjere at dei ikkje ser kva programmering eigentleg gjere i matematikkfaget. Dei fleste av elevane synst programmering gjere matematikk vanskelegare, og ingen har sagt at det gjere matematikk enklare. Nokre sa at det var artig å gjera nok litt annleis.

Programmering er tiltenkt som eit verktøy og hjelpemiddel, når nok skal vera eit verktøy må ein anta at det skal bli lettare når ein tek det i bruk. Ein grunn til at elevane ikkje ser på programmering som eit verktøy og då ein naturleg del av matematikkfaget er at dei ikkje greier å kople dei to område i hop til eitt. Dersom dei oppfattar det som er to forskjellige områder kan det lett oppstå motstridane tankar og meiningar utan at ein ser det.

Ein anna moglegheit er at elevane skil mellom den programmeringa dei gjere og det som andre gjere. Eg tolkar at når Tore seier at alt kan og bør løysast ved programmering, så meiner han eigentleg ikkje at alt han gjere bør løysast ved programmering. Det han meiner at alt enklare kan løysast med programmering som hjelpemiddel. Ikkje at all matematikk han skal løyse bør løysast ved programmering. Det er fordi Tore gjerne brukar eit halvt minutt lenger på ei oppgåve i staden for å programmere. Oppgåva han ville bruke eitt halvt minutt lenger på var ei ganske typisk programmeringsoppgåve (sjå vedlegg 3).

Det kan også tenkjast at elevane skil mellom kva som burde gjerast og kva dei sjølv vil gjere, når dei svare. Det kan vera at dei ser kva som er smartast å gjere, men ynskjer ikkje å gjere det sjølv og difor blir det oppfatta som sjølvmotseiingar.

Det er ikkje forska mykje på førestillingar elevar har om programmering i matematikkfaget. Eg har difor ikkje funne noko forskning som finn at elevar har førestillingar og meiningar som er sjølvmotseiande om programmering i matematikkfaget.

At elevar har sjølvmotseiingar i meiningane sine om programmering, tyder truleg på at det er nokre kunnskapshull som gjere at det er mogleg. Det kan og vera eit konsekvens av fleire av dei tidlegare funne og då spesielt at dei har hatt lite tid. Det betyr at dersom elevane hadde fått

meir kjennskap og blitt meir komfortable med programmering, kunne kanskje desse sjølvmotseiingane forsvunne.

6 Konklusjon

For å komme den digitale teknologiske revolusjonen i møte er det i den nye læreplanen (LK20) innført programmering. Meir og meir av samfunnet vårt basere seg på produkt som treng å bli programmert. For at alle skal kunne delta i samfunnet, bør ein ha ei forståing programmering. Programmering er innført som eitt tverrfagleg tema, men hovudtyngda er lagt til matematikk saman med algoritmisk tenking. Etter å ha undervist i vidaregåande skule parallelt med studiet mitt, har eg fått eitt inntrykk av at elevane kanskje ikkje har vore så positive til dette.

Med bakgrunn i den tanken, ville eg undersøkje om det stemmer og utforma to forskingsspørsmål. Der føremålet var å finne ut kva elevane tenkjer og kjenner på kring at det plutsleg er blitt programmering i matematikken? Dei to forskingsspørsmål som vart til, er:

1. *Kva er elevar sine kjensler knytt til programmering i matematikkfaget*
2. *Kva er elevar sine tankar kring programmering si rolle i matematikkfaget.*

For å kunne svare på desse spørsmål har eg brukt ein liten 1T-klasse som informantar. Der har eg sett på elevar sitt resultat i matematikk og programmering. Vidare har elevane svara på spørjeundersøkingar etter prøvar og til slutt intervjuet eg tre elevar for å få ei betre og djupare forståing av kva dei tenkjer. Sidan dette er elevar eg har undervist i 1T, har eg også observert dei når dei arbeider og eg har blitt kjent med dei. Dette gjer at eg har større kjennskap til elevane enn undersøkinga syner, det har truleg prega mine tolkingar i diskusjonen og konklusjonen.

Konklusjonane og kva dei implisere må sjåast i lys av kva elevar som var informantar i denne undersøkinga. Dette er elevar som er har valt 1T matematikk og følgd LK06 fram til dei starta på vidaregåande. Difor vil elevane som startar på vidaregåande i dei neste åra ha anna kunnskap, nok som kan prega deira meiningar og kjensler. Elevane i denne undersøkinga har valt 1T-matematikk. Dei vil ha ein vanskelegare matematikk, ein må difor kunne tru at elevane likar matematikk. Dette og at det er eit avgrensa tal, relativt homogene, elevar som er intervjuet gjere at overføringsevna og kor mykje det kan generaliserast til andre elevar og klassar truleg er avgrensa.

Den første konklusjonen er at den brå innføringa av programmering for elevar på vidaregåande som ikkje har programmert før gjere at mange elevar ikkje likar

programmering, fordi dei ikkje forstår det. Dei har ikkje den bakgrunns kunnskapen som trengst for at dei skal få det til. I tillegg skal elevane blir vurdert i programmering, kanskje som om dei har kunnskap om programmering frå grunnskulen, dette gjere at fleire føler at programmering blir eit hinder for å gjere det bra i faget. Dette ser ut til å vere viktige grunnar til at dei fleste av elevane i denne studia var negative til programmering. Alle elevane i denne undersøkinga hadde også lågare motivasjon for programmering enn for den tradisjonelle matematikken. Ein kan difor undrast om programmering gjer at motivasjon for matematikkfaget generelt blir dårlegare.

Den andre konklusjonen er at elevane ville truleg likt eller vore positiv til programmering om dei hadde forstått det betre. Eg finn at elevane meiner at dei har starta seint og har lite tid til programmering, det kan tolkast som dei kunne ynskja å lært meir programmering tidlegare. Elevane har ei mangelfull forståing av kva programmering er, og korleis ein kan og bør bruke det. Difor greier ikkje elevane å kople programmering til den algoritmiske tenkjaren og sjå på programmering som ein problemløysingsmetode. Det gjev belegg for å seie at programmering har ikkje oppnådd det som var målet for innføringa.

Den tredje konklusjonen er at elevane ikkje har tenkt over eller reflektert rundt rolla programmering har i matematikkfaget. Elevane meiner likevel at det er naturleg å ha programmering i matematikkfaget, det gjev meining for dei. Dette er samstundes som mange av dei ikkje vil ha programmering som ein del av matematikkfaget. Dette tydar nok på at elevane ser ei samanheng mellom matematikk og programmering, men likevel ikkje likar programmering som ein del av matematikkfaget.

Det gjev implisere at skal elevane få ei positiv oppfatning av, og noko igjen for å jobbe med programmering i matematikkfaget er det viktig at dei opplever meistring og har nok tid. Ut frå samtalar eg har med elevane gjennom året oppfattar eg at for elevar på vidaregåande som ikkje har tidlegare erfaring og kunnskap i programmering, innebere det også at dei må sleppe å tenkje på at dei skal bli vurdert i programmering, då mange er opptekne av resultat i matematikkfaget. Dette kan då bli eit stressmoment som gjere programmering vanskelegare. Elevane må få moglegheit til å bli kjent med kva programmering, og at det er meir enn berre å kode oppgåver. Truleg må fokuset då vera meir på den algoritmiske tenkjaren enn at dei skal lære tekst-programmering. Dei må få moglegheita til å utforske kva programmering er og korleis det blir brukt i den verkelege verda.

Dersom elevane får eitt større bilde av kva programmering er, vil dei nok vera i stand til å reflektere meir rundt programmering si rolle i matematikkfaget. Dette sidan dei då ville greie å sjå fleire konkrete eksempel når det gjev mening og når det ikkje gjev mening å bruke programmering. Det vil då forhåpentlegvis opplevast som meir meningsfullt for elevane at programmering er ein del av matematikkfaget.

Studia gjev likevel nokre implikasjonar eg kjem til å ta med meg ut i læraryrket.

Programmering krev noko anna av elevane enn den vanleg instrumentale matematikken som dei er vand med. Det gjere truleg at det kan vera vanskelegare å forstå programmering med den bakgrunn elevane har og difor kan gjera at dei har dårlegare motivasjon for programmeringa samanlikna med den tradisjonelle matematikken. Elevar har eitt inntrykk av programmering er koding, noko som gjere det truleg tørt og utfordrande for elevane. For å imøtekomme denne utfordringa er det nok viktig å la elevane få nok tid til å bli komfortabel med grunnprinsipp i programmering og gje dei meningsfulle oppgåver der fokuset er på meir enn sjølve kodinga. Korleis ein skal løys oppgåva og problemet burde vera like viktig som kva som står i koden.

6.1 Veggen vidare

Denne studia gjev eit augeblikk bilde av korleis situasjonen er i dag, i overgangen mellom den gamle læreplanen (LK06) og den nye læreplanen (LK20). Det betyr at om nokre år vil denne studia vera utdatert fordi elevane som begynner på vidaregåande og vel 1T matematikk har eitt heilt anna grunnlag i programmering og problemløysing. Difor ville det å gjennomføre ei likande studie om nokre år vore veldig interessant.

Denne studia er liten og gjort som ei kvalitativ studie på ein relativt homogen elevmasse. Det hadde vore spennande å gjort eit kvantitet undersøking på fleire skular, med fleire elevar. Om ein også kunne ha samanlikna 1T og 1P elevar sin oppfatning og tankar kring programmering, kunne ein fått eit klarare bilde på om matematikk-kunnskapen til elevane har mykje å sei for kjenslene og oppfatninga av programmering i matematikkfaget.

7 Referanselise

Algoritmisk tenking (u.å.) I *Wikipedia*. Henta 20.mai 2022

https://no.wikipedia.org/wiki/Algoritmisk_tenkning

Berg, T.K. (2021). Analog programmering. *Tangenten – tidsskrift for matematikkundervisning*, 32(3), 42–52

Bueie H. (2019) *Programmering for matematikklærere* . Universitetsforlaget

Busch T. (1992) *Overføring av Læring*. [Doktorgradsavhandling, Norges Handelshøgskole]

https://openaccess.nhh.no/nhh-xmlui/bitstream/handle/11250/162443/Busch_1992.pdf?sequence=1

Eliassen J. E, Bøe M. V., Nederbragt L., Berg M.M. & Gregers T.F. (2022) Motivasjon for beregningsorientert biologi og sammenhengen med matematikk R2 frå videregående opplæring. *Nordic Journal of STEM Education*, 4(2).

<https://doi.org/10.5324/njsteme.v4i2.4026>

Forsström S.E. & Kaufmann O.T. (2018) A literature review exploring the use of programming in mathematics education. *International journal of learning, teaching and educational research*, 17(12), pp 18-32

Gjøvik, Ø., Torkildsen, H. A. (2019). Algoritmisk tekning. *Tangenten – tidsskrift for matematikkundervisning*, 30(3). 31–37.

<http://www.caspar.no/tangenten/2019/Tangenten%203%202019%20Gj%C3%B8vik%20Torkildsen.pdf>

Gleiss M.S. & Sæther E. (2021) *Forskningsmetode for lærerstudenter: Å utvikle ny kunnskap i forskning og praksis*. Cappelen Damm Akademisk

Gómez-Serrano, J. (2019) Computer-assisted proofs in PDE: a survey. *SeMA* 76, 459–484

<https://doi.org/10.1007/s40324-019-00186-x>

Haraldsrud A. D., Sveinsson H. A., Løvold H.H (2020) *Programmering i skolen*, Univerisitetsforlaget

Hermansen S.S.(2021) *Hvilke forestillinger kan eleven ha om programmering i matematikk*, [Masteroppgåve, Norges Arkitske Universitet]. Munin.

<https://munin.uit.no/bitstream/handle/10037/22465/thesis.pdf?sequence=2>

Hovde, K.O., Grønmo, S.(2020). Algoritme i *Store norske leksikon* Henta 25. mars 2021 frå

<https://snl.no/algoritme>

Jacobsen D.I (2015) *Korleis gjennomføre undersøkelser?: Innføring i samfunnsvitenskapelig metode* (3.utg.) Cappelen Damm Akademisk

Jansen J.K.S (2019) Følelser I *Store norske leksikon*. Henta 19.mai 2022.

<https://snl.no/f%C3%B8lelse>

Kaufmann O.T. & Stenseth B. (2021) Programming in mathematics education. *International Journal of Matematical Education in Science and Techology*, 52(7), 1029-1048.

<https://doi.org/10.1080/0020739X.2020.1736349>

Kilhamn C., Rolandsson L, Bråting K. (2021) Programmering i svensk skolmatematik: Programming in Swedish school mathematics. *LUMAT: International Journal on Math, Science and Technology Education*, 9(1), 283–312.

<https://doi.org/10.31129/LUMAT.9.2.1457>

Kunnskapsdepartementet (2019). *Læreplan i matematikk (MAT01-05)*. Fastsett som forskrift. Læreplanverket for Kunnskapsløftet 2020

Kvale S. & Brinkmann S. (2015) *Det kvalitative forskningsintervju* (3.utg.) Gyldendal

Nosrati, M., & Wæge, K. (2015). Sentrale kjennetegn på god læring og undervisning i matematikk. *Hentet, 10, 2017*. <https://utdanningsforskning.no/artikler/2015/sentrale-kjennetegn-pa-god-laring-og-undervisning-i-matematikk/>

Nyutstumo S. (2021) *Programmering og problemløsning i matematikk* [Masteroppgåve, Noregs teknisk-naturvitenskapelige universitet]

Pacheco, A., Gomes, A., Henriques, J., de Almeida, A. M., & Mendes, A. J. (2008, June). Mathematics and programming: Some studies. In *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing* (pp. V-15). <https://doi.org/10.1145/1500879.1500963>

Postholm, M.B & Jacobsen, D.I (2018) *Forskningsmetode for masterstudenter i lærerutdanning*. Cappelen Damm Akademisk

Programmering. (2019) *I Wikipedia*. Henta 25. mars 2021 frå

<https://no.wikipedia.org/wiki/Programmering>

Programmeringsspråk (u.å.) *I Store norske leksikon*. Henta 28 mai 2022 frå

<https://snl.no/programmeringsspr%C3%A5k>

Rawson, K.A., Dunlosky, J., Sciartelli, S.M. (2013) The power of successive relearning: improving performance on course exams and long-term retention. *Educational Psychology Review* 25(4),

Rossen, E. (2019) programmering - IT *i Store norske leksikon* Henta 23.mars 2022

https://snl.no/programmering_-_IT

Rossen, E., Dvergsdal, H., Hannemyr, G., Busch, P. A (2022) Datamaskinens historie *i Store norske leksikon*. Henta 20.mai 2022 https://snl.no/datamaskinens_historie

Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., ... & Voll, L. O. (2016). Teknologi og programmering for alle-En faggjennomgang med forslag til endringer i grunnopplæringen-august 2016. <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2414205/teknologi-og-programmering-for-alle.pdf?sequence=3>

Sevik K. m. fl (2016) Programmering i skolen, Senter for IKT i utdanning.

https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Svartdal F. (2020) Læring. *I Store norsk leksikon*. Henta 24. mai 2022

<https://snl.no/l%C3%A6ring>

Teigen, K.A. Alnes, J.A. (2022) Mening. *I Store Norske Leksikon*. Henta 20. mai 2022

<https://snl.no/mening>

Tenkning(u.å.) *I Store Norske Leksikon*. Henta 25. mars 2021 frå <https://snl.no/tenkning>

Utdanningsdirektoratet (2019) *Algoritmisk tenking* Henta 18. januar 2022 [Algoritmisk tenking \(udir.no\)](https://www.udir.no/Algoritmisk-tenkning)

Utdanningsdirektoratet (2020) *Hva er nytt i matematikk ?* Henta 14. januar 2022
<https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/>

Utdanningsdirektoratet (2021a) *Hvorfor har vi fått nye læreplaner,* Henta 14. januar 2022
<https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvorfor-nye-lareplaner/>

Utdanningsdirektoratet (2021b) *Slik ble læreplanene utviklet,* Henta 14. januar 2022
<https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/slik-ble-lareplanene-utviklet/>

Verstad A. (2017) *Programmering + matematikk = sant?* [Masteroppgåve, Norges Arktiske Universitet]

Wæge, K. (2007). *Elevenes motivasjon for å lære matematikk og undersøkende matematikkundervisning* [Doktorgradsavhandling, Noregs tekniske-naturvitenskapelige universitet] <http://hdl.handle.net/11250/258129>

Wæge K. & Nosrati M.(2018) *Motivasjon i matematikk.* Universitetsforlaget AS

Zwick U. (2002) Computer assisted proof of optimal approximability results. In *SODA* (pp. 496-505)

8 Vedlegg

8.1 Samtykkeskjema

Vil du delta i forskingsprosjektet

«Bruk av programmering i matematikk»?

Dette er eit spørsmål til deg og dine føresette om du vil delta i eit forskingsprosjekt der hensikta er å undersøkje korleis elevar vel å bruke programmering for å løyse matematiske oppgåver. I dette skrive skal gje deg meir informasjon om prosjektet og kva deltakinga innebere for deg.

Hensikt

Hensikta med dette forskingsprosjektet er å undersøkje om elevane sjølve vel å bruke programmering i matematikk for å løyse oppgåver og problem. Grunnen for dette forskingsprosjektet er mi master oppgåve på studiet lektor i realfag.

Kven er ansvarlege for forskingsprosjektet?

Institutt for matematikk og statistikk ved fakultet for naturvitskap og teknologi ved UiT er ansvarleg for forskingsprosjektet.

Rettleiar for masteren er Jan Nyquist Roksvold ved institutt for lærarutdanning og pedagogikk.

Kvifor får du spørsmål om deltaking?

Då eg jobbar 50% ved NTG som lærar i blant anna 1T, ynskjer eg å bruke elevane i denne klassen for å samle inn data til oppgåva mi.

Kva innebere det for deg å delta?

Dersom du vel å delta i prosjektet vil eg analysere oppgåver på prøvar og innlevering som elevane har som ein del av undervegsvurdering i faget i hovudsak fram til jul. Dei oppgåvene som blir analysert er dei som er relevante med tanke på programmering. Eg vil og sjå på om det er noko samanheng mellom nivået til elevane, basert på resultatet frå prøvane/innlevering og val av bruk av programmering.

Det vil og vera eit spørjeskjema med fire spørsmål som er frivillig å svare på etter kvar prøve/innlevering.

Eitt eller to intervju der lyd vil bli tatt opp kan og vere aktuelt med nokre få av elevane. Intervjuet vil ha som føremål å finne ut kvifor eleven gjorde dei vala den gjorde på dei aktuelle oppgåvene som eg skal analysere på prøven. Det blir valfritt å gjennomføre intervjuet dersom du vil vere med på forskingsprosjektet.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Dersom du vel å delta, kan du når som helst trekkja samtykket tilbake utan noko grunngjeving. Alle dine personopplysning vil då bli sletta frå

prosjektet. Det vil heller ikkje ha noko negative konsekvensar for deg om du vel å ikkje delta eller seinare vel å trekkja deg.

Sidan prøvane og innleveringane som skal bli analysert er undervegsvurdering i faget, må alle gjennomføre desse uansett deltaking i prosjektet eller ei. Forskjellen er om dei blir brukt i mi masteroppgåve eller ei.

Ved eventuelt intervju vil dette skje når det passar for elev og lærar, men utan for vanleg undervising. Så eleven vil ikkje miste noko undervising.

Ditt personvern- korleis me oppbevare og brukar dine opplysningar

Me vil berre bruke opplysningane om deg til føremåla me har fortalt om i dette skrivet. Me behandlar opplysningane konfidensielt og i samsvar med personvernregelverket.

Det blir berre eg som har tilgang til personopplysningar. Dersom prøvar/spørjeskjema og resultat på prøvane blir delt med rettleiar vil alle personopplysningar vera fjerna og prøvane anonymisert. Prøvane og spørjeskjema vil bli anonymisert ved å gi nr til kvar deltakar, dokumentet som koplar nr og namn vil ligge saman med originalprøvane på NTG.

Det vil ikkje vera mogleg å gjenkjenning deltakarane igjen i publikasjonane. Deltakarane kan kanskje kjenne igjen sitt eige arbeid dersom det blir brukt som eksempel i publikasjonen.

Ved lydopptak vil det bli transkribert og anonymisert med det same nummeret som eleven allereie har.

Kva skjer med opplysningane dine når prosjektet blir avslutta?

Opplysningane anonymisert når prosjektet avsluttast / oppgåva er godkjent, noko som etter planen er 1.juni 2022. Då vil dokumentet som koplar nr og namn bli makulert, og eventuelle lydopptak sletta.

Dine rettigheter

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

- Innsyn i korleis personopplysningar som er registrert om deg, og få utlevert ein kopi av opplysningane
- Å få retta opp personopplysningar om deg
- Å få sletta personopplysningar om deg
- Å sende klage til Datatilsynet om behandling av dine personopplysningar

Kva gje oss rett til å behandle personopplysningar om deg?

Me behandlar opplysningar om deg basert på ditt samtykke.

På oppdrag frå IMS har NSD- Norsk senter for forskningsdata AS vurdert at behandlinga av personopplysningane i dette prosjektet er i samsvar med personregelverket.

Korleis kan eg finne ut meir?

Dersom du har spørsmål til studien, eller ynskjer å benytte deg av dine rettigheitar, ta kontakt med:

- Olaug G. Andersen på [oan043@uit.no](mailto: oan043@uit.no) eller på mobil 95090563
- Jan Nyquist Roksvold ved ILP på [jan.n.roksvold@uit.no](mailto: jan.n.roksvold@uit.no) eller +4777646141

Dersom du har spørsmål knytta til NSD [si](#) vurdering av prosjektet, kan du ta kontakt med:

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

Med vennleg helsing

Jan Nyquist Roksvold
(Rettleiar)

Olaug G. Andersen
Student

Samtykkeerklæring

Namn på elev:

Me har mottatt og forstått informasjon om prosjektet Bruk av programmering i matematikk og har fått anledning til å stille spørsmål. Eg samtykker til:

- mine prøvar og innleveringar kan analysert
- å delta med spørjeskjema i samband med prøvar/innlevering
- å delta i intervju

Eg samtykker mine opplysningar behandlast frem til prosjektet er avsluttet

(Signert av prosjektdeltakar, dato)

8.2 Intervjuguide

Intervjuguide

Innleiande spørsmål:

Har du programmert før du begynte på VGS?

Kva matematikk kjem du til å velja til neste år?

Kva er ditt mål med matematikk med å ta T-matematikk, både i eit kort og lenger perspektiv

Spørsmål:

1. Korleis vil du definere kva programmering er ?
2. Kva synst du om programmering ?
3. Kva er din motivasjon for å lære programmering?
4. Synst du programmering er matematikk, kvifor?
5. Kvifor trur du at programmering er blitt ein del av matematikkfaget?
6. Kva tenkjer du om at programmering er blitt ein del av matematikkfaget ?
7. Har du eksempel på når det kan vera lurt å nytte seg av programmering?
8. Når gjev det ikkje meining å bruke programmering i matematikk?
9. Kva er utfordringane med programmering og programmeringsoppgåvene i matematikk?
10. Kva er det enklaste med å løyse programmeringsoppgåvene i matematikkfaget?
11. Synst du det er nok tid til programmering i matematikk?
 - a. Til å lære
 - b. Til å bruke og utforske
12. Kva er det temaet/ emne du likar best i matematikk, og kvifor gjer du det?
13. Kan du samanlikne det temaet med programmering, er det noko likheitstrekk?
14. Korleis er din motivasjon/ interesse for matematikk samanlikna med programmering ?
15. Er det noko ved programmering som har overraska deg?
16. Er det noko som du har lyst å tilføre til dette intervjuet når du blir mint på at det eg skal undersøkje i min master er « Elevar sine tankar om programmering si rolle i matematikkfaget?»

8.3 Utforskningsoppgåver timen før intervju

Oppgåve 1

a) Finn summen av talla frå 1 til 10.

$$1+2+3+4+5+6+7+8+9+10 = ?$$

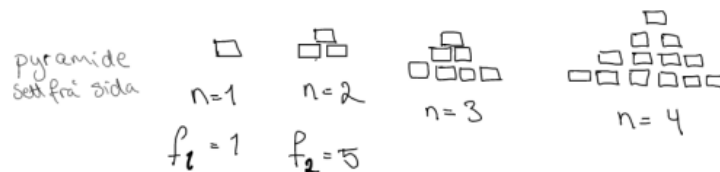
b) Finn summen av talla frå 1 til 100

c) Finn summen av talla frå 1 til 1000

Oppgåve 2

Du skal bygge ein firkant pyramid.

Skisse av
firkant pyramid



F_n er antall blokke i pyramiden og, der n er antall etasjar

- A) Kor mange blokke treng du om du pyramiden skal vere 10 etasjar høg
B) Dersom du har 1000 blokker, kor mange etasjar kan du bygge ?

Oppgåve 3

Finn dei fire perfekte tall. Eit tall er eit positivt heiltal som er lik summen av alle faktorane sine. 1 reknast som ein av faktorane, men talet sjølv reknast ikkje med.

Det første perfekte tallet er 6

faktorar i 6: 3, 2, 1, 6

sum av alle faktorar utan tallet sjølv:

$$3+2+1 = 6$$

Prøver 24

faktorar i 24: 1, 2, 4, 6, 12

sum: $1+2+4+6+12 = 25$ → ikkje eit perfekt tall

8.4 Meldeskjema frå NSD

29.05.2022, 13:47

Meldeskjema for behandling av personopplysningar



Meldeskjema

Referansenummer

882647

Hvilke personopplysningar skal du behandle?

- Navn (også ved signatur/samtykke)
- Lydopptak av personer

Prosjektinformasjon

Prosjekttittel

Masteroppgåve, Programmering i matematikk

Prosjektbeskrivelse

I masteren vil eg undersøkje om elevar i 1T på VGS brukar programmeringen dei har lærd i matematikktimane, til å løyse matematiske oppgåver. Dette vil skje gjennom innhentinga av oppgåver på prøvar eller innleveringar som enklast er løyst med programmering. Desse oppgåvane vil bli analysert, saman med resten av prøven for å danne ein oversikt over nivået til eleven, for å kunne undersøkje om det er noko samanheng mellom nivå og korleis dei vel å løyse oppgåvene.

Eg er læraren til elevane i 1T, så notatar frå timane, med planar og vurdering av økt kan bli brukt dersom det syner at har ei relevans for oppgåva.

Begrunn behovet for å behandle personopplysningene

I oppgåva er det ynskje å analysere oppgåver elevar har løyst på prøvar/innleveringar. Det vil bli anonimisert i oppgåva, men namn og siugnatur trengs i samtykkeskjemaet. Namnet til elevane vil bli lagra saman med prøvane til datainnsamlinga er ferdig for å halde styr på kven som er kven, sidan det vil bli innhenta oppgåver frå fleire prøvar og innleveringar.

Ekstern finansiering

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Olaug Grut Andersen, oan043@uit.no, tlf: 95090563

Behandlingsansvar

Behandlingsansvarlig institusjon

UiT Norges Arktiske Universitet / Fakultet for naturvitenskap og teknologi / Institutt for matematikk og statistikk

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Jan Nyquist Roksvold, jan.n.roksvold@uit.no, tlf: +4777646141

Skal behandlingsansvaret deles med andre institusjoner (felles behandlingsansvarlige)?

Nei

Utvalg 1

Beskriv utvalget

Elevar i 1 T, på VGS

Rekruttering eller trekking av utvalget

Elevane som blir spurd om å vere med i prosjektet er elevane min i 1T

Alder

16 - 16

Inngår det voksne (18 år +) i utvalget som ikke kan samtykke selv?

Nei

Personopplysninger for utvalg 1

- Navn (også ved signatur/samtykke)
- Lydopptak av personer

Hvordan samler du inn data fra utvalg 1?**Papirbasert spørreskjema****Grunnlag for å behandle alminnelige kategorier av personopplysninger**

Samtykke (art. 6 nr. 1 bokstav a)

Hvem samtykker for ungdom 16 og 17 år?

Ungdom

Personlig intervju**Grunnlag for å behandle alminnelige kategorier av personopplysninger**

Samtykke (art. 6 nr. 1 bokstav a)

Hvem samtykker for ungdom 16 og 17 år?

Ungdom

Annet

Beskriv

Matematikkprøver og innleveringar frå elevane og analysere desse

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Samtykke (art. 6 nr. 1 bokstav a)

Hvem samtykker for ungdom 16 og 17 år?

Ungdom

Informasjon for utvalg 1

Informerer du utvalget om behandlingen av opplysningene?

Ja

Hvordan?

Skriftlig informasjon (papir eller elektronisk)

Tredjepersoner

Skal du behandle personopplysninger om tredjepersoner?

Nei

Dokumentasjon

Hvordan dokumenteres samtykkene?

- Manuelt (papir)

Hvordan kan samtykket trekkes tilbake?

Ein beskjed på e-post eller på papir form

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

Prøvene bli godt gjennom med kvar enkelt elev som ein del av opplæringa til elevane, dei frå då innsikt i

opplysingane om seg sjølv, i tillegg kan elevane ta kontakt med meg eller min rettleiar for å innsyn/retta eller sletta opplysningar om seg sjølv

Totalt antall registrerte i prosjektet

1-99

Tillatelser

Skal du innhente følgende godkjenninger eller tillatelser for prosjektet?**Behandling**

Hvor behandles opplysningene?

- Mobile enheter tilhørende behandlingsansvarlig institusjon
- Maskinvare tilhørende behandlingsansvarlig institusjon

Hvem behandler/har tilgang til opplysningene?

- Student (studentprosjekt)

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

Nei

Sikkerhet

Oppbevares personopplysningene atskilt fra øvrige data (koblingsnøkkel)?

Nei

Begrunn hvorfor personopplysningene oppbevares sammen med de øvrige opplysningene

Sidan innhenting av datamaterialet er papir og digitale prøver som skal brukast i vurdering til elevanesstandpunkt, vil namnet deira stå saman med datamaterialet. Desse prøvane og innleveringane vil bli lagra i henhold til skuleeigar sine reglar.

Når materialet skal analyserast vil elevane bli anyomisert med nr.

Hvilke tekniske og fysiske tiltak sikrer personopplysningene?

- Andre sikkerhetstiltak
- Opplysningene anonymiseres fortløpende

Hvilke

Koblingsdokumentet mellom nummer og namn vil bli lagra i papirform på arbeidsplass med originalprøvane. Samtykkeskjemaet vil og ligge lagra der.

Varighet

Prosjektperiode

01.09.2021 - 06.07.2022

Skal data med personopplysninger oppbevares utover prosjektperioden?

Nei, alle data slettes innen prosjektslutt

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

Nei

Tilleggsopplysninger

