



Høgskulen på Vestlandet

Masteroppgave

MASIKT-OPG-OM-1-2022-VÅR-FLOWassign

Predefinert informasjon

Startdato:	18-05-2022 09:00	Termin:	2022 VÅR
Sluttdato:	01-06-2022 14:00	Vurderingsform:	Norsk 6-trinns skala (A-F)
Eksamensform:	Masteroppgave		
Flowkode:	203 MASIKT-OPG 1 OM-1 2022 VÅR		
Intern sensor:	(Anonymisert)		

Deltaker

Naun:	Renate Møgster Klepsvik
Kandidatnr.:	410
HVL-id:	147698@hvl.no

Informasjon fra deltaker

Antall ord *:	28236
----------------------	-------

Egenerklæring *: Ja

Jeg bekrefter at jeg har Ja registrert oppgavetittelen på norsk og engelsk i StudentWeb og vet at denne vil stå på vitnemålet mitt *:

Jeg godkjenner autalen om publisering av masteroppgaven min *

Ja

Er masteroppgaven skrevet som del av et større forskningsprosjekt ved HVL? *

Nei

Er masteroppgaven skrevet ved bedrift/virksomhet i næringsliv eller offentlig sektor? *

Nei



Høgskulen
på Vestlandet

MASTEROPPGÅVE

Programmering med PRIMM- modellen: ein
veg mot djupnelæring i matematikkfaget!

Programming with the PRIMM model: a path
towards in-depth learning in mathematics!

Renate Møgster Klepsvik

Master IKT i læring- teknologisk profil

Fakultet for lærarutdanning, kultur og idrett

Rettleiar: Anders Grov Nilsen

Innleveringsdato: 1.Juni 2022

Eg stadfestar at arbeidet er sjølvstendig utarbeida, og at referansar/kjeldetilvisingar til alle

kjelder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.

Føreord

Tida som masterstudent er no ved veggens ende og eg kan sjå attende på innhaldsrike år med både frustrasjon og meistring. Arbeidet med masteroppgåva har til tider vore krevjande, men òg oppbyggjande og lærerikt!

Takk, fyrst og fremst til klassen som stilte seg villig til å gjennomføra undervisningsopplegga, og til dei fem informantane som stilte til fokusgruppeintervju. Takk for interessa de har vist for forskingsprosjektet mitt, og for velvilje til å stille opp til det intervjuet som me gjennomførte og til disposisjon dersom eg skulle trenge noko meir. Det har betydd mykje at de stilte dykk så open og ærleg til å dela erfaringane og opplevingane dykkar med meg!

Ei stor takk skal òg rettast til rettleiar Anders Grov Nilsen, for tolmod, motivasjon og gode råd gjennom prosessen. Frå dei fyrste rotete utkasta der sjølv eg kunne slita med å forstå kva eg ville fram til, til det som i dag utgjer eit ferdig produkt. Takk for tips om både struktur og aktuell litteratur undervegs.

Renate Møgster Klepsvik

Storebø 30. mai 2022

Samandrag

Hausten 2020 gjorde Kunnskapsløftet 2020 (LK20) seg gjeldande i grunnskuleopplæringa. Som naturfag- og matematikklærer såg eg eit heilt nytt fokus på algoritmisk tenking, og programmering kom inn med eigne kompetansemål i både naturfag og matematikk. Kompetansar i algoritmisk tenking og programmering vert sett på som viktige verktøy for å oppnå djupnelæring, og dermed gjera elevane klar for «framtida sine kompetansar».

I denne masteroppgåva har eg teke utgangspunkt i problemstillinga «Korleis opplever elevane at programmering med undervisningsopplegg etter PRIMM- modellen bidreg til djupnelæring i matematikkfaget?» Gjennom to forskingsspørsmål, fem undervisningsøker med programmeringsoppgåver bygd opp etter PRIMM- modellen, ei kvantitativ likertundersøking og eit kvalitativt fokusgruppeintervju med fem informantar, har eg drøfta elevar sine opplevingar for å finna svar på problemstillinga mi.

Som teoretisk rammeverk har Lev Vygotsky sin sosiokulturelle læringsteori vore i fokus, då denne òg dannar eit fundament for PRIMM- modellen. PRIMM- modellen og grunntanken for denne har òg vore naturleg del av drøftingane mine. Vidare har algoritmisk tenking og Bloom's taksonomi stått sentralt i forskinga mi, då desse på mange måtar representerer henholdsvis nytt og gammalt syn på læring. Medan me historisk sett har hatt trua på å byggje stein på stein, ser me no ei satsing på å «fikle», «feilsøke» og «prøve igjen». Når vi ser dei i samanheng, kan vi likevel sjå at dei heng saman, og det treng ikkje vera anten ny eller gammal tradisjon som gjeld.

Sentrale funn i forskinga mi, viser at elevane takka være oppgåver med PRIMM- modellen, føler at dei kan prøve- og feile, og prøve igjen. Dei seier det er «gøy» å programmere når dei «får det til» og den primære metoden dei brukar er «prøve- og feile metoden» men når dei har fått det rett, reflekterer dei og analyserer svaret matematisk for å finne ein logikk dei kan ta med seg vidare.

Konklusjonen av studien min, er at elevane gjennom slikt arbeid, der dei uredd prøver seg fram, tester ut, får eit eigarskap til oppgåvene, og dermed ein sterk vilje til å meistre. Dei vert uthaldande, og søker etter samanhengar, både innafor fag og mellom fag. I konklusjonen viser eg òg at tankane og opplevingane til elevane kan knytast til utvikling av haldningsdimensjonen, dugleiksdimensjonen og kunnskapsdimensjonen av djupnelæring, og vi kan dermed sei at opplevingane elevane har fått med programmering i matematikk med PRIMM- modellen, bidreg til djupnelæring i matematikkfaget.

Abstract

In the autumn of 2020, the new Norwegian curriculum Kunnskapsløftet 2020 (LK20) became applicable in primary and lower secondary education of Norway. As a science and mathematics teacher, I saw a whole new focus on computational thinking and programming representing competence goals for both science and mathematics. Competences in computational thinking and programming are seen as important tools for achieving in-depth learning, and thus preparing students for «their future competencies».

In this master's thesis, I take as my starting point the problem «How do students experience that programming with teaching plans according to the PRIMM model contributes to in-depth learning in mathematics?» Through two research questions, five teaching sessions with programming assignments built according to the PRIMM model, a quantitative survey and a qualitative focus group interview with five informants, I have discussed students' experiences to find answers to my problem.

As a theoretical framework, Lev Vygotsky's socio-cultural learning theory is in focus, as this also forms a foundation for the PRIMM model. The PRIMM model and its basic ideas have a natural part of my discussions. Furthermore, computational thinking and Bloom's taxonomy have been central to my research, as these in many ways represent new and old views on learning. While we historically have had the belief in building stone upon stone, we now see an investment in "fiddling", "troubleshooting" and "trying again". When we see them in context, we can still see that they are connected, and there need not be either a new or old tradition.

Key findings in my research show that students, thanks to assignments with the PRIMM model, feel that they can try and fail, and try again. They say it is "fun" to program when they "get it right" and the primary method they use is the "trial and error method" but when they have got it right, they reflect and analyze the answer mathematically to find a logic they can use later.

The conclusion of my study is that the students through such work, where they fearlessly try their hand and test out, gain ownership of the tasks, and thus also gain a strong will to master. They are persistent, and search for connections, both within subjects and between subjects. In the conclusion, I also show that the thoughts and experiences of the students can be linked to the development of the attitude dimension, the diligence dimension and the knowledge dimension of in-depth learning, and we can thus say that the experiences the students have

with programming in mathematics with PRIMM model, contribute to in-depth learning in mathematics.

INNHALD

Føreord	I
Samandrag	III
Abstract	IV
INNHALD.....	VI
1.0 INNLEIING	1
1.1 Bakgrunn for vald tema	1
1.2 Problemstilling og forskingsspørsmål	3
1.4 Norsk skule i dag	4
1.4.1 Programmering i LK20	4
1.4.2 Algoritmisk tenking i LK20.....	5
1.4.3 Programmering og koding. Ei kort omgrepsavklaring	6
1.5 Oppgåva sin struktur.....	7
2.0 TIDLEGARE FORSKING	9
2.1 Tidleg forskning	10
2.2 Samanheng mellom programmering og djupnelæring?.....	11
2.3 Feilretting.....	11
2.4 Undervisningsopplegg	12
2.5 Elevane sitt læringsutbyte.....	13
2.6 Undervisningsplanlegging	14
2.7 Oppsummering av tidlegare forskning	14
3.0 TEORI	16
3.1 Sosiokulturell læringsteori.....	16
3.1.1 Den proksimale utviklingssona.....	16
3.1.2 Vygotsky sine fire kjenneteikn på læring	17
3.1.3 Kommunikasjon i eit sosiokulturelt læringsperspektiv.....	18
3.2 Blooms taksonomi	19

3.3 Djupnel�ring i LK20.....	20
3.3.1 Djupnel�ring og overflatel�ring	22
3.3.2 Programmering og djupnel�ring	24
3.4 PRIMM- modellen, ei sosiokulturell tiln�rming til l�ring.....	24
3.4.1 PRIMM- modellen og eleven sitt eigarskap til oppg�va	26
3.5 Sosiokulturell l�ring, Bloom’s taksonomi, djupnel�ring og PRIMM i samanheng.....	27
4.0 METODE	28
4.1 Metodisk tiln�rming.....	28
4.1.1 Kvalitativ metode.....	29
4.1.2 Metodetriangulering.....	29
4.1.3 Observasjon og forskarrolle	30
4.1.4 Hermeneutikk og fenomenologi.....	30
4.2 Utval	31
4.2.1 Fokusgruppe.....	32
4.2.2 Rolla som moderator	33
4.2.3 Intervju av fokusgruppa	34
4.2.4 Tema i fokusgruppeintervjuet	34
4.3 Transkribering av fokusgruppeintervju	35
4.4 Fenomenologisk reduksjon.....	36
4.5 Kvaliteten p� forskingsprosjektet	38
4.5.1 Reliabilitet.....	38
4.5.2 Validitet.....	38
4.6 Etske refleksjonar	39
4.6.1 Informert og fritt samtykke	39
4.6.2 Konfidensialitet og anonymitet og aktsamheit for deltakingrisiko.....	39
4.7 Utforming av undervisningsopplegg	40
4.8 Undervisnings�ktene	40

4.9 Evaluering av metodeval	41
5.0 RESULTAT	42
5.1 «Gøy»	42
5.2 «Få til»	44
5.3 Å arbeide med oppgaver bygd opp etter PRIMM- modellen	49
5.3.1 Samarbeid	49
5.3.2 Å byrje med ein ferdig kode	50
6.0 DRØFTING.....	52
6.1 F1: Å sjå samanhengar innafor matematikktemaet og til andre fagområde	54
6.1.1 Dugleiksdimensjonen og kunnskapsdimensjonen	54
6.1.2 «Gøy» å «få det til».....	56
6.1.3 Den algoritmiske tenkjaren og «prøve- og feilemetoden».....	58
6.2 F2: I møte med nye utfordringar.....	60
6.2.1 Haldningsdimensjonen.....	60
6.2.2 Den algoritmiske tenkjaren som «får det til».....	62
6.3 Programmering med PRIMM.....	63
6.3.1 Eigarskap med PRIMM- modellen	65
6.3.2 Samarbeid og sosiokulturell læring.....	67
6.4 Oppsummering av drøfting.....	70
7.0 AVSLUTNING	71
7.1 Konklusjon.....	71
7.1.1 F1: På kva måte ser elevane samanhengar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?	72
7.1.2 F2: Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?	72
7.1.3 Oppgaver med PRIMM- modellen	73
7.1.4 P1: Korleis opplever elevane at programmering med undervisningsopplegg etter PRIMM- modellen bidreg til djupnelæring i matematikkfaget?.....	74

7.2 Avgrensingar	75
7.3 Nytteverdi av forskingsprosjektet.....	76
7.4 Veggen vidare.....	76
8. KJELDER.....	78
Vedlegg 1 Informasjonsskriv	82
Vedlegg 2 Samtykkeerklæring	84
Vedlegg 3 Spørjeskjema distribuert i Survey- xact.....	85
Vedlegg 4 Undervisningsopplegg 1	86
Vedlegg 5 Undervisningsopplegg 2	87
Vedlegg 6 Undervisningsopplegg 3	88
Vedlegg 7 Undervisningsopplegg 4	89
Vedlegg 8 Undervisningsopplegg 5 (brukt i fokusgruppeintervju)	90
Vedlegg 9 Vurdering frå NSD	91

Oversikt over tabeller:

Tabell 1 Litteratursøk for studiet.....	9
Tabell 2 Kjenneteikn for djupnelæring versus overflatelæring.....	23
Tabell 3 Min mal for utarbeiding av undervisningsopplegg etter PRIMM- modellen	25
Tabell 4 Intervjuguide for intervju av fokusgruppa	35

Oversikt over figurer:

Figur 1 Den algoritmiske tenkjaren.....	6
Figur 2 Den proksimale utviklingssona	17
Figur 3 Bloom's taksonomi.	19
Figur 4 Djupnelæring som robuste mentale nettverk	22
Figur 5 Elevane sitt eigarskap til koden med PRIMM-modellen.....	26
Figur 6 Påstand: Det var lett å forstå dei ulike kommandoane i Python	46
Figur 7 Påstand: Det hjalp å vera to som samarbeidde då me jobba med oppgåvene.....	50
Figur 8 Eleven sitt eigarskap til koden med PRIMM- modellen	65
Figur 9 Den proksimale utviklingssona	68

1.0 INNLEIING

Hausten 2020 starta eg på masterstudiet IKT i læring ved Høgskulen på Vestlandet. Same hausten vart dei nye læreplanane LK20 (Utdanningsdirektoratet, 2020) innført i norsk skule. Nye læreplanar innebør ei omstilling både for lærarar og elevar, og eg såg fort at der var nokre område eg ville fordjupa meg i. Utdanningsdirektoratet grunngjev kvifor me trong nye læreplanar slik:

Samfunnet endrer seg raskt og det elevene lærer skal være relevant og framtidsrettet. Læreplanene er derfor endret slik at kompetansen elevene utvikler skal kunne brukes også på områder som i dag er ukjent. Å lære å lære er vesentlig i de nye læreplanene fordi det gir grunnlag for læring gjennom hele livet.

(Utdanningsdirektoratet, 2021)

Det å leggja betre til rette for djupnelæring for at elevane skal verta rusta til å sjå samanhengar og bruke det dei har lært i nye situasjonar, har vore ei av dei viktige endringane i dei nye læreplanane. Vidare er algoritmisk tenking og programmering lagt fram som ein del i fleire fag, fordi dei vert rekna som sentral kompetanse for framtida (Utdanningsdirektoratet, 2021).

1.1 Bakgrunn for vald tema

I takt med framveksten av Kunnskapsløftet 2020 (LK20) (Utdanningsdirektoratet, 2020), innsåg eg at eg som matematikklærer hadde ein jobb å gjera når det gjeld å byggja opp eigen kompetanse innafor fagfeltet programmering. Etter kvart har eg, gjennom lesing om temaet og eiga erfaring med programmering på 8. trinn, innsett at programmering berre er eitt av verktøya i arbeidet med algoritmisk tenking, og eg er blitt medviten at eg som lærar må sikra at elevane ikkje berre skal programmere for programmeringa si skuld, men som eit middel for å gjera dei til gode problemløysarar som ser samanhengar, kan utforska og kommunisera matematikk.

LK20 (Kunnskapsdepartementet, 2019) legg vekt på at elevane skal verta gode problemløysarar og forstå korleis matematikk heng saman med andre fag. Det å sjå samanhengar vert sett på som viktig for djupnelæring og forståing i faget. Vidare skal faget leggja til rette for at elevane utforskar og kommuniserer matematikk. Algoritmisk tenking og programmering høyrer under kjerneelementet utforsking og problemløysing, som vert beskrive slik:

Utforsking i matematikk handlar om at elevane leiter etter mønster, finn samanhengar og diskuterer seg fram til ei felles forståing.(...) Problemløysing i matematikk handlar om at elevane utviklar ein metode for å løyse eit problem dei ikkje kjenner frå før. Algoritmisk tenking er viktig i prosessen med å utvikle strategiar og framgangsmåtar for å løyse problem og inneber å bryte ned eit problem i delproblem som kan løysast systematisk.

(Kunnskapsdepartementet, 2019)

Det at elevane skal lære programmering og bruke algoritmisk tenking som problemløysingsmetode, heng òg saman med den nye satsinga på djupnelæring, som er eit av verdigrunnlaga i læreplanen sin overordna del (Kunnskapsdepartementet, 2017).

Utdanningsdirektoratet definerer djupnelæring som «å lære noe så godt at du forstå samanhengar og kan bruke det du har lært i nye situasjonar» (Utdanningsdirektoratet, 2019).

Med dette som bakteppe, las eg meg opp på temaet programmering i skulen og elevane sitt læringsutbytte av programmering. Gjennom litteratursøk, vart eg kjent med PRIMM-modellen, som er ei strukturert tilnærming til undervisningsprogrammering (Primmportal, 2022). I ein litteraturgjennomgang av programmering i skolen, utført av Dolonen et al. (2019), vert det konkludert at sidan engasjement i form av deltaking og samarbeid generelt gjev positivt læringsutbytte, bør PRIMM- modellen vera av spesiell interesse for dei som underviser i programmering. Med dette var mi interesse for PRIMM- modellen vekka, og eg bestemte meg for å teste ut eit undervisningsopplegg i programmering på 10. trinn, der eg nyttar PRIMM- modellen, og på den måten ha eit medvite forhold ikkje berre til programmering, men til korleis undervisninga vert lagt opp.

1.2 Problemstilling og forskingsspørsmål

Eg ville finna ut meir om korleis ungdomsskuleelevar sjølv opplever at dei lærer programmering på ein måte som sikrar forståing ikkje berre om sjølve programmeringsverktøyet, men òg varig kompetanse som er overførbar til andre lære- og utviklingssituasjonar. For å vurdere elevane sitt opplevde læringsutbyte, knytte eg læringa i programmering opp mot samanhengar innafor matematikkfaget sine kunnskapsområde og samanhengar mellom matematikk og andre fag sine kunnskapsområde. Det å sjå samanhengar innafor faget og mellom ulike fag, er del av djupnelæringsomgrepet, som eg ville ta med meg i forskingsspørsmåla til problemstillinga.

Eg var medviten at dette er eit forskingsprosjekt som kan vera vanskeleg å finna eit eintydig svar på. Etersom eg opplever at elevane ofte er opptekne av kvifor dei skal lære ulike tema i matematikk, og når dei får bruk for det, valde eg å forske på dette ut frå eit elevperspektiv. Det var eleven si oppleving av nytteverdi som skulle vera i fokus.

Med dette som bakteppe, kom eg fram til fylgjande problemstilling:

P: Korleis opplever elevane at programmering med undervisningsopplegg etter PRIMM-modellen bidreg til djupnelæring i matematikkfaget?

I formulering av forskingsspørsmåla som skulle hjelpe meg i finna svar på problemstillinga, freista eg å knyta desse til djupnelæring som overføring, og det å sjå samanhengar og bruka det ein har lært i programmering òg som problemløysingsstrategi i nye situasjonar og utfordringar:

F1: På kva måte ser elevane samanhengar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?

F2: Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?

1.4 Norsk skule i dag

1.4.1 Programmering i LK20

Dagens unge lærande møter eit anna digitalt samfunn enn det ein gjorde på 1980- talet.

Dessutan stiller samfunnsbiletet og arbeidslivet i dag nye krav, både til synet på utdanning og til bruk av digitale verktøy. Dette vert òg stadfesta både i forarbeidet til nye læreplanar, og i endelege læreplanar (NOU 2014:7; NOU 2015:8; Sanne, 2016; Sevik, 2016;

Utdanningsdirektoratet, 2020). Fagfornyinga har eit auka fokus på djupnelæring som eit grunnleggjande mål, og programmering vert mellom anna av Sevik et al. (2016) lagt fram som ein aktivitet som kan bidra til djupnelæring (Sevik, 2016).

I arbeidet med forslag til endringar i grunnopplæringa i LK20, vart det oppnemnd ei ekstern arbeidsgruppe som vart leia av Anders Sanne ved NTNU. Arbeidsgruppa la fram rapporten «Teknologi og programmering for alle» i august 2016 (Sanne, 2016). Rapporten gjev ei faggjennomgang som var delar av avgjerdsgrunnlaget om kva elevane vil trenge av kompetanse innafør teknologi og teknologirelaterte emne i framtida. Rapporten definerer programmering slik: «programmering vil si å bryte et gitt problem ned i et sett av kommandoer, og så få en datamaskin til å utføre disse kommandoene» (Sanne, 2016). Sanne et al. (2016) legg fram at å løyse problem gjennom å presisere sekvenser av kommandoer, vert kalla algoritmisk tenking. Algoritmisk tenking er òg teke opp i LK20 under kjerneelementet «utforskning og problemløysing» i matematikk 1-10 (Kunnskapsdepartementet, 2019). I tillegg til å vera omtalt som algoritmisk tenking i kjerneelementa for matematikk, har faget eigne kompetansemål som krev at elevane skal programmere. I ungdomsskulen skal dei:

Etter 8. trinn:

- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Etter 9. trinn:

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering

Etter 10. trinn:

- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

(Kunnskapsdepartementet, 2019)

Gjennom heile barneskulen finn ein òg kompetansemål knytt til algoritmisk tenking og programmering. På dei minste trinna er dette formulert som å kunne lage og følgje reglar og stegvise instruksjonar, og frå fjerde trinnet skal dei lage logaritmar og uttrykkje dei ved hjelp av lykkjer, vilkår og variablar (Kunnskapsdepartementet, 2019). Kompetansemåla på ungdomstrinnet har eit tydeleg preg av at programmering er eit verktøy dei skal bruke for å utforske, simulere, berekne og sjå samanhengar. Dette gjev ei stegvis innføring av programmering frå opplæring av programvare og algoritmisk tenking, til aktiv bruk i læresituasjonar og som verktøy til læring.

1.4.2 Algoritmisk tenking i LK20

Bruk av programvare i matematikkfaget er tett knytt opp til algoritmisk tenking. Å tenkje algoritmisk vil sei å bryte ned eit problem i delproblem, og å utvikle strategiar og framgangsmåtar for å løyse problema systematisk. Det handlar dessutan om å vurdere kva delproblem ein bør nytte teknologi til å løyse og kva som kan løysast på andre måtar (Utdanningsdirektoratet, 2019). Utdanningsdirektoratet har definert viktige kjenneteikn hos den algoritmiske tenkjaren: skapande, eksperimenterande og open for alternative løysingar. Den algoritmiske tenkjaren nyttar nysgjerrigheit og utforskande tilnærming både for å formulera problemet og for å løyse det (Utdanningsdirektoratet, 2019).

Utdanningsdirektoratet skildrar den algoritmiske tenkjaren som ein som må ha kognitiv kondis: Den algoritmiske tenkjaren må vera både systematisk og analytisk, i tillegg til å vera skapande, eksperimenterande og open for alternative løysingar. Dette krev ein nyfiken og eksperimenterande tilnærming for å formulera og løysa problem. Figuren under viser dei viktigaste hovudtrekka. Det å gjera feil er ein del av prosessen, og den algoritmiske tenkjaren må ha strategiar for å oppdage og rette opp i feila.



Figur 1 Den algoritmiske tenkjaren

(Utdanningsdirektoratet, 2019)

I skildringa av den algoritmiske tenkjaren, skil Utdanningsdirektoratet mellom arbeidsmåtar og nøkkelomgrep. Den algoritmiske tenkjaren nyttar arbeidsmåtarne fikle, skape, feilsøke, halde ut og samarbeid, og dugleikar innan logikk, algoritmar, dekomposisjon, mønstre, abstraksjon og evaluering. Under kjerneelementa i matematikkfaget i LK20 vert det presisert at algoritmisk tenking er viktig i prosessen med å utvikle strategiar og framgangsmåtar for å løyse problem, og inneber å bryte ned eit problem i delproblem som kan løysast systematisk (Kunnskapsdepartementet, 2019).

1.4.3 Programmering og koding. Ei kort omgrepsavklaring

Programmering og koding er omgrep som ofte vert brukt om kvarandre i skulen, og dermed har det oppstått ei forvirring og usemje om omgrepa (Statped, 2021). I denne oppgåva held eg meg til Statped (2021) sine omgrepsavklaringar i skiljet mellom omgrepa:

Programmering er ein form for algoritmisk tenking og handlar om å bryte komplekse problem ned til mindre problem, og deretter gje ein fullstendig og nøyaktig framgangsmåte for løysinga av problemet. Generelt kan dette overførast til situasjonar i dagleglivet. I denne

oppgåva forskar eg på programmering på ei datamaskin, og programmering rettar seg mot språket ein kan nytte til å kommunisere på ein måte som ei datamaskin forstår.

Omgrepet koding viser til instruksjonane ein legg inn, anten som tekst eller blokk, for å utføre ei handling på ei datamaskin. Å leggja inn koden og utvikla program, er såleis ein del av prosessen med å programmere.

Medan programmering er ein prosess som både handlar om å bryte ned eit problem i mindre problem, altså å definere steg for steg kva som må til for å få heilskapen, handlar kodinga om kvar instruks eller kommando som vert lagt inn på datamaskina.

Når eg i denne oppgåva skriv om «koden», handlar det om det som vert skrive inn som instruks til datamaskina. Når eg skriv om programmering, handlar det om heile prosessen der ein del er å finne ut korleis steg for steg skal kodast. Når eg refererer til «input» snakkar eg om sjølve koden som elevane skriv inn i Python, medan «output» er datamaskina si tolking av instruks, altså det som kjem fram som resultat av koden elevane har skrive inn.

1.5 Oppgåva sin struktur

I det fyrste kapittelet har eg gjort greie for bakgrunn for tema, formålet med forskingsprosjektet og problemstilling med tilhøyrande forskingsspørsmål. Dessutan har eg forklart omgrepa programmering og djupnelæring slik dei er meint i LK20 og klargjort nokre omgrep som vert nytta i teksten. I dei vidare kapitla vil eg ta føre meg fylgjande:

Kapittel 2: I kapittel 2 presenterer eg tidlegare forskning på området programmering i matematikkfaget.

Kapittel 3: Her presenterer eg det teoretiske grunnlaget for forskinga mi, med fokus på Lev Vygotsky sine sosiokulturelle læringsteoriar og den proksimale utviklingssona og betydinga av kulturelle artefakter, før eg presenterer Bloom's taksonomi som grunnlag for korleis ein kan forstå læring og undervisning som fremjar djupnelæring i faga. Vidare kjem eg inn på omgrepet djupnelæring i LK20, og ei gjennomgang av PRIMM-modellen og korleis denne kan hengja saman med djupnelæring i matematikk.

Kapittel 4: I fjerde kapittel tek eg for meg metode eg har brukt i forkinga mi, der eg kjem inn på prosessen med å utarbeide undervisningsopplegg, innsamling og analyse av datamateriell og vurdering av prosjektet med omsyn på reliabilitet, validitet og etikk. Avslutningsvis i kapittel 4 rettar eg eit kritisk blick mot vala eg har teke når det gjeld metode.

Kapittel 5: I kapittel 5 presenterer eg funn frå undersøkingane og legg fram kva funn eg har vald ut for vidare drøfting.

Kapittel 6: I det sjette kapittelet drøftar eg sentrale funn opp mot teoretisk grunnlag og tidlegare forskning.

Kapittel 7: Kapittel 7 inneheld konklusjon på resultat og drøftingar i forkinga, før eg seier nokre ord om vegen vidare på dette forskingsfeltet.

2.0 TIDLEGARE FORSKING

Programmering vert i LK20 introdusert som eit verktøy i matematikk i ungdomsskulen, og kompetansemåla i matematikkfaget har fokus på utforsking, simulering og å sjå samanhengar (Kunnskapsdepartementet, 2019). Dette er omgrep som òg knytast til djupnelæring, der spesielt det å sjå samanhengar, og «gjenbruk» i nye situasjonar vert peika på som idealet. Eg har gjennomført litteratursøk for å setja meg inn i kva tidlegare forskning seier om programmering som eit verktøy for å oppnå djupnelæring i matematikkfaget.

Framgangsmåten var søk gjennom ERIC og Oria for Høgskulen på Vestlandet, med søkeord programmering+djupnelære, algoritmisk tenking, datalogisk tänkande og computational thinking for ein tidsavgrensa periode etter 2006. Eg valde året 2006 fordi eg kjende til at Wing si «relansering» av omgrepet «computational thinking» i 2006 og ville ha eit søk som dekkja debatt på området i perioden frå 2006 og fram til programmering vart innlemma i læreplanane i fagfornyinga.

I tillegg har eg gjort nokre direktesøk både i Oria og i Google Scholar for å få fram litteratur og offentlege dokument som ikkje kjem fram i dette søket. Direktesøk er òg blitt nytta når eg har villa grave djupare i kjeldene til i aktuelle artiklar som har vore verdt å grave meir i.

Tema	Inkludert	Ekskludert
Database	Oria, ERIC, Google Scholar, Google	Andre
Tid	Etter 2006	Før 2006
Type publikasjon	Fagfelleverderte artiklar, bøker, avhandlingar, masteroppgåver, offentlege dokument, rapportar, presentasjonar.	Bacheloroppgåver
Fokus	Empiriske studie med fokus på opplæring i programmering på pc, samt djupnelæring	Problemløysing
Språk	Norsk, svensk, engelsk	Andre
Søkeord	«programmering + skole», «programmering + matematikk», «programmering + dybdelæring», Programmering. Dybdelæring. «algoritmisk tenkning», «computational thinking», «deep learning», datalogisk tänkande».	
Metode	Kvalitativ, kvantitativ, mixed	

Tabell 1 Litteratursøk for studiet

2.1 Tidleg forskning

Frå 1960- talet til midten av 1990 talet stod Seymour Papert for mykje av forskinga som vart gjort på programmering i skulen. Marvin Minsky introduserer Paperts prinsipp: «Noen av de mest avgjørende trinnene i mental vekst er ikke bare basert på å tilegne seg nye ferdigheter, men på å tilegne seg nye administrative måter å bruke det man allerede vet» (Minsky, 1988). Minsky (1988) nyttar dette prinsippet til å forklare Papert si oppfatning av at ein i læresituasjonar må leggja vekt på ikkje berre å samle kunnskap og dugleikar, men òg på betre måtar å bruke det ein allereie veit. Papert utfordra måten å tenkja utdanning på, og meinte datamaskina hadde ein sentral plass i framtidens skule. I sin presentasjon «En kritikk av teknosentrismen i å tenke på fremtidens skole» hevdar han at spørsmålet er ikkje kva datamaskina kan gjera med oss, men kva me kan lage av datamaskina (Papert, 1990). I presentasjonen rettar Papert (1990) fokus på at datamaskina må verta betrakta som noko anna enn ei informasjonskjelde, og utfordrar etablert skulesystem på kva målet med utdanning skal vera? «Ynskjer me borgarar som vil kjenna makta til å ta eigne avgjerder, eller ynskjer me disiplinerte borgarar som fylgjer instruksjonar og program som er sett opp for dei av andre?» spør han (Papert, 1990). Papert og teamet hans utvikla programmeringsspråket LOGOS, og gjennomførte ei rekke forskingsprosjekt på programmering i skulen ved Artificial intelligence Lab ved MIT (AI Lab) og seinare MIT medialab. Trass banebrytande arbeid innafor programmering og læring, lukkast han ikkje heilt med å endra måten å tenkja undervisning på. Då ein ikkje kunne dokumentera positiv læringseffekt og overføringseffekt, vart prosjekta i dåtidens skule lagt på is.

Papert nytta omgrep som prosesstenking om mykje av det same som Jeannette Wing snakka om då ho lanserte omgrepet «computational thinking» i 2006 (Grover & Pea, 2013). Wing hevda at «It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use» (Wing, 2006 referert i Grover & Pea, 2013, s. 38). Grover og Pea (2013) refererte til desse orda, då dei sju år seinare i ein litteraturgjennomgang gav ei oppsummering på status for computational thinking (CT) eller algoritmisk tenking. Dei la fram at Wing med desse orda gav eit nytt startskot for debatten om programmering som ein integrert del av utdanningsløpet, og at algoritmisk tenking handlar om å tenkje som ei datamaskin i problemløysing (Grover & Pea, 2013).

2.2 Samanheng mellom programmering og djupnelæring?

I ein litteraturgjennomgang av programmering i skolen, konstaterer Dolonen et al. (2019) at til tross for at tidlegare forskning syner at dugleikar i programmering i liten grad let seg overføra til andre problemstillingar eller fagområde, vert det i LK20 argumentert for nettopp denne effekten. Dolonen et al. (2019) meiner det er avgjerande korleis undervisningsopplegget vert lagt opp, og peikar på at der finns lite forskning på adekvate undervisningsopplegg. Dei legg fram at PRIMM- modellen kan vera lovande modell for undervisningsopplegg. PRIMM- modellen er utvikla spesielt med fokus på samarbeid og munnleg framstilling, i kombinasjon med prediksjon, utprøving og refleksjon. Ifylgje Dolonen et al. (2019), har PRIMM- modellen i tillegg vist seg å skapa djup og varig forståing i andre fag. Dolonen et al. (2019) sin presentasjon av PRIMM- modellen byggjer på mange måtar bru mellom programmering og djupnelæring, og spesielt med fokus på samarbeid og refleksjon, som er relevant for problemstillinga mi. Sentance et al. (2019) gjennomførte ein studie av bruken av PRIMM på 13 skular med 493 elevar i alderen 11-14 år og evaluerte resultat opp mot ei kontrollgruppe. Formålet med studien var å finne ut om bruken av PRIMM- tilnærming over 8-12 veker hadde noko å sei for elevane sitt læringsutbyte, og om det var verdifullt for lærarane. Studien viste at PRIMM yter eit verdifullt bidrag til utdanningsforskning ved å byggja vidare på tidlegare arbeid innan effektiv pedagogikk for lærarar, og oppmuntre til bruk av språk og dialog for å letta forståinga (Sentance et al., 2019). Elevane som fekk PRIMM- opplegget, presterte betre enn kontrollgruppa, og lærarane trakk fram spesielt at det er ein god undervisningsmetode for å undervise i blanda klasser.

2.3 Feilretting

Stenseth et al. (2019) beskriv eit undervisningsopplegg med PRIMM-modellen. Koden er i geometri, og elevane må kjenne matematikken for å forstå kva dei må endre på (Stenseth et al. 2019). Stenseth et al. argumenterer for algoritmisk tenking, som handlar om å ha metodar for å bryte ned eit problem i delproblem og løyse dei systematisk. Dei meiner feilretting, eller korleksjon, er underkommunisert i mandatet programmering har for algoritmisk tenking, og at ein bør fokusera meir på feilretting i positiv retning (Stenseth et al., 2019). Ved syntaksfeil eller køyretidsfeil vil ikkje programmet gje ønska svar. Feilretting handlar då om å gå inn i

koden og «finna feilen» før korreksjon. Stenseth et al. (2019) meiner denne typen feilretting kan underbyggje resonnement som inneber hypoteseformulering og hypotesetesting. Testgruppa til Stenseth et al. (2019) arbeidde i grupper på tre, og læraren observerte kommunikasjonen og resonnementa i gruppene. Det vart òg gjort videoopptak i klassen under økta. Elevane kom raskt i gang med oppgåvene, og fleire prøvde seg fram gjennom prøving og feiling. I etterkant ser Stenseth et al. (2019) at dei kanskje kunne ha presisert betre samanhengen mellom matematikkfaget og koden, slik at elevane dermed kunne sett at dei kunne rekne seg fram til kva endringar som var naudsynte. Då kunne dei fått ei tettare kopling mellom sjølve programmeringa og læring i matematikk.

2.4 Undervisningsopplegg

Fleire utdanningsforskarar rettar fokus på betydinga av korleis eit undervisningsopplegg i programmering vert bygd opp. Ellen Egebø Flø (2021) foreslår å arbeide tverrfagleg med programmering, for å sikre dei ulike måla som ligg i LK20. Ho argumenterer for at tverrfagleg arbeid etter ein modell som kombinerer naturvitskapleg metode med datatenking, kan sikra oppnåing av kompetansemål med programmering i faga, målet om opplæring i programmering for at elevane skal få fagleg utbyte av matematikkfaget, formuleringar i overordna del om at elevane skal utforske og oppleve skaparglede i matematikken, og ikkje minst det samla målet om at LK20 skal auka elevane si djupnelæring (Flø, 2021, s. 3). Flø presenterer ein modell som kombinerer naturvitskaplege metodar og designtenking, der elevane skal veksle mellom å undersøke, planlegge, gjennomføre, teste, evaluere, forbetre og dokumentere.

Det er eit viktig poeng at elevane ikkje skal arbeide seg kronologisk gjennom desse trinna, men navigere seg mellom dei og «lære av sine feil». Fasane har òg funksjon som støtte for lærarar som utviklar undervisningsopplegg. Flø (2021) rettar òg fokus mot å tilpasse oppgåvene til det elevane kan frå før, der ho viser til Harms et al. (2016) og Ericson et al. (2017).

Harms et al. (2016) seier at forskning på dette området ofte tek utgangspunkt i å redusera vanskegraden for elevane, ved at dei får oppgåver som ikkje krev at dei skal skrive koden frå botn. To av døma på slike typar opplegg som vert drege fram, er «Parson problem» som i

hovudsak går ut på at elevane får ei oppgåve der kodane er stokka om (puslespel), og PRIMM- modellen, som tek for seg ein miks av ferdig kode som skal analyserast, forklarast, endrast eller vidareførast (Harms et al., 2016 referert i Flø, 2021). Ericson et al. (2017) viser til at det kan verka vanskeleg for ein elev å skrive ein kode frå botn, og at arbeid med opplegg etter til dømes «Parson problem» gjev minst like mykje læringsutbyte for elevane (Ericson, 2017, referert i Flø, 2021).

2.5 Elevane sitt læringsutbyte

I norske læreplanar ligg algoritmisk tenking og programmering under kjerneelementet problemløysing og utforsking, sjølv om tidlegare forskning tyder på at algoritmisk tenking har lite effekt på problemløysing: I ei undersøking gjennomført på totalt 66 vidaregåande skular, delt inn i forsøksgruppe og kontrollgruppe, undersøkte Psycharis og Kallia (2017) effekten programmering har på elevane sine dugleikar til å resonnera, kjensle av meistring og problemløysing. Resultata av undersøkinga indikerer at programmering har positiv effekt på meistringskjensle og på evne til å resonnera, men mindre effekt på dugleikar innafor problemløysing (Psycharis & Kallia, 2017). Dette vert òg støtta av Sanford og Naidu (2017) som i tillegg seier at algoritmisk tenking er viktig og må koma inn tidleg i utdanningsløpet, men vil ikkje automatisk vera til hjelp ved problemløysing. Korleis opplæringa skjer vil vera av betydning, og Sanford og Naidu (2017) legg fram bruken av rekneark som ein god metode.

Sverige innførte programmering som obligatorisk del av teknikkfaget og matematikkfaget frå 1. juli 2018, og ligg to år framføre Noreg i dette arbeidet. Det kan vera naturleg å speide til svenskane, og kva erfaringar som er opparbeidd i den svenske grunnskulen. I ein kassstudie på 9. trinn har Maria Ueda (2021) forska på programmering og læringsutbyte frå eit elevperspektiv. Ho fylgde klassen under 6 leksjonar programmering i matematikk, og kombinerte observasjonar med elevintervju. Funna syner at elevane meiner det er meir kreativt å arbeide med programmering enn med «vanleg matematikk», og at elevane set pris på den fortløpande responsen programmet gjev dei. Dei ser likevel ikkje samanheng med læring i matematikk, bortsett frå bruken av variablar. Ueda (2021) legg fram at det ikkje held at programmeringsoppgåvene inneheld matematikk for at programmeringa skal gje eleven ei

oppleving av læring i matematikk. Desse funna stør opp om det Sanford og Naidu (2017) seier om betydninga av undervisningsform.

2.6 Undervisningsplanlegging

LK20 legg opp til at elevane skal utforske og jobbe med problemløysing i matematikkfaget ved hjelp av programmering. Mykje av litteraturen som er tilgjengeleg, drøftar i hovudsak kva type undervisningsmetodikk som gjev mest læringsutbytte. Metodikken varierer frå å la elevane fritt utforske og prøve seg fram i programmeringsarbeidet, til faste modellar der lærar instruerer og elevane kopierer. Forskarar argumenterer på ulikt vis for ulike typar undervisningsopplegg som skal sikra at elevane utforskar det matematiske temaet, framfor programmeringa i seg sjølv (Dolonen, 2019; Stenseth, 2019; Flø, 2021). Forsking gjennomført av Psycharis og Kallia (2017) tyder på at programmering kan ha positiv effekt på meistringskjensle og resonneringsevne, noko som igjen kan vera eit godt bidrag til djupnelæring i matematikkfaglege tema. Sanford og Naidu (2017) legg fram at programmering i seg sjølv ikkje har effekt på problemløysing, men at god opplæring og bruk av rekneark, kan bidra til å auke dugleikar i problemløysing.

2.7 Oppsummering av tidlegare forskning

I dette kapittelet har eg gjort greie for tidlegare forskning kring programmering og læring. Eg starta i kapittel 2.1 med Seymour Papert si tidlege forskning på programmering i andre halvdel av det 19.århundre, og Wing si relansering av dette tankesettet i 2006, før eg gjekk vidare med meir dagsaktuell forskning som søkjer samanhengar mellom programmering og djupnelæring og ulike undervisningsmodellar i kapittel 2.2. I kapittel 2.3 presenterte eg forskning som handlar om feilretting, i kapittel 2.4 undervisningsopplegg. I kapittel 2.5 og 2.6 tok eg for meg elevane sitt læringsutbytte og forskning som syner at måten lærar legg opp undervisninga har betydning for elevane sitt læringsutbytte.

Funn i tidlegare forskning (Papert, 1990; Grover & Pea, 2013; Psycharis & Kallia, 2017; Stenseth et al., 2019; Flø, 2021; Ueda, 2021) som syner at programmering bidreg til å auke

elevane si meistringskjensle og evne til resonnering, kan vera teikn på at programmering kan bidra til djupnelæring, fordi dette er dugleikar som kan gjera eleven betre i stand til refleksjon over eiga læring og det å sjå samanhengar. Litteratur frå tidlegare forskning peikar i retning av at programmering kan vera eit nyttig bidrag til djupnelæring i matematikkfaget, men er avhengig av korleis undervisninga vert planlagt og gjennomført. Å senke terskelen på den tekniske delen av programmering, til dømes ved å bruke PRIMM- modellen og ved å vera tydeleg på dei matematiske måla, kan vera ein måte å leggja til rette for at programmering vert eit eigna verktøy for å oppnå djupare læring i matematikkfaget. Denne kunnskapen tok eg med meg i det vidare arbeidet, og PRIMM- modellen og oppbygginga denne har etter eit sosiokulturelt læringsperspektiv, dannar grunnlaget for dei vala eg tek når eg i kapittel 3 vurderer teoretisk grunnlag for forskinga mi.

3.0 TEORI

I dette kapitlet vil eg greia ut om teori som er aktuell for forskingsprosjektet mitt. Eg finn det naturleg å byrje med ein presentasjon av det sosiokulturelle læringssynet til Lev Vygotsky i kapittel 3.1, før eg går vidare til Bloom`s taksonomi i kapittel 3.2. Vidare kjem ei gjennomgang av korleis ein skal forstå omgrepet djupnelæring i kapittel 3.3 før PRIMM-modellen og tankane bak oppbygningen av modellen vert presentert og forklart i kapittel 3.4.

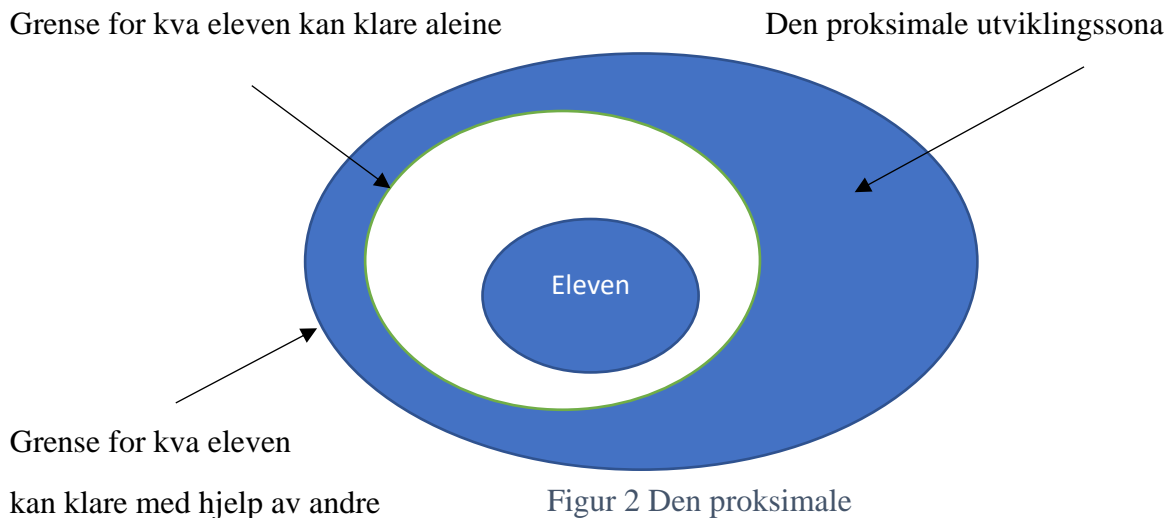
3.1 Sosiokulturell læringsteori

I eit sosiokulturelt perspektiv vil det vera slik at læring, som andre aktivitetar og handlingar, gjev mening og vert gjeve mening gjennom den sosiale og kulturelle konteksten. Kunnskap og kompetanse vert berre mediert gjennom ulike objekt, symbolsystem og særleg gjennom språket (Kluge, 2021, s. 41). I Vygotsky sine teoriar gjev det ikkje mening å skilje mellom avsendar og budskap, då heile den «menneskelege aktiviteten med eit kulturelt verktøy» må forståast for å gripa læringsprosessen.

Med eit sosiokulturelt læringssyn er det sosiale aspektet integrert i forståinga av korleis læring skjer. Sosial interaksjon og samarbeid er med andre ord ikkje verkemiddel for å oppnå læring, men ein føresetnad. Bruken av språk, teknologi og anna mediering, er sjølve kjerna av læringsprosessen (Kluge, 2021, s. 104). Kluge (2021) held fram at i ein viss forstand vert all teknologi som støttar samarbeid og fremjar diskusjonar, rekna som noko som støttar sosiokulturell læring, men det vil vera slik at teknologi vil gjera det i ulik grad, fordi variasjon i framstillingsformer, og samarbeidsformer rundt dei, kan bidra til å gje dei ei rikare forståing av fenomenet.

3.1.1 Den proksimale utviklingssona

Sentral i den sosiokulturelle læringsteorien, er utviklingssona (Kluge, 2021, s. 42). Den proksimale utviklingssona er den læringshorisonten av kva ein kan klare aleine ved å bygge på eksisterande kunnskap, og vidare kva ein kan klare ved hjelp av teknologi, lærar, medelev og annan støtte. Kunnskapssynet er dynamisk og dynamikken er på det sosiale nivået.



Figur 2 Den proksimale utviklingssona

(mi utforming av modellen)

I modellen skil ein mellom noverande kompetanse og kunnskap, det ein klarer på eigenhand, den nære utviklingssona der ein finn kunnskap og dugleikar ein har med litt støtte/hjelp utanfrå, og framtidens kunnskap som i dag er utafor rekkevidde (Kluge, 2021, s. 42).

3.1.2 Vygotsky sine fire kjenneteikn på læring

Strandberg (2008, ss. 25-26) beskriv dei fire kjenneteikna Vygotsky har på læring:

1. Aktivitetar som fører til læring, er alltid **sosiale**. Ein ting er kva eleven kan klare på eiga hand, men gjennom sosiale prosessar eller hjelp frå andre, vil eleven klare meir. Utvikling og læring krev sosiale prosessar. Jamfør figur 2, den proksimale utviklingssona (Strandberg, 2008).
2. Der finns **medierande artefakter**, som kan hjelpe oss i problemløysing. Strandberg (2008) beskriv medierande artefakter som reiskapar eller teikn, som til dømes kart, kompass og fingerrekning som «openberrar verda» for oss. Kart og kompass hjelper oss når vi er ute på tur. Fingerrekning hjelper oss når vi reknar. Poenget her er at det er den ytre aktiviteten som fører til indre tankearbeid. Når det gjeld programmering, kan ein her sei at datamaskina og programmeringsprogrammet er ein medierande artefakt.
3. Aktivitetar er alltid **situerte**. Aktivitetane eller læringa skjer på bestemte stader alt etter kva ein skal lære. Strandberg (2008) argumenterer for at det er lettare å lære seg tysk i Tyskland enn i Russland, og at det er lettare å bli ein god lesar i eit miljø som

har eit rikt utval av tekstar og bøker. På same vis kan ein sei at det er lettare å lære teknologi i eit teknologirikt miljø.

4. For det fjerde er **kreativitet** ein aktivitet som fører til læring. Menneske er i stand til å ta i bruk og gjera om læringssituasjonane ved relasjonar, hjelpemidlar og situasjonar.

3.1.3 Kommunikasjon i eit sosiokulturelt læringsperspektiv

Ei grunnleggjande oppfatning i eit sosiokulturelt perspektiv på utvikling og læring, er at menneske handlar innafor ramma for praksisar og kulturelle samanhengar og i direkte eller indirekte samspel med andre. Dette medfører at ein ofte snakkar om menneskeleg tenking som delar av, integrert i og bidrag til sosiokulturelle praksisar (Säljö, 2008, s. 107).

Samanhengane og interaksjon med andre, gjer at ein tenderer til å tenkja og handla på visse måtar. Som når ein er på skulen og fylgjer dei kommunikative spelereglane som er forventa at ein fylgjer. Då vil me òg gje og motta meining som fylgjer desse mønstra. Vygotsky beskriv ein dobbel *re*-presenterande funksjon med sin ofte siterte påstand:

Enhver funksjon i et barns kulturelle utvikling opptrer to ganger eller på to plan. Først finnes den på det sosiale planet, deretter på det psykologiske. Først opptrer den som en interpsykologisk kategori, og deretter i barnet som en intrapsykologisk kategori.

(Vygotsky, 1981 referert i Säljö, 2008, s. 108)

Säljö (2008) forklarar her at menneske lærer gjennom å delta i praktiske og kommunikative samspel med andre, og at måtane me resonnerer om og tolkar den verkelegheita me møter i interaksjon på, brukar me som ressursar for å forstå og kommunisera i framtidige situasjonar. Dei intrapsykologiske funksjonane (tenkinga) hos individet er i sosiokulturell forstand former for kommunikasjon (interpsykologiske funksjonar) som individet har støtt på, teke til seg, og brukar som ressursar i framtidige situasjonar, som varig personleg forståing.

3.2 Blooms taksonomi

Kva praksisar har me, og kva grunnleggjande tankar her me om korleis undervisning vert gjennomført, og korleis elevane byggjer kunnskap? Blooms taksonomi, eller «The Taxonomy of Educational Objectives» av Bloom, Engelhart, Furst, Hill og Krathwohl (1956) har lenge satt sitt preg på undervisning, didaktikk og vurdering i skulen. Opphavelag inneheldt taksonomien seks kognitive prosessar, frå enkle til meir kompliserte. Dei opphavelage namna på dei kognitive prosessane var: Kunnskap, gjera greie for, anvende, analyse, syntese og evaluering. Kwart trinn vert meir komplisert, og der ligg ein føresetnad om at ein må meistre trinna under for å koma høgare opp i taksonomien. Anderson og Krathwohl (2001) presenterte seinare taksonomien i verbform, der trinna fekk namna hugse, forstå, anvende, analysere, evaluere og skape, der dei tre fyrste er å rekne som lågare taksonomisk nivå, og dei tre siste som høgare taksonomisk nivå (Agarwal, 2019, s. 190). Høgare taksonomisk nivå, inneber mellom anna å kunna bryta noko i mindre delar, sjå samanhangar, tolka, samanlikna, differensiera, kritisera, generera, planleggja og produsera. Kjenneteikn ein kan kjenna att frå djupnelæring.



Figur 3 Bloom's taksonomi¹.

¹ Biletet er henta frå <https://www.universell.no/ikt-kartlegging/innhold-og-nivaa/>

Både den opphøvede modellen, frå 1956 og den reviderte frå 2001, har som føresetnad at ein må handtere eitt nivå før ein kan gå vidare til høgare nivå. Agarwal (2019) reiser spørsmålet om høgare kognitivt nivå berre kan utviklast ved å fyrst fokusera på dei lågare nivåa først, eller om dei kan utviklast gjennom å engasjera elevane i komplekse spørsmål og oppgåver som krev analyse, evaluering og skaparevne. Etter tre eksperiment konkluderte Agarwal (2019) med at øvingar som fokuserer på dei lågare kognitive nivåa, som hugse og forstå, ikkje påverkar evna til å analysera, evaluera og skapa. Agarwal konkluderer med at ein kan gå direkte inn på høgare kognitive nivå utan å gå innom lågare nivå først (Agarwal, 2019).

Som eg har vore inne på under algoritmisk tenking og programmering i matematikkfaget, er dette aktivitetar der ein skal fikle og bryte ned for å oppnå djupnelæring, som i Bloom's taksonomi ligg på høgare kognitivt nivå. Aktivitetar med programmering i matematikk kan dermed vera eit døme på aktivitetar som skal motivera elevar når dei byrjar midt i taksonomien og fyrst og fremst skal anvende, analysere, vurdere og skape.

3.3 Djupnelæring i LK20

Ludvigsenutvalet, som vart utnemnd av Regjeringa i 2013, hadde som mandat å vurdere grunnopplæringa sine fag opp mot krav til kompetanse i eit framtidig samfunn- og arbeidsliv. Utvalet gav ei delutreiing i NOU 2014:7 *Elevenes læring i fremtidens skole*, og hovudutreiing i NOU 2015:8 *Fremtidens skole- Fornyelse av fag og kompetanser*. Ludvigsenutvalet har definert djupnelæring slik:

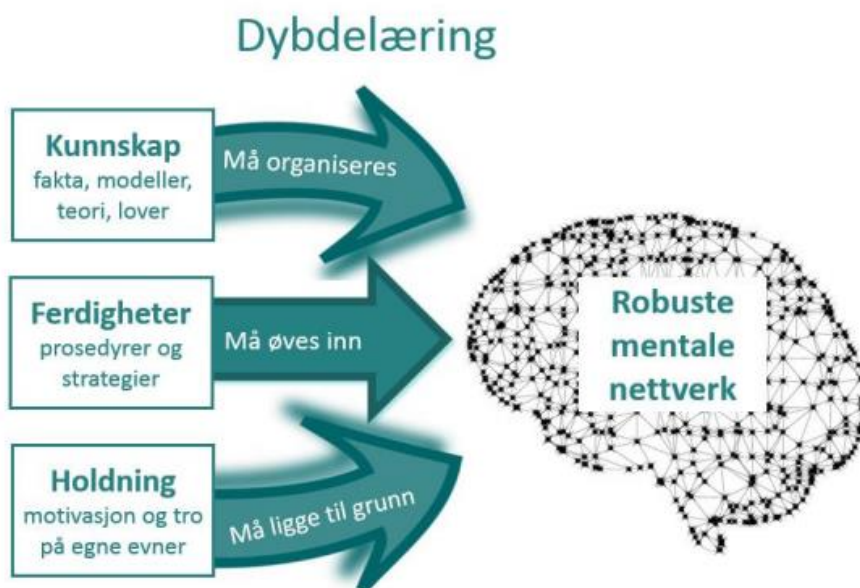
Dybdelæring er å gradvis utvikle kunnskap og varig forståelse av begreper, metoder og sammenhenger i fag og mellom fagområder. (...) det innebærer at vi reflekterer over egen læring og bruker det vi har lært på ulike måter i kjente og ukjente situasjoner, alene eller sammen med andre.

(NOU 2014: 7, s. 35).

Denne definisjonen hadde eg som grunnlag då eg utvikla forskingsspørsmåla mine, der eg spesielt har fokus på det å sjå samanhengar i fag og mellom fagområde i F1, og i F2 der eg rettar fokus mot det å bruke det ein har lært på ulike måtar i kjende og ukjende situasjonar.

Utdanningsdirektoratet delar tydinga av omgrepet i tre dimensjonar: Kunnskapsdimensjonen, dugleiksdimensjonen og haldningsdimensjonen. Det er denne forståinga av djupnelæring eg stettar i forskingsprosjektet, men Gamlem og Rogne (2019) nemner at internasjonalt, av mellom anna Fullan (2018) har omgrepet deep learning ei endå djupare betydning enn den norske. Medan den norske tolkinga fokuserer på eleven si læring, ser den internasjonale tolkinga i tillegg på skulestrukturar som skapar djupnelæring (Fullan, 2018; referert i Gamlem & Rogne, 2019). Dei ulike dimensjonane av omgrepet djupnelæring i norsk samanheng, kjem til syne i sjølve definisjonen. Kunnskapsdimensjonen handlar om å stegvis utvikle kunnskap og varig forståing, medan dugleiksdimensjonen syner til føresetnaden at vi reflekterer over eiga læring og brukar det på ulike måtar (...) aleine eller saman med andre.

Haldningsdimensjonen syner til dei overordna verdiane og prinsippa i overordna del av LK20, som set preg på djupnelæring slik at vi utviklar gode haldningar, evne til refleksjon, kritisk tenking og til å ta etiske vurderingar (Gamlem & Rogne, 2019, s. 7). I heftet «Dybdelæring i skolen» gjev Gamlem og Rogne ei gjennomgang av omgrepet djupnelæring generelt, og spesielt på korleis ein kan oppnå djupnelæring gjennom samarbeidslæring, arbeid med sjølvregulering og lærestrategiar, arbeid med kritisk tenking og gjennom metakognisjon (Gamlem & Rogne, 2019). I forhold til programmering, ser eg spesielt på samarbeidslæring som eit aktuelt område. Gamlem og Rogne (2019) syner til overordna del i LK20, der det vert presisert at elevar tenker, erfarer og lærer i samspel med andre gjennom læringsprosessar, kommunikasjon og samarbeid.



Figur 4 Djupnelæring som robuste mentale nettverk

Liv Oddrun Voll (2019, ss. 4-6) nyttar figuren over for å illustrera korleis djupnelæring består av dei tre dimensjonane kunnskap, dugleik og haldning. Kunnskapsdimensjonen består av fakta, modellar, teori og lovar som må organiserast. Dugleiksdimensjonen er prosedyre og strategiar som må øvast inn, og haldningsdimensjonen viser til motivasjon og tru på eigne evner, som må liggja til grunn. Voll (2019) understrekar at dette er tre prosessar som heng saman og påverkar kvarandre gjensidig.

3.3.1 Djupnelæring og overflatelæring

Djupnelæring handlar om å gradvis tileigne seg forståing av omgrep og samanhengar innfor eit fagområde, og om å forstå korleis tema og lærestoff går på tvers av dei ulike fagområda. Djupnelæring handlar òg om å utvikle elevane sine evner til å analysera, løyse problem og reflektera over eiga læring. Ludvigsenutvalet konkluderte med at det var for mykje overflatelæring og for lite djupnelæring i skulen til å møte framtida sine krav til kompetanse i arbeids- og samfunnslivet (NOU 2015: 8, s. 8). Utvalet meinte at dagens unge vil måtte halde seg til eit samfunn i stor og hurtig utvikling, med rask utvikling i teknologi, utfordringar knytt til berekraft, store kulturelle, religiøse og demografiske utfordringar, og treng dermed kompetanse som er tilpassa dette. Ludvigsenutvalet (NOU 2015: 8) framstiller djupnelæring som ein kontrast til overflatelæring, og hevdar vidare at det som skil eksperten frå nybyrjaren,

er at eksperten kan bruka djupneforståing til å raskt tolka og trekka nye slutningar. For å utdjupa skilnaden mellom overflatelæring og djupnelæring, har utvalet teke utgangspunkt i R.Keith Sawyer sin tabell (Sawyer, 2006 referert i NOU 2014:7):

Djupnelæring	Overflatelæring
Elevar relaterer nye idear og omgrep til tidlegare kunnskap og erfaringar.	Elevar arbeidar med nytt stoff utan å relatere det til kva dei kan frå før.
Elevar organiserer eigen kunnskap og omgrepssystem som heng saman.	Elevar handsamar lærestoff som separate kunnskapselement.
Elevar ser etter mønster og underliggjande prinsipp. Elevar vurderer nye idéar og knyt dei til konklusjonar. Elevar forstår korleis kunnskap vert til gjennom dialog og vurderer logikken i eit argument kritisk.	Elevar memorerer fakta og utfører prosedyre utan å forstå korleis eller kvifor. Elevar har problem med å forstå nye idéar som er ulike frå dei dei har møtt før. Elevar handsamar fakta og prosedyre som statisk kunnskap, overført frå ein allvitande autoritet.
Elevar reflekterer over eiga forståing og eigen læreprosess.	Elevar memorerer utan å reflektere over formålet eller over eigne lærestrategiar.

Tabell 2 Kjenneteikn for djupnelæring versus overflatelæring

(Henta frå NOU 2014:7)

Gilje og Bolstad (2019) argumenterer for at denne tabellen kan vera misvisande på ein slik måte at tradisjonell klasseromundervisning vert sett på som overflatelæring. Dei meiner dessutan at Ludvigsenutvalet har misforstått når dei set om det engelske omgrepet deep learning til djupnelæring, fordi, som eg har vore inne på, det engelske omgrepet har ei breiare tyding enn det norske (Gilje & Bolstad, 2019).

Tor J. Skjelde forklarar skiljet mellom overflatelæring og djupnelæring med at å lære for å hugse kan vera slitsamt, og at læringa då kan føra til stress. Viss det å attfortelja vert eit mål i seg sjølv, har overflatelæring for stor plass i læringa. Han hevdar at overflatelæring berre er det fyrste steget i læringa mot å utvikla eleven sin kompetanse, og at denne læringa dannar grunnlaget for djupnelæring (Skjelde, 2017).

3.3.2 Programmering og djupnel ring

Sevik (2016) argumenterer for at programmering har god effekt b de p  tilpassa oppl ring, og   byggja bru mellom teori og praksis, og mellom djupnel ring og overflatel ring. N r elevane programmerar, f r dei fortl pande tilbakemelding p  om det dei har gjort er rett, og dersom programmet ikkje verkar etter intensjonen, f r dei som regel indikasjonar p  kva som kan vera feilen. Programmering har  g l g terskel for   koma i gang, og «h gt tak» n r det kjem til kor langt elevane kan drive det med avanserte koder og komplekse prosjekt. Dette gjer at   jobba med programmering kan gje ein alternativ inngang til   jobbe med faglege problemstillingar for dei som strever, og samtidig gje st rre faglege utfordringar til dei som treng det. Dette er og med p    danne eit fundament for elevane si vidare l ring. Sevik (2016) viser til at programmering byggjer bru mellom overflatel ring og djupnel ring, og gjev moglegheiter for   knyte teori opp mot ein praktisk kontekst der elevane skapar teknologi. Det at programmering utfordrar elevane med ulike problemstillingar som krev kritisk tenking og resonnering, legg til rette for at elevane m  bruke relevante strategiar for   l yse utfordringane dei m ter i arbeidet med programmering (Sevik, 2016, s. 16).

3.4 PRIMM- modellen, ei sosiokulturell tiln rming til l ring

PRIMM er ei tiln rming til planlegging av undervisnings kter og aktivitetar som inneheld stega predict- run- investigate- modify- make, som er arbeidsprosessane i PRIMM- modellen. Dei ulike stega eller fasane i modellen kan forklarast slik:

- **Predict:** Elevane f r utlevert ein ferdig kode, og skal freiste   avgjera kva som skjer n r ein k yrer koden.
- **Run:** Her skal elevane laste ned eller skrive av koden, for s    k yre han for   evaluere om dei tenkte rett i fase 1 predict.
- **Investigate** er tredje steg i modellen. Her skal elevane verta betre kjent med koden. Denne fasen kan og b r helst varierast med ulike typar oppg ver, som gjerne kan innebera   lage feil i koden eller gje namn til dei ulike variablane.
- **Modify** er steg fire i modellen. Her f r elevane i oppg ve   gjera bestemte endringar i koden slik at «output» vert noko anna.

- **Make** er femte og siste steg i modellen. Her skal elevane lage ein heilt ny kode, med utgangspunkt i den koden dei fyrst fekk utlevert. Det kan vera ei oppgåve der dei kan låne noko frå den opphavslege koden, eller ei heilt ny oppgåve. I begge høve skal den nye koden innebera noko nytt som krev ein ny funksjon.

(Primmportal, 2022)

Nr	Fase	Utfordring	Elevoppgåve
1	Predict (lese og tolke koden)	Elevane får utlevert ein ferdig kode som fungerer, men ser ikkje sjølve programmet og kva som skjer.	Elevane skal lese og forstå koden. Setja ord på og forklare og grunngje kva som vil skje når dei køyrer koden. (Notere eventuell usemje med partner)
2	Run (køyr koden)	Elevane køyrer koden og registrerer kva som skjer.	Diskuter med samarbeidspartner. Hadde de rett? Var det noko som var annleis enn de hadde sett føre dykk i fase 1?
3	Investigate (Undersøkje koden)	Her kan lærar leggje inn feil i koden, eller eventuelt be elevane finne andre måtar å kode for å få same «output».	Elevane må undersøkje og forstå koden, og sjå kva som skjer undervegs, for å identifisere kva som kan gjerast annleis.
4	Modify	Her må elevane gjera om på koden for å få ny «output».	Rekning: Elevane kan bruke det dei har lært i det matematikkfaglege temaet for å veta kva som skal til. Programmering: Elevane må finne ut kva dei må endre i koden for å få den ønska endringa som oppgåva ber om.
5	Make (lage eigen kode)	Elevane skal lage eigen kode, som har fellestrekk med den opphavslege koden, men med eigen kravspesifikasjon som skil seg frå den opphavslege koden.	Her må elevane dra nytte av det dei har lært av den koden dei fekk ferdig utlevert, og sjølv byggje opp ein kode frå botn.

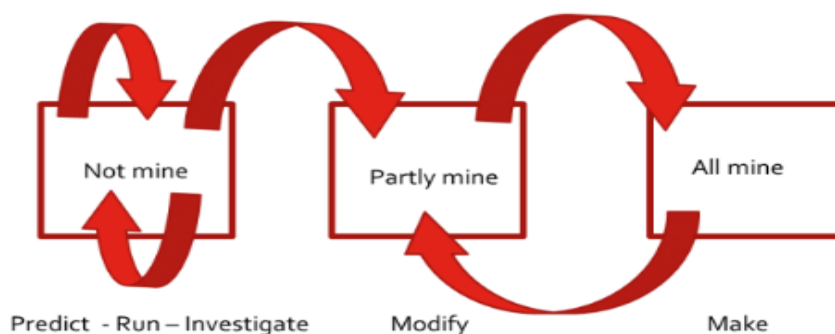
Tabell 3 Min mal for utarbeiding av undervisningsopplegg etter PRIMM- modellen

PRIMM-modellen er utvikla av Sue Sentance, som tok utgangspunkt i Lev Vygotsky sine sosiokulturelle læringsteoriar som vektlegg språk, formidling og overføring av kunnskap og dugleikar frå det sosiale til det kognitive planet (Sentance et al., 2019). For å minske startvegring og sikre motivasjon hos elevane, gjev lærar som brukar PRIMM- modellen elevane eit ferdig program dei skal analysere (predict), før dei får køyre det (run).

Elevane skal arbeide parvis med ei datamaskin, der dei må samarbeide og kommunisere om oppgåvene. Gjennom opplegget får elevane i oppgåve å gjere bestemte endringar i programmet, og må då gå inn i koden for å undersøkje korleis endringa skal gjerast (investigate). Ved å utføre endringane (modify), får dei meir eigarskap til koden. Heil til slutt skal elevane lage sin eigen kode frå botn, med utgangspunkt og fellestrekk til den dei fekk utlevert i starten. Gjennom alle trinna i undervisningsopplegget er samarbeid og kommunikasjon med ein eller fleire medelevar viktig (Primmportal, 2022).

3.4.1 PRIMM- modellen og eleven sitt eigarskap til oppgåva

Gjennom å arbeide seg gjennom ei oppgåve i par, vil elevane gå frå å fyrst gjera seg kjend med ein kode dei sjølv ikkje har eigarskap til i dei tre fyrste trinna, til å delvis eige koden i fase tre dan han eller ho modifierer eller endrar koden etter instruksjonar, til, i det femte steget, å sjølv lage ein eigen kode som han eller ho ha fullt eigarskap til. Som figur 4 under viser, vil elevane kunna veksle mellom dei to siste stega i modellen («modify» og «make»), alt etter kor mykje dei nyttar den opphavslege koden når dei skal byggje opp sin eigen kode i siste fase «make».



Figur 5 Elevane sitt eigarskap til koden med PRIMM-modellen

(Primmportal, 2022)

3.5 Sosiokulturell læring, Bloom's taksonomi, djupnelæring og PRIMM i samanheng

I dette kapitlet har eg presentert Lev Vygotsky sin læringsteori med fokus på kommunikasjon, kulturelle artefaktar og den proksimale utviklingssona, før ei utgreiing om djupnelæring i norsk skule, med definisjonar og skiljet mellom djupnelæring og overflatelæring. Vidare vart Bloom's taksonomi og utviklinga og fundamentet bak oppgåver med PRIMM- modellen. Desse heng saman på fleire måtar: PRIMM- modellen er utvikla med utgangspunkt i Lev Vygotsky sine sosiokulturelle teoriar, spesielt med tanke på viktigheita av språk, formidling og overføring av kunnskap og dugleikar frå det sosiale til det kognitive planet. Ser ein på beskrivingane av djupnelæring og overflatelæring, og dei tre dimensjonane kunnskap, dugleik og haldningar, kan det vera her PRIMM-modellen er tenkt å byggja bru mellom overflatelæring og djupnelæring, ved kunnskapsdimensjonen og dugleiksdimensjonen. Bloom's taksonomi kan vera aktuell på fleire måtar. Når elevane arbeider med programmering i matematikk, er mi antaking at dei lærer på fleire plan. Når det gjeld sjølve programmeringsjobben, har dei støtte og hjelp frå PRIMM- modellen, og ein kan sei dei byrjar på lågaste trinn i taksonomien. Ser ein på det matematikkfaglege i oppgåvene, kan ein sei at elevane byrjar midt i taksonomien, der dei skal anvende og analysere. Denne jobben kan krevje at elevane anten tileignar seg kunnskap på området, eller aktiverer tidlegare lært stoff. Ser ein slik på det, kan PRIMM- modellen og eigarskap til oppgåva spele ei rolle for om og korleis elevane tilnærmar seg matematikkfagleg djupnelæring. Viss elevane opplever eigarskap til matematikkoppgåva frå starten av, tenker eg at det kan spela inn på haldningsdimensjonen, fordi det gjev ein motivasjon og sterkt ynskje om å klare å løyse oppgåva.

4.0 METODE

Forskinga mi har eit klart elevperspektiv, og eg ville læra meir om elevane sine opplevingar av djupnelæring gjennom programmering i matematikkfaget. I dette kapittelet vil eg gjera greie for sjølve forskingsprosessen, val av metode og førebuing til datainnsamling. Deretter kjem eg inn på korleis sjølve datainnhentinga har gått føre seg, val av informantar og informasjon til desse. Vidare vil eg gjera greie for handsaming og analyse av datamateriale. Studien sin reliabilitet, validitet og etiske omsyn vert handsama

4.1 Metodisk tilnærming

Innan forskning, skil ein gjerne mellom to hovudparadigme, kvantitativ og kvalitativ forskning. Medan kvantitativ forskning er deskriptiv, eller beskrivande, beveger kvalitative studie seg utover det deskriptive, ettersom forskaren freistar å forklare kva som skjer og kva det kan tyde. I kvalitativ forskning er det forskaren som ved hjelp av eigne erfaringar, observasjonar og teoriar, som tolkar dei empiriske data i kontekstuell samanheng (Postholm M. , 2010, s. 29).

I pedagogisk forskning er det i all hovudsak kvalitative fenomen og prosessar ein studerer. Postholm peikar på at det betyr ikkje at ein bør avgrensa seg til kvalitative metodar og data (Postholm & Jacobsen, 2019, s. 41). Eg ville gjennomføra eit undervisningsopplegg med PRIMM- modellen, og deretter gjennom intervju læra meir om korleis elevane opplevde læring under øktene. Metoden min er primært kvalitativ, og hovudvekta vert lagt på det kvalitative intervjuet av fokusgruppa. For å sikra at eg fekk tak i den relevante informasjonen i intervjuet, valde eg å fylgja Postholm og Jacobsen (2019) sitt syn på at ein bør sjå kvalitativ og kvantitativ metode som komplementære metodar som utfyller kvarandre og kan gje inspirasjon til ytterlegare refleksjon og diskusjon (Postholm & Jacobsen, 2019, s. 41). Eg laga difor ei kvantitativ spørjeundersøking som heile klassen svara på kort etter den siste undervisningsøkta. Resultata frå denne undersøkinga var med å danna noko av grunnlaget for emna for diskusjon og refleksjon i eit seinare intervju. Denne trianguleringa gav meg høve til å skaffa meg ein brei oversikt i svara frå eit standardisert kvantitativ undersøking, samtidig som det kvalitative fokusgruppeintervjuet gjorde meg meir rusta til å utdjupe og fortolke svara. For å få ein systematisert observasjon av korleis elevane kommuniserer medan dei arbeider med ei programmeringsoppgåva, fekk dessutan fokusgruppa ei femte

programmeringsoppgåve bygd opp etter PRIMM-modellen, som dei løyste munnleg under intervjuet (vedlegg 8).

4.1.1 Kvalitativ metode

Postholm (2010, s. 35) definerer kvalitativ forskning som ei undersøking av menneskelege/sosiale prosessar i deira naturlege setting, der forskaren freistar å danna eit heilskapleg bilete av deltakarane sitt perspektiv. Slike undersøkingar vert òg kalla naturalistiske, fordi målet med studien er å utforske sosial handling i ein verkeleg situasjon (Postholm M. , 2010, s. 28). Ifylgje Postholm (2010), kan slike kvalitative undersøkingar aldri vera heilt fastlagde på førehand. Når ein forskar brukar induktiv metode, må forskaren fokusera fullt og heilt på respondenten sitt perspektiv. Forskaren må vera open for at nye forhold kan gjera seg gjeldande og bli teke med i forskingsbiletet. Gjennom interaksjon mellom antakingar og data, dannar forskaren ei ny forståing av forskingsfeltet og respondenten sine uttaler (Postholm M. , 2010, s. 35).

Ettersom eg var ute etter elevane sine opplevingar med programmeringsøktene, og ville stilla meg så open som mogleg til deira tankar og refleksjonar, avgjorde eg at kvalitativt intervju skulle få hovudfokus frå mi side. For å sikre at min forkunnskap eller observasjonar frå øktene ikkje skulle verta for styrande for intervjuet, kombinerte eg med ei kvalitativ likertundersøking i klassen, som var med å danna grunnlaget for tema for intervju. Intervjuet hadde form som eit ope fokusgruppeintervju, der refleksjonane mellom informantane og svara frå likertundersøkinga i klassen utgjorde fundamentet, saman med nokre fokustema eg hadde med meg i form av ein semistrukturert intervjuguide.

4.1.2 Metodetriangulering

Hovudvekta i forskinga mi vart lagt på dei kvalitative intervju av informantane som inngjekk i fokusgruppa. For å sikra meg at eg tok opp dei mest relevante tema i fokusgruppa, var det i tillegg interessant å vita litt om korleis dei andre elevane i klassen opplevde øktene og korleis dei reflekterte over eiga læring. Både for å sjå etter eventuelle meiningsfortettingar, men òg for å eventuelt gje meg ein peikepinn før fokusgruppeintervjuet, gjennomførte eg ei likertundersøking i Survey-xact² (vedlegg 3) som heile klassen svara på. Denne var utforma

² <https://www.survey-xact.dk/servlet/com.pls.morpheus.web.pages.CoreUserOrganizationList>

med likert-verdiane usamd- litt usamd- korkje/eller- litt samd- samd, og eg nytta svara i fokusgruppeintervjuet for å be informantane kommentere og reflektere over nokre av svara. Ei slik måling gav meg kvantitative verdiar i det kvalitative forskingsarbeidet mitt, og hjelpte meg å utforme fokusgruppeintervjuet på ein måte som sikra at eg tok tak i dei rette elementa for å finna svar på problemstillinga mi.

Likertskjema vart utforma med utgangspunkt i problemstillinga og forskingsspørsmåla mine, og hadde rangerte verdiar som svaralternativ. I tillegg til spørsmål til kvart av forskingsspørsmåla, laga eg nokre meir generelle spørsmål om opplevingane elevane hadde med programmeringsøktene. Hensikta med desse spørsmåla var å stilla meg så open som mogleg, og få ei beskriving av korleis elevane opplevde øktene med programmering.

4.1.3 Observasjon og forskarrolle

Dei fire undervisningsøktene overtok eg klasseleiaransvaret i klassen: Som forskar hadde eg det Postholm kalla fullstendig medlemsrolle (2010, s. 66). Eg sette dei i gang med oppgåvene, og inviterte til diskusjonar i gruppene når dei stod fast. Postholm (2010, s. 66) skriv at når forskaren deltek i forskingsdeltakarane sine aktivitetar, kan det verka forstyrrende på kontakten at forskaren tek notat undervegs. Eg gjennomførte ikkje systematisk observasjon under øktene, men noterte meg korte feltnotat rett i etterkant av øktene, der eg såg noko eg ville fylgja opp vidare i intervjusituasjon. Slike observasjonar var meir inntrykk som kunne fylgjast opp vidare enn sjølvstendig materiale for drøfting og konklusjonar. Dei har vore med å meg i utforminga av intervjuguiden for gruppeintervjuet.

4.1.4 Hermeneutikk og fenomenologi

Omgrep som fortolking og meining er sentrale i hermeneutikken, som vert definert som «læren om fortolkningen av tekster» (Kvale & Brinkmann, 2015, s. 73). Alle data som vert omforma til skriftleg materiale, skal ifølge Postholm (2010, s. 99) oppfattast som tekst. Elevane og deira opplevingar er sentrale for forskinga mi. Dette krev at eg som forskar er open for tankar og erfaringar elevane deler, og at eg evnar å byggje vidare på dette med opp med relevante oppfylggingsspørsmål. Det innebør òg at eg som forskar klarar å forholde meg til den hermeneutiske sirkelen, og er medviten at den førforståinga og erfaringane eg har med

meg inn i studien, kan vera med på å farga tolkinga mi. Postholm viser til Gudmundsdottir (2001) i forklaringa av den hermeneutiske sirkelen som sjølvve kjerna i meiningsskapande og meiningsutviklande prosessar (Postholm M. , 2010, s. 178). Det var viktig for meg som forskar å vera tydeleg med elevane på førehand om at dette er eit forskingsprosjekt som skal bidra til at eg og andre får betre forståing for elevane sine opplevingar i møtet med programmering som verktøy i matematikkfaget.

Fenomenologiske studie skildrar meininga menneske legg i ei oppleving, knytt til ei bestemt erfaring av eit fenomen. Her vert dette knytt til opplevinga informantane har frå øktene med programmering i matematikk. I fylgje Postholm (2010, s. 43) er intervju vanlegvis den einaste strategien for datainnsamling som kan takast i bruk ved fenomenologiske studie. Eg var på jakt etter elevane sine eigenføyte opplevingar i øktene med programmering, og måtte stilla meg open for andre sine opplevingar. Då var det eit naturleg val for meg at det måtte bli gjennom intervju.

Innafor fenomenologien skil ein mellom eit sosiologisk og eit psykologisk, individuelt perspektiv. Postholm (2010, s. 41) forklarar at det som skil dei to retningane er at målet for den sosial-fenomenologiske tilnærminga er at forskaren undersøker grupper av individ og korleis dei bevisst utviklar meining i ein sosial interaksjon, medan i den individfokuserte tilnærminga er individet i fokus, og målet er å gripe enkeltmennesket sine opplevingar, men samtidig prøve å finne ut av om det same fenomenet vert opplevd av fleire enkeltindivid. Eg har ei psykologisk tilnærming på forskinga mi, med individ-perspektiv, samtidig som eg freistar å finne ut om enkeltindividet sine opplevingar gjeld for fleire. Ved bruk av fokusgruppeintervju, der fleire er med samtidig, kunne eg kanskje risikere at nokon ikkje våga å dele. Eit pluss er derimot at eg gjennom aktuelle emne og opne spørsmål, kan få elevane til å samtale og dermed dele tankar om noko dei elles ikkje hadde kome på, men som ein annan deltakar i fokusgruppa byrjar å snakke om. På den måten var det ein metode for å få enkeltindividet sine opplevingar drøfta av fleire.

4.2 Utval

Eg sende ut førespurnad til lærarar på 10.trinn i mitt nærrområde, og det enda med at eg fekk gjennomføre forskingsprosjektet i ein 10.klasse bestående av 17 elevar. Eg presenterte meg

for klassen, og gav klassen eit informasjonsskriv og samtykkeskjema som skulle signerast. Fire elevar i klassen fylgjer ikkje ordinær undervisning i matematikkfaget, men får undervisning med tilpassa kompetansemål i mindre gruppe. Desse elevane fekk informasjonsskriv, men ikkje samtykkeskjema. Dei valde sjølv å bli med i undervisningsøktene fordi dei var interesserte, men deltok ikkje i spørjeundersøkinga.

Før første økt hadde eg fått inn signert samtykke frå heile klassen, anten i papir, digitalt eller ein kombinasjon av desse, der føresette hadde svara digitalt og eleven på papir.

Eg laga eit eige felt for avkryssing i samtykkeskjema, der dei som var interesserte i å delta i fokusgruppeintervju, måtte krysse av for det. Eg håpa at fem til åtte elevar ville melde seg. Fem elevar valde å krysse av for at dei ville delta i fokusgruppa, og eg fann det tilstrekkeleg. I fylgje Halkier (2010, s. 38) kan det vera ein fordel med små grupper viss ein vil arbeide vidare med datamaterialet i djupna med omsyn på sosiale forhandlingar, språkleg uttrykk og meiningsdanning. Eit minus ved for lita gruppe kan vera at dei kan bli mindre dynamiske spesielt viss deltakarane liknar kvarandre. Dette var dermed noko eg måtte vera observant på i gjennomføringa.

4.2.1 Fokusgruppe

Ein av fordelane ved å bruke fokusgruppeintervju er, ifylgje Halkier (2010, ss. 13-15), at samhandling i gruppa kan gje kunnskap om kompleksiteten i sosiale praksisar som kan vera vanskelegare å få fram ved individuelle intervju eller styrte gruppeintervju. Ein annan fordel med fokusgruppeintervju kan vera at praten går lausare og dermed at refleksjonane til elevane kjem meir fram. I fylgje Halkier (2010, s. 31) er fokusgruppe spesielt eigna til å produsera data om innhaldsmessige betydingsmønstre i forteljingar, vurderingar og forhandlingar som ei gruppe menneske kan bli samde om, men mindre eigna til å få fram data om livsverda hos individa. Som kommentarar og utdjuping av resultat frå spørjeundersøkingane, var difor fokusgruppeintervju ein nyttig metode. Viss eg var meir oppteken av å finna ut om enkelt deltakarane sine personlege erfaringar og opplevingar aleine, kunne individuelle intervju vore eit betre val.

Fokusgruppeintervju sikra at eg både var open og mottakeleg, men òg kunne styre emna slik at eg sikra svar på problemstillinga. Intervju av fokusgrupper skil seg frå individuelle intervju og gruppeintervju. Kjenneteiknet på ei fokusgruppe er kombinasjonen av gruppesamhandling

og forskarstyrt emnefokus (Halkier, 2010, s. 10). Delar av fokusgruppeintervjuet eg gjennomførte, handla om å reflektere over resultatane av spørjeundersøkinga, der eg såg tendensar eg ville utdjupa. Eg innleia intervjuet med at det er tankane elevane har etter programmeringsøktene som var viktig for meg, og at elevane måtte kjenna seg fri til å sei det dei tenkte. For å sikra at eg fekk med meg alt, og samstundes ha fokus på samhandlinga i gruppa, tok eg lydopptak av intervjuet. På den måten kunne eg gje fokusgruppa full merksemd under intervjuet, noko som både viser interesse for det som vart sagt, men òg gav ei større opning for at eg fekk med meg kroppsspråk under sjølve intervjuet og eventuelt nøling eller toneleie ved transkripsjon av lydopptaket.

Dei fem informantane vart invitert til fokusgruppeintervju om deira oppleving av matematikkfagleg læring gjennom øktene med programmering som verktøy. Før eg landa på fokusgruppe, var eg inne på tankar om både individuelle intervju og gruppeintervju med semistrukturert intervjuguide. Fokuset mitt på elevane sine opplevingar, gjorde at eg ville stilla meg så open som mogleg, og det er noko fokusgrupper legg opp til. Hovudforskjellen på gruppeintervju og fokusgruppe, er ifylgje Halkier (2010, s. 9) kvar samspelet skjer. I eit gruppeintervju vil samspelet skje mellom intervjuar og intervjuobjekta, men i fokusgruppa skjer samspelet mellom intervjuobjekta, og intervjuaren har rolla som moderator som legg fram emna for diskusjon og ser til at alle kjem til ordet med refleksjonane sine.

Fokusgruppeintervju er ei open tilnærming, som stiller krav til forskaren. Eg hadde ein semistrukturert intervjuguide som utgangspunkt, men måtte vera budd på å gå vidare med uføresette utsegn og refleksjonar elevane kom med, og at analyse av intervjuet ville ta tid.

4.2.2 Rolla som moderator

I fokusgruppeintervju har eg som forskar rolla som moderator. Som moderator har forskaren i fylgje Halkier (2010, s. 57) to oppgåver: å få deltakarane til å snakke saman, og handtera den sosiale dynamikken mellom dei. Halkier (2010) viser til Puchta og Potter (2005), som presenterer eit sosialkonstruktivistisk perspektiv på fokusgrupper, og argumenterer for at moderator må handtera alle fire oppgåvene: 1. Sørge for at gruppa får eit uformelt preg, 2. Få deltakarane til å delta aktivt, 3. Sørge for at deltakarane uttrykker relevante meiningar og erfaringar, 4. Få deltakarane til å uttrykke så mange forskjellige meiningar og erfaringar som mogleg (Halkier, 2010, s. 58). Desse fire punkta skreiv eg inn i intervjuguiden for å sikra at eg gjennom heile intervjuet var medviten ansvaret mitt som moderator.

4.2.3 Intervju av fokusgruppa

Fokusgruppa mi bestod av fem informantar, derav ein gut og fire jenter. Sjølv om eg på førehand var medviten at det kunne vera utfordrande å få fem femtenåringar til å styre diskusjonane og refleksjonane meir eller mindre sjølv, kjende eg på at det til tider var utfordrande å få fram ulike refleksjonar. Eg måtte vera aktiv lyttar som stilte oppfølgingsspørsmål og passa på at alle deltok. Likevel vart det fleire gonger slik at når eg stilte oppfølgingsspørsmål der eg bevisst retta meg mot heile gruppa, var det den som hadde snakka før oppfølgingsspørsmålet som svara først. Kjensla eg hadde under sjølv intervjuet vart ikkje heilt stadfesta då eg hørde gjennom intervjuet i etterkant. Eg eg at det var kome nokre gode refleksjonar i gruppa, som eg kunne bruke til vidare tolking knytt opp mot forskningsspørsmål og problemstilling.

Med fokus på elevane sine opplevingar i programmeringsøktene, måtte eg syta for at emna elevane skulle reflektere over i fokusgruppeintervjuet kunne knytast til djupnelæring. For å sikra dette, laga eg ein semistrukturert intervjuguide med tema for intervjuet. Sjølv om eg ville at informantane i fokusgruppa skulle reflektere så fritt som mogleg, var det viktig at eg som moderator sikra at refleksjonane faktisk kunne nyttast til å svara på problemstillinga mi. Eg la difor inn oppfølgingsspørsmål knytt til læring, i tillegg til at tema kunne knytast til dei to forskningsspørsmåla mine. Intervjuguiden ligg i tabell 4 side 35.

4.2.4 Tema i fokusgruppeintervjuet

I tillegg til Survey- diagramma frå spørjeundersøkinga i klassen, hadde eg på førehand laga ein oversikt over tema knytt til forskningsspørsmåla mine. Denne tok utgangspunkt i både tidlegare forskning og i klassen sine svar i spørjeundersøkinga. Dessutan var eg budd på å fylgje opp med oppfølgingsspørsmål undervegs. Det var ikkje lett å førebu seg på oppfølgingsspørsmål, men der freista eg å knytte refleksjonar opp mot læring. Under intervjuet merka eg meg at ordet «gøy» var blitt brukt av fleire av deltakarane gjentatte gongar, og la difor inn eit tilleggsspørsmål om det omgrepet, og kva det kan ha å sei for læring.

Tabellen under (tabell 4) viser emna eg på førehand hadde bestemt meg for å utforske i fokusgruppeintervjuet. Under intervjuet kom det fram refleksjonar og omgrep som eg på førehand ikkje hadde tenkt gjennom, men som eg valde å utforske vidare fordi dei kan knytast

til tidlegare forskning som kan vera relevant for problemstillinga mi. Den gjentatte bruken av ordet «gøy» er eitt døme på dette. Eit anna er refleksjonane elevane hadde om å «få til noko». Ettersom desse omgrepa kan knytast til dei tre dimensjonane av djupnelæring, la eg til nokre spørsmål for å få fram elevane sine refleksjonar kring desse områda, med ordbruken «gøy» og «få til». Ettersom det var utfordrande å gjennomføra pilotintervju, gjorde eg ei avtale med informantane at me skulle møtast i intervjusituasjon ein til to gonger. På den måten sikra eg meg at viss intervjuguiden ikkje var tilstrekkeleg, eller eg ikkje fekk svar som kunne hjelpa meg å finna svar på problemstillinga mi, kunne eg justere tema og spørsmål og ta eit intervju til med fokusgruppa. Dette vart ikkje naudsynt, men det gav meg ein tryggleik i arbeidet.

Forskingsspørsmål	Emne for refleksjon	Aktuell oppfølging (?)
Generell oppleving og læring <i>Elevane si generelle oppleving av undervisningsøktene med programmering, og til læring (følge opp spørjeundersøking)</i>	*Refleksjonar knytt til korleis dei opplevde undervisningsøktene. *Refleksjonar knytt til læring under øktene. *Refleksjonar knytt til prøve/feile- metode. *Refleksjonar knytt til samarbeid. (jobbe i par) *Refleksjonar knytt til PRIMM (ferdig kode)	*Utdjupe *La alle svare
F1: <i>På kva måte ser elevane samanhengar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?</i>	*Kva tenkjer de om å bruke det de lærte i desse øktene til å løyse andre matematiske problemstillingar? *Kor motivert er de til å jobba meir med programmering i matematikkfaget?	*Kva tenkjer de om det å bryte ned til fleire delproblem? *Kva/korleis lærer du?
F2: <i>Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?</i>	*Reflekter over påstanden: programmering er eigentleg ein måte å tenka på som kan brukast i nesten alle samanhengar. *Kan du bruka tankegangen i programmering når du skal til dømes arrangera ein klassefest eller laga ein presentasjon i samfunnsfag?	*(styre refleksjonane unna matematikk)

Tabell 4 Intervjuguide for intervju av fokusgruppa

4.3 Transkribering av fokusgruppeintervju

Under fokusgruppeintervjuet nytta eg funksjonen «Taleopptak» på mobilen for å få lydopptak av intervjuet. Opptak gjorde meg friare under sjølve intervjuet til å kunne eg vera meir til

stades i samtala og refleksjonane til deltakarane i fokusgruppa, og eg kunne høyre gjennom intervjuet fleire gongar i ettertid. Ved sjølv transkriberinga, skrudde eg ned hastigheita på avspelinga, slik at eg fekk skriva meir fortløpande i takt med tale. I tillegg nytta eg spolefunksjonen ofte, for å sikra at eg fekk alt korrekt. Opptaket vart straks etter transkripsjon overført til skykonto med 2-trinnspålogging for å tryggja personvern.

I transkriberingsprosessen omgjorde eg dei munnlege refleksjonane til skriftleg datamateriale, noko som i fylgje Kvale & Brinkmann (2015, s. 206) gjer intervjuar betre eigna for analyse. Deretter såg eg etter meiningsfortetting i svara. Når ein skal trekke konklusjonar frå forskinga, er det viktig å knyte desse opp mot problemstilling og forskingsspørsmåla. I fylgje Postholm (2010, s. 91) går analyseprosessen føre seg både under og etter datainnsamling, og det kan bli aktuelt å finna kategoriar under forskingsprosessen som eg ikkje har tenkt ut på førehand, knytt opp til problemstillinga og forskingsspørsmåla mine.

4.4 Fenomenologisk reduksjon

Kvalitative analysemetodar har til hensikt å sortera datamaterialet for å gjera det meir forståeleg og rapportvenleg. Dette stiller krav til forskaren, som må ha klart for seg kva ein ser etter, allereie før innhenting av materiale. Postholm (2010, s. 86) beskriv kvalitative analyser som gjentatte og dynamiske prosessar, som startar med det fyrste intervjuet eller det første møtet forskaren har med forskingsobjektet, og aldri blir heilt avslutta.

I fenomenologisk analyse freistar forskaren å klarleggja meininga, strukturen og essensen av det erfarte, opplevde fenomenet (Patton, 2002 referert i Postholm, 2010, s. 98). I fenomenologiske studie vert ofte Moustakas (1994) sin modifiserte tilnærming av metoden, kalla Stevick-Colaizzi-Keen- metoden nytta (Postholm M. , 2010, s. 44). Metoden har ein firetrinns prosedyre for fenomenologisk reduksjon:

1. **Samle inn alt materiale** som er knytt til forskningsemnet: I forskinga mi var det resultatata frå spørjeundersøkinga og full versjon av transkripsjon av fokusgruppeintervjuet som utgjorde materialet, i tillegg til eit blikk på det som er gjort av tidlegare forskning på området.

2. **Horisontalisering**, som inneber at all data i første omgang vert handsama som likeverdige. Så skal ein utelate uttaler som er irrelevante for emne og problemstilling. Dette arbeidet gjorde eg ved å lytte gjentatte gongar til fokusgruppeintervjuet samt å lese transkripsjon av intervjuet. Noko av arbeidet vart òg gjort før intervjuet, då eg plukka ut dei mest relevante svara frå spørjeundersøkinga som eg ville ta med meg og få utdjupa av fokusgruppa. I dette arbeidet kom eg fram til tre overordna tema som eg ynskte å gå vidare med: Tydinga av omgrepet «gøy», tydinga av å «få til noko», og erfaringane dei deler om å arbeide med oppgåver som er bygd opp etter PRIMM-modellen. Eg markerte utsegn knytt til dei tre kategiane i tre ulike fargar i transkripsjonane av intervjuet.
3. Det tredje trinnet består i å **samle alle uttaler under ulike tema, og vidare organisera desse tema til ein samanhengande tekst, eller tekstuell beskriving av fenomenet, som inneheld det som er blitt erfart eller opplevd**. Dette arbeidet vert lagt fram i neste kapittel, som presentasjon av resultat. Her samla eg alle uttaler knytt til dei tre kategoriane, for å få sett dei meir i eit organisert system.
4. **Strukturell beskriving av fenomenet**. Her må forskaren finne ut korleis denne erfaringa er opplevd. Denne delen inneber meir analyse frå mi side, og eg som forskar måtte sjå elevane sine uttaler i lys av teori og tidlegare forskning og nasjonale rammeverk for å fortolke opplevingane dei snakka om. Dette arbeidet vert presentert i drøftingsdelen i kapittel 6.

Heilt til slutt vert denne tekstkuelle og strukturelle beskrivinga foreint i ein heilskapleg forklaring av essensen av fenomenet.

Eg brukte denne firetrinnsprosedyren i analysearbeidet mitt, fordi eg meiner han på ein fin måte legg opp til at eg tek omsyn til å få fram både einskilddeltakaren sine opplevingar og forskaren sine fortolkingar av utsegna frå deltakarane.

4.5 Kvaliteten på forskingsprosjektet

Ifylgje Postholm (2010, s. 169) er dei tradisjonelle krava til reliabilitet og validitet problematiske i kvalitativ forskning, sidan eit møte med forskaren og informanten alltid er ein unik, tidsbestemt situasjon. Reliabilitet handlar om reproduksjon av resultat, og passar difor ikkje inn i biletet med eit kvalitativt intervju der intervjuobjekta sin sensitivitet er ein variabel faktor (Postholm, 2010, s. 169).

4.5.1 Reliabilitet /pålitelegheit

Reliabilitet har med forskingsresultata sin konsistens og pålitelegheit å gjera, og vert ofte handsama i samanheng med spørsmålet om eit resultat kan verta reproduisert seinare av andre forskarar (Kvale & Brinkmann, 2015, s. 276). Sidan dette ikkje samsvarar med logikken i kvalitative intervju, og det ikkje er mogleg å gjenta eit identisk intervju, skriv Postholm (2010) at det er meir vanleg å erstatte omgrepet reliabilitet med pålitelegheit i fenomenologiske studie. Ifylgje Postholm (2010, s. 170) er forhold som kan trua pålitelegheita til forskingsprosjektet mitt: at informanten prøver å stille seg i eit «godt lys» og dermed ikkje alltid svarar ærlig, at informanten legg vekt på positive opplevingar, og undertrykker dei negative, og at eg og informanten kjenner til og tolkar omgrep ulik, som kan føra til mistydingar.

I arbeidet mitt sikra eg pålitelegheita ved å presentere likert-diagramma for fokusgruppa, og i transkripsjonsprosessen spela eg gjennom lydopptaka mange ganger, både medan eg transkriberte og etter at intervjuet var ferdig transkribert, for å sikra at det var samsvar mellom tekst og tale. Informantane i fokusgruppa fekk lese gjennom det transkriberte intervjuet og stadfeste at transkripsjonane var rett i forhold til det som var sagt og meint i intervjuet.

4.5.2 Validitet

Validitet handlar om å forsikra seg om at ein konkret undersøker det som ein har sett føre seg å undersøkje (Halkier, 2010, s. 127). Halkier (2010, s. 127) skriv at validitet vidare inneber at dei teoretiske omgrepa og kategoriane og metodiske verktøya som er brukt, heng saman med kunnskapsinteressene i prosjektet, noko som igjen stiller krav til at eg stiller dei rette spørsmåla i spørjeundersøkinga og set dei rette emna for fokusgruppeintervjua. Dessutan inneber validitet at fortolkingar og analyser òg til ein viss grad harmonerer med det empiriske feltet.

Validiteten har eg sikra ved å presentera funn og resultat i kapittel 5, for så å analysere desse opp mot problemstilling og forskingsspørsmål i kapittel 6. Ikkje alle funn var som eg hadde forventa, men dei var likevel aktuelle for ein vidare analyse basert på teoretisk rammeverk og tidlegare forskning presentert i kapittel 2 og 3.

4.6 Etiske refleksjonar

Kvale og Brinkmann kallar intervju ei moralsk undersøking (2015, s. 95), og forklarar dette med at det er knytt moralske spørsmål både til intervjuundersøkinga sine midlar og mål. Dette vert forklart med at det menneskelege samspelet i intervju påverkar intervjupersonane, og kunnskapen som vert produsert i intervju, påverkar synet vårt på situasjonen. Å sikre at deltakar opplever respekt og tryggleik, er viktige faktorar for å sikre kvalitet i tråd med etiske prinsipp. Eg har fylgd retningslinjene til Den nasjonale forskningsetiske komitè for samfunnsvitenskap og humaniora (NESH, 2021), med spesielt fokus på det Befring (2016, s. 31) legg fram som dei viktigaste normene: Informert og fritt samtykke, konsidensialitet, særleg omsyn til born og utsette grupper, og aktsamheit for deltakingrisiko.

I dei neste delkapitla gjer eg greie for korleis eg har sikra informert og fritt samtykke i 4.6.1, og konsidensialitet og anonymitet og aktsamheit for deltakingrisiko i 4.6.2.

4.6.1 Informert og fritt samtykke

Ifylgje Befring (2016, s. 31), skal informantane ikkje berre vera informert, dei skal òg ha forstått informasjonen. Eg sendte ut og gjekk gjennom eit generelt informasjonsskriv og samtykkeskjema til heile klassen (vedlegg 2 og 3). I tillegg vart det sendt digitalt til foreldra. Her fekk elevar og føresette informasjon om lagring av personopplysningar, og stadfesta aktiv deltaking i intervju og refleksjonar, godkjenne opptak av intervjuet samt få informasjon om lagring av personopplysningar. Informasjonsskjemaet gav òg informasjon om at deltaking er anonym og konsidensielt. Sjølvne spørjeundersøkinga i SurveyXact var utforma slik at dei måtte krysse av for å delta i undersøkinga for å koma vidare i spørjeskjemaet. Elevane som i tillegg ville inngå som informantar i fokusgruppa, måtte krysse av for dette i samtykkeskjema, som måtte signerast av elev og føresette.

4.6.2 Konsidensialitet og anonymitet og aktsamheit for deltakingrisiko

Eg var medviten risikoen for at elevane kunne føla eit behov for å «svare rett» eller imponera meg som forskar. Eg starta difor intervjuet med å informera deltakarane om at det er

informasjonen og refleksjonane dei kjem med som er viktig, og at deltakinga var anonym. Informantane i fokusgruppa vart informert om at eg ville nytta fiktive namn på dei i teksten min, og at dersom nokon ville trekke seg frå prosjektet før det var ferdig gjennomført, var det deira rett å gjera det. Forskingsprosjektet vart meldt til NSD (Norsk senter for forskingsdata) der krava til etikk vart teke omsyn til i form av informasjonsskriv og samtykkeerklæring til alle involverte den 6. desember 2021, og vart vurdert i samsvar med personvernvurderinga den 6.januar 2022.

4.7 Utforming av undervisningsopplegg

Før undervisningsøktene laga eg klar fem undervisningsopplegg med estimert arbeidstid på 60 minutt inkludert innleiing og oppsummering. Eg brukte PRIMM- modellen som mal, og laga oppgåver som passa for kvart av trinna i modellen. Oppgåvene var utforma for å supplere kompetansemål i matematikk generelt, og ikkje dei spesifikke kompetansemåla der programmering inngår. Fire av oppgåvene var i temaet geometri, og ei i temaet statistisk analyse. Felles for alle oppgåvene er at dei kan bruke matematikk som dei har lært tidlegare til å løyse utfordringane. Undervisningsopplegga ligg som vedlegg 4-8.

4.8 Undervisningsøktene

Opplegget vart gjennomført i fire undervisningsøkter a 60 min over to veker. Klassen var informert på førehand om kva timar som skulle brukast til gjennomføringa. To av øktene var i matematikktimar, medan to av dei var opphavelig samfunnsfag på timeplanen.

Eg innleia fyrste økta med å sei at det var eit poeng at dei skulle snakke saman og programmere på éin PC. Klassen bestod no av 17 elevar, ettersom dei fire elevane som normalt har eiga matematikkundervisning, ville delta på dette. Dermed vart det sju par og ei tremannsgruppe.

Eg hadde funksjon som tilretteleggjar og igangsetjar av oppgåvene, og hjelpte til når elevar bad om hjelp, då helst ved å gje hint som kunne hjelpe gruppa vidare i arbeidet. I etterkant av siste økta fekk heile klassen eit kvantitativt spørjeskjema til utfylling. Dette vart gjennomført i ein klasstime på skulen. Resultata frå denne undersøkinga vart ein del av grunnlaget for fokusgruppeintervjuet eg gjennomførte veka etter.

4.9 Evaluering av metodeval

Val av metode har vore ein nøye gjennomtenkt prosess, og eg var innoom mange alternative metodealternativ før eg avgjorde at det måtte bli læringsøker fylgt opp gjennom fokusgruppeintervju. Fokusgruppeintervju er ein god metode når forskaren ynskjer å ha ei så open tilnærming som mogleg. Den kvantitative undersøkinga i klassen var eit innslag som skulle ha som funksjon å hjelpe meg å sjå eventuelle tendensar som eg burde få refleksjonar rundt i fokusgruppa.

Fokusgruppa mi bestod av fem personar. I fylgje Halkier (2010, s. 39), er der ein tommelfingerregel som seier at fokusgrupper bør bestå av seks til 10-12 personar, men at litteraturen viser til vellukka fokusgrupper som er både mindre og større enn dette, og at viktigare enn storleiken på gruppa, er det at ein vel rett i forhold til tema for intervju. Eg frykta litt at fokusgruppa mi var for lita, og vurderte å prøve å få eit par elevar til å bli med. Årsaka til at eg ikkje gjorde det, var at no hadde dei fem som var med, meldt seg frivillig, noko som kunne tyde på at dei var interessert og motivert til å delta. Dette ville eg ikkje risikere å øydelegge ved å lokke nokon som kanskje ikkje var interessert til å bli med, fordi eg skulle ha eit vist tal på intervjuobjekt. Denne avgjerda meiner eg var rett, for intervjuet gjekk fint.

Problemstillinga mi, med fokus på elevane sine erfaringar knytt opp mot djupnelæring, har bydd på nokre utfordringar. Det same gjeld forskingsspørsmåla der eg valde å fokusera på det å sjå samanhengar og å bruke i nye utfordringar. Elevane har lite kjennskap til djupnelæring som omgrep, og då dei skulle reflektere over tema knytt til forskingsspørsmåla om å sjå samanhengar og å bruke i nye utfordringar, svara dei fyrst ut frå om dei kunne bruka Python i andre fag og utfordringar, og ikkje ut frå om samanhengar innafor matematikkfaget eller frå matematikk til andre fag. Verdien av nokre av svara frå elevane i fokusgruppa var difor ved fyrste augekast liten, men etter å ha høyrd intervjuet og lese transkripsjonane nokre gonger, var der svar å ta tak i som kunne nyttast i ei analyse.

5.0 RESULTAT

Målet med forskinga mi er å finna ut korleis elevane si oppleving av djupnelæring er, etter økter med programmering med PRIMM- modellen i matematikkfaget. Som beskrive i kapittel 4.4, nytta eg Stevick-Colaizzi-Keen- metoden (Postholm M. , 2010, s. 44). Trinn tre i modellen består i å samle alle uttaler under ulike tema, og vidare organisera desse tema til ein samanhengande tekst, eller tekstuell beskriving av fenomenet. Eg har vald å dele dette kapittelet inn etter hovudkategoriane eg kom fram til. I delkapittel 5.1 presenterer eg funn knytt til kategorien «Gøy», før eg i kapittel 5.2 presenterer funn knytt til «få til». I kapittel 5.3 presenterer eg funn knytt til arbeidet med PRIMM- modellen.

Informantane i fokusgruppa her er gjeve fiktive namn for å sikra anonymiteten deira.

5.1 «Gøy»

Gjennom intervjuet nytta elevane ofte omgrepet «gøy» for å stadfeste at øktene med programmering var noko dei likte. Dette kjem fram allereie heilt i starten av intervjuet, då eg spurde korleis dei hadde opplevd øktene:

- Hedda: Det var gøy. Litt då. Å lage sånne former og sånt. Ikkje berre jobbe i boka.
- Erik: Konkurrerte om å klare å sjå kva som skjedde. Om me forstod kodane.
- Brita: Gøy å lære noko nytt. Gjera noko anna.
- Rebekka: Det eg tenkte fyrst var: Kvifor har me bruk for det? Og så no skjønar eg litt kvifor det kan vera lurt å lære.
- Anna: Ja, du var ikkje akkurat positiv i starten.
- Rebekka: Nei, men alle kan ikkje vera like positiv då!
- Hedda: Det var veldig gøy å få ting til. Når du klarte å laga ein perfekt sirkel... eller firkant. Fekk til oppgåva, då. Såg jo med ein gong om det stemte. Gøy utfordring som virka veldig vanskeleg fyrst, men så skjønte eg...

Litt ut i intervjuet, spør eg om kva dei tenkjer om at me programmerte i heilt andre matematikkfaglege tema enn det dei heldt på med i matematikkfaget på det tidspunktet. Spørsmålet gav litt delte synspunkt:

- Erik: Eg trur det er best å programmere i geometri når me heldt på med det i matte, og sånn median når me har om median og sånt i matten.
- Hedda: Men det er litt «digg» viss me har sånn algebra, og så er det masse stress, så er det bra å få ein pause frå det og heller programmere firkantar og sånt.
- Brita: Viss du greier å hugse kva du heldt på med i matten når du får programmering i temaet, så lærer me jo.

Omgrepet «digg» som Hedda nyttar her, vert brukt til å forklare at ho set pris på ei slik avveksling.

På spørsmål om dei var nyfiken på å bruka programmering meir i matematikk og andre fag, fekk eg svara:

- Erik: Fyrst var mange negative. Eg trur dei fleste i klassen blei positivt påverka av kor bra dei fekk det til.
- Hedda: Men eg trur òg det blei litt sånn, me hadde det i andre timar òg, og ikkje alltid i mattetimar. Så då blei det ekstra matte, og me har allereie masse matte.

Her svarar dei etter kor positivt eller negativt innstilt dei var. I det vidare utdjupar dei litt:

Hedda: Når det er gøy så er du jo meir interessert, og då er det lettare å lære.

Brita: Gøyare å gjera viss du skjønar kva du gjer. Matte er gøy når du forstår kva du jobbar med!

Heilt til slutt i intervjuet, då eg spurde kva dei tenkte om at dei fire elevane som normalt får spesialpedagogisk undervisning ute av klasserommet var med, nytta Erik ordet «gøy» her og:

Erik: Ja det var gøy at dei var med! Dei er jo ikkje i klassen så ofte, og aldri i mattetimane.

Dette utsegnet tek oss vidare til neste delkapittel der eg presenterer funn knytt til å få til noko.

5.2 «Få til»

Som det kjem fram i førre delkapittel, koplur ofte fokusgruppa kva som er gøy til det dei får til. Det ser me og av korleis elevane fylgjer opp Erik sitt svar om at det var gøy å ha dei andre elevane med på dette:

Hedda: Bra at dei kan trivast i klassen og få til det same som oss.

Rebekka: Sikkert bra følelse for dei.

Anna: Jonas var jo kjempegod! Han likar å sitte med koding og sånt. Har erfaring. Kunne meir frå før.

Tidleg i intervjuet tok Anna opp at ein såg om ein fekk det til:

Anna: Såg om me fekk det til. Det var masse av testing og finne ut om du fekk rett svar eller ei. Og så når du fann svaret til slutt so «åja». Det knytt til det.

Moderator: Tenkjer du på når du fann feil i koden?

Erik: Eg las det som stod i feilmeldinga og så prøvde eg meg fram. Og når eg til slutt fekk rett så «åhh», det var sånn, ein lettare måte å gjera det på.

På spørsmålet om korleis dei arbeidde med oppgåvene, fekk eg svar som tydeleg er knytt til geometrioppgåvene. Også her viser dei til at dei såg om dei fekk det til:

Hedda: Det va berre å prøve seg fram.

Erik: I starten var det sånn prøving og feiling men etterkvart fant me ut at me kunne rekne.

Brita: Ja, eg trur, -eg og Anna jobba ilag-, og me rekna litt, og så gjetta me litt, og så rekna me for å få det rett.

Hedda: Eg trur eg prøvde meg fram.

Erik: Pytagoras. Der kan du lære viss du først prøver og feilar, og så kan du sjå korleis du kan rekne for å få det til...

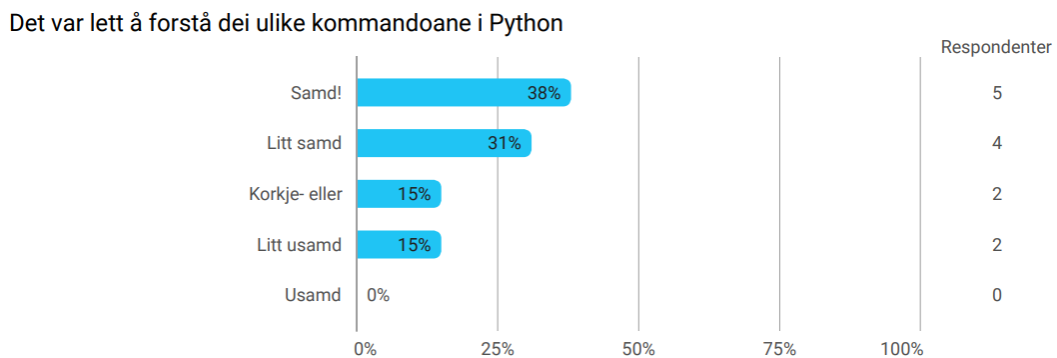
Då eg bad fokusgruppa reflektere rundt den fyrste økta og korleis det var å setja seg inn i dei ulike kommandoane i Python, svara dei:

Rebekka: Det gav liksom ikkje meining.

Hedda: Fekk litt sjokk. Har aldri gjort noko sånt, så fekk me et ark og skulle finna ut av.

Erik: Som då alle trudde det var ei trapp i staden for ein firkant...

Moderator: Kommenter klassen sine svar...(moderator legg fram klassen sine svar på påstanden «det var lett å forstå dei ulike kommandoane i Python (figur 5)).



Figur 6 Påstand: Det var lett å forstå dei ulike kommandoane i Python

Rebekka: Nokon fekk det til og nokon sat litt lengre.

Hedda: Litt difor eg ikkje var så gira.

Då eg bad informantane reflektere over påstanden frå spørjeskjema: «Programmering er eigentleg ein måte å tenkje på som kan brukast i nesten alle samanhengar», kom det ettertenksamt frå Erik:

Erik: Først var mange negative. Eg trur dei fleste i klassen blei positivt påverka av kor bra dei fekk det til.

Her snakkar han utifrå kor bra dei fekk det til, og viser til den tekniske biten i Python. Resten av fokusgruppa fylgjer opp og stadfestar dette:

- Hedda: Er jo sikkert mogleg. Eg meiner...viss du får det til.
- Hedda: Ville vore rart å skulla liksom finna ut kvifor 2. verdskrig skjedde med å bruka Python liksom
- Rebekka: Ja, då hadde me heller gått ifrå å lære oss noko om 2. verdskrig til korleis me skal bruka...ehh
- Erik: Ja, liksom det tekniske

Seinare i intervjuet, er Erik likevel inne på eit svar som kan lenkast til påstanden:

- Erik: Når me lagar sånne punkt. Det er jo likskap. For eksempel når me lager ein PowerPoint-presentasjon. Då deler du temaet i små segment, og så deler du det opp i fleire småsegment.
- Moderator: Korleis lærer de av den prosessen?
- Erik: Ja eg føler eg går meir i djupna då. Kva er hovudpunkta og kva er hovudsmåsegmenta.

Litt uti intervjuet gav eg elevane ei ny oppgåve bygd opp etter PRIMM- modellen, som dei skulle løyse munnleg, utan å faktisk gjennomføre det på PC (vedlegg 8). Oppgåva inneheldt kommandoen «in range 4» som elevane ikkje var kjent med frå før. Etter litt diskusjonar i gruppa, kom dei fram til at det tyder at det neste skulle gjentakast fire gongar. Det neste trinnet i modellen er «investigate» og oppgåva som låg der bad elevane undersøkje koden og vurdere om noko kunne vore koda annleis.

- Brita: Kunne skrive alt fire gongar i staden for å bruke linje 3.
- Erik: Ja, viss du vil gjera det meir avansert
- Brita: Eg trur det er lettare for folk å forstå viss du skriv det fire gongar.

Brita meiner ein heller burde skrive alt fire gongar, og at det ville vore lettare for folk å forstå. Erik på si side, har her fokus på sjølve koden, og at ei slik endring ville gjort koden meir avansert.

På trinnet «modify» i modellen, skal dei gjera naudsynte endringar for å programmere ein regulær femkant. Etter å ha fått avklart kva ein regulær femkant er, spurde eg dei korleis dei ville gått fram for å løyse denne oppgåva:

- Erik: Eg trur me får det til viss me endrar til 5 på linje 3 in range (5) og side 5. Då har me det!
- Brita: Gradane. 90 blir for masse.
- Hedda: 120 eller noko.
- Brita: Nokon og 60 eller noko...?
- Moderator: Viss de skulle gjort denne no, korleis ville de gått fram for å finne ut av det?
- Anna: Prøvd meg fram.
- Erik: Ja, det er jo berre å prøva og feila!
- Brita: Eg trur eg ville kanskje... prøvd litt. Men under 90 grader for det MÅ det jo vera. Men så når eg fann det ut so ville eg tenkt på det, og hugsa det til neste gong, trur eg.
- Erik: Ja, då hadde eg i alle fall hugsa det!
- Hedda: Ja, då hadde eg sikkert forstått det. Når eg fekk det til i Python, liksom.

Eit av spørsmåla eg la inn under intervjuet, var basert på førre kategori «gøy». Det var eit område eg ville finna ut meir om kva dei la i. Eg spurde difor kva dei tenkjer at det at det er

gøy å jobbe med noko, har noko å sei for om dei får det til? Delar av svara på spørsmålet ligg under kapittel 5.1 «Gøy». Dei utdjupa vidare:

Anna: Var jo litt frustrasjon i starten... Og så blei det gøy!

Rebekka: Ja, eg trur det, for du sit igjen med det at når du var så irritert over at du ikkje fekk det til, så blei det gøy når du fekk det til, og så tenker du at dette hugsar du.

5.3 Å arbeide med oppgaver bygd opp etter PRIMM- modellen

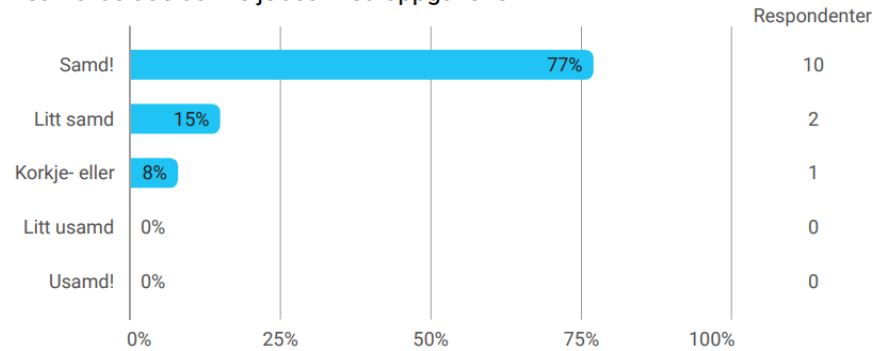
PRIMM-modellen er ein del av problemstillinga mi, der eg vil finna ut korleis elevane opplever at programmering etter PRIMM- modellen bidreg til djupnelæring i matematikkfaget. Det var difor naturleg å byggja opp nokre av tema i spørjeundersøkinga og fokusgruppeintervjuet med spørsmål om erfaringane dei opplever etter å ha arbeidd med programmering på denne måten.

5.3.1 Samarbeid

Det fyrste temaet knytt til PRIMM-modellen, handla om samarbeid, og påstanden i spørjeskjemaet var: «Det hjelpte å vera to som samarbeidde då me jobba med oppgåvene».

Klassen sine svar er vist i figur 7. Eg bad fokusgruppa kommentere med eigne refleksjonar.

Det hjalp å vera to som samarbeidde då me jobba med oppgåvene



Figur 7 Påstand: Det hjalp å vera to som samarbeidde då me jobba med oppgåvene

Hedda: Eg syns det var lettare å jobba med fleire.

Rebekka: Det var jo heilt nytt, så det hadde vore litt skummelt å sete der heilt aleine.

Erik: Det hjelper å ha et nytt blick på kodinga, korleis kan du sjå dette på ein annan måte.

Hedda: Sånn viss du kan litt, men ikkje alt, og så er det nokon andre som kan noko du ikkje kan, so kan du liksom setja det saman på ein måte.

5.3.2 Å byrje med ein ferdig kode

På påstanden om at det å byrja med ein ferdig kode gjorde det lettare å forstå, nikkar alle samtykkande til Rebekka sin kjappe respons:

Rebekka: Det blei jo lettare. Viss du berre hadde levert ei oppgåve at me skulle lage ein firkant, hadde me berre sete der som eit spørjeteikn. Vertfall eg!

Då dei fekk ei ny og ukjent oppgåve som var bygd opp etter PRIMM- modellen, viser elevane at dei har kjennskap til måten å jobba på, og kjem raskt i gang med dei ulike trinna i

modellen, med tilhøyrande oppgåver. Eg fylgde opp Rebekka sitt svar, og spurde om dei no, etter å ha arbeidd på denne måten i nokre veker, ville ha handtert å gå rett på å lage ein ny kode:

Erik: Nei.

Hedda: Veit ikkje.

Brita: Trur eg hadde huska litt meir og kanskje greidd litt.

Hedda: Hadde klart litt, fekk jo på ein måte formel. Usikker på korleis du starta. Korleis du får pennen ned på papiret.

Erik: Må liksom memorera nokre sånne standardar. Er ikkje heilt klar for å jobba heilt på eigenhand.

6.0 DRØFTING

Postholm definerer deskriptiv analyse til å omfatte analyseprosessar som strukturerer datamaterialet og gjer datamaterialet oversikteleg, forståeleg og rapportvenleg, og teoretisk analyse som inneber at forskaren tek i bruk substantiv teori for å analysere delar av materialet (Postholm M. , 2010, s. 86). Eg drøftar funna i lys av teoretisk rammeverk, tidlegare forskning og styringsdokument. For å gjera datamaterialet meir oversikteleg og rapportvenleg, har eg vald å freista å finna kategoriar som kan hjelpa meg å forstå kva elevane legg i omgrepa «gøy» og «få til» og på kva måte dette kan hjelpa meg å svara på problemstillinga og forskingsspørsmåla i forskinga mi.

P: Korleis opplever elevane at programmering med undervisningsopplegg etter PRIMM-modellen bidreg til djupnelæring i matematikkfaget?

F1: På kva måte ser elevane samanhengar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?

F2: Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?

Eg har identifisert tre hovudkategoriar frå informantane for vidare drøfting: Informantane sine gjentatte uttrykk om «gøy» ser ut til å vera noko eg kan knyte til forskingsspørsmål 1, medan kategorien «få til» kan knytast til forskingsspørsmål 2. Den tredje kategorien eg gjekk vidare med, var oppgåver bygd opp etter PRIMM- modellen.

Djupnelæring er ikkje eit kjent omgrep for elevane. Eg må difor tolke elevsvar frå deira ståstad, og å knyte dei opp mot dei ulike dimensjonane av djupnelæring. Eg nytta fenomenologisk reduksjon, og gjekk frå det informantane kalla «gøy» og «få til», samt refleksjonane deira frå å arbeide med PRIMM- modellen, opp mot dei to forskingsspørsmåla mine, samt definisjonar og forklaringar av djupnelæringsomgrepet.

I kapittel 3.3 gjorde eg greie for omgrepet «djupnelæring» slik det vert forklart av Utdanningsdirektoratet (2020), som det «å *gradvis utvikle kunnskap og forståelse av begreper, metoder og sammenhenger i fag og mellom fagområder*» og vidare inndelingane i dei tre dimensjonane kunnskap, dugleik og haldning. Det å synast det er «gøy» å arbeide med programmering eller å «få det til», er ikkje nødvendigvis noko som sikrar læring i matematikkfaget. Eg har vald å strukturera drøftingskapittelet etter forskingsspørsmåla i forskingsprosjektet mitt, med underkapittel der eg drøftar korleis elevane sine refleksjonar rundt «gøy» og «få til» kan gje svar på desse.

I delkapittel 6.1 tek eg for meg forskingsspørsmål 1 og refleksjonar frå elevane som kan analyserast i retning for å gje svar på dette. Elevane sine opplevingar av at programmering er «gøy» kan vera med på å gje svar på dette spørsmålet, spesielt når ein ser «gøy» opp mot kunnskapsdimensjonen og dugleiksdimensjonen av djupnelæring.

Delkapittel 6.2 tek føre seg forskingsspørsmål 2 kva elevane kan leggja i det å «få til noko» og kva dette kan ha å seie for elevane si læring. «Få til» drøftar eg opp mot haldningsdimensjonen av djupnelæring.

I delkapittel 6.3 drøftar eg elevane sine utsegn knytt til PRIMM- modellen spesielt samarbeid opp mot sosiokulturell læringsteori og tidlegare forskning.

I delkapittel 6.4 kjem ei oppsummering av drøftingane.

6.1 F1: Å sjå samanhengar innafor matematikktemaet og til andre fagområde

På direkte spørsmål om å sjå samanhengar, svara Erik kjapt at då dei fekk den siste oppgåva, såg dei at dette var noko heilt anna enn dei oppgåvene dei hadde jobba med tidlegare. Han held fram at sidan mange no forstod korleis det fungerte (Python), kunne dei ta det i bruk i andre typar oppgåver. Svaret hans peikar meir på om han kan bruke den tekniske dugleiken å programmere i andre tema, og eventuell gjenbruk av ein kode viss han får nye oppgåver. Då eg prøvde å utdjupe spørsmålet, gjekk refleksjonane i gruppa meir over på om dei bør jobbe med programmering medan dei held på med det matematikkfaglege temaet i matematikk, eller om det bør gjerast som eit avbrekk utan omsyn til det dei jobbar med i faget. Det herska semje om at det kanskje var lettast å programmere når dei heldt på med det same temaet, men samtidig «digg» å gå ein avbrekk frå det dei eigentleg held på meg i matematikkfaget.

Sjølv om òg desse refleksjonane startar med den tekniske biten å programmere, kjem Brita og Hedda til slutt inn på at viss dei har matematikken med seg, altså viss dei er medviten det matematiske temaet, kan dei lære av å programmere. Då eg gjekk nærare inn på kva elevane faktisk gjer når dei programmerer, at dei bryt det ned i små delar og tek det bit for bit, reflekterer dei over det, og Erik kjem etter kvart inn på at det er på ein måte det han gjer når han lagar ein PowerPoint i samfunnsfag, fordi han må bestemme seg for det han kallar hovudsegment og småsegment. Han meiner sjølv at han då går meir i djupna, deler opp eit tema i dei viktige deltema han må ha oversikt på. Erik viser her eit teikn på at han beveger seg fekk frå den tekniske utfordringa med programmering, og ser at det handlar om å bryte ned eit problem til fleire delproblem, noko som er eit teikn i positiv retning når det gjeld algoritmisk tenking og vegen mot djupnelæring: Samanlikninga til å lage ein PowerPoint om eit stort tema i samfunnsfag er god med tanke på nedbryting og deretter systematisk og stegvis tilnærming til heilskapen. Dette kan sjåast som eit ledd i utvikling av kunnskapsdimensjonen i djupnelæring:

6.1.1 Dugleiksdimensjonen og kunnskapsdimensjonen

Voll (2019) forklarar dugleiksdimensjonen med prosedyrar og strategiar som må øvast inn. Eit døme kan vera å kunna bruka Pytagoras' læresetning i praksis. Dette har elevane jobba med i matematikkfaget tidlegare, og gjennom programmeringsoppgåvene der elevane valde å

rekne, var det Pytagoras' læresetning dei praktiserte. Det at Erik uttrykker at han meiner ein kan lære Pytagoras' gjennom å programmere, kan ha samanheng med den eine oppgåva dei arbeidde med, men det er òg eit teikn på at han har sett nytteverdien i å støta på ei utfordring i programmeringsaktiviteten, og at han såg lenkja til matematikk generelt, og spesielt til Pytagoras' læresetning som her kunne hjelpa han å løysa utfordringa.

Kunnskapsdimensjonen handlar om å stegvis utvikle kunnskap og varig forståing (Gamlem & Rogne, 2019). Informantane sine uttaler om at når det er «gøy», er dei motiverte til å jobba vidare med oppgåvene, anten ved prøving og feiling eller ved å rekne. Dette kan tyda på at elevane har meir uthald og løysingsvilje når dei arbeidar med programmering. Då kan programmering med oppgåver bygd opp etter PRIMM- modellen vera eit bidrag inn mot kunnskapsdimensjonen fordi oppgåvene legg til rette for at elevane vil arbeide med oppgåvene over tid, der elevane ser på og evaluerer dei ulike delkomponentane av programmet. Informantane sine refleksjonar på dette området, stemmer godt overeins med intensjonane med PRIMM- modellen, som var å minske startvegring og sikra motivasjon hos elevane (Sentance et al., 2019).

Det at informantane seier dei er motiverte når dei arbeider med programmering i matematikkfaget, kan vera med på å gjera dei meir uthaldande og dermed auke sjansen for den stegvise utviklinga av kunnskap og forståing. Informantane svara òg at når dei fyrst hadde fått rett «output» i programmet, undersøkte dei koden for å sjå korleis dei kunne ha rekna det, noko som igjen kan auke sjansane for ein varig forståing, sidan dei sjølv meiner at dei vil hugse det, fordi dei har eit klart bilete av det og vil hugse «prøve- og feileprosessen». Dette kan vera med på å gje programmeringsaktiviteten ein eigen verdi i faget, som dei ikkje får under «vanleg rekning», som elevane kallar ordinære matematikkøktar. Alle programmeringsøktene klassen har arbeidd med, har matematikk innafor tema elevane har arbeidd med i ordinær undervisning førre skuleår. Det at dei hugsar at dei har gjort noko slikt, gjorde kanskje programmeringsøktene til ei repetisjonsøkt som sette i gang tankane til at dei har litt forkompetanse som måtte hentast fram. Dette kan ein sjå i lys av forskinga til Psychiaris og Kallia (2017), som konkluderte med at programmering kan ha positiv effekt på elevane si meistringskjensle og evna til å resonnera. Når informantane mine kjenner meistringskjensle, resonnerer dei ved å ta i bruk matematikk som dei har jobba med tidlegare, eller ved å leite fram ny kompetanse som kan hjelpa dei å løysa den utfordringa dei står ovanfor, slik dei gjer når dei undersøker koden etter å ha fått han rett.

Sjølv om det kjem tydeleg fram at tekstprogrammering i Python var noko heilt nytt for elevane, og dei ofte snakkar om «den tekniske biten» framfor læring i matematikkfaget, er der òg klare teikn til at dei meiner dei lærer betre når det er lystbetonte aktivitetar, og etter kvart som dei fekk litt taket på det dei omtalar som «den tekniske biten» var det ein arbeidsmåte dei likte. Voll (2019) omtalar Kunnskapsdimensjonen som fakta, modeller, teori og lovar som må organiserast. Oppgåvene elevane jobba med, kravde at elevane anten henta fram gammal kunnskap i geometri, eller sette seg inn i det på ny. Det at elevane trong denne kunnskapen for å «få det rett» som dei seier, kan ha vore med på å organisere kunnskapen då dei aktiverte han i programmeringsarbeidet. Informantane seier sjølv at viljen til å klare å løyse oppgåvene var stor, noko som kan vera eit teikn på at dei opplever at PRIMM- modellen har fungert etter sin hensikt for dei i dette arbeidet, og at programmeringsarbeidet har bidrege til utvikling av både dugleiksdimensjonen og kunnskapsdimensjonen av djupnelæring.

6.1.2 «Gøy» å «få det til»

Hedda: Når det er gøy så er du jo meir interessert, og då er det lettare å lære.

Hedda meiner ein vert meir interessert, og lærer lettare, når det ho arbeidar med er «gøy». Òg fleire andre gonger kjem det fram at elevane meiner det er gøy å gjera noko anna enn det dei pleier. Erik var fleire gonger i intervjuet opptatt av at det utvikla seg konkurrans mellom gruppene om å vera dei som først såg kva som vart «output» i PRIMM-fasen «predict». Både «gøy» og konkurrans kan knytast til motivasjon i den forstand at det gjev farge eller glød til oppgåvene ein held på med. Det er noko som får ein til å halde fokus og ville fortsetje med oppgåvene. Desse utsegna samsvarar med konklusjonane i Ueda (2021) sin kasusstudie, der elevane spesielt sette pris på den fortløpande responsen som gjorde at dei følte seg meir kreative når dei programmerte enn med «vanleg matematikk». Den fortløpande responsen er òg nemnt av Sevik (2016) som eit av argumenta for programmering i skulen.

Uttaler som kan knytast til «gøy» var gjennomgåande i fokusgruppeintervjuet. Sjølv om eg ikkje hadde utforma spørsmål direkte knytt til motivasjon, vinkla informantane refleksjonane

mot kva som var «gøy». Dette kom allereie på første spørsmålet då eg bad dei reflektere over korleis dei hadde opplevd øktene:

- Hedda: Det var gøy. Litt då. Å lage sånne former og sånn. Ikkje berre jobbe i boka.
- Erik: Konkurrerte om å klare å sjå kva som skjedde. Om me forstod kodinga.
- Brita: Gøy å lære noko nytt. Gjera noko anna.

Diskusjonane mellom informantane tyder her på at dei ville ha prøvd seg fram til dei såg at dei fekk det til. Det verkar òg som dei etter at dei har fått det til, ville ha studert koden for å finna matematikken. Brita er usikker på kor mange grader vinklane i ein regulær femkant har, men ho brukar matematikk til å resonnerer seg fram til at dei må vera under 90° . Viss ho hadde plotta koden i programmet, ville ho ha oppdaga at resonnementet hennar var feil. Utifrå det ho seier, kan det vera rett å anta at ho då ville justert seg ved å prøve seg fram, og etter kvart når ho fekk det til, ville ho i alle fall ha hugsa kor mange grader vinklane i ein regulær femkant har. Det kan òg vera dette Erik viser til når han seier at då hadde han hugsa det seinare. Hedda sluttar seg til dette resonnementet, og uttaler at når ho har fått det til i Python etter prøving og feiling, så ville ho sikkert ha forstått det. Dette støttar opp om Stenseth et al. (2019) sine argumentasjonar for at feilretting har ein eigenverdi i seg sjølv. Feilretting kan seiest å koma til syne på to område i forskingsprosjektet mitt: For det første ligg det inne som ein del av PRIMM- modellen, der elevane i steget «modify» skal justera koden slik at han får ein annan output. Vidare kan informantane sine utsegn om «prøving og feiling» som dei sjølv seier dei valde som metode, sjåast på som ein form for feilretting. Dei prøvde seg fram, undersøkte feilmeldinga, og prøvde igjen. Ser ein dette opp mot Bloom's taksonomi (Agarwal, 2019), kan ein sei at matematikkoppgåvene elevane fekk, ikkje starta på lågaste kognitive nivå, men på trinna anvende eller analysere altså i grensa mellom overflatelæring og djupnelæring. Slik informantane forklarar at dei har jobba, kan det sjå ut som dei brukte ulike metodar for å «få det til», men når dei fekk det til, ville dei ha forstått det. Når informantane seier at «då ville dei forstått det» og «då ville dei hugsa det til seinare» kan dette vera teikn på at dei meiner dei då hadde fått ei varig forståing av temaet, og at dei kan ta det i bruk ved nye utfordringar. Gjennom ei praktisk oppgåve i programmering, står dei midt i taksonomien, og

gjer det dei kan for å løyse utfordringa. Dette kan vera å henta fram kunnskap og forståing som ligg på eit lågare nivå i taksonomien. Når utfordringa er løyst, vil dei danne seg knaggjar som gjer at dei veit korleis dei skal løyse ei liknande utfordring i framtida. Arbeidsmetodane elevane refererer til, kan her vera eit døme på Paperts prinsipp (Minsky, 1988): Dei opplever mental vekst ved å tileigne seg nye administrative måtar å bruke det dei allereie veit. Når dei møter utfordringar dei treng verktøy for å løyse, er gammal kunnskap verktøyet, og det at dei føler at dei får reelle knaggjar å feste kunnskapen til, gjer at dei vil hugse det for framtida.

Gjennom intervjuet fekk eg inntrykk av at elevane hadde eit sterkt ynskje om å «få det til» når dei programmerte, og at den kjappe tilbakemeldinga dei får i Python gjorde dei uthaldande i arbeidet. Dette kom òg fram i temaet om dei lærte matematikk medan dei programmerte. I spørjeundersøkinga i klassen var dette den påstanden som fekk mest spreing i svara. Då fokusgruppa fekk dette spørsmålet, slutta elevane seg til Erik sitt utsegn:

Erik: Pytagoras. Der kan du lære viss du først prøver og feilar, og så kan du sjå korleis du kan rekne for å få det til....

Det kan sjå ut som motivasjonen ligg i å få til programmet, og at viss dei har prøvd og feila litt før dei fekk det til, vil dei studere koden for korleis dei kan rekne for å få det til. Det dei i praksis gjer då, er å leite etter samanhengar som dei kan ta med seg vidare. Nettopp desse samanhengane er det eg ser etter i forskingsspørsmål 1 som handlar om elevane si oppleving av å sjå samanhengar i og mellom fagområde i matematikken.

Dermed kan det at elevane tykkjer det er «gøy» å jobbe med programmering i matematikkfaget, vera eit godt bidrag for F1: å sjå samanhengar innafør matematikktemaet og til andre fagområde.

6.1.3 Den algoritmiske tenkjaren og «prøve- og feilemetoden»

Programmering som aktivitet er gjennomgåande arbeidsmåtane som er beskrivne i den algoritmiske tenkjaren i kapittel 1.4.2: Dei fiklar, skapar og søkjer etter feil, i tillegg til at

aktivitetane stiller krav til uthald og samarbeid (Utdanningsdirektoratet, 2019). Nøkkelomgrepa som ligg til den algoritmiske tenkjaren, logikk, algoritmar, dekomposisjon, mønstre, abstraksjon og evaluering, er aktivitetar oppgåvene i seg sjølv kan innby til. Sidan elevane her melder at dei i stor grad nytta seg av ein metode der prøving og feiling var sentral, er det interessant å sjå at sjølv om dei prøvde seg fram, nytta dei litt matematikk i resonnementa sine for kva dei skulle prøve, kva steg dei skulle byrje med. Prøve og feile kan sjåast på ein ei form for fikling, slik det er omtala i den algoritmiske tenkjaren (Utdanningsdirektoratet, 2019). Informantane fortel at når det vart feil, såg dei nøye på koden (feilsøking) og justerte «inputen» for å «få det rett». I denne prosessen aktiverer dei algoritmar og dekomposisjon, der dei tek for seg delkomponent for delkomponent. Fleire gonger i intervjuet svarar elevane at når dei «fekk det til» studerte dei koden for å sjå korleis ein kunne rekne for å få det rett. Utifrå svara i fokusgruppa, kan det sjå ut som etter nokre økter, var det ikkje berre prøving og feiling som var metoden dei nytta seg av. Då eg spurte korleis dei gjekk fram i oppgåva med sirkelen (oppgåve 3, vedlegg 6), svara dei slik:

Erik: Markus rekna for oss der. Og så fekk me han til slutt til

Brita: Me fekk den til. Me rekna!

På oppfølgingsspørsmål om rekning versus prøve og feile, kom dei med ulike svar:

Hedda: Det var berre å prøva seg fram.

Erik: I starten var det sånn prøving og feiling men etterkvart fann me ut at me kunne rekna.

Brita: Ja, eg trur, -eg og Anna jobba ilag-, og me rekna litt, og så gjetta me litt, og så rekna me for å få det rett.

Hedda: Eg trur eg prøvde meg fram.

Refleksjonane tyder på at dei i alle fall nytta evaluering av svara sine til å reflektere over matematikken i programmet. Hedda held fram at ho prøvde seg fram, medan Erik, Brita og

Anna alle har veksla litt mellom rekning og prøving, men at dei sjølv meiner det har vore ei utvikling der dei etter kvart valde å prøve seg med matematikken. Utifrå svara frå informantane mine, ser det ut som den algoritmiske tenkjaren har vore aktiv i arbeidet med programmering; dei har fikla, søkt etter feil, lett etter mønstre, dekomponert og ikkje minst har dei vore uthaldande i samarbeidet. Alle svara ser ut til å ha ei kopling mellom det å prøve seg fram i det tekniske programmet, teste ulike svaralternativ, og bruke «outputen» dei får som hint vidare. Deretter vil dei prøve å finne ut korleis dei kan rekne seg fram til det, slik at dei har ein metode neste gong dei møter ei liknande utfordring.

Dette er teikn på at arbeidet med programmeringsoppgåver basert på PRIMM- modellen kan vera eit viktig bidrag for F1 ved at elevane gjennom algoritmisk tenking ser samanhengar både innafor det aktuelle matematikkfaglege temaet og til andre tema i skulematematikken.

6.2 F2: I møte med nye utfordringar

I spørjeundersøkinga i klassen hadde eg påstanden «Programmering er eigentleg ein måte å tenkje på som kan brukast i nesten alle samanhengar». Denne påstanden var meint å trigga den algoritmiske tenkjaren og måten å jobbe systematisk bit for bit. Utifrå refleksjonane i fokusgruppa, ser det ut som elevane tolka spørsmålet som at dei skulle programmere i Python i alle fag. Det at informantane her går langt inn i ein diskusjon om korleis dei skulle ha brukt Python til å finne årsaker til 2. verdskrigen, tyder på at dei har teke spørsmålet bokstaveleg til å gjelde å bruke Python, og ikkje eit tankesett som kan brukast i dagleglivet eller i arbeid med andre fag. Ein samheng Erik ser, er at når han lagar ein PowerPoint, jobbar han litt på same måte som når han programmerer, i og med at han bryt det heilskapen ned i delar. Det Erik seier om å gå meir i djupna ved å definere hovudpunkta og hovudsegmenta kan tolkast som ein prosess der ein dekomponerer og deler opp heilskapen i mindre delar, slik det er beskrive om algoritmisk tenking (Utdanningsdirektoratet, 2019).

6.2.1 Haldningsdimensjonen

Informantane mine seier at dei lærer meir når dei føler at dei får det til. Dei svarar òg at dei analyserer koden for å sjå korleis dei kunne ha rekna, noko som òg stadfestar Psychiaris og

Kallia (2017) sine funn om at programmering med PRIMM har positiv effekt på resonneringsevne. Dette tyde på at viss oppgåvene er medvite bygd opp til at elevane skal sjå samanhengar, og knyta program og kode til matematikkfaget, kan det ha positiv effekt på alle dei tre dimensjonane av djupnelæring, men spesielt på haldningsdimensjonen.

Haldningsdimensjonen av djupnelæring handlar om å utvikla gode haldningar, evne til refleksjon og kritisk tenking (Gamlem & Rogne, 2019). Det at informantane fleire gonger i intervjuet konstaterte at dei fekk ei meir positiv innstilling til oppgåvene når dei såg at dei fekk det til, kan vera eit teikn på at det at dei fekk det til, kan vera eit positivt bidrag for haldningsdimensjonen. Fyrst skjøna dei ikkje vitsen, men så fekk dei det til, og då gav det meining. Fleire utsegn frå informantane tyder på at den kjappe tilbakemeldinga i Python var med på å gje meistringskjensle fordi då var det «berre å sjå på feilmeldinga og korrigere koden», så fekk dei det til:

Erik: Eg las det som stod i feilmeldinga, og så prøvde eg meg fram, og så når eg til slutt fekk rett.

Eit siste moment som er verdt å merke seg når det gjeld å få det til, er at dei fire elevane som ikkje følgjer ordinær undervisning i matematikkfaget, deltok ivrig i øktene med programmering. Fokusgruppa kommenterte dette:

Hedda: Bra at dei kan trivast i klassen og få til det same som oss.

Rebekka: Sikkert bra følelse for dei.

Rebekka snakkar her om meistringskjensla dei fire elevane må ha kjent på, når dei fekk til det same som resten av klassen. I dette tilfelle var det heilt utan tilpassing av oppgåvene frå mi side, men det stettar likevel resultata frå forskingsprosjektet til Sentance et al. (2019), der lærarane trakk fram spesielt at PRIMM- modellen er ein god undervisningsmetode for å undervise i blanda klasser.

Haldningsdimensjonen av djupnelæring, har vist seg å bli eit av kjernefunna i forskinga mi. Informantane snakkar om at det vart «gøy» når dei «fekk det til». Både det at det er «gøy», og ein «får det til», kan knytast til djupnelæring. Voll (2019) forklarar at haldningar som må liggje til grunn for læring, er motivasjon og tru på egne evner. Desse heng saman akkurat som informantane sine samanhengar mellom «gøy» og «få det til». Informantane forklarar at i starten, då dei ikkje fekk det til, var det mykje frustrasjon, men når dei fekk det til, var det «gøy», og dei likte å jobbe med det. Dette igjen gav motivasjon og uthald til vidare arbeid.

Spesielt med tanke på haldningsdimensjonen, kan det sjå ut som det at oppgåvene var bygd opp etter PRIMM- modellen, har spela ei viktig rolle for F2 om korleis elevane møtte dei nye utfordringane. PRIMM- modellen senka inngangsterskelen og bidrog til at elevane fekk kjenne på meistring på eit tidleg stadium i programmeringsopplæringa. Dette kan ha bidrege til at dei òg fekk meir tru på egne evne og motivasjon til å jobba vidare med fagstoffet i Python.

6.2.2 Den algoritmiske tenkjaren som «får det til»

Gjennom heile intervjuet oppfattar eg at informantane ser ein samanheng mellom meistring og haldningar. I starten av den fyrste økta omtalar informantane seg som umotiverte, og hadde vanskar med å koma i gang med oppgåvene. Dei forklarar det med at dei ikkje skjønnte vitsen og hadde jo aldri gjort noko slikt før. Dette kan tolkast som at dei kjente seg usikre og at dette igjen spela inn på haldningane dei hadde til å jobbe med oppgåvene. I intervjuet forklarar dei det med at det var mykje frustrasjon i starten, men at når dei lukkast, var det ei god kjensle.

Brita svarar at matte er gøy når du får det til.

Den algoritmiske tenkjaren (Utdanningsdirektoratet, 2019) må vera systematisk og analytisk i arbeidet sitt, og ikkje minst, vera skapande, eksperimenterande og open for alternative løysingar. I tillegg må han innsjå at å gjera feil undervegs er ein viktig del av prosessen, og ha strategiar for å oppdaga at noko er feil og retta opp i feila (Utdanningsdirektoratet, 2019).

Informantane i fokusgruppa er fleire gongar inne på korleis dei prøvde seg fram, eller brukte prøve- og- feilemetoden til å koma fram til rett svar:

Anna: Såg om me fekk det til. Det var masse testing og finne ut om du fekk rett svar eller ei. Og så når du fann svaret til slutt så: «åja».

Denne måten å arbeide på kan ved fyrste augekast verka lite akademisk, då dei ikkje refererer til matematikk eller annan teori for korleis dei arbeidde. Likevel, når ein ser det i lys av den algoritmiske tenkjaren (Utdanningsdirektoratet, 2019), kan ein seie at elevane gjennom denne måten å arbeide på, er innom mange av trekka til den algoritmiske tenkjaren: Gjennom «masse testing» eksperimenterer elevane seg gjennom ulike trinn for å «sjå om dei fekk rett svar». Ein kan anta at dette er gjort systematisk, og analytisk. Det at Anna seier at når du fann svaret til slutt, så «Åja», kan vera eit teikn på at der var ein viss form for analyse og vilje til å lære noko om kvifor det var nettopp slik det skulle gjerast. Eg tolkar her «åja» som at når dei «fekk det til», så gjekk det eit lys opp for dei om kvifor denne løysinga var rett.

Informantane sine refleksjonar kring «å få det til» stadfestar delvis det Psychiaris og Kallia (2017) kom fram til i undersøkinga om elevane sitt læringsutbytte. Undersøkinga deira indikerte at programmering har positiv effekt på meistringskjensle og evne til å resonnera, men mindre effekt på problemløysing. Informantane i fokusgruppa mi stadfestar dette i refleksjonane sine om at det var gøy når dei fekk det til.

Effekt på problemløysing har ikkje vore del av denne studien, men svara frå informantane kan tyde på at dei kjenner seg meir budd på å møte nye utfordringar når dei får det til, fordi dei gjennom algoritmisk tenking då analyserer og dekomponerer for å finna logikk dei kan bruke seinare. Dette er eit positivt funn som er med på å svare på Forskingsspørsmål 2 om dei kan bruke det dei har lært i heilt nye utfordringa.

6.3 Programmering med PRIMM

Alle undervisningsopplegga var bygd opp etter PRIMM- modellen, som er utvikla med utgangspunkt i det sosiokulturelle læringssynet. Nokre av spørsmåla i spørjeundersøkinga og tema for intervjuet, var knytt direkte til denne måten å arbeide på. På spørsmåla knytt til oppgåvetypen der dei får utlevert ein ferdig kode som utgangspunkt, var klassen sine svar

eintydig på at det å byrja med ein ferdig kode gjorde det lettare å forstå. Informantane i fokusgruppa støtta dette:

Rebekka: Det blei jo lettare. Viss du berre hadde levert ei oppgåve at me skulle lage ein firkant, hadde me berre sete der som eit spørjeteikn. Vertfall eg!

Eg var medviten at dette svaret kan hengja saman med at undervisningsøktene mine var elevane sitt fyrste møte med programmering i Python, og fylgde opp med spørsmål om dei trur dei kunne greidd ei slik oppgåve no.

Brita: Eg trur eg hadde huska litt meir og kanskje greidd litt.

Hedda: Hadde klart litt, fekk jo på ein måte forma. Usikker på korleis ein startar. Korleis du får pennen ned på papiret.

Erik: Må liksom memorera nokre sånne standardar. Er ikkje heilt klar for å jobba heilt på eigenhand.

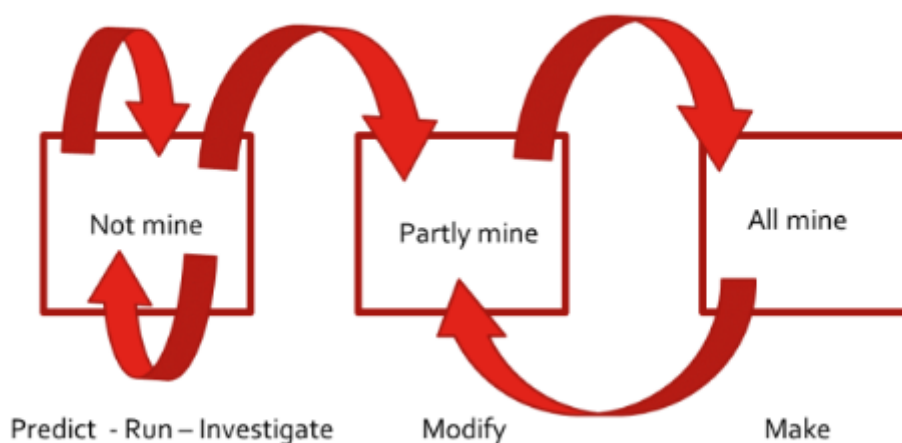
På spørsmål om dei etter kvart kunne ha klart det, for eksempel i siste økta, utan å fyrst få ein ferdig kode å undersøke, er dei litt spreidd i svara. Medan nokon trur dei kunne klart litt, verker andre litt meir usikker. Både Hedda og Erik viser til det tekniske når det gjeld standardar og korleis ein skal få pennen på papiret. Erik snakkar i tillegg om å hugse nokre standardar, òg her snakkar han i forhold til programmeringsspråket Python, og ikkje til matematikkfaglege standardar.

Når eg freistar å tolke desse svara, må eg sjå det i lys av at svara kan vera farga av den manglande erfaringa dei har frå tekstprogrammering. Igjen er informantane litt låst i konteksten at det tekniske var nytt for dei. Svara deira stør opp om det både Harms (2016) og Ericson et al. (2017) har vore inne på om at programmeringsoppgåver der ein senkar vanskegraden i starten, slik at elevane ikkje må skrive koden frå botn, kan gje like stort læringsutbyte for elevane. Det at terskelen er låg, sikrar at elevane kjem i gang og jobbar med

oppgåvene. Skal ein dømme etter informantane mine sine svar, ville dei ikkje kome i gang utan støtta frå PRIMM, og då ville ein heller ikkje kunna snakke om læringsutbyte. I dei neste delkapitla drøftar eg PRIMM- modellen sitt bidrag i læreprosessen, der eg fyrst går inn på elevane sitt eigarskap til koden i 6.3.1, før eg i 6.3.2 drøftar PRIMM- modellen som artefakt i eit sosiokulturelt perspektiv.

6.3.1 Eigarskap med PRIMM- modellen

Informantane og klassen er samde om at det gjorde det lettare å ha ein ferdig kode som utgangspunkt, og at dei ved å gå inn i koden kunne gjera dei rette endringane. Viss ein ser på steget «modify» i PRIMM- modellen, der elevane får i oppgåve å gjera naudsynte endringar i koden, kan dette steget samanstillast litt med feilretting som Stenseth et al. (2019) meiner me må fokusera meir på. Det er slikt arbeid Stenseth et al. (2019) meiner kan underbyggje resonnement som inneber hypoteseformulering og hypotesetesting. I presentasjonen min av PRIMM- modellen i kapittel 3.4, presenterte eg modellen som syner korleis elevane stegvis tek eigarskap til ein ferdig presentert kode. Modellen syner at på steget «modify» i PRIMM- modellen, har eleven eit delvis eigarskap til koden, fordi det er her dei gjer endringane med utgangspunkt i ein ferdig kode. Eg ser på modellen igjen (figur 8) for å tolke elevane sine læringsprosessar.



Figur 8 Eleven sitt eigarskap til koden med PRIMM- modellen

Informantane sine refleksjonar tyder på at det er dette punktet eller denne delen av programmeringsarbeidet dei meiner er «gøy» å «få til». Gjennom resonnering og «prøve- og feile- metode» tek dei delvis over eigarskapet av koden, og det er ei god kjensle når dei klarar det. Det at dei får eigarskap til koden, og dermed til sjølve utfordringa som skal løysast, kan òg vera med å spela ei rolle for den stegvise tilnærminga til djupnelæring. Som eg har sett på i både kapittel 6.1 og 6.2, refererer elevane ofte til at viljen til å «få det til» var stor, og det var «gøyare» med programmering enn «vanleg rekning». Dette seier meg at det kan vera slik at dei ved vanleg rekning kanskje ikkje får eigarskap til oppgåva på same måte som det dei gjer når dei programmerer. Dette kan igjen sjåast i samanheng med Bloom's taksonomi (Agarwal, 2019): Det at matematikkoppgåva er bygd opp slik at elevane ikkje byrjar på lågaste kognitive nivå, men på eit høgare nivå der det kan vera naudsynt å hente fram det dei kan anvende av kunnskap frå lågare kognitivt nivå, kan spela ei rolle. Elevane står midt i ei utfordring dei kjenner eigarskap til, og vil løyse denne. Det kan vera eit bidrag inn mot det å gje oppgåvene meining, som er noko elevane ofte etterspør.

Det kan likevel sjå ut som dei ikkje kjenner seg heilt klar til å lage ein eigen kode som er heilt deira, utan å ha noko å ta utgangspunkt i. For desse elevane, som er nybyrjarar innan tekstprogrammering, ser det ut som dei meiner det er «gøy» å programmere med oppgåver som er bygd opp etter PRIMM- modellen, men det er ikkje sikkert dei hadde meint det same utan den støtta dei får frå PRIMM.

Dermed kan ein sei at det at oppgåvene var bygd opp etter PRIMM- modellen, og det at elevane får eigarskap til utfordringa dei skal løyse, spelar ei stor rolle for korleis elevane opplever læreprosessen.

Sett opp mot sosiokulturelt læringsperspektiv, ser ein her at ikkje berre datamaskina og programmeringsspråket Python, men òg PRIMM- modellen for korleis oppgåva er bygd opp, ser ut til å fungere som ein medierande artefakt (Strandberg, 2008) for elevane på dette stadiet. I neste delkapittel ser eg nærare på kva samarbeid og sosiokulturell læring har betydd for elevane i programmeringsøktene.

6.3.2 Samarbeid og sosiokulturell læring

PRIMM- modellen er bygd opp med utgangspunkt i Lev Vygotsky sin sosiokulturelle læringsteori, der språk og kommunikasjon er ein viktig del av læreprosessen (Primmportal, 2022). På spørsmål knytt til det å samarbeide i par, er både klassen og informantane i fokusgruppa samde om at det var ein god måte å jobbe på. Informantane mine knyt parsamarbeidet tydeleg til meistring og det at dei utfyller kvarandre i det dei kan, og kan setja det saman.

Hedda: Eg syns det var lettare å jobba med fleire

Rebekka: Det va jo heilt nytt, so hadde vore litt skummelt å sete der heilt aleine

Erik: Det hjelper å ha et nytt blick på kodinga, korleis kan du sjå dette på ein annan måte

Hedda: Sånn viss du kan litt, men ikkje alt, og så er det nokon andre som kan noko du ikkje kan, so kan du liksom setta da samene på ein måte.

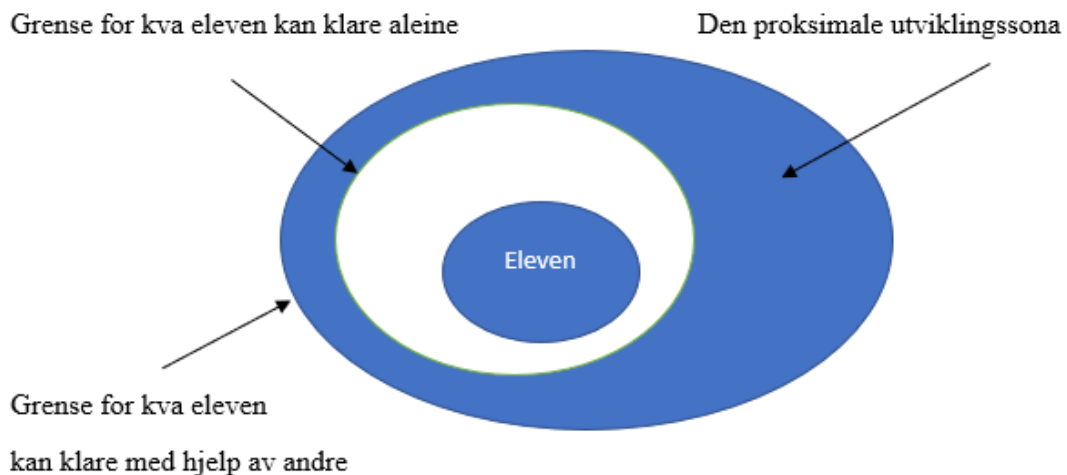
Svara tyder på at informantane meiner der er klare grenser for kva dei kan klare aleine, og at dei saman med ein medelev kan få til meir, og sjå ting frå ulike perspektiv. Rebekka seier i tillegg at det hadde vore «skummelt» å jobbe med det aleine. Dette svaret er kanskje eit teikn på at ho òg kjente på at medeleven hadde ein støttefunksjon som gjorde at Rebekka òg fekk meir tru på eigne refleksjonar gjennom samarbeidet. Med den støtta ho hadde frå programmet som gav svar undervegs og medeleven som både støtta og bidrog med eigne innspel, bevega Rebekka seg frå det ho kan klare aleine og ut i den proksimale utviklingssona for det ho kan klare med støtte og hjelp frå andre (Kluge, 2021, s. 42).

Sjølv om informantane på mange måtar knyt svara sine direkte til den tekniske programmeringsjobben, finn ein ytringar som tyder på at svara deira kan knytast til det reint matematikkfaglege òg.

Erik: Markus rekna for oss der. Og så fekk me han til slutt til.

Brita: Me fekk han te. Me rekna!

Uttalane er svar på spørsmålet mitt om korleis dei hadde gått fram for å løyse oppgåva der dei skulle programmere ein innskriven sirkel i kvadrat (oppgåve 3, vedlegg 6). Svara frå informantane kan tyde på to ting: For det fyrste har det gått føre seg eit samarbeid i gruppene. I Erik si gruppe kan det sjå ut som ei arbeidsdeling, ettersom han seier at det var Markus som rekna. Han er likevel tydeleg i neste setning, der han seier «og så fekk me han til slutt til». Sjølv om det var Markus som rekna, og Erik ikkje sjølv gjennomførte denne prosessen, ser det ut som han likevel føler at han har vore med på laget og var med på å få til oppgåva. Svara hans tyder på at han har kjent eigarskap til oppgåva, jamfør kapittel 6.3.1 «Eigarskap med PRIMM- modellen.» Både her, og i dei innleiande kommentarane om at det hadde «vore skummelt aleine» kan ein sjå informantane sine opplevingar i lys av Vygotsky si proksimale utviklingssone. Elevane ser ut til å ha ei oppleving av at sidan dei jobba saman i par, greidde dei meir enn det dei hadde gjort dersom dei skulle gjera alt på eiga hand, anten det var på grunn av at dei fekk fagleg støtte som Erik snakkar om, eller, for Rebekka, emosjonell støtte fordi «det hadde vore skummelt aleine».



Figur 9 Den proksimale utviklingssone

Det kvite feltet i modellen markerer horisonten for kva ein elev kan oppnå på eiga hand, medan det ytste blå feltet synleggjer den proksimale utviklingssona for kva eleven kan klare med hjelp frå andre. Ser me på PRIMM- oppgåvene elevane har jobba med, kan det sjå ut som dei sjølv opplever at dersom dei skulle ha jobba aleine med oppgåvene, hadde dei heldt seg i den kvite sona aleine, men saman med ein makker, har samarbeidet bidrege til at dei kunne utfylle og hjelpe kvarandre fordi der ein var usikker på noko, hadde kanskje den andre eit anna blick på det.

Informantane ser ut til å ha litt ulike refleksjonar kring dette med samarbeid: Medan Rebekka meiner det hadde vore skummelt å sitja aleine med oppgåvene, har Erik og Hedda fokus på at dei ved samarbeid kan utfylle kvarandre på kva dei meistarar, anten ved «nye augo» eller ved at nokon andre får til noko av det du sjølv ikkje klarar. Sjølv om grunngevingane for at dei set pris på samarbeidsforma er ulik, ser det likevel ut til at alle refererer til at dei føler støtte og tryggleik ved å vera to. Säljö (2008, s. 107) forklarar at menneske lærer gjennom å delta i praktiske og kommunikative samspel med andre, og at måtane me resonnerer om og tolkar den verkelegheita me møter i interaksjon på, brukar me som ressursar for å forstå og kommunisera i framtidige situasjonar. Säljö (2008) forklarar her Vygotsky sin teori om at barns utvikling opptrer på to plan. Informantane mine fortel om det dei fekk til då dei jobba i par, der dei opplevde læring på det sosiale planet. Vidare seier dei at dei når dei får det til ved programmering, vil dei analysere koden og finne ut korleis dei kunne ha rekna, slik at dei kan bruka det i seinare utfordringar. Denne delen av aktiviteten kan seiast å vera på det psykologiske planet, fordi måten ein tolkar og resonnerer i ein interaksjon, blir ressursar for å forstå og kommunisera i framtidige situasjonar.

Informantane er klare på at det at samarbeid er ein del av oppgåvene, har vore avgjerande både for korleis dei har kome i gang med oppgåvene og for kor mykje dei har meistra. Vidare fortel dei om prosessar der dei prøver å analysere svara sine når dei har fått rett i koden, slik at dei kan bruke det i seinare utfordringar. Ser ein desse samla, kan det sjå ut som elevane ikkje berre fann tryggleik fagleg og emosjonelt for å få løyst oppgåvene, men at det òg har fungert slik at dei har resonnert på det psykologiske planet for å sikra varig forståing. Slik kan det sjå ut som det at oppgåvene er bygd opp med fokus på sosiokulturell læringsteori, er med på å sikra djupnelæring i matematikkfaget.

6.4 Oppsummering av drøfting

I dette kapitlet har eg drøfta funn opp mot teori, tidlegare forskning og rammeverk, der eg i kapittel 6.1 og 6.2 freista å gje meining til det informantane mine omtalar som «gøy» og « få til» og kva betydning det kan ha for læreprosessane deira. Vidare har eg i kapittel 6.3 drøfta kva informantane tenkjer om oppgåver som er bygd opp etter PRIMM- modellen.

«Gøy» er i drøftingane mine blitt knytt til kunnskapsdimensjonen og dugleiksdimensjonen av djupnelæring. Som beskrive i kapittel 3.3 djupnelæring (figur 7, side 21), er desse viktige prosessar for robuste mentale nettverk, og utgjer saman med haldningsdimensjonen ein heilskap av djupnelæring. Haldningsdimensjonen, som handlar fyrst og fremst om motivasjon og trua på egne evne, kunne kanskje òg vore knytt til «gøy», men den ser eg fyrst og fremst tydeleg kjem til uttrykk når informantane snakka om å «få til» og om PRIMM- oppgåver.

Drøfting av funna i forkinga mi, har dermed gjeve nokre positive svar på problemstillinga om elevane opplever djupnelæring i matematikk etter å ha arbeidd med programmering med PRIMM. I neste kapittel gjev eg ei avsluttande oppsummering der drøftingane vert knytt direkte til problemstilling og forskingsspørsmål

7.0 AVSLUTNING

Gjennom dette forskingsprosjektet har eg freista å finna svar på problemstillinga:

P: Korleis opplever elevane at programmering med undervisningsopplegg etter PRIMM- modellen bidreg til djupnelæring i matematikkfaget?

I dette kapittelet vil eg gje ei oppsummering av dei funna eg har gjort, før eg trekk fram kva avgrensingar studien har. Avslutningsvis ser eg på kva som er vegen vidare og kva vidare forskning på dette feltet bør ha fokus på.

7.1 Konklusjon

Forskingsspørsmåla som har hjulpet meg å finna svar på problemstillinga har vore:

- F1: «På kva måte ser elevane samanhengar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?»
- F2: «Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?».

Dei to forskingsspørsmåla dekkjer viktige element i djupnelæring: I F1 ser eg etter kva elevane tenkjer om å sjå samanhengar innafor matematikkfaget, medan F2 fokuserer meir på læring som overføring til andre utfordringar. I tillegg til dei to forskingsspørsmåla, har det vore naturleg å tolke funn opp mot det å arbeide med oppgaver bygd opp etter PRIMM- modellen, for å kunna svare på heile problemstillinga mi. I dette kapittelet oppsummerer eg hovudfunn knytt til F1, F2 og PRIMM, før eg legg fram konklusjon på funn knytt til problemstillinga P1.

7.1.1 F1: På kva måte ser elevane samanhengar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?

Spesielt det informantane fortel om at dei prøvde seg fram og når dei fekk det til, undersøkte dei koden for å finne ut korleis dei skulle ha rekna, er teikn på at elevane ser på programmering som ei nyttig støtte når dei skal lære matematikk og sjå samanhengar i faget.

Informantane mine svarar positivt på at dei kan bruke «dette» i mange samanhengar. Dei viser i tillegg til at den siste oppgåva handla om noko heilt anna (statistikk) enn dei første (geometri) men no hadde dei skjønt det. Sjølv om elevane her snakkar om det tekniske programmet, kan det reknast som positivt at dei ser at noko dei har lært på eitt område, kan koma til nytte på eit anna område. Dette ser ein òg av svarea deira om korleis dei arbeidde: Dei veksla litt mellom prøve- og feilemetode og rekning, med forklaringar på at når dei fekk det til ved å berre prøve seg fram, ville dei undersøkje korleis dei kunne ha rekna, og då ville dei hugse! Dette er eit teikn på at dei gjennom programmering får trening i å hente fram den kunnskapen som er relevant for den utfordringa dei står ovanfor. Arbeidet er då med på å utvikle kunnskapsdimensjonen og dugleiksdimensjonen av djupnelæring: Dei utviklar stegvis kunnskap gjennom å prøve seg fram og korrigere seg sjølv, noko som bidreg til kunnskapsdimensjonen. Vidare analyserer dei svaret når dei får det til, slik at dei hugsar det til neste gong. Då jobbar dei med dugleiksdimensjonen som handlar om innøving av strategiar og modellar. Det at dei seier at viss dei får det til når dei programmerer, vil dei hugse det, er eit teikn på at dei meiner at dugleiksdimensjonen vert styrka meir når dei får det til ved programmering enn ved «vanleg rekning».

7.1.2 F2: Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?

Kategorien av funn som har hjelpt meg å svare på dette forskingsspørsmålet, handlar om å «få til noko». Denne kategorien har eg knytt til at det at elevane føler seg trygge og «får det til», har positiv effekt på haldningane til aktiviteten, og spelar igjen inn på korleis dei møter nye utfordringar. Dersom elevane er motiverte til å jobbe med noko fordi dei «får det til», og dei kjenner meistring, kan dette ha ein positiv effekt i nye utfordringar. Dei ser òg at ein måte å tenkje på, kan brukast i andre samanhengar. Informantane gjev òg tydelege svar på at dei set

pris på den kjappe tilbakemeldinga dei får når dei programmerer, og at dei brukar feilmeldinga til å korrigere seg sjølv. Dette er teikn på at programmering, på grunn av dei kjappe tilbakemeldingane, bidreg til at elevane vågar å prøve seg i nye utfordringar, for viss det blir feil, er det berre å rette opp i feilen.

Programmeringsøktene i forskingsprosjektet mitt var ei heilt ny utfordring då eg møtte elevane fyrste økta. Dei har sjølv beskrive kor blanda kjenslene var då, og refererer til at «dei hadde aldri gjort noko slikt før». Etter kvart som dei kom i gang og «fekk det til» var haldningane endra. Utifrå dette kan det sjå ut som programmeringsøktene har hatt positiv effekt på haldningsdimensjonen i djupnelæring, og når motivasjon og trua på eigne evne er på plass, er dette eit viktig grunnlag som må vera på plass for evnene til å overføre det du har lært i ei utfordring og bruke det når du møter heilt nye utfordringar.

7.1.3 Oppgåver med PRIMM- modellen

Det kan sjå ut som PRIMM- modellen bidreg til ein positiv haldningsendring hos elevar som i utgangspunktet kjenner låg meistring og motivasjon, fordi modellen legg til rette for ein bratt læringskurve med låg inngangsterskel.

I tillegg har eg fått tydelege svar frå elevane om at det å senke inngangsterskelen ved å bruke PRIMM- modellen, fungerte som ei god tilnærming i denne gruppa med lita eller inga erfaring frå programmering spesielt fordi modellen inviterer til utforsking i steget «modify» utan at han krev at elevane byrjar frå botn med ein gong. PRIMM- modellen ser ut til å ha positiv effekt av fleire årsaker:

1. **Låg inngangsterskel** fordi det er lettare å byrje med ein ferdig kode og ta utgangspunkt i denne i dei neste stega av oppgåva.
2. Elevane sine svar støttar idèen i PRIMM-modellen om at dei får **kjapt eigarskap til oppgåva eller utfordringa** dei skal løyse, noko som er med på å gje arbeidet mening.

3. **Parsamarbeid** vert referert til som positivt både fordi det gjev emosjonell støtte og tryggleik, og fordi elevane i par kan utfylle kvarandre fagleg og teknisk.

7.1.4 P1: Korleis opplever elevane at programmering med undervisningsopplegg etter PRIMM- modellen bidreg til djupnelæring i matematikkfaget?

Problemstillinga mi har vore tredelt: For det fyrste handlar forskinga om læring i matematikkfaget. For det andre, på kva måte programmering med PRIMM- modellen bidreg til djupnelæring. Den tredje delen av problemstillinga, og kanskje den største, handlar om djupnelæring. Eg ville ikkje berre måle læringseffekt, men korleis elevane opplever læring som kan knytast til omgrepet djupnelæring. Det har difor vore naturleg med ein brei definisjon på djupnelæring, og resultatata frå spørjeundersøking og fokusgruppeintervjuet er vorte knytt opp mot programmering som eit middel for algoritmisk tenking og djupnelæring.

Gjennom forskingsspørsmåla som er knytt til djupnelæring, der F1 handlar om å sjå samanhangar innafor matematikkfaget og F2 handlar om overføring av kunnskap og dugleikar frå eitt område til eit heilt anna. Dei tre dimensjonane av djupnelæring, kunnskapsdimensjonen, dugleiksdimensjonen og haldningsdimensjonen har vore til hjelp for å identifisera uttaler frå informantane som kan knytast til forskingsspørsmål. I dei tre føregåande delkapitla har eg gjeve greie for korleis informantane mine sine opplevingar kan knytast til forskingsspørsmåla og PRIMM- modellen sin oppbygning. Saman er desse med på å gje svar på problemstillinga om korleis elevane opplever at programmering med undervisningsopplegg etter PRIMM- modellen bidreg til djupnelæring i matematikkfaget. Funna kan oppsummerast slik:

- Elevane er tydelege på at dersom aktiviteten er «gøy» lærer dei betre, fordi dei vert meir uthaldande og vil halde fram med aktiviteten over lengre tid. I drøftinga mi har eg sett på det at elevane meiner at dei lærer betre når ein aktivitet er «gøy» er eit godt utgangspunkt for elevane si utvikling av kunnskap, fordi det at dei vil halde ut med aktiviteten over lengre tid kan sikre den stegvise utviklinga av forståing, som inngår i kunnskapsdimensjonen i djupnelæring.

- Elevane er tydelege på at matematikk og programmering er gøy når dei «får det til», noko eg tolkar som at ein låg inngangsterskel slik PRIMM- modellen inviterer til, kan vera nyttig for at elevane skal koma i gang med og halde fram med programmeringsarbeidet. I drøftinga mi såg eg nærare på korleis det å «få det til» spelar ei positiv rolle når det gjeld haldningsdimensjonen i djupnelæring. Det at elevane har motivasjon og trua på egne evne («få til») er ei viktig føresetnad for, og ein del av djupnelæringa.
- Informantane sine utsegn om at dei undersøkte koden og matematikken når dei til slutt fekk rett «output», er klare teikn på at elevane ynskjer å finne ut kva strategi eller framgangsmåte dei skulle ha brukt. Det at dei seier at dei vil hugsa det i ettertid, er eit teikn på at dugleikskompetansen, med prosedyre og modellar, vert utvikla gjennom programmeringsøktene.
- Elevane sette pris på å arbeide i par fordi dei då både fekk moralsk støtte frå kvarandre i ei ny arbeidsform, men ikkje minst fordi dei kunne utfylle kvarandre fagleg. Vidare meiner dei at det at ein startar med ei ferdig kode i PRIMM- modellen, var viktig for at dei skulle kome i gang med arbeidet, og forstå kva dei skulle gjera. Her spelar bruken av oppgåver bygd opp etter PRIMM- modellen ei positiv rolle, spesielt grunna låg inngangsterskel, eigarskap til oppgåva og støtta gjennom parsamarbeid.

Med bakgrunn i dei fire punkta over, kan ein konkludera med at elevane i denne studien opplever at programmering med undervisningsopplegg etter PRIMM- modellen er eit godt bidrag til djupnelæring i matematikkfaget.

7.2 Avgrensingar

Forskningsprosjektet hadde som mål å finna svar på elevane si eiga oppleving av djupnelæring gjennom programmering i matematikkfaget. Faktisk læringsutbyte vart ikkje målt, og var ikkje ein del av forskningsprosjektet. Ei heller handlar forskinga om mine (forskaren sine) observasjonar i klasserommet under programmeringsøktene. Der observasjonar er brukt i forskinga, er det på ein slik måte at observasjonar har bidrege til emne teke opp i fokusgruppeintervjuet, der eg søkte forklaringar på det observerte.

Eg har ikkje gått i djupna i det reint faglege når det gjeld programmering, og heller ikkje i detalj på korleis elevane meistra bestemte oppgåver. Dette fordi oppgåvene og forskingsprosjektet hadde ei meir heilskapleg tilnærming på korleis programmering generelt kan opplevast som eit verktøy i matematikken, med tanke på algoritmisk tenking og djupnelæring.

7.3 Nytteverdi av forskingsprosjektet

Dette forskingsprosjektet har gjeve meg positive indikasjonar på elevane sine opplevingar og erfaringar av djupnelæring etter programmering med PRIMM- modellen i matematikkfaget. Som realfagslærer med lite kjennskap til programmering før Fagfornyninga, har prosjektet gjeve meg ein god start inn i programmeringsverda, der eg har hatt stor nytte av å bruka PRIMM- modellen som mal for undervisningsplanlegging. Dette har vore nyttig for meg som lærar, og då er det motiverande og oppløftande at elevane ikkje berre set pris på denne typen oppgåver, men òg føler at det bidreg til læring. Funna i forkinga mi gjev positive indikasjonar på at elevane opplever at dei lærer, og dei seier tydeleg at PRIMM- modellen fungerer som gode «støttejul» med låg inngangsterskel, oppleving av eigarskap til oppgåva og gode erfaringar frå samarbeidslæring. Eg håpar og trur at dette òg kan hjelpa andre matematikklærarar og skuleklassar som slit med å koma i gang med programmering.

Forskinga mi er gjort av ein lærar med minimal erfaring frå programmering og i ein klasse med manglande forkunnskapar frå programmering. Dette er sjølv sagt med på å farga nytteverdien både for lærar og elevar. Resultata av forskingsprosjektet må sjåast i lys av at prosjektet er gjennomført blant elevar med særst lite erfaring med tekstprogrammering.

7.4 Veggen vidare

Eg har freista å få fram elevstemma, og elevane sine opplevingar av djupnelæring ved programmering med PRIMM- modellen. Eg er glad og stolt av dette valet, for eg har som grunnfilosofi at viss elevane skal vera motiverte til å lære, må dei kjenne meistring og føle nytteverdi av det dei gjer. Gjennom arbeidet med denne masteroppgåva, har eg lært mykje av elevane om kva dei meiner «gjev mening» utover det å bli ferdig med oppgåva dei må gjera. I forkinga mi svarar dei at fyrst skjønna dei ikkje vitsen, men så fekk dei det til og då skjønna

dei kvifor dette var lurt å kunne. Eg håpar andre vil ta opp tråden, slik at me får fleire elevstemmer om kva som fungerer og kva som er med på å gje elevane lærelyst på ein slik måte at det å jobba med programmering kan bidra til å auka kompetansen i faga. Som fleire har vore inne på før meg (Psychiaris & Kallia, 2017; Stenseth et al., 2019; Dolonen et al., 2019; Flø, 2021) : det viktigaste er kanskje ikkje at elevane arbeidar med programmering, men korleis undervisningsopplegga vert lagt opp for å sikra den faglege læringa. Elevane frå forskingsprosjektet mitt ser ut til å vera samde om at PRIMM- modellen kan vera ein god metode for undervisningsplanlegging. Både fordi han gjer at dei kjem fort i gang og fordi dei har lyst til å halde fram fordi dei meiner programmeringsoppgåvene er «gøy» når dei «får dei til». Dette igjen er eit positivt teikn med tanke på djupnelæring.

Forskinga mi stør opp om tidlegare forskning når det gjeld programmering som bruyggjar til djupnelæring. Dolonen et al. (2019) si litteraturgjennomgang konkluderte med at PRIMM- modellen kan skapa djup og varig forståing i andre fag. Dette kjem kanskje ikkje så tydeleg fram i forkinga mi, men dersom prosjektet hadde vore større og vart over lengre tid, skal ein ikkje sjå vekk frå det. Feilretting, som Stenseth et al. (2019) argumenterer for, kjem tydeleg til syne som verdifullt for elevane i forkinga mi. Dette ser me av refleksjonane deira kring korleis det var å byrja med ein ferdig kode som dei skulle justere, og når dei seier at det var motiverande å få feilmelding på koden fordi då kunne dei «finne ut av feilen». Når det gjeld elevane sitt faktiske læringsutbyte, har ikkje dette vore ein del av forkinga mi, men ut frå informantane sine svar, kan det sjå ut til at deira opplevingar av læring fell saman med elevane frå Ueda (2021) si forking, der konklusjonen var at elevane ikkje såg den store samanhengen med matematikkfaget, men at dei meinte det meir kreativt å programmere enn å jobba med «vanleg matematikk», og at dei òg sette pris på den fortløpande responsen dei fekk frå programmet.

Fagfornynging LK20 med nye forventingar til både djupnelæring og programmering i matematikkfaget har no vore gjeldande i 2 skuleår. Vidare forskning må ta omsyn til kor lenge læreplanen har vore verksam og med det kor etablert og medvite arbeidet med både djupnelæring og programmering er.

8. KJELDER

- Agarwal, P. (2019). Retrieval Practice & Blooms Taxonomy: Do Students Need Fact Knowledge Before Higher Order Learning? *Journal of Educational Psychology*, ss. 189-209. doi:<https://psycnet.apa.org/doi/10.1037/edu0000282>
- Befring, E. (2016). *Forskningsmetoder i utdanningsvitenskap*. Cappelen Damm AS.
- Dolonen, J., Kluge, A. L., & Mørch, A. (2019, November 8). *Litteraturgjennomgang av programmering i skolen*. Hentet fra Vitenarkiv: <https://www.duo.uio.no/handle/10852/76290>
- Ericson, B. J., Margulieux, L. E., & Rick, J. (2017, November 1). Solving Parson problems versus fixing and writing code. ss. 20-29. doi:<https://doi.org/10.1145/3141880.3141895>
- Everett, E., & Furseth, I. (2012). *Masteroppgaven. Hvordan begynne og hvordan fullføre?* Oslo: Universitetsforlaget.
- Flø, E. E. (2021). Programmering i LK20. *Tangenten- tidsskrift for matematikkundervisning* 32(1), ss. 3-9.
- Fullan, M. Q. (2018). *Dybdelæring*. Cappelen Damm Akademisk.
- Gamlem, S., & Rogne, W. (2019). *Dybdelæring i skolen (2.utgave)*. Pedlex.
- Gilje, Ø., & Bolstad, B. (2019, September 5). *Utdanningsforskning*. Hentet fra Utdanningsforskning.no: <https://utdanningsforskning.no/artikler/2019/dybdelaring-i-rufsete-farvann/>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational researcher* Vol. 42(1), ss. 38-43. doi:<https://doi-org.galanga.hvl.no/10.3102%2F0013189X12463051>
- Halkier, B. (2010). *Fokusgrupper*. Oslo: Gyldendal Akademisk.
- Harms, K., Chen, J., & Kelleher, C. L. (2016, August 25). Distractors in parson problems decrease learning efficiency for young novice programmers. ss. 241-250. doi:<https://doi.org/10.1145/2960310.2960314>
- Kluge, A. (2021). *Læring med digital teknologi*. Oslo: Cappelen Damm.

- Kunnskapsdepartementet. (2017, September 1). *Overordnet del –verdier og prinsipper for grunnopplæringen*. Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. Hentet fra <https://www.regjeringen.no/no/dokumenter/verdier-og-prinsipper-for-grunnopplaringen/id2570003/>
- Kunnskapsdepartementet. (2019, November 15). *Læreplan i matematikk 1.-10. trinn (MAT01-05)*. Fastsatt som forskrift. Læreplanverket for Kunnskapsløftet 2020. Hentet fra <https://data.udir.no/kl06/v201906/laereplaner-lk20/MAT01-05.pdf?lang=nob>
- Kvale, S., & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg., Vol. 3). Oslo: Gyldendal akademisk.
- Minsky, M. (1988). *Paperts prinsipp*. Hentet fra papert.org: <http://papert.org/articles/PapertsPrinciple.html>
- NESH (2021, Desember 16). *Samfunnsvitenskap og humaniora*. Hentet fra Retningslinjer: <https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora/>
- NOU 2014: 7. (2014). *Elevenes læring i fremtidens skole- Et kunnskapsgrunnlag*. Hentet fra <https://www.regjeringen.no/contentassets/e22a715fa374474581a8c58288edc161/nou/pdfs/nou201420140007000dddpdfs.pdf>
- NOU 2015: 8. (2015). *Fremtidens skole: Fornyelse av fag og kompetanser*. Hentet fra <https://www.regjeringen.no/no/dokumenter/nou-2015-8/id2417001/?ch=1>
- Papert, S. (1990). *A Critique of Technocentrism*. Hentet fra <http://www.papert.org/articles/ACritiqueofTechnocentrism.html>
- Postholm, M. (2010). *Kvalitativ metode- en innføring med fokus på fenomenologi, etnografi og kasesstudier. 2.utgave*. Oslo: Universitetsforlaget.
- Postholm, M. B., & Jacobsen, D. I. (2019). *Læreren med forskerblick*. Oslo: Cappelen Damm.
- Primmportal. (2022). *Primmportal*. Hentet fra [Primmportal.com](https://primmportal.com/): <https://primmportal.com/>
- Psycharis, S., & Kallia, M. (2017, Juli 27). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), ss. 583-602. doi:<https://doi.org/10.1007/s11251-017-9421-5>

- Puchta, C., & Potter, J. (2005). *Focus Group Practice*. London: Sage.
- Sanford, J., & Naidu, J. (2017, Mars 31). Mathematical modeling and computational thinking. *Contemporary Issues in Education Research*, ss. 159-168.
doi:<https://doi.org/10.19030/cier.v10i2.9925>
- Sanne, A. e. (2016). *Teknologi og programmering for alle*. Hentet fra <https://www.udir.no/globalassets/filer/tall-og-forskning/forskningsrapporter/teknologi-og-programmering-for-alle.pdf>
- Sentance, S. (2017). <https://blogs.kcl.ac.uk/cser/>. Hentet fra Informatikkutdanning @ King's: <https://blogs.kcl.ac.uk/cser/2017/09/01/primm-a-structured-approach-to-teaching-programming/>
- Sentance, S., Waite, J., & Kallia, M. (2019, April 26). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer science education* 29(2), ss. 136-176.
doi:<https://doi.org/10.1080/08993408.2019.1608781>
- Sevik, K. (2016, November 1). *Programmering i skolen*. Hentet April 10, 2021 fra https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Skjelde, T. (2017, August 14). *Utdanningsnytt*. Hentet fra Utdanningsnytt.no: <https://www.utdanningsnytt.no/fagartikkel-pedagogikk/ja-takk-begge-deler-bade-overflatelaering-og-dybdelaering/172112>
- Statped. (2021, 03 01). *Statped.no*. Hentet fra Læringsressurser: <https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/hva-er-programmering/>
- Stenseth, B., Kaufmann, O., & Forsström, S. (2019). Programmering og matematikk. *Tangenten- tidsskrift for matematikkundervisning* 30(2), ss. 7-12.
- Strandberg, L. (2008). *Vygotsky i praksis. Blant pugghester og fuskelapper*. Gyldendal Akademisk.
- Säljö, R. (2008). *Læring i praksis- et sosiokulturelt perspektiv*. Stockholm: Bokförlaget Prisma.

- Ueda, M. (2021). *Programmering i matematik ur elevenas perspektiv: En fallstudie i en niondeklass*. Hentet fra <https://www.diva-portal.org/smash/get/diva2:1535173/FULLTEXT02.pdf>
- UiO. (2021). *Institutt for pedagogikk*. Hentet fra Programmering og Skaperglede i skolen: <https://www.uv.uio.no/iped/forskning/prosjekter/morch-prokraft/>
- Utdanningsdirektoratet. (2019, Mars 27). *Den algoritmiske tenkeren*. Hentet fra kvalitet og kompetanse: <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algorithmisk-tenkning/>
- Utdanningsdirektoratet. (2019, Mars 13). *Dybdeløring*. Hentet fra Læring og trivsel: <https://www.udir.no/laring-og-trivsel/dybdelaring/>
- Utdanningsdirektoratet. (2020). *Fagspesifikk støtte*. Hentet fra Læring og trivsel: <https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/>
- Utdanningsdirektoratet. (2020). *Læreplanverket*. Hente fra: <https://www.udir.no/laring-og-trivsel/lareplanverket/>
- Utdanningsdirektoratet. (2020). *Støttmateriell til overordnet del: dybdeløring*. Hentet fra Udir.no: <https://www.udir.no/laring-og-trivsel/lareplanverket/stottmateriell-til-overordnet-del/film-dybdelaring/>
- Utdanningsdirektoratet. (2021, Juni 24). *Hvorfor nye læreplaner*. Hentet fra Læring og trivsel : <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvorfor-nye-lareplaner/>
- Voll, L. O. (2019, Mai 29). *Realfagsløyper*. Hentet fra Dybdeløring: https://realfagsloyper.no/sites/default/files/2020-03/artikkel%20dybdel%c3%a6ring_0.pdf
- Wing, J. (2006, Mars 1). Computational thinking. *Communications of the AMC Vol.49(3)*, ss. 33-35.

Vedlegg 1 Informasjonsskriv

Vil du delta i forskingsprosjektet?

«Styrkar programmering djupnelæring i matematikkfaget?»

Dette er ein førespurnad til deg om å delta i eit forskingsprosjekt der føremålet er å finna ut korleis elevane sjølv opplever at arbeid med programmering i matematikkfaget kan bidra til læring. Prosjektet er del av student Renate Møgster Klepsvik sitt masterarbeid, som skal gjennomførast skuleåret 2021/2022 og ferdigstillast 01.06.2022. I dette skrivet får du informasjon om måla for prosjektet og kva deltaking vil bety for deg.

Formål

Målet med denne studien er å læra meir om korleis elevane sjølv opplever undervisningsøkter med programmering i matematikk, og om elevane opplever læring i programmering, matematikk eller andre emne gjennom desse øktene. Opplysningane vil bli brukt som grunnlag for den vidare studien. All deltaking er konfidensiell. I sjølve avhandlinga vil all data verta anonymisert slik at svara dine ikkje skal kunne rettast tilbake til deg på nokon måte

Kven er ansvarleg for forskingsprosjektet?

Institutt for pedagogikk, religion og samfunnsfag ved Høgskulen på Vestlandet er ansvarleg for prosjektet.

Kvifor får du spørsmål om å delta?

Kontaktlæraren i klassen din har samtykka til at forskingsprosjektet kan gjennomførast i klassen din. Det er likevel du sjølv som avgjer om du vil delta, og eg må ha samtykke frå kvar og ein som ynskjer å delta i dette forskingsprosjektet.

Kva betyr det for deg å delta?

Prosjektet vil vare i 4-6 skuletimar over to veker, der du og dei andre deltakarane i prosjektet skal jobbe i par med ulike matematikkoppgåver i programmeringsspråket Python. Etter at alle øktene er gjennomført, vil du få eit digitalt spørjeskjema der du skal svare på ulike påstandar og dela dine erfaringar og di oppleving av korleis det var å jobba med programmering og matematikk i desse undervisningsøktene. Både undervisningsøktene og spørjeskjema vil verta gjennomført i skuletida. I tillegg kan det vera aktuelt å delta i fokusgruppeintervju kort tid etter at undervisningsøktene er gjennomført og spørjeskjema er analysert. Dersom du ynskjer å delta i slike intervju, kryssar du av for det i samtykkeerklæringa.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Viss du vel å delta, kan du kortid som helst trekke samtykket tilbake utan å gje noko grunn. Alle personopplysingar vil då verta sletta. Det vil ikkje ha negative konsekvensar for deg viss du ikkje vil delta, eller seinare vel å trekkje deg.

Personvern – korleis vi oppbevarer og bruker opplysingar

Vi vil berre bruke opplysingane om deg til dei formåla vi har fortalt om i dette skrivet. Vi handsamar opplysingane konfidensielt og i samsvar med personvernregelverket.

Du vil som informant ikkje kunne gjenkjennast i avhandlinga då det vil verta brukt fiktive namn. Berre alder og kjønn vil verta oppgitt.

Kva skjer med opplysingane dine etter at prosjektet er avslutta?

Opplysingane vert anonymisert etter at prosjektet er avslutta og oppgåva er godkjent. Etter planen er dette 30.06.2022. Data vil verta lagra ut desember 2022, for eventuell vidare forskning.

Dine rettar

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

- Innsyn i kva personopplysingar som er registrert om deg, og å få utlevert ein kopi av opplysingane.
- Å få retta personopplysingar om deg.
- Å få sletta personopplysingar om deg, og
- Å sende klage til Datatilsynet om handsaminga av dine personopplysingar.

Kva gjev oss rett til å handtera personopplysningar om deg?

Vi handsamar opplysingar om deg basert på ditt og føresette sitt samtykke.

På oppdrag frå Høgskulen på Vestlandet har NSD – Norsk senter for forskingsdata AS vurdert at handsaming av personopplysningar i dette prosjektet er i samsvar med personvernregelverket.

Kvar kan eg finna ut meir?

Viss du har spørsmål til studien, eller ynskjer å nytta deg av rettane dine, ta kontakt med:

- Høgskulen på Vestlandet ved
 - student Renate Møgster Klepsvik, e-post: renate.klepsvik@aukevoll.kommune.no og
 - rettleiar Anders Grov Nilsen, e-post: anders.nilsen@hvl.no

Viss du har spørsmål knytt til NSD si vurdering av prosjektet, kan du ta kontakt med:

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

Med venleg helsing

Anders Grov Nilsen
(Rettleiar)

Renate Møgster Klepsvik
(Student)

Vedlegg 2 Samtykkeerklæring

Deltakar sitt namn: _____

Eg har motteke og forstått informasjon om prosjektet «*Styrkar programmering djupnelæring i matematikkfaget?*», og har fått høve til å stilla spørsmål. Eg samtykker til:

- å delta i undervisingsøker med programmering
- å delta i /svare på spørjeskjema i etterkant av undervisingsøktene med programmering
- å delta i fokusgruppeintervju i tillegg til spørjeskjema
 - At det vert teke lydopptak av fokusgruppeintervju

Eg samtykker til at mine opplysingar vert handsama fram til prosjektet er avslutta 30.06.2021

(Signert av prosjektdeltakar (elev) og føresette)

Vedlegg 3 Spørjeskjema distribuert i Survey- xact

Påstandar med likert- verdiar: usamd- litt usamd- korkje eller- litt samd- samd

Generelle spørsmål:

1. Eg klarte meg bra med oppgåvene
2. Det hjalp å vera to som samarbeidde då me jobba med oppgåvene
3. Det var lett å forstå dei ulike kommandoane
4. Å byrja med ein ferdig kode, gjorde det lettare å forstå
5. Ein må kunna ein del matematikk på førehand for å klara å programmera

F1: På kva måte ser elevane samanhangar innafor matematikktemaet og til andre fagområde etter å ha arbeidd med programmering?

1. Eg lærte matematikk medan eg programmerte
2. Eg brukte det eg kan i matematikk til å løyse utfordringane i programmering
3. Eg fekk djupare matematisk forståing gjennom å jobba med programmering

F2: Kva tenkjer elevane om å kunna ta i bruk det dei har lært i programmering når dei møter heilt nye utfordringar?

1. Det eg lærte under programmeringsøktene, kan eg òg nytta i andre samanhengar
2. Eg er motivert til å jobba meir med programmering i matematikk
3. Eg er nyfiken (nysgjerrig) på korleis eg kan bruka programmering i andre fag
4. Programmering er eigentleg ein måte å tenke på som kan brukast i nesten alle samanhengar

Heilt til slutt:

Takk for di deltaking!

Vedlegg 4 Undervisningsopplegg 1

```

1 import turtle
2 tommy = turtle.Turtle()
3 tommy.shape("bird")
4
5 tommy.forward(100)
6 tommy.left(90)
7 tommy.forward(100)
8 tommy.left(90)
9 tommy.forward(100)
10 tommy.left(90)
11 tommy.forward(100)
12 tommy.left(135)
13 tommy.forward(141.42)
    
```

Trinn i PRIMM	Elevoppgåve	Elevnotatar! Diskusjon i par:
Predict	Kva trur de vil skje når de køyrer denne koden? Ta linje for linje for dykk, og fortel kva som skjer!	
Run	Køyr koden og observer kva som faktisk skjer!	
Investigate	Sjå nøye på koden. Er det noko som kunne vore koda annleis eller enklare?	
Modify	Gjer naudsynte endringar i koden slik at kvadratet får to diagonalar	
Make	Programmer eit nytt kvadrat med halvparten så stort areal, med diagonal som går nøyaktig frå hjørne til hjørne. (Bruk rekning til å finne nøyaktige mål!)	

Vedlegg 5 Undervisningsopplegg 2

```

trinket Run ? Modules
main.py
1 import turtle
2 tommy = turtle.Turtle()
3 tommy.shape("bird")
4 tommy.penup()
5 tommy.goto(-100, 0)
6 tommy.pendown()
7
8 tommy.forward(200)
9 tommy.left(120)
10 tommy.forward(200)
11 tommy.left(120)
12 tommy.forward(200)
13 tommy.penup()
14
15 tommy.goto(-50, -100)
16 tommy.pendown()
17 tommy.forward(100)
18 tommy.left(120)
19 tommy.forward(100)
20 tommy.left(120)
21 tommy.forward(100)
22 tommy.penup()
23
24 tommy.penup()
25 tommy.goto(0, -75)
26 tommy.color("blue")
27 tommy.write("Formlike trekantar", align="center", font=(None, 16, "bold"))
28 tommy.goto(50, -30)

```

Trinn i PRIMM	Elevoppgåve	Elevnotatar! Diskusjon i par:
Predict	Kva trur de vil skje når de køyrer denne koden? Ta linje for linje for dykk, og fortel kva som skjer!	
Run	Køyr koden og observer kva som faktisk skjer!	
Investigate	Sjå nøye på koden. Er det noko som kunne vore koda annleis eller enklare?	
Modify	Gjer naudsynte endringar i koden og lag midtnormal i kvar av figurane.	
Make	Programmer to formlike figurar der arealet av den vetle er nøyaktig halvparten av arealet av den største.	

Vedlegg 6 Undervisningsopplegg 3

```

1 import turtle
2 tommy = turtle.Turtle()
3 tommy.shape("bird")
4
5 def draw_circle(turtle, color, size, x, y):
6     turtle.penup()
7     turtle.color(color)
8     turtle.fillcolor(color)
9     turtle.goto(x,y)
10    turtle.begin_fill()
11    turtle.pendown()
12    turtle.circle(size)
13    turtle.penup()
14    turtle.end_fill()
15    turtle.pendown()
16
17 draw_circle(tommy, "red", 58, 0, 0)
18 tommy.forward(100)
19 tommy.left(120)
20 tommy.forward(200)
21 tommy.left(120)
22 tommy.forward(200)
23 tommy.left(120)
24 tommy.forward(100)
25 tommy.penup()
26 tommy.goto(-50, -40)

```

Trinn i PRIMM	Elevoppgåve	Elevnotatar! Diskusjon i par:
Predict	Kva trur de vil skje når de køyrer denne koden? Ta linje for linje for dykk, og fortel kva som skjer!	
Run	Køyr koden og observer kva som faktisk skjer!	
Investigate	Sjå nøye på koden. Kan du forklare linje 17?	
Modify	Gjer naudsynte endringar i koden slik at sirkelen vert svart.	
Make 	Programmer ein innskreven sirkel i eit kvadrat!	

Vedlegg 7 Undervisningsopplegg 4

```

1 from statistics import mean, mode, median
2 # definerer eit datasett
3 datasett = [1, 2, 3, 3, 4, 4, 4, 5, 5, 6]
4 typetal = mode(datasett)
5 median = median(datasett)
6 snitt = mean(datasett)
7
8 print("typetalet er:", typetal)
9 print("Medianen er:", median)
10 print("Gjennomsnittet er:", snitt)
11 print("Variasjonsbreidda er:", max(datasett)-min(datasett))
12

```

Trinn i PRIMM	Elevoppgåve	Elevnotatar! Diskusjon i par:												
Predict	Kva trur de vil skje når de køyrer denne koden? Ta linje for linje for dykk, og fortel kva som skjer!													
Run	Køyr koden og observer kva som faktisk skjer!													
Investigate	Sjå nøye på koden. Er det noko som kunne vore koda annleis eller enklare?													
Modify	Du oppdagar ein feil med datasettet! 2 skal ikkje vera med, og det manglar ein 6. Korriger datasettet og køyr berekningane på ny.													
Make	Tala viser tal på mål i nokre fotballkampar. Bruk Python3 og finn gjennomsnitt, median, typetal og variasjonsbreidd for talet på mål! <div style="border: 1px solid black; padding: 10px; margin: 10px 0; text-align: center;"> <table border="0"> <tr> <td>1</td><td>6</td><td>3</td><td>8</td></tr> <tr> <td>7</td><td>2</td><td>2</td><td>4</td></tr> <tr> <td>8</td><td>1</td><td>7</td><td>5</td></tr> </table> </div>	1	6	3	8	7	2	2	4	8	1	7	5	
1	6	3	8											
7	2	2	4											
8	1	7	5											

Vedlegg 8 Undervisningsopplegg 5 (brukt i fokusgruppeintervju)

```

1 from turtle import*
2 speed(10)
3 for side in range(4):
4     forward(100)
5     left(90)
6 side = 100
7 omkrins = side * 4
8 areal = side * side
9 print("Omkrinsen er", omkrins)
10 print("Arealet er", areal)
11

```

Trinn i PRIMM	Elevoppgåve	Elevnotatar! Diskusjon i par:
Predict	Kva trur de vil skje når de køyrer denne koden? Ta linje for linje for dykk, og fortel kva som skjer!	
Run	Køyr koden og observer kva som faktisk skjer!	
Investigate	Sjå nøye på koden. Er det noko som kunne vore koda annleis?	
Modify	Gjer naudsynte endringar i koden slik at kvadratet vert ein regulær femkant	
Make	Programmer eit rektangel med dobbelt så stort areal som det opphavslege kvadratet, med like sidekantar som kvadratet. Berekn omkrins og areal av rektangelet	

Vedlegg 9 Vurdering frå NSD

1. Meldeskjema
2. Masteroppgåve - P: Korleis opplever elevane at programmering i matematikk etter PRIMM- modellen bidreg til djupnelæring?
3. Vurdering

Vurdering

Skriv ut

Referansenummer

208777

Prosjekttittel

Masteroppgåve - P: Korleis opplever elevane at programmering i matematikk etter PRIMM- modellen bidreg til djupnelæring?

Behandlingsansvarlig institusjon

Høgskulen på Vestlandet / Fakultet for lærerutdanning, kultur og idrett / Institutt for pedagogikk, religion og samfunnsfag

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Anders Grov Nilsen, anders.nilsen@hvl.no, tlf: 53491511

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Renate Møgster Klepsvik, reate.klepsvik@austevoll.kommune.no, tlf: 91524239

Prosjektperiode

01.09.2021 - 01.06.2022

Vurdering (1)

06.01.2022 - Vurdert

Det er vår vurdering at behandlingen vil være i samsvar med personvernlovgivningen, så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet den 06.01.2022 med vedlegg og meldingsdialog. Behandlingen kan starte.

TYPE OPPLYSNINGER OG VARIGHET Prosjektet vil behandle alminnelige personopplysninger frem til 01.06.2022

LOVLIG GRUNNLAG Forskningsdeltagerne i prosjektet er 15år. Prosjektet vil innhente samtykke fra de registrerte og fra deres foresatte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 nr. 11 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse, som kan dokumenteres, og som den registrerte og de foresatte kan trekke tilbake. For alminnelige personopplysninger vil lovlig grunnlag for behandlingen være samtykke, jf. personvernforordningen art. 6 nr. 1 a.

PERSONVERNPRINSIPPER Personverntjenester vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen: - om lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte/foresatte får tilfredsstillende informasjon om og samtykker til behandlingen - formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål - dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet - lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet.

DE REGISTRERTES RETTIGHETER Vi vurderer at informasjonen om behandlingen som de registrerte/ de foresatte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13. Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18) og dataportabilitet (art. 20). Vi minner om at hvis en registrert /foresatt tar kontakt om rettighetene, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER Vi legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32). For å forsikre dere om at kravene oppfylles,

må prosjektansvarlig følge interne retningslinjer/rådføre dere med behandlingsansvarlig institusjon.

MELD VESENTLIGE ENDRINGER Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilken type endringer det er nødvendig å melde: <https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema> Du må vente på svar før endringen gjennomføres.

OPPFØLGING AV PROSJEKTET Vi vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet. Kontaktperson hos personverntjenester: Kaja Amundsen Lykke til med prosjektet!