

Python som medierende artefakt i matematikkundervisning

En kasusstudie av programmering og algoritmisk tenking som
verktøy i undervisning av stokastiske hendelser i en S1-klasse

MARKUS TVEIT

Veileder

Said Hadjerrouit

Universitetet i Agder, [2022]

Fakultet for teknologi og realfag

Institutt for matematiske fag

Forord

Det at jeg nå leverer denne masteroppgaven betyr at jeg er ferdig med fem år på lektorstudiet. Studiet har i likhet med arbeidet med denne oppgaven vært både lærerikt og krevende, og har rustet meg til å bli en best mulig lærer for mine fremtidige elever. Jeg har ved flere anledninger underveis klødd meg litt i hodet og lurt på hvordan en matematiker skal kunne skrive en så stor oppgave som dette, og hadde det ikke vært for all den gode hjelpen jeg har fått er det ikke sikkert det hadde gått. Derfor er det på sin plass å rette takk til flere personer her.

Først og fremst rettes en stor takk til min veileder Said Hadjerrouit som har gitt meg mange gode råd og innspill gjennom hele arbeidet med oppgaven. Det å få innspill av en person med god erfaring innen forskning og akademisk skriving har vært gull verdt for en student som er mest komfortabel med å arbeide med tall.

Videre rettes en takk til faglærer og elever i klassen jeg gjennomførte studien i samarbeid med. Takk for positiviteten ovenfor prosjektet og takk til hver enkelt elev for viljen til å delta. Uten deres samarbeid hadde ikke oppgaven vært stort mer en samling teori og en hypotetisk metodedel.

Takk til alle som har vært villige til å korrekturlese oppgaven og luke ut feil og tungvinne formuleringen. Oppgaven hadde ikke hatt den samme kvaliteten uten dere.

Til slutt rettes også en takk til mine medstudenter som har gjort disse årene til noe mer enn bare en utdanning. Det å ha slike støttespillere har vært utrolig viktig i tunge eksamensperioder og ikke minst denne masterperioden. Studiet hadde ikke vært det samme uten dere.

Kristiansand, mai 2022

Markus Tveit

Sammendrag

Et av de nye ordene som nevnes i læreplanene i matematikk i fagfornyelsen som innføres gradvis etter 2020 er programmering. Dette er et nytt område for mange lærere og elever, noe som gjør at innføringen kan by både på store muligheter, men også problemer. For å bli klar over noen av fordelene valgte jeg i denne studien å undersøke følgende forskningsspørsmål:

- 1. Hvilke av Niss matematiske kompetanser arbeider elevene med når de tar i bruk programmering og algoritmisk tenkning som medierende artefakter i matematiske oppgaver knyttet til sannsynlighetsregning?**
- 2. I hvor stor grad gjenspeiles stegene i algoritmisk-tenkningsprosess i elevenes arbeid med oppgaver som tar i bruk programmering i Python?**

For å foreta denne studien har jeg som teoretisk rammeverk basert meg på et sosiokulturelt syn på læring. Jeg har også basert meg på teorier knyttet til algoritmisk tenkning, og tatt i bruk Niss åtte matematiske kompetanser for å undersøke forskningsspørsmålene.

Studien er gjennomført som en kvalitativ kasusstudie hvor jeg har samarbeidet med en S1-klasse. Klassen ble delt inn i fire grupper med to elever i hver gruppe, hvor jeg sammen med hver gruppe gjennomførte et oppgavebasert intervju. Her arbeidet elevene med et oppgavesett som tok i bruk programmering for å løse oppgaver knyttet til stokastiske hendelser, hvor de også ble intervjuet underveis i arbeidet.

Resultatene fra gjennomføringen av studien peker mot at det å ta i bruk simulering i Python som medierende artefakt gir rom for at elevene arbeider med flere av Niss kompetanser. Blant de mest tydelig er arbeidet med tankegangskompetanse, kommunikasjonskompetanse, resonnementskompetanse, representasjonskompetanse og modelleringskompetanse. Elevenes arbeid viser tegn til at det å ta i bruk alle stegene knyttet til algoritmisk tenkning gir dem rom til å bygge en mer helhetlig modelleringskompetanse, samt å gi dem gode strategier som er med på å bygge deres problemløsningskompetanse.

Elevenes arbeid bærer preg av at stegene i algoritmisk tenkning ikke er en del av deres intuitive prosess i arbeidet mot en ferdig kode, men at de innehar nok kompetanse til å gjennomføre stegene når de først blir orientert i den retningen av lærer eller oppgavedesignet.

Implikasjonene dette kan ha for oss som lærere er at innføringen av programmering i læreplanen og bruken av programmering, og i den sammenheng også algoritmisk tenkning, som medierende artefakt byr på gode muligheter til å hjelpe elevene våre med å bygge flere sentrale matematiske kompetanser. Det er derimot også tydelig at om vi skal gi dem en mest mulig helhetlig kompetanse må vi gi dem gode strategier når det gjelder konstruksjon av koder og gi dem god innsikt i stegene i algoritmisk tenkning og hvordan dette knyttes til programmeringssituasjoner.

Summary

One of the new words mentioned in the new Norwegian curriculum in mathematics which is gradually implemented after 2020 is the word programming. This is a new area for a lot of teachers and students. This means that the implementation of programming can lead to both big learning opportunities, but also some problems. To bring awareness to some of the positive opportunities I have chosen to investigate the following research questions in this study:

- 1. Which of Niss' mathematical competencies do students work with when they use programming and computational thinking as mediating artefacts in mathematical tasks about probability?**
- 2. To what degree the steps of computational thinking be seen in the students' work in programming in Python?**

To conduct this study, I have chosen to base my theoretical framework on a sociocultural view of learning. I have also based the study on theory on computational thinking and used Niss' eight mathematical competencies to investigate the research questions.

The study is designed as a qualitative case study where I have worked together with a Norwegian S1 mathematics class. The class was split into four groups with two students in each group. Each of these groups took part in a task-based interview. In this interview the students worked on two tasks that required the use of programming and was based on stochastic variables. The students were also interview while working.

The results of the study show that the use of simulation in Python as a mediating artefact gives the students room to work on multiple of Niss' competencies. Among the most clearly shown was thinking mathematically, communicating about mathematics, reasoning mathematically and representing mathematical entities. The work done by the students also shows that using all the steps in computational thinking allows the students to get a more complete modelling competency and helps them learn new strategies to improve their ability to solve mathematical problems.

The work done by the students shows that the steps in computational thinking is not an intuitive part of their work towards a working code, but all groups showed that they were able to do the steps once they were asked by the interviewer or the tasks.

The implications of this study for teachers is that the implementation of programming into the curriculum and the use of programming, and with it computational thinking, as a meditating artefact makes it possible for students to work with multiple competencies at the same time. It is also clear that in order to allow the students to work with more aspects of the different competencies we have to give them good strategies when it comes to creating codes and give them good insight into the steps of computational thinking and how they are relevant in programming tasks.

Innhold

Forord	iii
Sammendrag	v
Summary	vii
1 Innledning	1
1.1 Bakgrunn for oppgaven	1
1.2 Problemstilling og forskningsspørsmål	1
1.3 Kort oversikt over resten av oppgaven	2
2 Teoretisk rammeverk	3
2.1 Syn på læring	3
2.2 Algoritmisk tenking	4
2.2.1 Abstraksjon	5
2.2.2 Dekomponering	5
2.2.3 Algoritmer	5
2.2.4 Koding	5
2.2.5 Simulering	5
2.2.6 Evaluering	5
2.3 Algoritmisk tenking og programmering i syn av sosiokulturell teori	6
2.4 Matematisk kompetanse	6
2.4.1 Å spørre og svare, i, med, om matematikk	7
2.4.2 Å kunne bruke språk og redskaper i matematikk	7
2.5 Tidligere forskning på programmering og algoritmisk tenking i matematikkundervisning	8
3 Metode	11
3.1 Syn på forskning	11
3.2 Forskningsdesign	11
3.3 Forskningsmetode	12
3.4 Valg av tema for oppgavene	12
3.5 Valg av programmeringsspråk	13
3.6 Oppgavesettet	14
3.6.1 Oppgave 1	14
3.6.2 Oppgave 2	16
3.7 Utvalg og gruppesammensetning	17
3.8 Gjennomføringen av forskningen	18
3.9 Analysemetode	18
3.9.1 Transkribering	18
3.9.2 Analyse av transkribert materiale	19
3.10 Forskningsetikk	20
3.11 Resultatenes reliabilitet og validitet	20
4 Resultater og analyse	23
4.1 Gruppe 1	23
4.1.1 Oppgave 1	23

4.1.2 Oppgave 2	25
4.2 Gruppe 2	26
4.2.1 Oppgave 1	27
4.2.2 Oppgave 2	28
4.3 Gruppe 3	30
4.3.1 Oppgave 1	30
4.3.2 Oppgave 2	31
4.4 Gruppe 4	33
4.4.1 Oppgave 1	33
4.4.2 Oppgave 2	34
4.5 Oppsummering av resultater og analyse.....	36
5 Diskusjon	39
5.1 Simulering med ferdigskrevne programmer som medierende artefakt for å bygge tankegangskompetanse	39
5.2 Algoritmisk tenkning som medierende artefakt for å gi elevene en helhetlig modelleringskompetanse	40
5.3 Abstraksjon og dekomponering som medierende artefakter for å arbeide med matematisk kompetanse ..	41
5.4 Python som medierende artefakt for å bedre matematisk kommunikasjon og resonnering	42
5.5 Programmering som medierende artefakt for å bygge elevers hjelpemiddelkompetanse.....	43
5.6 Lærer og/eller oppgavedesign i rollen som proksimal utviklingszone	44
5.7 Simulering som mulig løsning på problemer med tid.....	44
6 Avslutning	47
6.1 Konklusjon.....	47
6.2 Studiens begrensninger	47
6.3 Implikasjoner og videre forskning	48
6.4 Eget utbytte	49
7 Referanseliste	51
8 Vedlegg.....	55
8.1 Informasjonsskriv/ Samtykkeerklæring.....	55
8.2 Godkjenning fra NSD.....	58
8.3 Oppgavesettet.....	61
8.4 Intervjuguide	63
8.5 Datamateriale gruppe 1	64
8.5.1 Transkribering og analyse av lydopptak.....	64
8.5.2 Kode fra oppgave 2.....	76
8.6 Datamateriale gruppe 2.....	76
8.6.1 Transkribering og analyse av lydopptak.....	76
8.6.2 Kode fra oppgave 2.....	85
8.7 Datamateriale gruppe 3.....	86
8.7.1 Transkribering og analyse av lydopptak.....	86
8.7.2 Kode fra oppgave 2.....	96
8.8 Datamateriale gruppe 4.....	96
8.8.1 Transkribering og analyse av lydopptak.....	96
8.8.2 Kode fra oppgave 2.....	109

1 Innledning

1.1 Bakgrunn for oppgaven

De siste årene har det vokst frem et internasjonalt fokus på å fremme programmering i skolen, med argumenter som at programmering er en nødvendig ferdighet i det moderne samfunnet, fremtidig behov for kompetanse i næringslivet og evnen til å forstå hvordan et stadig mer digitalisert samfunn fungerer (Senter for IKT i utdanningen, 2016, s. 6). Dette er et tydelig signal om at vi som lærere må ha et fokus på å få inn programmering i utdanningen om vi skal kunne si at vi aktivt arbeider for å utdanne elever med kompetanse som kan brukes i hverdagen de møter etter endt skolegang.

Det å ta i bruk programmering i undervisningssituasjoner er ikke en ny tanke. Allerede i 1980 kom Seymour Papert med sin LOGO-arena, hvor barn programmerte en skilpadde som kunne bevege seg. I sin artikkel skriver han at når elever programmerer skilpaddens handlinger og hvordan den skal tenke, lærer de også selv å tenke. Dette setter elevene i en posisjon til å aktivt bygge sine egne intellektuelle strukturer (Papert, 1980, s. 19). I en litteraturstudie 15 år senere poengterer Yelland (1995, s. 866) at det fortsatt er mye debatt om hvilke kognitive gevinster det å bruke LOGO i undervisningen kan ha, og at fremskritt i teknologi potensielt vil gjøre det lettere å studere dette.

I nyere tid har elevene i norsk skole hatt få eller ingen muligheter til å lære seg programmering i grunnskolen, og kan først på videregående selv velge enkelte teknologirettede fag. Dette har ført til at elevene utdannes til å bli konsumenter av teknologi heller enn å kunne bli skapere av teknologi (NOU 2013:2, kap. 6). Dette er et problem som har blitt tatt hensyn til i fagfornyelsen som trer i kraft fra 2020 (innføres gradvis, så noen fagkoder innføres senere), ved at programmering og algoritmisk tenkning nevnes eksplisitt som en av grunnene til at en fornyelse av læreplanen var nødvendig (Utdanningsdirektoratet, 2021a). I matematikkfaget spesifikt forsvares innføringen av programmering og algoritmisk tenkning slik: "I læreplanen er algoritmisk tenkning synliggjort fordi dette er en viktig problemløsningsstrategi. Når elevene bruker programmering til å utforske og løse problemer, kan det være et godt verktøy for å utvikle matematisk forståelse" (Utdanningsdirektoratet, 2020a).

Dermed er det tydelig at det å få en forståelse for hvordan programmering kan implementeres i matematikkundervisningen ikke bare er viktig fordi det er nevnt i læreplan, noe som i og for seg hadde vært nok, men også fordi det er en ferdighet som kreves i flere yrker og områder i et moderne samfunn. Siden jeg selv har lite kjennskap til programmering gjennom eget studieløp virket det å vie masterprosjektet mitt til dette temaet som en god mulighet til å innarbeide gode ferdigheter innen programmering, samtidig som jeg får innsikt i hvordan jeg kan bruke det i undervisningen i mitt fremtidige klasserom.

1.2 Problemstilling og forskningsspørsmål

For å snevre inn omfanget av oppgaven, noe som er nødvendig med tanken på prosjektets tidshorison, har jeg valgt å rette meg inn mot et spesifikt kompetansemål i den nye læreplanen og undersøke om det kan gi rom for bruk av programmering på en måte som kan gi god læring. Kompetansemålet jeg vil ta for meg lyder slik: "*bruke digitale verktøy til å simulere og utforske utfall i stokastiske forsøk, og forstå begrepet stokastiske variabler*" (Utdanningsdirektoratet, 2020b) og er en del av læreplanen i matematikk S1. Dette er et av de kompetansemålene som nevner digitale verktøy og simulering eksplisitt, og som derfor er naturlig å bruke i en oppgave med denne tematikken.

Problemområdet jeg ønsker å undersøke i denne oppgaven er hvor egnet programmering er som medierende artefakt i matematikkundervisningen, hvor et medierende artefakt er et verktøy som står sentralt i sosiokulturell læringsteori og vil bli beskrevet grundigere i kapittel 2. Det jeg her legger i begrepet egnethet er hvilke matematiske kompetanser bruken av det medierende artefaktet lar elevene arbeide med og bygge. Dette er for meg som lærer et veldig viktig felt å danne en mening om, da et godt medierende artefakt kan fremme læring hos elevene, og et dårlig artefakt i verste fall kan føre til redusert læring.

For å undersøke problemområdet ønsker jeg å undersøke og svare på følgende forskningsspørsmål:

- 1. Hvilke av Niss matematiske kompetanser arbeider elevene med når de tar i bruk programmering og algoritmisk tenkning som medierende artefakter i matematiske oppgaver knyttet til sannsynlighetsregning?**
- 2. I hvor stor grad gjenspeiles stegene i algoritmisk-tenkningsprosess i elevenes arbeid med oppgaver som tar i bruk programmering i Python?**

Av disse to kan det første sees på som et hovedspørsmål for å undersøke problemstillingen jeg har valgt å fordype meg i, da det er her kompetansene til elevene vil bli undersøkt. Det andre spørsmålet blir da et nødvendig underspørsmål for å bekrefte at det faktisk er i sammenheng med bruken av algoritmisk tenkning kompetansene kommer frem.

Begreper som er tatt i bruk i forskningsspørsmålene vil bli forklart i detalj i teorikapittelet, og måten spørsmålene blir lest og besvart må derfor forstås i lys av hvordan begrepene blir definert der.

1.3 Kort oversikt over resten av oppgaven

Videre i oppgaven vil jeg i kapittel 2 presentere det teoretiske rammeverk jeg har basert meg på i utformingen av oppgavene elevene skal arbeide med og som jeg vil legge til grunn for analysen av elevenes arbeid. Jeg vil her også gi en forklaring på sentrale begreper jeg tar i bruk i oppgaven og presentere noe av forskningen som tidligere er gjort om programmering i matematikkundervisning.

Etter det vil jeg i kapittel 3 presentere forskningsmetoden jeg har tatt i bruk og forklare hvorfor den var gunstig med hensyn til mine forskningsspørsmål, samt hvilke etiske valg jeg har tatt og hva som styrker og svekker oppgavens reliabilitet og validitet. Jeg vil også her presentere oppgavesettet elevene har arbeidet med, og begrunne hvorfor oppgavene er konstruert som de er.

Videre vil jeg i kapittel 4 presentere resultatene fra gjennomføringen av oppgavesettet og intervjuene, og analysere disse i lys av teorien jeg har basert meg på. I denne delen vil jeg gi en deskriptiv presentasjon av funnene, uten å drøfte for mye rundt dem.

Når analysen er presentert vil jeg i kapittel 5 videre drøfte funnene mine i lys av teorien og den tidligere forskningen som er gjort, og bruke dette til å diskutere forskningsspørsmålene jeg har satt ut for å undersøke.

Til slutt vil jeg i kapittel 6 komme med en konklusjon til forskningsspørsmålene og presentere hvilke implikasjoner jeg mener dette medfører for lærere og innføringen av programmering i matematikkundervisningen. Jeg vil her også presentere eget utbytte av arbeidet med oppgaven og studiens begrensninger.

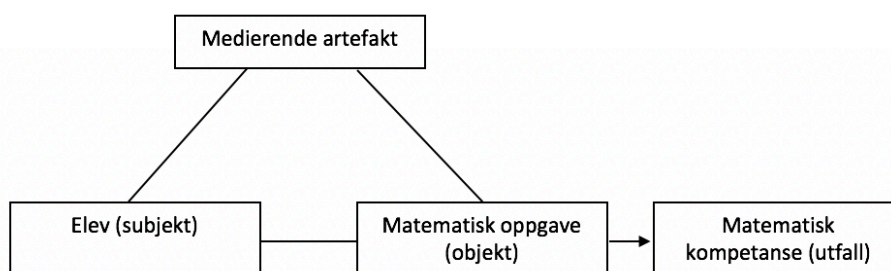
2 Teoretisk rammeverk

I dette kapittelet kommer jeg til å presentere det teoretiske rammeverket jeg baserer meg på i gjennomføringen av denne studien. Det teoretiske rammeverket jeg velger å basere meg på er teorier som jeg ser på som viktige for å undersøke problemstillingen jeg har satt ut for å undersøke, og som vil gi meg et grunnlag til å både gjennomføre og analysere et forskningsopplegg som kan gi svar på mine forskningsspørsmål. Videre i kapittelet vil jeg i tillegg presentere hvilke resultater tidligere forskning innenfor samme område har gitt, noe som gir meg et grunnlag å sammenligne mine funn opp imot.

2.1 Syn på læring

Som lærer er en av mine viktigste oppgaver å stimulere til læring hos elevene i mitt klasserom. Dette betyr at jeg må ha en bevissthet rundt hvordan mennesker lærer og hvordan et klasserom på best mulig måte kan styres for å legge til rette for dette. Hvordan læring foregår har vært mye forsket på og debattert rundt, noe som har ført til mange teorier som har som mål å beskrive læringsprosessen til mennesker. Jeg vil her ikke diskutere de ulike synene, da det hadde vært en for stor oppgave i et teorigapittel i en masteroppgave, men heller presentere synet jeg baserer meg på i denne studien.

En av disse læringsteoriene er den sosiokulturelle læringsteorien som ble introdusert av Lev Vygotsky og som har fått godt fotfeste i vestlig psykologi fra slutten av 1990-tallet (Imsen, 2014, s. 184). I sosiokulturell læringsteori ser man ikke på læring som noe som skjer individuelt hos hver enkeltperson, men som noe som skjer når individet står i samspill med sine omgivelser (Imsen, 2014, s. 183). Skal vi studere læring er det derfor viktig å studere samspillet mellom den som lærer og omgivelsene personen befinner seg i. Den som lærer tar da i bruk omgivelsene i sin egen utvikling (Imsen, 2014, s. 188). Dette medfører at vi som lærere må være fokusert på at omgivelsene elevene befinner seg i faktisk stimulerer til læring. I en undervisningssituasjon sett fra et sosiokulturelt ståsted vil man si at læringen skjer som et samspill mellom tre deler: subjektet (eleven), objektet (aktiviteten som gjennomføres) og som et hjelpemiddel i subjektets arbeid med objektet finner man det medierende artefaktet (et hjelpemiddel/verktøy). Alle disse brikkene kombinert fører til et utfall (læring) hos subjektet (Imsen, 2014, s. 205), som demonstrert i figur 2.1. Denne måten å se elevenes læring på, i samspill med omgivelsene, ser jeg på som et veldig gunstig syn på læring for å undersøke mine forskningsspørsmål, da det gir et fokus på hvordan eleven samhandler med programmeringen for å fremme kompetanse. Denne studien vil derfor forsøke å finne ut hvor egnet programmering og algoritmisk tenkning er som et slikt medierende artefakt for å hjelpe elever med å lære og arbeide med matematiske kompetanser.



Figur 2.1: Sammenhengen mellom subjekt, objekt, utfall og medierende artefakt i sosiokulturell læringsteori

Et annet begrep som knyttes til sosiokulturell læringsteori som er relevant for min undersøkelse er den proksimale utviklingssone. Den proksimale utviklingssone handler om at i sosiale

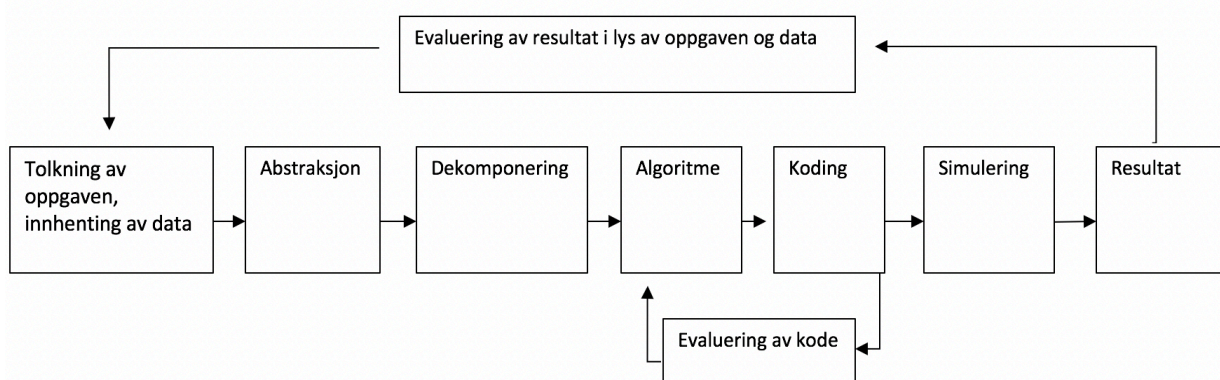
omgivelser vil det være ting eleven kan gjøre i samspill med andre som han enda ikke har lært å gjøre på egenhånd. Når dette er tilfellet vil det eller den som hjelper eleven med å gjøre noe han/hun enda ikke kan klare selv, virke som en medierende hjelper for eleven og skape en proksimal utviklingszone (Imsen, 2014, s. 192). Det vil derfor være viktig for oss som lærere å være bevisste på å skape en så god proksimal utviklingszone for våre elever som mulig, så de kan ha et rom til å utvikle seg og lære nye ting i klasserommet.

2.2 Algoritmisk tenking

Når jeg i denne oppgaven bruker begrepet algoritmisk tenkning er det en oversettelse av det engelske begrepet “computational thinking” som tas i bruk i mye forskning knyttet til programmering i skoleverket. Wing (2008, s. 3717) beskriver i sin artikkel algoritmisk tenkning som en analytisk form for tenkning. Den har til felles med matematisk tenkning måten vi går frem for å løse problemer, og har til felles med ingeniørers tenkemåte måten de designer og evaluerer sammensatte systemer. Når det kommer til hvorfor algoritmisk tenkning er en gunstig tankegang å lære, nevner Wing (2006, s. 33) at denne måten å tenke på kan gi oss mulighet og mot til å angripe oppgaver og designe systemer som er større enn hva noen av oss kunne klart med hodene våre alene.

Barr og Stephenson (2011, s. 50) nevner dekomponering, abstraksjon, algoritmer, automasjon og simulering, samt innhenting, tolkning og representasjon av data som kjerneegenskaper når man skal innføre algoritmisk tenking i K-12 (1-12 trinn i amerikansk skolesystem). Dette har mange fellestrekk med elementene Utdanningsdirektoratet (2021b) trekker fram i sin artikkel om algoritmisk tenking, som også nevner det å se etter mønstre og evaluering av egne algoritmer underveis som viktige prosesser. Det er disse kjerneegenskapene jeg vil velge å fokusere på i mitt videre arbeid. Dette er dels grunnet at jeg selv er enig i at det er de mest elementære delene av algoritmisk tenkning, og dels fordi et prosjekt med det omfanget jeg skal til med ikke vil tillate å gå i detalj på alle aspekter innenfor et så bredt felt som algoritmisk tenkning.

I min forskning vil jeg, basert på artiklene nevnt i dette avsnittet, se på algoritmisk tenkning som prosessen beskrevet i figur 2.2, og vil videre beskrive hvert av stegene mer detaljert.



Figur 2.2: Stegene i en algoritmisk-tenkingsprosess. Inspirert av Hansen og Hadjerrouit (2021, s.253).

Jeg vil her også poengtere at det ut ifra beskrivelsene som har blitt gitt av algoritmisk tenkning er tydelig at det er et tema og en måte å tenke på som ikke bare gjelder programmering. Et eksempel på dette kan være at det å skape og følge oppskrifter på kjøkkenet også regnes som å tenke algoritmisk. Man kan derfor ikke si at algoritmisk tenkning er det samme som programmering, men at det er en sammenheng mellom dem og at det å programmere er et resultat av å kunne tenke algoritmisk (Hansen & Hadjerrouit, 2021, s.252). Dette vil være med på å begrense omfanget av hvor dypt jeg i min oppgave undersøker algoritmisk tenkning hos elevene, da jeg bare ser på det i lys av matematiske programmeringsoppgaver.

2.2.1 Abstraksjon

Abstraksjon handler om å trekke ideer ut fra det som skjer i den fysiske verden, og på den måten generalisere hendelsen (Wing, 2008, s. 3717-3718). Prosessen knyttet til abstraksjon blir derfor å bestemme hvilke detaljer i en situasjon man velger å legge fokus på, og hvilke man velger å se bort i fra. Å ha gode ferdigheter innen abstraksjon blir da å vite hvilke detaljer man kan utelate i en modell, uten at modellen mister vesentlige detaljer eller deler. Her vil det også være viktig å være nøye med å velge en passende representasjon for modellen, da ulike representasjoner gjør det enklere å presentere ulike situasjoner (Csizmadia et al., 2015, s. 7).

2.2.2 Dekomponering

Dekomponering vil si å dele et større problem opp i flere små biter som er enklere å løse hver for seg (Barr & Stephenson, 2011, s. 52). Dette vil være en viktig egenskap å inneha da oppgaver knyttet til programmering fort kan bli veldig store om programmene skal representere en kompleks situasjon. Dette kan føre til at oppgaven kan bli vanskelig å løse om alle sidene av problemet skal angripes samtidig. Å inneha gode dekomponeringsferdigheter er derfor en viktig egenskap å ha om man ønsker å lage store og mer komplekse systemer (Csizmadia et al., 2015, s. 8)

2.2.3 Algoritmer

Misfeldt og Ejsing-Duun (2015, s. 2526) beskriver en algoritme som en systematisk beskrivelse av en problemløsningsstrategi. De trekker frem oppskrifter på mat som et godt eksempel på en algoritme, som beskriver steg for steg hva som skal gjøres for å ende opp med det ønskede resultatet. I deres artikkel som fokuserer på algoritmisk tenkning med programmering som mål skriver de videre at algoritmer da vil handle om evnen til å utvikle, ta i bruk og utnytte maskiner som kan gjennomføre de konstruerte algoritmene, og gi et ønsket svar eller resultat.

2.2.4 Koding

Når man har designet en algoritme for å løse et problem som skal ta i bruk et dataprogram som hjelpemiddel er neste steg å oversette denne algoritmen til en kode i et programmeringsspråk som datamaskinen kan forstå (Csizmadia et al., 2015, s. 9). Wing (2006, s. 35) beskriver denne prosessen som en måte å få datamaskinen til å tenke slik som vi vil, noe som gir oss muligheten til å bygge systemer og løsningsmetoder vi ikke hadde turt å prøve på før datamaskinene ble til, der nå bare fantasien setter grenser. Her vil det kreves av den som løser oppgaven at han/hun innehar nok kunnskap om programmeringsspråket som skal tas i bruk til å oversette algoritmen til en fungerende kode.

2.2.5 Simulering

I og med at kompetansemålet i læreplanen som oppgavene i denne studien vil gå ut ifra eksplisitt nevner at elevene skal ta i bruk simulering for å utforske stokastiske hendelser, er det naturlig at også oppgavene de kommer til å gjennomføre krever at programmet de lager ender opp med å simulere en situasjon. Vansteenkiste (1975, s. 83) beskriver simulering som et godt hjelpemiddel for å forklare, illustrere og demonstrere sammenhenger i matematiske systemer. Simulering kan derfor være en god mulighet for å visualisere matematiske hendelser og sammenhenger for elevene.

2.2.6 Evaluering

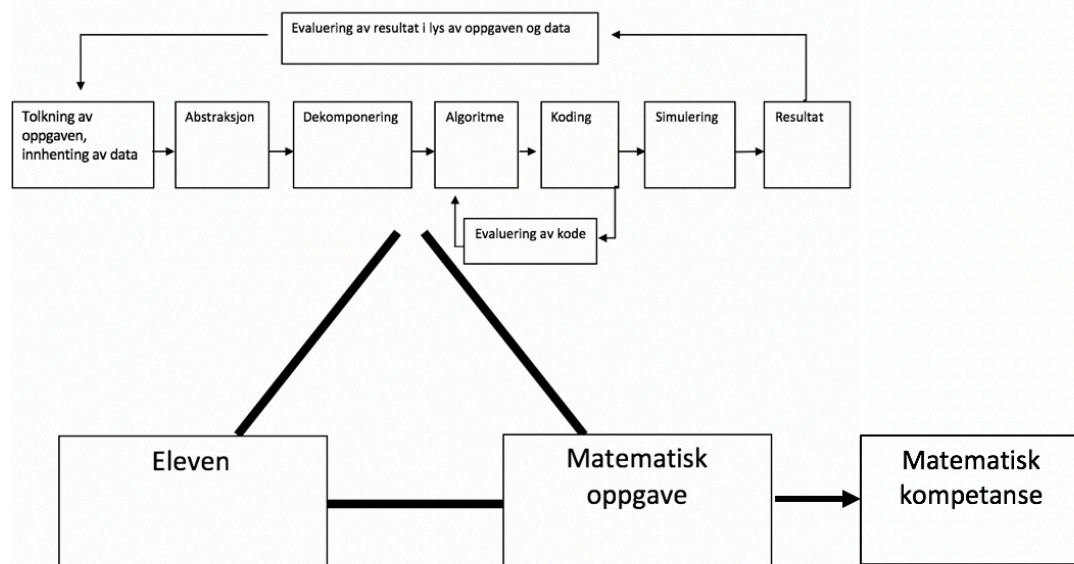
Evaluering handler om å forsikre seg om at løsningen man har endt opp med er korrekt og passende for oppgaven man satt ut for å løse. Aspekter ved dette er blant annet å gå gjennom algoritmen steg for steg, forsvare valg tatt under konstruksjonen av algoritmen og valg av hva du har utelatt i den, undersøke om koden man har produsert faktisk passer til situasjonen den

skulle behandle og undersøke om algoritmen/programmet i det hele tatt gjør det man ønsker at den skal gjøre (Csizmadia et al., 2015, s. 15) . Om det underveis i evalueringsprosessen viser seg å dukke opp feil eller mangler i algoritmen går man tilbake til der hvor feilen oppsto og retter opp i feilen, før man igjen kan evaluere algoritmen på nytt.

2.3 Algoritmisk tenking og programmering i syn av sosiokulturell teori

Teorien som er nevnt over om algoritmisk tenking og den sosiokulturelle læringsteorien, sammen med Niss matematiske kompetanser som beskrives senere i teorikapittelet, vil danne det teoretiske rammeverket jeg vil basere denne studien på og drøfte resultatet i lys av.

I dette rammeverket vil prosessen knyttet til algoritmisk tenkning, som med mitt fokusområde ender opp med en kode i Python, være det medierende artefaktet som er et hjelpemiddel for at eleven i arbeidet med de matematiske oppgavene skal oppnå og arbeide med matematisk kompetanse. Dette forholdet er illustrert i figuren under (figur 2.3).

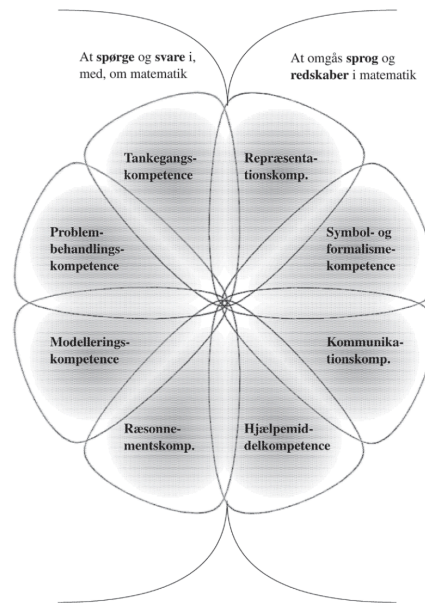


Figur 2.3: Algoritmisk tenking som medierende artefakt for å oppnå matematisk kompetanse fra et sosiokulturelt perspektiv

Her vil altså algoritmisk tenkning og programmering i Python betraktes som et medierende artefakt, som elevene kan bruke til å utvikle sine matematiske kompetanser. Om dette er tilfelle vil det bety at Python er et nyttig verktøy å ha med seg som lærer, da det kan bidra til å gi eleven økt kompetanse i faget i tillegg til at de som tidligere nevnt får kompetanse i programmering, som blir mer og mer relevant i et moderne arbeidsmarked og hverdagsliv.

2.4 Matematisk kompetanse

For å kunne svar på forskningsspørsmålene jeg ønsker å undersøke er det nødvendig å presisere hva jeg legger i begrepet kompetanse. Jeg kommer til å basere meg på kompetansene som ble presentert i det danske KOM-prosjektet, hvor målet var å gi en definisjon for hva det vil si å mestre matematikk. Prosjektet var rettet inn mot skolen, og var ment som en måte å kunne beskrive mestringsen på tvers av nivåer og alderstrinn. I prosjektet ble det presentert åtte kompetanser delt inn i to kategorier: å spørre og svare i, med, om matematikk og å kunne ta i bruk matematisk språk og redskaper (Niss, 2003). Disse kompetansene vises i figur 2.4, og vil bli beskrevet mer detaljert i de neste avsnittene.



Figur 2.4 En visuell fremstilling av kompetanse og kompetansekategoriene (Niss & Jensen, 2002, s. 45).

2.4.1 Å spørre og svare, i, med, om matematikk

Tankegangskompetanse omtales som å kunne stille spørsmål og vite hvilke typer svar som er karakteristiske for matematikkfaget. Det handler også om å kunne generalisere resultater og forstå omfanget og begrensningene til matematiske konsepter (Niss, 2003, s. 7).

Problembehandlingskompetanse handler om å kunne identifisere, stille og spesifisere matematiske problemer, samt å kunne løse ulike problemer stilt av en selv eller av andre. Denne kompetansen har en del til felles med tankegangskompetanse, men spørsmålene som stilles her er nærmere knyttet til selve problemløsningsprosessen (Niss, 2003, s. 7).

Modelleringskompetanse vil si å kunne konstruere egne modeller og å kunne analysere allerede eksisterende modeller. I tillegg til å kunne konstruere egne modeller nevnes å kunne evaluere egne modeller og kunne kommunisere godt om modellene og deres resultater (Niss, 2003, s. 7-8).

Resonnementekompetanse vil si å kunne følge og drøfte matematiske argumenter. Det handler også om å vite hva et matematisk bevis er og ikke er, samt å kunne komme med egne argumenter og sette dem sammen til matematiske bevis (Niss, 2003, s. 8).

2.4.2 Å kunne bruke språk og redskaper i matematikk

Representasjonskompetanse handler om å kunne ta i bruk og forstå ulike representasjoner av matematiske objekter, fenomener og situasjoner. Elevene skal også ha en forståelse for sammenhengen mellom ulike representasjoner og ha evnen til å ta i bruk og bytte mellom representasjonsformer (Niss, 2003, s. 8).

Symbol- og formalismekompetanse beskrives som å kunne tolke og forstå bruken av matematiske symboler og formelt matematisk språk. Elevene skal også klare å oversette fra daglig språk til formelt matematisk språk, samt å kunne manipulere og håndtere uttrykk som inneholder matematiske symboler (Niss, 2003, s. 8).

Kommunikasjonskompetanse brukes om elevens evne til å kunne uttrykke seg selv på en teknisk og teoretisk korrekt måte, og også kunne forstå andres kommunikasjon. Kommunikasjonen kan være muntlig, skriftlig eller visuell (Niss, 2003, s. 8).

Hjelpemiddelkompetanse handler om å vite om hvilke hjelpemidler som finnes til ulike matematiske prosesser, og å inneha ferdighetene til å ta disse i bruk. Elevene skal også kunne reflektere rundt bruken av hjelpemidlene og kunne se deres begrensninger (Niss, 2003, s. 9)

2.5 Tidligere forskning på programmering og algoritmisk tenking i matematikkundervisning

I og med at dette er en masteroppgave med et relativt begrenset omfang tidsmessig, vil det å se på tidligere forskning på samme felt som jeg vil undersøke kunne gi gode innspill og en pekepinn på hvordan mine funn forholder seg til tidligere resultater. En av artiklene jeg har sett på som kan gi et fundament for videre tanker er et litteratursøk av Forsstrøm og Kaufmann (2018). De tar for seg 15 artikler knyttet til programmering i matematikkundervisning, og viser til at resultater tyder på bedring av matematiske ferdigheter og motivasjon når elevene tar i bruk programmering. Av de kvantitative undersøkelsene de undersøkte viste alle en forbedring i matematiske resultater, spesielt innenfor geometri og problemløsning. De peker derimot også på at lite av forskningen som er utført kan generaliseres utover tilfellene de utforsker, noe som vil si at det ikke er noen garanti for at mine funn kommer til å støtte opp under deres.

I en studie av Psycharis og Kallia (2017) hvor en eksperimentell gruppe lærte en del av læreplanen ved å ta i bruk programmering, for så å bli sammenlignet med en gruppe som lærte samme tema uten hjelp av programmering, så forskerne tegn til flere positive forskjeller i favør gruppen som tok i bruk programmering i undervisningen. Et av disse punktene var en økt mestringstro hos elevene, som er en viktig faktor for elevers resultater i matematikk. De fant også tegn til at elever får økt kompetanse i resonnering når undervisningen inneholdt programmering. Det fantes i tillegg tegn til at elevenes kompetanse i problemløsning økte, men ikke i like stor grad som mestringstroen og resonneringsevnen.

Lie et al. (2017) finner i sin forskning tegn på at oppgaver som tar i bruk programmering i blokkprogrammeringsspråket Scratch gir tegn til økt kompetanse innen kommunikasjon, resonnering, problemløsning og modellering. I deres studie tok de i bruk åpne problemløsningsoppgaver og observerte blant annet at elever får til å overføre deres tanker rundt en oppgave til matematiske kommandoer, enten gjennom matematisk resonnering eller prøving og feiling. Dette peker på at elever har muligheter til å overføre tankene sine til kode selv om de ikke er eksperter på programmering, om det så er gjennom å prøve og feile til det blir rett.

Det finnes også statiske funn som peker mot at det å gi elever oppgaver som er designet for å fremme prosessene i algoritmisk tenkning, fremfor mer tradisjonelle lærebokoppgaver, gir elevene økt prestasjon i problemløsning. Det betyr at prosessene i algoritmisk tenkning kan være gunstige mekanismer for elevene å inneha når de skal arbeide med problemløsningsoppgaver (Costa et al., 2017, s. 8).

Et annet viktig aspekt å legge merke til i implementeringen av algoritmisk tenking og programmering poengteres i artikkelen til Hansen og Hadjerrouit (2021). De observerte tegn til at mangel på matematisk tenkning kan være et hinder for å oversette et matematisk problem til en algoritme og kode i et programmeringsspråk. Dette vil bli et viktig fokuspunkt i min konstruksjon av oppgaver, da jeg ønsker å undersøke elevers bruk av programmering, noe som ikke bør bli hindret av utilstrekkelig matematisk tenkning. Hansen og Hadjerrouit (2021, s.259) observerte også at mange studenter hopper for fort frem til konstrueringen av kode uten egentlig å ha arbeidet grundig nok med å lage en fornuftig og korrekt algoritme, noe som kan lede til problemer med å overføre algoritmen til en brukbar kode.

I en studie gjennomført i Sverige hvor forskerne undersøkte lærernes holdninger til programmering i skolen trekker flesteparten av lærerne frem at mangel på tid og kontinuitet vil

være et stort problem for innføringen av programmering i undervisningen (Humble et al., 2020, s. 121). De trekker frem at det vil være tidkrevende og vanskelig å få god kontinuitet i bruken av programmering, noe som vil være viktig da programmering faktisk krever en forståelse for et nytt språk og en ny måte å tenke på. Som en mulig løsning på dette trekkes blokkprogrammering frem som en måte å innføre programmering for elever som ikke har erfaring med dette fra før, da denne typen programmeringsspråk har en lav inngangsterskel, da elevene ikke trenger å lære å skrive selve kommandoene selv (Humble et al., 2020, s. 123). Det er derimot ikke denne typen programmeringsspråk utdanningsdirektoratet legger opp til at S1-elever skal ta i bruk, noe som presenteres mer i detalj i metodekapittelet.

I 2016 oppnevnte utdanningsdirektoratet en ekstern arbeidsgruppe ledet av Anders Sanne (2016, s. 38) for å foreta en gjennomgang av teknologi i grunnopplæringen. Blant funnene gjort i denne undersøkelsen så gruppen at en stor andel av elever har for dårlig digital kompetanse, og at blant dem med god digital kompetanse hadde skolen bare moderat betydning for deres utvikling. Videre anbefaler gruppa at skoleverket oppretter et helt eget teknologifag, der målet er å gi elevene en god digital kompetanse for å møte fremtidens arbeidsmarked. De referer også til at forskning peker mot at det ikke er gunstig å bare implementere programmering inn i allerede eksisterende fag, men at det er vil gi best utfall å opprette et helt nytt teknologifag (Sanne et al., 2016, s. 76). Funnene fra denne gruppen vil bety at slik programmering innføres nå ikke er optimalt med tanke på elevenes læring.

På tross av at mange artikler vektlegger at funn fra forskning rundt programmering i matematikk sjeldent kan generaliseres i nevneverdig grad, synes jeg allikevel at den store mengden funn som peker på at algoritmisk tenkning og programmering kan gi økt kompetanse hos elevene viser at det er verdt å prøve å implementere det i skoleverket. Her kommer også forskningen med enkelte fallgruver man bør unngå, og det å ha kunnskap om dette kan gjøre implementeringen lettere. Kunnskapsgrunnlaget som er lagt av blant annet forskningen som er nevnt over er noe jeg forhåpentligvis kan bygge videre på med resultatene fra denne oppgaven.

3 Metode

I dette kapittelet vil jeg presentere valgene jeg har tatt når det kommer til gjennomføringen av min forskning, og forsvare hvorfor nettopp disse valgene ble tatt. Jeg vil først presentere valg knyttet til metode, for så å presentere utvalg av deltakere, tema og oppgavesettet jeg har konstruert for gjennomføringen. Videre vil jeg presentere hvordan jeg analyserte data fra gjennomføringen. Til slutt vil jeg drøfte de etiske valgene jeg har tatt og se på validiteten og reliabiliteten til resultatene jeg får fra forskningen.

3.1 Syn på forskning

Før jeg presenterer hvilke valg jeg har tatt når det kommer til forskningsmetode og forskningsdesign ønsker jeg først å presentere mitt syn på hvilke typer kunnskap man kan få ut av et forskningsprosjekt som dette. Det å drøfte alle slike syn vil være både en epistemologisk og ontologisk debatt, som antakeligvis kunne vært en mastergrad i seg selv. Derfor velger jeg heller å presentere mitt syn, slik at leseren kan ha dette i bakhodet ved mine senere valg knyttet til metode og hvordan jeg tolker mine funn og resultater.

I dette arbeidet kommer jeg til å ha en interpretivistisk holdning til forskning. Å ha en interpretivistisk holdning vil si at forskere i sosiale situasjoner ikke kan se på hendelser uten å tolke dem i lys av personers subjektive syn på situasjonen. Forskerne er da ikke i like stor grad interessert i å hvorfor menneskers oppfører seg som de gjør, men er i stedet mer interessert i å forstå hvordan de oppfører seg i gitte situasjoner. Det vil føre til at man ikke kan se på resultater fra forskningen som frittstående fakta, man må også sette seg inn i den sosiale bakgrunnen til dem man studerer (Bryman, 2016, s. 26-27).

Dette synet på forskning passer godt med den sosiokulturelle læringsteorien jeg har basert meg på, som igjen er valgt med tanke på mine forskningsspørsmål, da begge krever at man ser på individet som en del av et sosialt samspill. Dette mener jeg passer bra i forskning knyttet til skole og undervisning, hvor alle elever vil ha forskjellige syn på blant annet hvor viktig det er for dem å prestere bra og lære mest mulig. Det vil også være vesentlige forskjeller i hvordan ulike lærere underviser, og derfor også forskjeller i hvordan elever angriper ulike oppgaver. Dette vil påvirke måten de arbeider med oppgavene jeg kommer til å gi dem, og dermed også resultatene dette arbeidet gir.

3.2 Forskningsdesign

Et av valgene man står overfor når man skal gjennomføre et forskningsprosjekt er hvorvidt man ønsker å designe forskningen etter kvalitative eller kvantitative prinsipper, eller å prøve å finne en blanding av dem (Bell & Waters, 2018, s. 24). Bell og Waters (2018, s. 24) beskriver kvantitativ forskning som forskning som samler inn fakta i større mengder og/eller sammenligner fakta fra ulike datasett. Denne typen forskning baserer seg ofte på fastsatte teoretiske rammeverk og tar i bruk fastsatte design, med mål om å produsere kvantifiserte og forhåpentligvis generaliserbare resultater. Bell og Waters (2018, s. 24) beskriver videre kvalitativ forskning som en type forskning der forskeren er mer opptatt av å se på individers forståelse av verden rundt dem, noe som ofte fører til et datasett basert på færre individer. Forskere kan her ofte starte med et bredere felt de ønsker å studere for så å spisse det inn etterhvert som forskningen går fremover. Denne typen forskning stiller også spørsmål ved hvorvidt man kan bruke statistiske metoder når det er mennesker som undersøkes.

Noen av fordelene ved å bruke kvalitative metoder er at man forstår funnene i konteksten de ble til i og man får muligheten til å se meninger og tankegangen bak resultater og handlinger hos enkeltindivider (Bryman, 2016, s. 401). Disse fordelene ser jeg på som gunstige når jeg skal drøfte funnene mine i lys av sosiokulturell læringsteori, som beskrevet i teoridelen handler

mye om hvordan læring blir til gjennom samhandling med omverdenen. Jeg ser også på dette som fordelaktig når jeg skal se etter kompetansene elevene innehar, da mye av dette kan komme frem ved å se tankegangen til elevene og å se på samtalene elevene har med hverandre.

Basert på dette har jeg valgt å ta i bruk kvalitativ forskning i denne studien da det å undersøke individers opplevelse av å bruke programmering i matematikk, samt å kunne få et innblikk i deres tanker og handlingsmønster, er mest hensiktsmessig med tanke på mine forskningsspørsmål.

Mer spesifikt vil oppgaven designes som en kasusstudie. En kasusstudie er en studie som studerer kompleksiteten og den spesifikke naturen til et enkelt «kasus», som kan være en person, en gruppe, en skole eller et samfunn (Bryman, 2016, s. 60). I denne oppgaven vil kaset være en skoleklasse. Bell og Waters (2018, s. 28-29) poengterer at kasusstudier kan legge til rette for at spesifikke aspekter av en situasjon kan studeres nøye. En slik studie kan også få frem samhandlinger blant individer og hvordan innføringen av systemer kan påvirke en gruppe, noe som kan være vanskelig i en storskala undersøkelse. Poengene Bell og Waters kommer med peker på at en kasusstudie kan være et passende design for mitt problemområde og mine forskningsspørsmål.

3.3 Forskningsmetode

For å samle inn data som kan svare på mine forskningsspørsmål har jeg valgt å gjennomføre et oppgavebasert intervju. Et oppgavebasert intervju (kjent som task-based interview på engelsk) er en type intervju som er designet slik at dem man intervjuer ikke bare er i kontakt med intervjueren, men også med en form for oppgave eller problem som intervjuobjektene skal arbeide med. Oppgaven er her nøye planlagt for å passe inn med det intervjueren ønsker å undersøke. Ved å ta i bruk denne typen intervju vil forskeren etter gjennomføringen sitte igjen med en observasjon av intervjuobjektene handlinger, samtaler og svar på oppgavene det arbeides med samt svar på spørsmål intervjueren har kommet med underveis (Maher & Sigley, 2020, s. 821-822). Goldin (2000, s. 520) legger frem flere områder hvor oppgavebaserte intervjuer er gunstige, blant annet for å undersøke psykologien bak å lære matematikk og som verktøy for å undersøke elevers matematiske kunnskap. En av grunnene til dette er at man kan skape en matematisk situasjon som man i stor grad kan kontrollere gjennom design av oppgavene som skal gjennomføres. Poengene Goldin legger frem viser at oppgavebasert intervju er en metode som passer godt for å undersøke mitt problemområde om metoden brukes på en god måte.

3.4 Valg av tema for oppgavene

Etter at jeg hadde bestemt problemområde for oppgaven og hvilken forskningsmetode jeg ønsket å ta i bruk, var neste steg å finne ut hvilket matematisk tema og kompetansemål fra læreplanen jeg ville rette oppgavene i oppgavesettet mot.

Som nevnt tidligere i oppgaven er programmering nå eksplisitt nevnt i den nye læreplanen, så man kan i grunn knytte programmering opp mot alle kompetansemålene, men noen virker med første øyekast mer egnet til bruk av koding enn andre. Et slikt kompetansemål, og det eneste kompetansemålet som eksplisitt nevnte å ta i bruk digitale verktøy og simulering, er kompetansemålet som lyder slik: «bruke digitale verktøy til å simulere og utforske utfall i stokastiske forsøk, og forstå begrepet stokastiske variabler» (Utdanningsdirektoratet, 2020b). Faglærer i klassen jeg skulle gjennomføre forskningen i var enig i at det var et passende tema, og et tema som passet godt inn med tanke på hvor elevene var i læreplanen i tidsrommet jeg ønsket å gjennomføre intervjuene.

3.5 Valg av programmeringsspråk

Som nevnt er programmering gjennom fagfornyelsen blitt en del av læreplanen. Det betyr at skoleverket må finne en fornuftig måte å innføre dette, og blant de valgene som må tas kommer valget av programmeringsspråk.

Det finnes i dag mange alternativer å velge mellom når man skal velge hvilket programmeringsspråk man ønsker å ta i bruk, alt fra å ta i bruk ferdige kommandoer som bygges sammen som legoblokker til språk som krever at du skriver ut koden selv. Av eksempler trukket frem i forskning nevnes ofte Python som et egent språk om man ønsker at eleven skal lære å skrive koder selv. Blant fordelene med Python nevnes at kodene er enkle å lese og intuitive å forstå, samt at det i Python allerede eksisterer gratis «biblioteker» som inneholder koder elevene kan ta i bruk (Mészárosóvá, 2015, s. 6).

Som et eksempel på et blokkprogrammeringsspråk trekkes ofte Scratch frem som et gunstig alternativ. En av fordelene med dette språket som inngangsport til programmering er at det eliminerer problemet med feilstavede kommandoer og feil tegnsetting, da kommandoene allerede kommer ferdig skrevne i form av blokker elevene skal sette sammen. Dette lar elevene fokusere på problemløsning, noe som kan være motiverende for elever med lite erfaring innen programmering (Mladenović et al., 2016, s. 1622-1623).

I et sett med eksempeloppgaver laget av utdanningsdirektoratet legges det opp til at Python er språket som kommer til å bli aktuelt ved eksamen i S1-matematikk, hvor blant annet en del 1-oppgave (uten hjelpemidler) ser slik ut:

Oppgave 5

```
1 def f(x):
2     return x/(1+x**2)
3
4 h = 0.0001
5 x = 0
6 while (f(x+h)-f(x))/h > 0:
7     x = x + 0.01
8 print("x=", x)
```

En elev har skrevet programkoden ovenfor.

- Hva ønsker eleven å finne ut?
- Forklar hva som skjer når programmet kjøres. Hva blir resultatet?

Eksempel på en oppgave fra del 1 på eksamen i S1(Utdanningsdirektoratet, 2021c)

Som et eksempel på en oppgave som kan komme på del 2 (med hjelpemidler) gis følgende oppgave, som krever at elever konstruerer en kode selv etter et par deloppgaver hvor de har regnet på en konto med renter:

Halvor ønsker å kunne regne ut hvor mange år det tar før beløpet han har på konto når en viss størrelse K .

- Lag et program som Halvor kan bruke.
Input skal være rentefot, innskudd og K .
Output skal være antall år det går før beløpet er større eller lik K .

Du kan for eksempel starte slik:

```
1 rentefot = 1.8
2 innskudd = 10000
3 K = 250000
4
5
6
7
8
9
10
11
12 print("Det tar", år, "år")
```

Eksempel på en oppgave fra del 2 på eksamen i S1(Utdanningsdirektoratet, 2021c)

Det kommer fra disse oppgavene tydelig frem at det Python som er det mest aktuelle programmeringsspråket med tanke på eksamen for elever i matematikk S1. Det var også dette språket læreren i klassen jeg skulle gjennomføre min studie i hadde brukt i sin undervisning. Disse to faktorene var avgjørende for at jeg valgte å ta i bruk Python som programmeringsspråk i oppgavesettet elevene skulle gjennomføre i denne studien.

3.6 Oppgavesettet

Før jeg satte i gang med konstruksjonen av oppgavesettet elevene skulle gjennomføre, hadde jeg først en samtale med faglæreren til klassen for å finne ut hvilket matematisk nivå elevene kunne forventes å ha både når det kom til programmering og stokastiske hendelser. Det kom da frem at ved tidligere arbeid med Python har elever i liten grad konstruert egne koder da de ikke hadde tilstrekkelig kjennskap til språket og programmering generelt. De har da heller sett på og tolket eksisterende koder, og videre sett etter hva som eventuelt er feil ved dem eller hvilke påbyggingsmuligheter de har.

Betydningen dette har for oppgavesettet jeg konstruerte er at det ville vært urimelig å be elevene konstruere lange og avanserte koder. Kodene de selv skal produsere vil derfor ikke kreve kjennskap til kompliserte kommandoer i Python, men heller se på om elevene klarer å tenke algoritmisk i enklere situasjoner og konstruere koder hvor de allerede har sett de nødvendige kommandoene.

Oppgavesettet er delt inn i to oppgaver, som hver har som mål å undersøke spesifikke deler av algoritmisk tenkning, programmering og matematisk kompetanse.

3.6.1 Oppgave 1

Etter samtalen med faglærer ble det som nevnt klart at elevene er vant til å først møte en tilnærmet ferdig kode som de så skal prøve å forstå og eventuelt bygge videre på. Oppgave 1 vil derfor basere seg på en ferdig kode som skal tolkes, og som videre kan brukes som en referanse og inspirasjon når de selv skal skrive egne koder i oppgave 2. Dette medfører at elevene i oppgave 1 ikke vil få bruk for alle stegene i algoritmisk tenkning, men hovedsakelig ta i bruk simulering og evaluering.

Oppgave 1 er delt inn i fem deloppgaver, og er bygget opp med mål om å gi elevene gode innblikk i stokastiske hendelser og hvordan antall forsøk vil påvirke resultatet. Dette vil skje spesifikt gjennom at simuleringene vil inneholde flere og flere myntkast utover i oppgaven.

Deloppgave a lyder slik:

I mattetimen får dere beskjed om å lage et program som skal simulere det å kaste en mynt, og som viser om dere får kron eller mynt. Programmet dere lager blir seende slik ut:

```
1 import random
2
3 mynt = ["Kron","Mynt"]
4 kast = random.choice(mynt)
5
6 print(kast)
7
```

Utsnitt fra oppgave 1 (Vedlegg 8.3)

a) Er det noen fordeler med å bruke programmering i slike forsøk?

Hovedmålet med denne oppgaven er å teste elevenes hjelpemiddelkompetanse, og se om de selv ser nytteverdien av å bruke programmering som hjelpemiddel i denne konteksten, eller om

de i denne situasjonen heller ville tatt i bruk en fysisk mynt. Her er det også mulig at noen av elevenes holdninger til programmering kommer frem.

Deloppgave b ser slik ut:

b) Hvordan ville dere gått frem for å få programmet til å kaste mynten to ganger?

Denne oppgaven tester om elevene har forstått koden de fikk presentert, ved å se på om de kan bygge videre på den ved å ta i bruk de samme kommandoene. Det vil derimot ikke være mulig å direkte kopiere den allerede eksisterende koden, da de må lage et nytt kast som programmet klarer å skille fra det første. Dette krever en viss forståelse for oppbygning av koder og algoritmer datamaskinen kan forstå. Denne oppgaven er i tillegg ment som en hjelp til koden de skal lage selv i oppgave 2.

Deloppgave c lyder slik:

Etter å ha kjørt dette programmet et par ganger ser dere at vi ikke har fått akkurat like mange kron som mynt, noe man kanskje skulle tro fra det vi vet om sannsynlighet. For å teste dette videre bestemmer dere dere for å lage et program som simulerer 50 myntkast for å se om det da blir et jevnere utfall. Programmet blir sendt slik ut, men fungerer ikke:

```
1 import random
2
3                                     # Presiserer at alle
4 Kron = 0                             #variabler starter på 0
5 Mynt = 0
6
7 while kast < 50:                       #Dette forteller hvor mange
8     kast += 1                           # kast som skal gjennomføres
9
10    utfall = random.randint(1, 2)        #random.randint(1,2) får
11    if utfall == 1:                       #programmet til å velge et
12        Kron += 1                          #heltall fra og med 1 til og
13    else:                                  #med 2
14        Mynt += 1
15
16 print("Vi fikk til sammen,", Kron, "kron og", Mynt, "mynt etter", kast, "kast")
17
```

Utsnitt fra oppgave 1 (Vedlegg 8.3)

c) Finn feilen i koden, og endre den så simuleringen fungerer.

Ett av målene med denne oppgaven er å teste elevenes evaluering av koder. Dette er en kode som ikke fungerer fordi variabelen «kast» ikke er satt til null fra starten av, og programmet vet derfor ikke hvor det skal begynne å telle fra. Her vil det bli mest interessant å høre om elevene har en strategi når de evaluerer koden, noe som vil være en god egenskap å ha i arbeidet med å lage egne koder.

Deloppgave d ser slik ut, og referer til koden de rettet på i forrige deloppgave:

d) Kjør simuleringen flere ganger, og se på resultatet. Vil du basert på det du ser si at det å regne med sannsynlighet er en fornuftig måte å anslå virkelige hendelser på?

Med denne oppgaven ønsker jeg å undersøke flere av kompetansene som er nevnt i teorikapittelet, og som knyttes til forskningsspørsmålet om matematisk kompetanse. En av kompetansene som kan komme frem her er elevenes resonnementskompetanse, da elevene her må fremme egne argumenter basert på funnene fra koden de har tatt i bruk. Oppgaven vil også teste elevens tankegangskompetanse, ved at svaret de gir vil handle om omfanget og begrensningene til sannsynlighetsregning. Måten de uttrykker seg rundt simuleringens resultater vil i tillegg teste kommunikasjonskompetansen deres.

Til sist kommer deloppgave e, som lyder slik:

- e) Hvordan kan koden endres dersom du ville demonstrert at simuleringen ikke motbeviser at det er 50% sannsynlighet for begge utfallene?

Det jeg ønsker å undersøke med denne oppgaven er om elevene etter å ha gjennomført simuleringene, om ikke de kan det fra før, har fått en forståelse av at tallene vi tar i bruk når vi regner med sannsynlighet er basert på om vi hadde gjort forsøket uendelig mange ganger. Om elevene ser dette gjennom simuleringen, vil det peke mot at simulering kan være et godt egnet medierende artefakt i arbeidet med å undervise stokastiske variabler. Her vil de kunne endre antall kast i den forrige koden for å visualisere fenomenet selv.

Denne deloppgaven vil ikke bare teste om programmering egner seg som en måte å lære bort dette temaet, men vil også teste elevenes tankegangskompetanse, resonnementskompetanse og kommunikasjonskompetanse som i forrige oppgave.

3.6.2 Oppgave 2

Etter å ha sett kodene i oppgave 1 og fått tid til å arbeide med og evaluere dem, skal elevene i oppgave 2 forsøke å lage egne koder. Dette gjør at elevene i motsetning til i oppgave 1 får rom til å ta i bruk alle stegene i algoritmisk tenkning. Oppgaven de får presentert ser slik ut:

Oppgave 2: Terningkast

Dere er på hyttetur og har nettopp satt dere ned for å spille yatzy, men finner fort ut at ingen av terningen er der de skal. Det dere derimot har er en pc, og får en ide om at dere bare kan lage et program som kan simulere terningkastene.

- a) Prøv å lag et slikt program (Vanlig yatzy har fem terninger som kastes samtidig)

Hint: Når man jobber med programmering er et lurt skritt å ta det vi kaller dekomponering, som betyr å dele problemet opp i mindre deler som hver for seg er lettere å løse. Det kan være lurt å prøve på her også.

Denne oppgaven gis både for å gi elevene mulighet til å utvikle egne ferdigheter knyttet til å programmere selv, samtidig som at den gir meg mulighet til å analysere elevenes arbeid med alle stegene i algoritmisk tenkning, enten på eget initiativ eller etter spørsmål fra meg.

I arbeidet med denne oppgaven vil elevene ha mulighet til å vise flere av Niss matematiske kompetanser. To åpenbare kompetanser som kan vises i dette arbeidet er problembehandlingskompetanse og modelleringskompetanse, da elevene gjennom å arbeide med dette problemet lager en modell for kast av terninger. Om elevene klarer å lage koden får de også vist representasjonskompetanse, da dette er en måte å representere den fysiske hendelsen å kaste fem terninger. Det å ta i bruk programmering her viser i tillegg at elevene innehar god hjelpemiddelkompetanse, da de får innblikk i hvordan programmering kan brukes som hjelpemiddel til å løse matematiske problemer.

En mulig måte å løse denne oppgaven på, og den måten som ligger tettest opp mot oppgave 1 ser slik ut:

```

1 import random
2
3 terning= ["1", "2", "3", "4", "5", "6"]
4
5 kast1= random.choice(terning)
6 kast2= random.choice(terning)
7 kast3= random.choice(terning)
8 kast4= random.choice(terning)
9 kast5= random.choice(terning)
10
11 print(kast1 + kast2 + kast3 + kast4 + kast5)

```

Et forslag til hvordan elevene kan løse oppgave 2

Hintet som legges til i slutten av oppgaven er lagt til fordi elevene har begrenset erfaring med programmering og veien mot en ferdig kode fra før. Hintet vil derfor kunne hjelpe dem med å huske at dekomponering kan være en god måte å angripe en slik oppgave på.

En mulig måte å dekomponere dette problemet på er å først fokusere på å lage en fungerende kode som viser ett terningkast, før man bygger videre til fem kast. En kode med bare ett kast kan se slik ut:

```

1 import random
2
3 terning= ["1", "2", "3", "4", "5", "6"]
4 kast=random.choice(terning)
5 print(kast)
6

```

Kode som simulerer ett terningkast

Oppgavesettet ble før gjennomføringen med elevene pilotert sammen med medstudenter ved UiA. Her var hovedfokus på formuleringer og vanskelighetsgrad på programmeringen som krevdes. Det kom fra denne piloteringen frem at det å spørre elevene om å eksplisitt drøfte stegene i algoritmisk tenkning kunne være fornuftig for å hjelpe elevene i gang med oppgave 2 og for å få stegene tydelig frem i datamaterialet.

Oppgavesettet i sin helhet slik det ble presentert for elevene kan sees i avsnitt 8.3.

3.7 Utvalg og gruppesammensetning

Klassen jeg gjennomførte forskningsopplegget med var en ganske liten klasse med åtte elever. Samtlige elever hadde lyst til å være med på forskningen, noe som passet bra med mine tidsbegrensninger, da det gjorde det mulig å ha fire grupper med to elever i hver gruppe. Det at hele klassen deltok i forskningen medførte at jeg ikke aktivt trengte å ta valg knyttet til hvem som ble med i utvalget, noe som igjen medførte at mine preferanser eller tilfeldige valg ikke påvirket hvem som endte opp som datagrunnlag for denne oppgaven. Dette gjorde at jeg dekket hele populasjonen jeg var inne og undersøkte. Det å få dekket et representativt utvalg av populasjonen man setter ut for å undersøke nevnes av Bell og Waters (2018, s. 31) som et av de viktigste aspektene ved å velge et utvalg til forskningen. Det at jeg tok med hele klassen gjør at jeg med trygghet kan si at mine funn kommer fra et representativt utvalg av klassen jeg har vært inne i, men kan derimot ikke automatisk si at jeg dekker et representativt utvalg av alle S1 elever i landet. Bryman (2016, s. 417) poengterer at det å ikke ha for mange intervjuer å gjennomføre og analysere gir forskeren mulighet til å gå grundigere inn i hvert intervju, noe som åpner for en mer grundig analyse. Med den mengden intervjuer jeg gjennomførte vil det å kunne analysere alle grundig fortsatt være mulig med tanke på tiden jeg har til rådighet for denne oppgaven.

Når det kommer til gruppesammensetning er et viktig spørsmål hvorvidt man ønsker at deltakerne i gruppene skal kjenne hverandre, og i så fall i hvor stor grad. En av fordelene med å ha grupper hvor deltakerne kjenner hverandre er at det kan føre til at samtaler blir mer naturlige, da personene allerede er vant til å arbeide og kommunisere sammen (Bryman, 2016, s. 510). I samtale med faglærer til klassen kom det frem at dette kunne være en fornuftig løsning da elevene er unge, og det er mer sannsynlig at de er komfortable om de arbeider sammen med noen de kjenner. Dette vil kunne føre til at de velger å delta mer aktivt i samtaler og arbeidet med oppgavene, spesielt med tanke på at de blir observert av meg som for dem er en relativt ukjent person. Det som derimot kan skje når gruppene fordeles etter dette kriteriet er at noen av gruppene kan ha en deltaker som er mer dominerende enn den andre (Bell & Waters, 2018, s. 215). Dette var noe jeg hadde i tankene under min forskning, og jeg forsøkte aktivt å få frem tanker og kompetanser fra alle gruppelemmene.

3.8 Gjennomføringen av forskningen

For å forsikre meg om at jeg får dekket alt jeg ønsker å undersøke for å besvare forskningsspørsmålene, konstruerte jeg en intervjuguide som jeg hadde med meg under elevenes arbeid med oppgavene. Bryman (2016, s. 469) beskriver en intervjuguide som en liste med temaer eller spørsmål intervjueren har foran seg under intervjuet. Dette skal være spørsmål som er åpne nok til at intervjuobjektene skal ha rom til å svare ut ifra deres perspektiver. Dette krever en fleksibilitet i intervjuet. Intervjuguiden jeg har laget baserer seg på å gi elevene mulighet til å vise de matematiske kompetansene de innehar, da noen av dem kanskje ikke dukker naturlig opp i deres drøfting av oppgavesettet. Intervjuguiden kan sees i sin helhet i vedlegg 8.4.

Før gjennomføringen observerte jeg to introduksjonstimer for temaet sannsynlighet for å se hvordan læreren presenterte det for elevene, så jeg kunne legge opp oppgavene etter hva læreren hadde presentert og hvordan læreren formulerte temaet. Det å delta i disse timene var for meg viktig for å vite mer om hvilket grunnlag elevene hadde, og for å unngå at oppgavene mine hadde for høyt eller lavt nivå. I denne timen presenterte læreren kort om sannsynlighet, stokastiske variabler og litt om relativ frekvens, og hvordan man regner ut dette. Denne timen ga en bekreftelse på at mine oppgaver virket til å ligge på rett vanskelighetsgrad og tematikk.

Selve gjennomføringen av intervjuet foregikk ved at jeg tok hver gruppe ut av klasserommet og inn i ett grupperom hvor elevene arbeidet med oppgavene og ble intervjuet samtidig som lyd ble tatt opp. Jeg hadde 45 minutter til rådighet i hver gruppe, og gjennomføringen varte mellom 35 og 45 minutter.

3.9 Analysemetode

3.9.1 Transkribering

Etter gjennomføringen av intervjuene transkriberte jeg lydopptakene så fort som mulig, for å ha situasjonene og stemningen friskt i minne. Noen av fordelene med å ta i bruk transkribering er at det hjelper til der hukommelsen kan utelate viktige detaljer fra samtaler, og at det lar forskeren studere samtaler til elevene flere ganger (Bryman, 2016, s. 479). I tillegg fører det til at andre får tilgang på dataen forskeren bruker, noe som gjør det mindre aktuelt å kritisere forskeren for måten han/hun selv har tolket datasettet, da de selv kan undersøke samme datasett. Det å gjøre transkriberingen grundig er derfor noe jeg vil vie mye tid til, for å forsikre meg om at datasettet representerer den faktiske situasjonen på en best mulig måte. Det må også nevnes at i overgangen fra en virkelig situasjon til å bare sitte igjen med transkribert tale forsvinner deler av vår kommunikasjon, som f.eks. kroppsspråk og gestikulering. Jeg vil prøve å gjøre opp for dette ved å skrive ned ting jeg ser som bør nevnes, og ved å legge inn transkriberingen hvor

det er pauser og elevene stopper opp. Tegnene transkriberingen vil inneholde utover ordene deltakerne sier forklares i tabell 3.1.

Tegn	Betydning
..	Pause på 1-2 sekunder
...	Pause på 3 sekunder eller mer
(...)	En beskrivelse av noe jeg observerer som ikke er verbalt
[...]	Kommentar til det elevene sier. For eksempel om det ikke kommer tydelig frem hva de sikter til når de prater.
?	Indikerer nå noe sies som et spørsmål eller på en spørrende måte
!	Indikerer at deltakeren sier noe med tydelig entusiasme

Tabell 3.1

For å sikre anonymiteten til deltakerne vil de i transkripsjonene ikke stå med sine faktiske navn. Jeg har derfor valgt å navngi deltakerne som 1A og 1B i gruppe 1, 2A og 2B i gruppe to og tilsvarende for gruppe 3 og 4. Selv vil jeg stå oppført som intervjuer.

3.9.2 Analyse av transkribert materiale

Etter å ha transkribert lydfilene fra intervjuene har jeg gjennomgått datamaterialet i to omganger, en i de faktiske intervjuene og gang som lydfile under transkriberingen. Dette ga meg god innsikt i elevens arbeid og samtaler, og ga meg et godt grunnlag for å analysere funnene mine i lys av forskningsspørsmålene jeg har ønsket å undersøke og teorien jeg har basert meg på. For å gjøre transkriberingene enklere å analysere valgte jeg å fargekode elevens uttalelser for å sortere dem inn i grupper som var aktuelle for å besvare mine spørsmål. Bell og Waters (2018, s. 271) beskriver koding som det å sette merkelapper på deler av dataen man samler inn for å samle sammen data til ideer som kan hjelpe deg på vei til å trekke konklusjoner.

Når man skal lage slike koder som merkelapper for transkripsjonen er induktiv og deduktiv metode to alternativer for å konstruere koder. I induktive metoder bruker man datamaterialet man har og ser etter mønstre i dataene for å finne passende koder. I deduktive metoder tar man i bruk allerede eksisterende teori som basis for å lage koder, som man så leter etter i eget datasett (Bell & Waters, 2018, s. 23). Med tanke på at mine forskningsspørsmål er tett knyttet opp mot eksisterende teori ser jeg på en mer deduktiv metode i samspill med de empiriske dataene som mest gunstig for denne studien, og tar derfor i bruk en slik metode i analysen av mitt datasett.

Fargekodene jeg valgte å ta i bruk representerer hver av de åtte kompetansene til Niss som nevnes i teoridelen og forskningsspørsmålene. Jeg har også i transkripsjonen valgt å skrive inn når elevene tar i bruk de ulike stegene i prosessen knyttet til algoritmisk tenkning. Rent praktisk er disse skrevet inn som en kommentar ved siden av uttalelsene og fargekodene. Hver fargekode har fått sin egen kolonne i et Excel-dokument. I uttalelser hvor noen av kompetansene ikke vises vil derfor rutene forbli ufargede. De åtte kompetansene til Niss har jeg kodet ved hjelp av fargekoder som beskrevet i tabell 3.2:

Kompetanse	Fargekode
Tankegangskompetanse	TAN
Problembehandlingskompetanse	PRO
Modelleringskompetanse	MOD
Resonnementskompetanse	RES
Representasjonskompetanse	REP
Symbol- og formalismekompetanse	SYM
Kommunikasjonskompetanse	KOM
Hjelpemiddelkompetanse	HJE

Tabell 3.2

Et eksempel på hvordan analysert og transkribert materiale da vil se ut er utdraget under fra gruppe 4:

188	Intervjuer	bra, er dere fornøyd med det koden viser?								
189	4A	Ja, og det enkleste er jo bare å ta et ark og skrive ned dem man vil beholde								

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.8.1)

3.10 Forskningsetikk

For å forsikre meg om at forskningen ble gjennomført på en etisk forsvarlig måte søkte jeg først til Norsk senter for forskningsdata (NSD), noe som også er lovpålagt for å samle inn persondata. I søknaden beskrev jeg hvordan jeg planla å gjennomføre forskningen for å være sikker på at den var i tråd med reglene som finnes om personvern og datainnsamling. Det var først når NSD godkjente planen min at jeg satte i gang med gjennomføringen av intervjuene. Et av stegene som ble gjennomført for å få godkjenning fra NSD var at alle deltakerne før gjennomføringen kunne begynne måtte signere på et samtykkeskjema (vedlegg 8.2) hvor det ble forklart hvilke data som kom til å bli samlet inn. Alle elevene som var under 18 år måtte da også få signatur fra foresatte før datainnsamlingen kunne begynne.

Det var også viktig å sikre at anonymiteten til elevene kunne garanteres. Et av stegene som tas for å hindre at elever skal kunne gjenkjennes er at elevene i alle transkripsjoner, oppgavebesvarelser og i selve forskningsoppgaven anonymiseres. Jeg har dermed etter beste evne fjernet alt i datasettet som kan peke mot identiteten til elevene etter at lydopptaket fra intervjuene er slettet, noe som skal gjøres rett etter endt transkribering. Jeg har også valgt å ikke nevne hvilken skole elevene går på eller navn på faglærer, da dette kunne vært med på å identifisere klassen, og dermed også enkeltelever.

3.11 Resultatenes reliabilitet og validitet

Spørsmålet om forskningens reliabilitet handler om hvorvidt forskningen ville fått samme resultat om den ble gjennomført av andre forskere og på et annet tidspunkt (Bryman, 2016, s. 41). Bryman (2016, s. 383) peker på at dette er et spesielt stort problem i kvalitativ forskning, da det vil være umulig å fryse eller gjenskape den eksakte sosiale situasjonen og de sosiale forholdene forskningen originalt ble utført i. Derfor må en forsker som skal prøve å gjenskape resultatene være nøye med å ta en så lik sosial rolle som den originale forskningen som mulig, og reflektere over hvordan forskeren selv påvirker forskningssituasjonen.

Et problem Bell og Waters (2018, s. 183) trekker frem når det kommer til reliabiliteten til resultater som stammer fra intervjuer, er at meningene som ligger bak svarene er i stadig endring. Dette kan skje av flere grunner som for eksempel at ny informasjon kan ha dukket opp,

eller at en endret situasjon gjør at intervjuobjektene ønsker å fremstå annerledes. Dette kan medføre at utfallene fra en lignende gjennomføring kunne gitt annerledes svar om spørsmålene hadde blitt stilt av faglærer, eller om de ikke hadde vært en del av prosjekt som tydelig hadde innføring av programmering som tema. Det å gjennomføre samme forskningen på et annet tidspunkt og en annen sosial situasjon kan derfor potensielt gi enkelte forskjeller fra mine funn noe som viser at reliabiliteten til denne forskningen blir svekket av min forskningsmetode. For å motvirke dette er det viktig at en forsker som vil teste funnene fra denne oppgaven inntar en lik sosial rolle som intervjueren har tatt i denne gjennomføringen, noe jeg har forsøkt å gjøre mulig ved å konstruere en intervjuguide som beskriver rollen jeg har inntatt og spørsmålene jeg har kommet med, samt presentert hva jeg ønsker å undersøke i hver av oppgavene.

Et annet steg i evalueringen av forskning, og et steg som av Bryman (2016, s. 41) omtales som kanskje enda viktigere enn reliabiliteten, er å se på forskningens validitet. Å undersøke forsknings validitet handler om å se på integriteten til konklusjonene som trekkes i forskningen. Bryman (2016, s. 41) nevner flere aspekter ved validitet, blant annet intern validitet. Intern validitet vil si å sjekke om konklusjoner som trekkes rundt sammenhenger mellom ulike variabler kan stamme fra andre forhold enn det forskningen påstår at det stammer fra. Jeg har i denne oppgaven forsøkt å styrke studiens interne validitet ved å gjennomgå situasjonen jeg undersøker grundig i analysen av mitt datagrunnlag, og forsøkt å gjennomgå hvilke faktorer som kan påvirke sammenhenger som trekkes derfra. Jeg har gjennom dette forsøkt å forsikre meg om at sammenhenger som trekkes i min forskning skyldes de variablene jeg påstår at de skyldes ved å belyse andre mulige variabler og hvordan de kan påvirke funnene, som for eksempel elevenes bakgrunnskunnskaper og holdninger til programmering. Det er derimot ingen garanti for at min vurdering av variabler er korrekt, noe som kan medføre at jeg undervurderer viktigheten av eller utelater enkelte faktorer.

Et annet viktig aspekt ved validitet som nevnes av Bryman (2016, s. 41) er begrepsvaliditet. Dette handler om hvorvidt undersøkelsesmetoden faktisk undersøker de begrepene den setter ut for å undersøke. Dette er et viktig aspekt å ha et bevisst forhold til da mange begreper i forskning ikke direkte lar seg observere, spesielt innenfor psykologi. For å forsikre meg om at min forskning har så god begrepsvaliditet som mulig har jeg satt meg godt inn i begrepene fra det teoretiske rammeverket, for så å konstruere et oppgavesett og en intervjuguide som legger opp til at de teoretiske begrepene rundt algoritmisk tenkning og Niss matematiske kompetanser skal bli en del av det empiriske arbeidet, noe som gjør at de ønskede begrepene blir tydeligere og enklere å fange opp i analysen av datasettet.

Ekstern validitet nevnes også som en viktig del av validiteten til en studie, og beskrives som i hvor stor grad resultatene av en studie kan generaliseres utover det enkelte forskningstilfellet (Bryman, 2016, s. 42). Når det kommer til hvorvidt resultatene jeg kommer til er generaliserbare, poengterer Goldin (2000, s. 528-529) at ved bruk av oppgavebaserte intervjuer vil ingen intervjuer være like og det vil være flere variabler forskere ikke kan kontrollere. Dette gjelder også mine intervjuer, noe som svekker resultatenes generaliserbarhet. For å gjøre forskning generaliserbar i størst mulig grad nevner Goldin (2000, s. 530) videre at man må beskrive metodene og funnene sine på en måte som åpner opp for at andre forskere kan gjennomføre liknende metoder, noe som vil gjøre det mulig å sammenligne funn for så å enten underbygge eller motsi tidligere funn. Det å gjøre mine valg knyttet til metode så klare som mulig har derfor vært et viktig fokusområde i formuleringen av metodekapittelet, så funnene og konklusjonene jeg trekker ut ifra dem i størst mulig grad kan føre til reel læring og kunnskap som ikke bare gjaldt situasjonen jeg undersøkte.

4 Resultater og analyse

I dette kapittelet vil jeg presentere resultatene fra gjennomføringen av oppgavesettet og intervjuene med elevene. Datamaterialet vil presenteres gruppevis, og jeg vil forsøke å presentere det så komplett og presist som mulig. Resultatene vil derfor ikke diskuteres og drøftes før i neste kapittel. Jeg velger å presentere funnene fra hver gruppe hver for seg, da gruppesammensetningene hadde enkelte forskjeller i bakgrunnskunnskaper. I presentasjonen vil jeg også skille mellom oppgave 1 og 2 da de krever ulike deler av algoritmisk tenkning.

Resultatene vil knyttes opp mot teorien om matematisk kompetanse og algoritmisk tenkning ved hjelp av analyseverktøyet nevnt i avsnitt 3.9.2. Til slutt vil jeg kort oppsummere hvilke sammenhenger mellom algoritmisk tenkning og de nevnte matematiske kompetansene som kommer frem i analysen.

4.1 Gruppe 1

I gruppe 1 hadde ingen av elevene tidligere kjennskap til programmering i andre fag enn matematikk. De vil derfor ha mindre erfaring med å skrive koder og å konstruere algoritmer enn enkelte av de andre gruppene. Dette kan medføre at denne gruppen kan fungere som et eksempel på hvordan programmering og algoritmisk tenkning tas i bruk blant elever dersom de bare eksponeres for programmering gjennom matematikkundervisningen.

4.1.1 Oppgave 1

Allerede fra starten av intervjuet kom det frem at elevene føler at de aldri grundig har lært å programmere, og at de bare har vært innom det i liten grad. Dette stemmer godt overens med det faglæreren til klassen har sagt.

1	Intervjuer	Har dere hatt noe programmering før?																		
2	1A	Altså, vi har jo hatt det her [I mattetimen]																		
3	1B	Men aldri lært det liksom																		HJE

Utdrag fra åpningen av intervjuet med gruppe 1 (Vedlegg 8.5.1)

Elev 1B viser her evnen til å tenke kritisk rundt eget bruk av hjelpemidler ved å ytre at han/hun ikke har lært å bruke hjelpemiddelet grundig. Dette viser at eleven innehar god kompetanse når det kommer til å reflektere rundt eget bruk av hjelpemidler, og i hvor stor grad han/hun klarer å ta i bruk hjelpemidlene på en god måte.

For å fremheve noen av kompetansene elevene viste gjennom arbeidet med oppgave 1 har jeg valgt å trekke frem to sekvenser der elevenes bes om å drøfte funnene de fikk fra simuleringene i lys av det de allerede vet om sannsynlighet. I det første utdraget bes elevene tolke funnene sine fra de første myntkastsimuleringene i oppgave 1D.

128	Intervjuer	Med det dere fikk fra denne koden, vil dere si at sannsynlighet er en fornuftig måte å beregne virkelige hendelser på?																		
129	1B	Den treffer jo sånn cirka, men det er jo ikke hver gang	TAN		MOD	RES														KOM
130	Intervjuer	Men hadde dere turt å vedde penger på det																		
131	1B	Nei, det hadde ikke jeg veddet noe på																		
132	1A	Men vi får det jo en del																		
133	1B	Men kan ikke stole helt på det	TAN																	

Utdrag fra elevenes diskusjon av oppgave 1D (Vedlegg 8.5.1)

Her viser elevene at de gjennom bruk av simulering som medierende artefakt ser noen av begrensningene ved sannsynlighetsregning, noe som er med på å styrke deres tankegangskompetanse. De arbeider her også med å være kritiske til modellen de har laget, samt å kommunisere om både modellen, funnene de har gjort ved hjelp av modellen, og videre

knytte dette opp mot teori. Elevenes resonnementskompetanse styrkes også her da de arbeidet med å se etter hva som kan brukes i matematiske argumenter for å bygge opp under matematiske ideer.

I det neste utdraget bes elevene oppsummere hva simuleringene har vist dem etter at de har gjennomført alle deloppgavene til oppgave 1.

159	Intervjuer	Kunne dere laget en setning som oppsummerte hva vi hadde funnet ut om sannsynlighet ved å bruke denne koden?								
160	1A	Det er jo 50-50 hver gang, men jo flere ganger du kaster jo mer sentrerer du til sammen for å likt resultat mellom kron og mynt	TAN		RES		SYM	KOM		

Utdrag fra elevenes oppsummering av hva de har lært i oppgave 1 (Vedlegg 8.5.1)

Her ser vi igjen at simuleringene hjelper elevene med å se omfanget og begrensningene ved temaet stokastiske hendelser, noe som er med på å bygge opp tankegangskompetansen deres. Også her får elevene arbeidet med kommunikasjon, og hvordan de kan presentere funnene sine på en mer formell matematisk måte gjennom å oppsummere funnene sine i en setning. Elevene får i tillegg bygd videre på sin resonnementskompetanse gjennom å kunne bruke simuleringer og forsøk som basis for å argumentere rundt matematisk teori. Styrkingen av både kommunikasjonskompetanse og resonnementskompetanse knyttes opp til simulering her fordi simuleringen gir et svar som ikke er ferdig uttrykket i setninger og teoremer, noe som gjør at elevene må formulere disse selv ovenfor både seg selv, medelever og intervjuer.

Elevene i denne gruppen viser ved flere anledninger gode ferdigheter når det kommer til å konstruere koder gjennom å ta i bruk algoritmer og koding. Vi kan se i utdraget under at det å ta i bruk disse stegene i algoritmisk tenkning er et godt medierende artefakt for å bedre elevenes modelleringskompetanse og hjelpemiddelkompetanse. Modelleringskompetansen bygges her ved at elevene bevisst tenker gjennom hva de ulike stegene i algoritmen betyr for modellen. Hjelpemiddelkompetansen bygges gjennom at de analyserer koder og på den måten lærer relevante kommandoer og tegnsetting i Python som kan hjelpe dem når de skal skrive egne koder.

109	Intervjuer	Hva er det egentlig kast betyr i vår kode?								
110	1B	Det er jo hvor mange ganger den skal, så er det fra 1 til 50?	MOD						Algoritme	
111	Intervjuer	Så hva er da lurt å starte med kast som								
112	1B	en eller null?	MOD					HJE	Algoritme	
113		(skriver inn kode med kast som 1)						HJE	Koding	
114	Intervjuer	Hvor mange kast har vi fått nå?								
115	1A	49.								
116	1B	så da er det null	MOD						Evaluering	
117		(Endrer koden)						HJE	Koding	

Utdrag fra elevenes arbeid med oppgave 1C (Vedlegg 8.5.1)

Det kommer også frem i intervjuet at elevene evner å tenke kritisk rundt egnetheten og bruken av hjelpemidler. Et eksempel på det er utdraget under hvor elevene blir spurt om programmering var et fornuftig hjelpemiddel i de tidlige myntkastene. Elevene svarte her at ved få myntkast var det ikke fornuftig å gå gjennom prosessen med å lage en kode når man raskere og enklere bare kunne kastet en mynt. De ser derimot verdien allerede her av å kunne ta i bruk simulering når mynten skal kastes flere ganger. Dette viser at de klarer å se både positive og negative sider ved hjelpemidlene de tar i bruk.

24	Intervjuer	Så da er jo spørsmålet, er det egentlig noen fordel med å bruke simulering her istedenfor å bare kaste en mynt?							
25		...							
26	Intervjuer	Føler dere det er noe poeng når dere ser på denne koden?							
27	1A	Når det bare er ett myntkast mener jeg man bare kunne gjort det, men hvis man skulle sett på sannsynlighet med mange kast hadde det vært lurt å gjøre sånn [simulering]			MOD	REP			HJE

Utdrag fra elevenes arbeid med oppgave 1A (Vedlegg 8.5.1)

4.1.2 Oppgave 2

Det at elevene ikke tidligere har hatt en grundig gjennomgang av programmering betyr også at de ikke har god kjennskap til stegene i algoritmisk tenkning. Dette medførte at de ikke tok i bruk abstraksjon og dekomponering på eget initiativ ved starten av oppgave 2. Arbeidet til elevene viser derimot at de har kompetanser til å gjennomføre disse prosessene når de først blir spurt om dem, noe som kan sees i utdraget under. Her viser elevene at når de først blir spurt om disse prosessene kommer de med fornuftige tanker og valg. Elevene i denne gruppen innehar altså kompetanse nok til å gjennomføre både abstraksjon og dekomponering, men ikke kompetanse nok om selve prosessen rundt algoritmisk tenkning til å automatisk gjennomføre stegene.

175	Intervjuer	Hva er det essensielle å få med							
176	1B	Altså vi vil jo bare ha med tallet	PRO	MOD	REP				Abstraksjon
177	Intervjuer	Det å ta bort sånn unødvendig informasjon kaller vi abstraksjon i programmering							
		Da kan neste steg være å prøve å gjøre problemet litt enklere for oss selv. Å hoppe rett inn i et program med fem terninger kan være litt vanskelig. Kan dere prøve å gjøre det problemet litt mindre?							
178	Intervjuer								
179	1B	Vi kan begynne med en terning kanskje?	PRO	MOD					Dekomponering
180	1A	Så kan vi heller legge på flere kast	PRO	MOD					Dekomponering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.5.1)

Fra utdraget ser man at det å bruke abstraksjon som medierende artefakt gir elevene mulighet til å bygge videre på flere kompetanser. En av disse er problemløsningskompetanse som bygges gjennom at elevene får arbeidet med hvordan man kan gjøre matematiske problemer enklere å angripe. Gjennom abstraksjon styrker også elevene sin modelleringskompetanse ved at de får en grundigere forståelse av stegene som må gjøres på vei mot en ferdig modell av en reell situasjon. Begge disse kompetansene bygges også gjennom at elevene gjennomfører en dekomponering av problemet i oppgaven.

Vi kan i tillegg se at bruken av abstraksjon fører til at elevene kan arbeide med representasjonskompetanse ved at de lærer hvordan man kan representere reelle situasjoner gjennom å simulere situasjonen i Python. De arbeider med å se hvilke faktorer som er viktige for å få en korrekt representasjon av situasjonen og dermed også hvilke svakheter representasjonen kan ha, ved at de ser hva simuleringen ikke tar hensyn til.

Koden elevene endte opp med i oppgave 2 ble seende slik ut:

```

1 import random
2
3 terning = ["1", "2", "3", "4", "5", "6"]
4 kast = random.choice (terning)
5 kast2 = random.choice (terning)
6 kast3 = random.choice (terning)
7 kast4 = random.choice (terning)
8 kast5 = random.choice (terning)
9
10 print(kast+kast2+kast3+kast4+kast5)
11
12

```

Elevenes kode fra oppgave 2 (Vedlegg 8.5.2)

Da elevene skulle evaluere koden de hadde laget ble følgende sagt:

236		(elevene tester koden)							
237	1B	Dette ble helt feil		MOD					Evaluering
238	1A	Ja							Evaluering
239	Intervjuer	Hvorfor mener dere det?							
240	1B	Nei nei, det ble jo ikke det. Den viser alle tallene		MOD	REP				Evaluering
241		(Tester koden)							
242	Intervjuer	Er dere fornøyd med det dere har nå eller vil dere pynte litt på det?							
243	1B	Hadde det vært bedre å ha det under hverandre?							
244	Intervjuer	Enten det eller så kan dere legge inn litt tekst							
245	1B	Jeg er egentlig veldig fornøyd							Evaluering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.5.1)

Det blir tydelig her at elevene intuitivt gjennomfører en evaluering av koden etter at de tester den for første gang. Det kan sees ved at de har en forventning om hvilke resultater de ønsker å få, da dette ved første øyekast ikke var det programmet endte opp med å vise. Selv om de først ikke forstår resultatet viser de god modelleringskompetanse gjennom å aktivt evaluere modellen de har laget og å undersøke hvordan den passer som modell for den gitte situasjonen. De finner videre ut at programmet de laget faktisk viste det de ønsket, og fortsetter evalueringen med å se hvordan koden kunne gitt et resultat som var enklere å forstå.

En annen situasjon som er verdt å trekke frem er utdraget under, hvor elev 1B lurer på om man bare vet hvordan kommandoer i Python ser ut. Det er ikke noe som kan forventes av personer med lite erfaring innen koding. Det at eleven er usikker på dette er et tegn på at elevene ikke har mye erfaring fra å skrive koder på egenhånd, hvor det kan kreves at man må søke på internett eller lignende for å finne gunstige kommandoer til kodene sine. Dette er strategier som er viktige for elevene å ha om de skal konstruere kompliserte koder selv, og som vi som lærere bør gi dem tidlig i møtet deres med programmering.

165	1B	Nei, for jeg tenkte i stad at jeg ikke skjønner hvordan man finner while og randint							
166	Intervjuer	Det er noe man bare må lære seg. Man kan enten finne det på google, eller i et register. Det blir litt samme som å lære et nytt språk							
167	1B	okey, det er sånn ja							

Utdrag fra samtale mellom oppgavene (Vedlegg 8.5.1)

4.2 Gruppe 2

I gruppe 2 hadde elev 2A tidligere tatt valgfaget teknologi og forskningslære, hvor eleven hadde arbeidet programmering. I dette faget tok elevene i bruk blokkprogrammeringsspråket Micro:bit, hvor elevene tar i bruk ferdigskrevne kommandoer. Derfor hadde ikke eleven erfaring med å skrive egne kommandoer, men hadde fått litt forståelse av hva koder består av og hvordan de bygges opp av ulike kommandoer.

4.2.1 Oppgave 1

En situasjon som er verdt å trekke frem for å analysere hvilke kompetanser elevene får vist og arbeidet med når de tar i bruk simulering i Python, er situasjonen under hvor elevene skal drøfte om funnene deres fra simuleringen støtter det de vet om sannsynlighet fra før:

84	Intervjuer	Men av og til er vi litt fra 50-50. Men synes dere allikevel at sannsynlighet er en fornuftig måte å anslå virkelige hendelser på?								
85	2B	Nei det er jo det jo ikke, det er jo ikke sikkert	TAN	MOD	RES				KOM	
86	2A	Det vil jo ikke være helt nøyaktig	TAN	MOD	RES				KOM	
87	Intervjuer	Nei, dere ville ikke veddet penger på at etter 50 kast havnet dere på 25 hver								
88	2A	Nei								
89	2B	Nei								
90	Intervjuer	Men så sier vi jo allikevel i matte at det er 50% sannsynlighet for begge. Så kunne vi gjort noe med koden for å vise at matten ikke er helt gal								
91	2B	Øke hvor mange kast	TAN	MOD						
92	Intervjuer	Har du noen tanker på hvorfor det er lurt?								
93	2B	Nei altså, jo flere kast man har, altså så vil det ha mindre å si om det er en forskjell eller ti forskjell	TAN	MOD	RES			SYM	KOM	

Utdrag fra elevenes samtale rundt oppgave 1D (Vedlegg 8.6.1)

Her kan vi se i linje 85, 86 og 93 at elevene innehar god tankegangskompetanse knyttet til stokastiske hendelser etter å ha gjennomført simuleringen flere ganger. De viser gjennom disse linjene at de har en god forståelse for begrensningene til sannsynlighetsregning. De viser også evner til å forstå i hvilke tilfeller sannsynlighetsregning passer godt med virkelige situasjoner. Disse ytringene underbygger analysen fra gruppe 1 som viste at bruk av simulering som medierende artefakt kunne være gunstig for å bygge opp elevenes tankegangskompetanse rundt det gjeldende temaet.

De samme linjene kan også brukes for å demonstrere at elevene innehar en god modelleringskompetanse. De forstår omfanget og begrensningen av modellen de tar i bruk, og kommuniserer godt om styrker og svakheter ved denne.

Elevene viser i dette utdraget gode evner til å resonnerer rundt funnene de har fått fra simuleringen, samt gode evner til å kommunisere disse funnene på en måte som er forståelig for en person som ikke har sett simuleringen eller ikke kjenner til koden i Python.

Det at elevene innehar de nevnte kompetansene støttes av bolken under hvor elevene bes om å oppsummere hva simuleringene har vist i en setning:

111	Intervjuer	så vi ser at ideen med å legge til flere kast var veldig god. Men kunne dere oppsummert det koden har vist oss til nå med en setning?								
112	2A	Det er veldig tungvint å kaste terningen 100 000 ganger, så dette er et godt verktøy til det			MOD		REP			HJE
113	2B	Og at vi må kaste mange ganger før den havner på 50 [%]	TAN		MOD	RES		SYM	KOM	

Utdrag fra samtale rundt hva elevene lærte i arbeidet med oppgave 1 (Vedlegg 8.6.1)

Her kan vi igjen se at elevene viser gode evner til å resonnerer rundt funnene sine, og å kommunisere dem på en forståelig måte. Tankegangskompetansen som ble vist i det forrige utdraget vises igjen her, nå med flere simuleringen bak utsagnet. Her viser i tillegg elevene god hjelpemiddelkompetanse ved å forstå i hvilke situasjoner simulering er et godt hjelpemiddel, etter å tidligere ha kommentert at det ikke er et nødvendig hjelpemiddel ved få kast.

I linje 112 viser elevene god evne til å bytte mellom ulike representasjoner, og forstår når de ulike representasjonene er nyttige. Representasjonskompetansen til elevene bygges også gjennom hele oppgaven, da elevene lærer å representere fysiske hendelser gjennom å simulere dem i Python.

4.2.2 Oppgave 2

Selv om en av elevene i denne gruppen tidligere har hatt et teknologifag, bærer arbeidet til gruppen preg av at elevene ikke er vant til å ta i bruk hele prosessen rundt algoritmisk tenkning. Dette er noe elevene presiserer selv i utdraget under hvor en av elevene sier at de vanligvis baserer kodene sine på en ferdig mal:

114	Intervjuer	Ja, det er jo veldig fint, men her har vi jo fått litt hjelp, så nå tenkte jeg at vi skulle prøve å lage en kode fra bunnen av. Har dere gjort det mye?																	
115	2B	Nei																	
116	2A	Ikke uten en mal å se på																	Algoritmetankegang mangler HJE ofte

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.6.1)

På tross av dette viser gruppen evner til både dekomponering og abstraksjon når de først blir spurt om det. I dette tilfellet blir intervjuer en tilrettelegger for en proksimal utviklingszone. Det betyr at begge disse prosessene er noe de evner å gjøre, selv om de ikke vanligvis tar dem i bruk i arbeidet mot en ferdig kode. I intervjuutdragene med elevene som vises under ser man at elevene klarer å både abstrahere og dekomponere problemet, men at de først må orienteres i den retningen av intervjueren.

121	Intervjuer	Så da starter vi litt fra scratch. I et virkelig terningkast så er det ganske mye som skjer. Du skal kaste en terning med en viss fart, før den treffer et bord der det kanskje står andre ting også, før du skal se hvilken siden av terningen som peker opp, hvor mye av disse tingene fra kastet er viktig å få med i koden?																	
122	2B	Ingenting																	Abstraksjon
123	Intervjuer	For målet med koden																	
124	2A	Er jo bare utfallet																	Abstraksjon

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.6.1)

Som i analysen av svarene til gruppe 1 ser vi at gjennom å gjennomføre en abstraksjon i denne oppgaven får elevene arbeidet med både problemløsningskompetanse og modelleringskompetanse. Dette skjer gjennom å lære å ta i bruk abstraksjonen som et steg i en problemløsningsprosess som innebærer å lage en modell av en reell situasjon. Vi ser fra utdraget under at også dekomponering fungerer som et godt medierende artefakt for å hjelpe elevene med å bedre begge disse kompetansene.

		Så er ofte neste steg å bryte ned oppgaven. For mange programmeringsoppgaver kan være veldig store, så det kan være lurt å bryte ned problemet til noe som er litt enklere å løse og som vi heller kan bygge videre på. Kan dere gjøre det her?							
127	Intervjuer								
128	2A	Vi kan jo begynne med 1			PRO	MOD			Dekomponering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.6.1)

Når det kommer til konstruksjon av algoritmer og å oversette dem til koder viser elevene på tross av at de selv tenker at de ikke får til å lage egne koder, at de får til å lage en algoritme for å løse problemene de får presentert. De klarer videre å overføre denne algoritmen til en kode i Python. Et eksempel på det er utdraget under hvor elevene skal lage en simulering som skal vise fem terningkast:

130	2A	Da kan vi jo begynne med å definere dem da			PRO	MOD			Algoritme
131	2B	Ja							
132	2A	De ulike tingene							
133	2B	Men der står det jo ikke noe om hvor mange kast [I koden fra 1A]							
134	2A	Men da kan jo bare ta flere				MOD			Algoritme
135	2B	Kan jo bare ta fem kast ja							Algoritme

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.6.1)

Her viser elevene tydelig at de forstår oppbygningen av algoritmen som kreves for å simulere situasjonen de er ute etter å simulere. Det at elevene får arbeidet med å konstruere slike algoritmer viser at denne typen oppgaver er gode hjelpemidler for elevene for å bygge modelleringskompetanse.

Den ferdige koden elevene endte opp med i oppgave 2 ble seende slik ut:

```

1 import random
2 terning=["1","2","3","4","5","6"]
3 kast=random.choice(terning)
4 kast2=random.choice(terning)
5 kast3=random.choice(terning)
6 kast4=random.choice(terning)
7 kast5=random.choice(terning)
8
9 print(kast,"-",kast2,"-",kast3,"-",kast4,"-",kast5)
10

```

Elevenes kode fra oppgave 2 (Vedlegg 8.6.2)

Etter at elevene kjørte gjennom koden første gang ble det tydelig at også elevene i gruppe 2 intuitivt gjennomførte en evaluering av koden, noe som kan sees i utdraget under:

185	2A	ja, det er vel ikke sånn side på noen terninger?							Evaluering
186	Intervjuer	Hvis du kjører den en gang til							
187	2A	åja, åja, sånn ja							
188	Intervjuer	Er dere fornøyd							
189	2B	Jeg er veldig fornøyd				MOD			Evaluering
190	2A	Det går jo an å forstå det liksom, det så jo bare litt knotete ut				MOD			Evaluering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.6.1)

I likhet med gruppe 1 slet elevene i gruppe 2 med å forstå modellen etter første test, da resultatet de fikk opp ikke lignet en vanlig terning. Elevene viser gjennom å være misfornøyd her at de hadde en klar tanke om hva de ville at modellen deres skulle vise, men at det ved første øyekast ikke så ut til å stemme. De viser derimot etter å ha kjørt koden flere ganger at de forstår

modellen de har laget og er fornøyd med hva den viser, noe som tyder på at evaluering er å et godt medierende artefakt for å bygge videre på elevenes modelleringskompetanse.

4.3 Gruppe 3

I gruppe 3 hadde ingen av elevene tidligere erfaring med programmering i andre fag, og hadde i likhet med elevene fra gruppe 1 dermed opparbeidet seg kompetansen de innehar gjennom arbeidet de har gjort i mattetimene. Det kan føre til at denne gruppens resultater kan være et godt grunnlag for å analysere hvordan programmering fungerer som hjelpemiddel om man bare har mattetimene til rådighet for å innføre programmering i skolen.

4.3.1 Oppgave 1

I oppgave 1 er også for denne gruppen de mest interessante situasjonene å hente frem for å undersøke hvilke kompetanser elevene viser i sitt arbeid, utdragene hvor elevene bes om å formulere hvordan funnene deres stemmer opp mot det de vet fra før i matematikken.

78	Intervjuer	Da kan dere jo gå videre til neste oppgave. Kunne dere svart med en setning der?								
79		...								
80	3A	Hvis man tar det mange ganger får man jo en anelse på hva det skal være, men det er jo helt tilfeldig i kastene	TAN		MOD	RES		SYM	KOM	

Utdrag fra elevenes diskusjon rundt oppgave 1D (Vedlegg 8.7.1)

I dette utdraget viser elev 3A tydelig etter å ha gjennomført simuleringen flere ganger at han/hun har fått dannet seg et bilde av omfang og begrensninger ved stokastiske hendelser, som peker mot at simulering er et godt medierende artefakt for å bygge og arbeide med tankegangskompetanse. Videre viser eleven at han/hun også evner å kommunisere dette på en forståelig og korrekt måte, som kan knyttes til både kommunikasjonskompetanse og formalismekompetanse. Det er derimot tydelig at det å skulle formulere noe på en matematisk korrekt måte er noe eleven her har en del å gå på, da elevens uttalelse ikke kan kalles formelt matematisk språk. Det er derimot tydelig at eleven forstår hva som må med i et formelt matematisk utsagn, og at det å måtte formulere slike setninger etter å ha gjennomført simuleringer er en god arena for elevene å øve seg på denne typen matematiske formuleringer.

Eleven viser også at han/hun har en god forståelse av resultatene fra sin modell, og å videre kunne analysere disse resultatene for å komme med styrker og svakheter i modellen. Dette peker mot at simulering er et godt medierende artefakt for å lære elevene disse aspektene av modelleringskompetanse.

Kompetansene elev 3A viste i oppgave 1D får elev 3B også vist i arbeidet med oppgave 1E hvor følgende ble sagt:

85	3B	Hvis vi tar mange flere kast								
86	Intervjuer	Har du noen tanker på hvorfor det er sånn?								
87	3B	Da er det jo større sjans for at man ikke bare ikke hadde flaks, at det blir mer riktig på en måte	TAN			RES		SYM	KOM	

Elevenes tanker rundt oppgave 1E (Vedlegg 8.7.1)

Her kommer en enda grundigere forklaring på hvorfor konklusjonen de har dratt om sannsynlighetsregning faktisk er holdbar. Dette tyder på at elevenes bruk av simulering har gitt en større dybdeforståelse av sannsynlighetskonseptet, noe som stryker simulerings plass som et godt medierende artefakt for å lære dette temaet og bygge grundig tankegangskompetanse.

Elevene får vist sine evner til å resonnerer ved å ta i bruk simuleringens resultater som basis for sine utsagn. Dette er med på å bygge elevenes resonneringskompetanse.

4.3.2 Oppgave 2

Som nevnt i de tidligere gruppene har heller ikke den gruppen grundig kjennskap til stegene i algoritmisk tenkning, og setter derfor ikke i gang med abstraksjon og dekomponering uten først å bli orientert i den retningen av intervjuer, som må være tilstede for å skape en proksimal utviklingszone. Når de derimot blir spurt om abstraksjonsprosessen og dekomponeringsprosessen av intervjuer viser de at de har kompetanse nok til å gjennomføre begge prosessene på en fornuftig måte. Det kan sees i utdragene under:

129	Intervjuer	Når man kaster en terning skal man for eksempel kaste den med en viss fart fra en viss høyde før den treffer et bord og ruller bortover og viser et tall. Hvor mye av denne situasjonen er det viktig å få med i simuleringen?							
130	3B	Tallet som kommer opp	PRO	MOD	REP				Abstraksjon
131	3A	mhm							
132	Intervjuer	Ja, er det noe annet vi vil ha med?							
133	3B	Nei	PRO	MOD					Abstraksjon
134	3A	Nok terninger	PRO	MOD	REP				Abstraksjon
<i>Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.7.1)</i>									
137	Intervjuer	et fint neste steg kan være å bryte ned problemet litt til noe vi kan bygge videre på siden mange problemer kan være vanskelig å hoppe rett inn i							
138	3B	Ja							
139	Intervjuer	Har dere en tanke på hvordan dere kunne gjort det her?							
140	3A	Kan jo begynne med å bare ta en terning	PRO	MOD					Dekomponering

Vi kan se fra begge disse utdragene at abstraksjon og dekomponering er noe deltakerne i gruppen evner å få til, men at de ikke har tilstrekkelig forståelse av prosessen knyttet til algoritmisk tenkning til at de gjennomfører disse prosessene automatisk i arbeidet med programmeringen.

Elevene i denne gruppen viser gjennom hele arbeidet med oppgavesettet gode ferdigheter når det kommer til å ta i bruk koding og å tenke frem algoritmer som passer til situasjonen de ønsker å representere i programmet sitt. I utdraget under ser man tegn til at å ta i bruk det å tenke i algoritmer som medierende artefakt kan bidra til å bygge elevenes problemløsningskompetanse gjennom at de lærer å lete seg frem til algoritmer og løsningsmetoder som kan være passende til å løse problemet de står ovenfor. Man ser også at algoritmer er med på å bygge elevenes modelleringskompetanse gjennom at de lærer å finne fornuftige fremgangsmåter til å bygge modellene sine.

171	Intervjuer	Så bra, da har vi sett at vi klarer det. Så da kan vi bygge videre til det originale problemet							
172	3A	Da må vi vel ha fem terninger da?	PRO	MOD					Algoritme
173	3B	Skal vi bare skrive den om og om igjen da?	PRO					HJE	Algoritme/koding
174	3A	Ja							
175		(Legger til flere kast i koden)						HJE	Koding

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.7.1)

Koden elevene endte opp med i oppgave 2 ble seende slik ut:

```

1 import random
2
3 terning = ["1", "2", "3", "4", "5", "6"]
4
5 kast1 = random.choice(terning)
6 kast2 = random.choice(terning)
7 kast3 = random.choice(terning)
8 kast4 = random.choice(terning)
9 kast5 = random.choice(terning)
10
11
12
13 print ("Tallet ble", kast1, kast2, kast3, kast4, kast5)
14

```

Elevenes kode fra oppgave 2 (Vedlegg 8.7.2)

Etter å ha skrevet koden ferdig gjennomførte elevene en test av programmet, og utdraget under er elevenes første reaksjon på simuleringen:

180	3B	Skal vi se hva som skjer						
181		..						
182	3B	4 1 2 3 5						
183		..						
184	3B	Det funket jo veldig fint		MOD				Evaluering
185	3A	mhm						
186	Intervjuer	Gjør koden det dere vil den skal gjøre?						
187	3B	Ja, fem kast		MOD	REP			Evaluering
188	3A	ja		MOD	REP			Evaluering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.7.1)

Vi ser også i denne gruppa at elevene har en tanke om hvilke resultater de ønsker at programmet skal gi. Det at denne evaluering skjer intuitivt hos elevene viser at denne måten å arbeide på er et godt medierende artefakt for at elevene skal arbeide mot helhetlig modelleringskompetanse, hvor også evalueringen av egne og andres modeller er et sentralt element. Elevene får her også arbeidet med sin representasjonskompetanse da de må bedømme hvorvidt simuleringen de har laget faktisk er en god representasjon av hendelsen de er ute etter å representere.

Noe som skiller denne gruppen fra de andre er at elevene i denne gruppen ser ut til å ha en mer positiv holdning til bruken av programmering i mattetimene. I utdraget under trekker elevene frem positive sider ved å ta i bruk Python, og sier at det er et hjelpemiddel de gjerne ønsker å ta i bruk så lenge oppgavene ikke er så vanskelige at de ikke klarer å tenke seg frem til mulige løsninger. Dette tyder på at det å gi elevene en forståelse av programmerings nytte utenfor mattetimene kan gi elevene en god holdning til programmering når det skal implementeres i ulike matematiske temaer.

199	Intervjuer	ja, det er ofte det man må gjøre egentlig. Er dette noe dere kunne tenke dere å ha mer av i mattetimene?						
200	3B	Så lenge det ikke er for vanskelig. Noe av det kan man jo tenke seg til						HJE
201	3A	Så er det jo sikkert noe som blir veldig sentralt i fremtiden						HJE
202	Intervjuer	Ja, det kommer jo godt med i mange yrker						
203	3A	mhm. Så å få litt grunnlag skader jo ikke det						HJE

Utdrag fra samtale med gruppe 3 etter at alle oppgavene var ferdige (Vedlegg 8.7.1)

4.4 Gruppe 4

I gruppe 4 hadde begge elevene tidligere hatt valgfaget teknologi og forskningslære, og hadde som den ene deltakeren i gruppe 2 kjennskap til blokkprogrammering fra dette faget. Det at begge hadde denne forkunnskapen kan føre til at de stiller sterkere enn gruppene som ikke hadde slik kompetanse på forhånd. I denne gruppen hadde deltaker 4A også tatt i bruk programmering i Python i fysikk, selv om de i liten grad hadde konstruert egne koder.

4.4.1 Oppgave 1

Som i de andre gruppene får også gruppe 4 resultater fra simuleringen som bygger opp under forståelsen de allerede har om sannsynlighetsregning, gjennom å visuelt se simuleringen av hendelsen og ikke bare gjennom det læreren forteller muntlig eller på tavla. Dette peker mot at bruken av simulering i dette tilfellet både var et godt medierende artefakt for å bygge videre på elevenes tankegangskompetanse, men også på deres kommunikasjonskompetanse gjennom at de selv må formulere funnene sine og knytte dem opp mot teori. Et eksempel på det er utdraget under hvor elevene presenterer sine funn knyttet til antall kast i myntkastoppgaven.

125	Intervjuer	Ja, nå er vi jo veldig mye nærmere. Har dere noe tanker om hvorfor det er sånn?								
126	4A	Det er jo sånn at det blir mer riktig beregning. Det kan jo bare være flaks de få gangene	TAN		MOD	RES			KOM	
127	4B	Ja								
128	Intervjuer	Ja, de små forskjellene har ikke så mye å si lengre. Så hvis vi skulle oppsummert hva vi har funnet ut i denne oppgaven, hva ville dere sagt da?								
129	4A	Du må ha mange forsøk om det skal bli mest mulig nøyaktig [50%]	TAN		MOD	RES			KOM	

Utdrag fra diskusjon etter at eleven har fullført oppgave 1 (Vedlegg 8.8.1)

De samme kompetansene viser også elevene at de får brukt når de blir spurt om hvordan funnene de gjør i de første myntkastene stemmer overens med det de vet om sannsynlighet fra før, i utdraget under.

105	Intervjuer	Men stemmer noe bedre med det vi vet fra matematikken?								
106	4B	Nei, det er jo litt tilfeldig fra hver gang vi prøver	TAN		MOD	RES			KOM	
107	Intervjuer	Det stemmer jo kanskje ikke helt med det vi tenker. Så da kan dere gå videre til neste side								
108		...								
109	4A	Nei, det kommer an på hvor stor								
110	4B	Det kommer an på antall	TAN			RES				
111	4B	Hvis man har veldig lave tall vil det ikke alltid stemme	TAN			RES			KOM	

Utdrag fra diskusjon rundt oppgave 1D og 1E (Vedlegg 8.8.1)

Elevene må i begge utdragene over fremme egne argumenter for utsagnene de kommer med, da det er elevene selv som har kommet frem til funnene gjennom simuleringene sine. Denne prosessen er en god måte å få elevene til å arbeide med resonnementskompetanse, da elevene selv skal hente matematiske argumenter og trekke konklusjoner ut fra eget arbeid uten å få teorien eksplisitt presentert fra andre.

Det å påstå at oppgaven har ført til en økt forståelse for sannsynlighet støttes av utdraget under fra tidlig i oppgavesettet, hvor elevene ikke er like sikre i påstanden om at det er lik sannsynlighet for å få alle utfall. Dette utsagnet sett i lys av de senere utsagnene som ble trukket frem over viser at elevene har fått mer selvillit i måten de kommuniserer tankene sine på, noe som påker mot en bedret forståelse av tema.

52	Intervjuer	Ja, det ser jo fint ut. Men her ser vi jo at ikke alltid får en av hver. Hvordan stemmer det med det dere vet fra matematikken?																			
53	4A	Det er jo egentlig like sannsynlig																			KOM

Utdrag fra diskusjon etter oppgave 1B (Vedlegg 8.8.1)

4.4.2 Oppgave 2

Som i de tidligere gruppene viser også elevene i gruppe 4 at de mestrer abstraksjon og dekomponering i oppgave 2 når de først blir spurt om det, men det virket også her som om disse prosessene var nye, på tross av at dette var gruppen med mest erfaring med programmering. Det virket som om abstraksjon spesielt var en uvant måte å tenke på, da elevene først svarte usikkert og med korte setninger, som sett i utdraget under. Det virket derimot som om dekomponeringen gikk veldig raskt.

132	Intervjuer	Så det som er en god plass å begynne når man skal lage en kode er å se på den virkelige situasjon vi vil simulere. I et terningkast skal man jo kaste terningen med en viss fart og fra en viss høyde før terningen skal treffe et bord, trille bortover og ende opp med en av sidene opp. Hvor mye av dette føler dere at det er nødvendig å ha med i koden?																			
133	4A	Veldig lite			PRO	MOD			REP												Abstraksjon
134	4B	Bare tallet			PRO	MOD			REP												Abstraksjon
135	Intervjuer	For poenget med koden er jo?																			
136	4B	Å få opp noen tall						MOD					REP								Abstraksjon

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.8.1)

139	Intervjuer	Noe annet som er greit å gjøre er å se om vi kan bryte ned problemet litt. For det kan jo være vanskelig å hoppe rett i det store problemet. Så spørsmålet her er jo om vi kan bryte problemet ned til noe vi heller kan bygge videre på.																			
140	4A	Kan jo lage et program som vi heller kan kjøre flere ganger			PRO	MOD			REP												Dekomponering
141	4B	Ja, der vi kaster en terning			PRO	MOD			REP												Dekomponering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.8.1)

Som drøftet i de andre gruppene viser denne gruppen gjennom abstraksjon og dekomponering at de innehar og får arbeidet med både modelleringskompetanse gjennom å skape egne modeller, og representasjonskompetanse gjennom å se om modellen faktisk representerer det den gir seg ut for å representere. Vi ser også at algoritmisk tenkning i denne oppgaven er et godt medierende artefakt for å bedre elevenes problemløsningskompetanse, da de lærer gode strategier til hvordan man kan angripe større og mer komplekse modelleringsoppgaver.

Elevene i denne gruppen viser at de innehar gode kodeferdigheter, noe som ikke er uventet i og med at begge deltakerne har erfaring med programmering fra tidligere fag. Dette fører til at elevene i terningkastoppgaven har flere mulige løsningsmetoder de kan ta i bruk, som kan sees i utdraget under. De velger til slutt å definere flere kast, noe som virker fornuftig siden de der presenterer hele algoritmen som trengs, noe som peker mot at de også innehar en del kunnskap om å tenke i algoritmer, i hvert fall tilstrekkelig kunnskap for den gitte oppgaven. Gjennom det å konstruere algoritmer for å lage simuleringer ser vi også at elevene får arbeidet med modelleringskompetansen da det å bygge disse algoritmene er et steg på veien mot å lage egne modeller av reelle situasjoner.

145	4B	Vi må definere dem								Algoritme
146	4A	Vi kan jo skrive terning også 1, 2, 3, 4		MOD					HJE	Algoritme
147		(Definerer terningen i koden)								Koding
148	4A	Så må vi gå tilbake igjen og se her								
149	4B	ehhm, så må vi bare ha kast da								
150	4A	Så må vi bare ha kast1, kast2, kast3		MOD					HJE	Koding/algoritme
151	4B	Så må vi ha en while-løkke							HJE	Koding
152	4A	Men vi kan bare skrive kast1, kast2, kast3 også							HJE	Koding
153	4B	nedover her?								
154	4A	Ja								

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.8.1)

Koden elevene i gruppe 4 endte opp med i oppgave 2 ble seende slik ut:

```

1 import random
2
3 terning = ["1","2","3","4","5","6"]
4 kast1 = random.choice(terning)
5 kast2 = random.choice(terning)
6 kast3 = random.choice(terning)
7 kast4 = random.choice(terning)
8 kast5 = random.choice(terning)
9 print (kast1,kast2,kast3,kast4,kast5)
10

```

Koden elevene i gruppe 4 endte opp med i oppgave 2 (Vedlegg 8.8.2)

Etter å ha fullført koden og rettet opp i en liten skrivefeil i en av kommandoene ble følge evaluering av koden gjennomført:

188	Intervjuer	bra, er dere fornøyd med det koden viser?								
189	4A	Ja, og det enkleste er jo bare å ta et ark og skrive ned dem man vil beholde		MOD	REP				HJE	Evaluering

Utdrag fra elevenes arbeid med oppgave 2 (Vedlegg 8.8.1)

Elevene viser her som i de andre gruppene at å bruke evaluering som medierende artefakt gir elevene rom til å bygge videre på sin modelleringskompetanse ved å måtte evaluere egne

modeller og se på deres egnethet. Vi ser at elevene her viser god representasjonskompetanse ved å presisere at koden simulerer terningkastene som den skal, men at det ikke er mulig å ta var på terninger, noe som er vanlig i yatzy. Her kommer også elev 4A med en løsning på dette problemet ved å ta i bruk et ark, som viser at han har gode oversikt over hjelpemidler som kan tas i bruk for å løse problemet han står ovenfor.

Elevene i gruppe 4 viser gode evner til å reflektere rundt eget bruk av programmering som hjelpemiddel, og viser at de reflekterer på en kritisk måte. Dette demonstreres i utdragene under:

207	4B	Ja, for nå skriver jeg bare mye av, og så satser jeg på det beste																				HJE	
208	4A	Da lærer man ikke så mye av det																					HJE
209	4B	Så tar det så lang tid																					
<i>Utdrag fra samtale etter at elevene hadde gjennomført oppgavesettet (Vedlegg 8.8.1)</i>																							
215	4A	Ja, vi har fått veldig lite forklaring og hva det er godt for og sånn																					HJE
216	4B	Ja, da synes jeg det er mye bedre med blokkprogrammering, det er mye lettere å forstå. Da skjønner jeg faktisk litt hva jeg gjør																					HJE
<i>Utdrag fra samtale etter at elevene hadde gjennomført oppgavesettet (Vedlegg 8.8.1)</i>																							

Elevene viser her at de har gode evner til å reflektere rundt egen bruk av hjelpemidler, og at programmering i mange tilfeller ikke er et hjelpemiddel de aktivt har et forhold til, men noe de bare kopierer fra en mal. De presiserer også at de ofte ikke forstår hva bruken er godt for, og at de ikke føler at de lærer noe særlig av å ta det i bruk.

Kommentaren til 4B om at blokkprogrammering kjennes mer fornuftig er også verdt å notere seg, da det er et verktøy eleven selv ser på som mer fornuftig å bruke. Dette begrunnes med at eleven selv forstår mer av kodingen og kan arbeide mer selvstendig enn å bare klippe og lime fra en allerede gitt mal. Å ta i bruk blokkprogrammering nevnes også av Humble et al. (2020) som en mulig måte å gjøre innføringen av programmering i skoleverket enklere.

Disse kommentarene er spesielt interessante for oss som lærere da de sier mye om hvordan elevene selv responderer på hjelpemidlene vi tilbyr dem, og hvilken holdning denne måten å implementere programmering i undervisningen kan føre til. Selv om denne oppgaven handler om hvilke kompetanser man kan se ved bruk av programmering og algoritmisk tenkning synes jeg disse kommentarene var verdt å ta med, da de er viktige innspill for lærere.

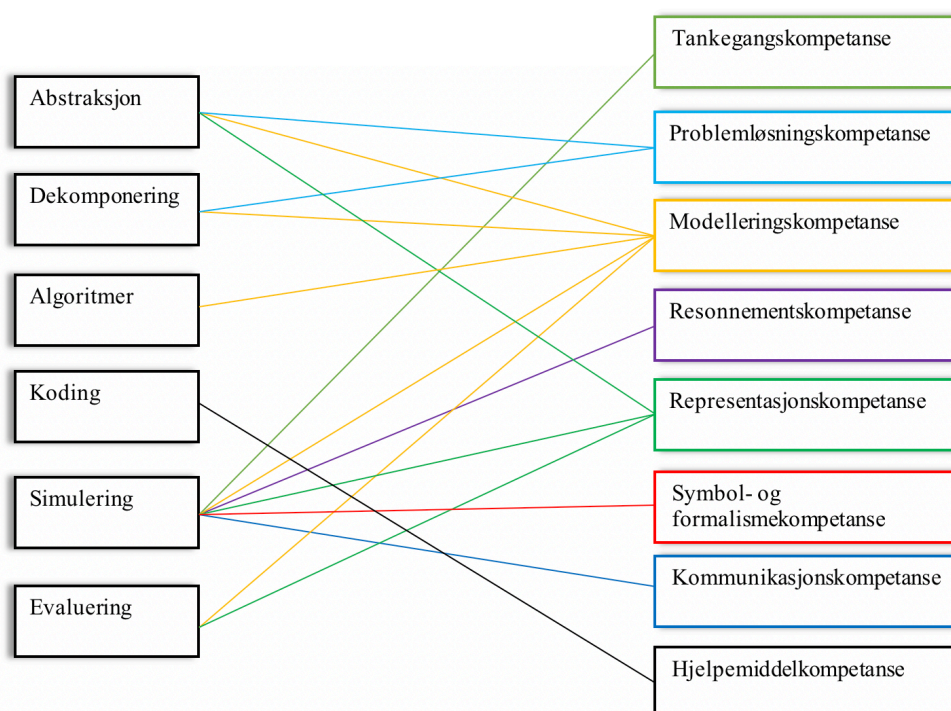
4.5 Oppsummering av resultater og analyse

Elevene i samtlige av gruppene viste tegn til at de mestret de ulike stegene innenfor algoritmisk tenkning, men at de i variert grad var vant til å gjennomføre stegene i prosessen i måten de arbeidet i mattetimene. Dette var spesielt tydelig i denne elevgruppen ved abstraksjon og dekomponering, som var prosesser de ikke hadde gjennomført mye av tidligere. De viste allikevel at de innehadde tilstrekkelig med kunnskap til å gjennomføre prosessene da de først ble orientert i den retningen. Det samme gjaldt det å lage koder fra bunnen av. Elevene uttrykket at de var vant til å basere seg på allerede eksisterende maler, men viste at de hadde god forståelse for enkel koding ved at de i stor grad forstod hva kodene gjorde, og at de klarte å endre allerede kjente koder til å passe en litt endret situasjon, som å overføre ideer fra myntkast til terningkast.

Resultatene fra denne forskningen peker mot at å bruke simulering som medierende artefakt kan være gunstig for å arbeide med flere kompetanser hos elevene. I denne studien er de mest tydelige kompetansene som kommer frem i elevenes arbeid tankegangskompetanse, modelleringskompetanse, resonnementskompetanse, representasjonskompetanse, symbol- og formalismekompetanse og kommunikasjonskompetanse. Simuleringen kan da være enten med en ferdig kode som elevene bruker for å gjennomføre simuleringen, eller at elevene følger en ferdig mal og bare trenger å kopiere allerede skrevne koder for å lage programmet bak simuleringen.

Denne studien viser også at det å ta i bruk alle stegene knyttet til algoritmisk tenkning for å lage et program som simulerer en reell hendelse er et godt medierende artefakt for å gi elevene mer grundig kompetanse enn ved bare å ta i bruk simulering alene. Resultatene viser blant annet at elevene får en mer helhetlig modelleringskompetanse når de aktivt gjennomfører abstraksjon, dekomponering, å tenke i algoritmer og evaluering gjennom at de får arbeidet med ferdigheter knyttet til flere aspekter ved modelleringsprosessen. Elevene viste også tegn til å arbeide med problemløsningskompetanse gjennom å tenke aktivt gjennom valg knyttet til dekomponering og abstraksjon for å gjøre oppgavene lettere å arbeide med. Algoritmisk tenkning virker også til å være et godt medierende artefakt når målet er å gi elevene en god representasjonskompetanse gjennom at de aktivt må velge hva som er viktig å ta med i en representasjon av en hendelse, samt å evaluere hvorvidt en måte å representere en hendelse på er passende.

Resultatene av studien er demonstrert under i figur 4.1, hvor linjene som er trukket fra de ulike stegene i algoritmisk tenkning viser hvilke kompetanser man ser at elevene tar i bruk når de arbeider med de ulike trinnene.



Figur 4.1: Illustrasjon av sammenhengen mellom algoritmisk tenkning og Niss matematiske kompetanser

Studien viste i tillegg at elevene gjennom å arbeide med disse oppgavene fikk arbeidet med hjelpemiddelkompetanse. Alle gruppene viste jevnt over gode ferdigheter når det kom til å reflektere rundt egnetheten av Python som hjelpemiddel, samt å kunne være kritisk til hvordan de selv tar i bruk hjelpemiddelet. Det er knyttet til denne kompetansen de største forskjellene mellom gruppene kommer til syne, da elevenes positivitet til å ta bruk Python varierte i stor

grad. Elevenes arbeid som kan knyttes til hjelpemiddelkompetanse kommer ikke klart nok frem i figuren over, da det ofte ikke var knyttet til et spesifikt steg i algoritmisk tenkning, men var et resultat av samtaler rundt oppgavene de har gjennomført og hvordan de gjennomførte dem. Dette peker mot at arbeidsmåten som en helhet er et godt medierende artefakt for å gi elevene god hjelpemiddelkompetanse.

5 Diskusjon

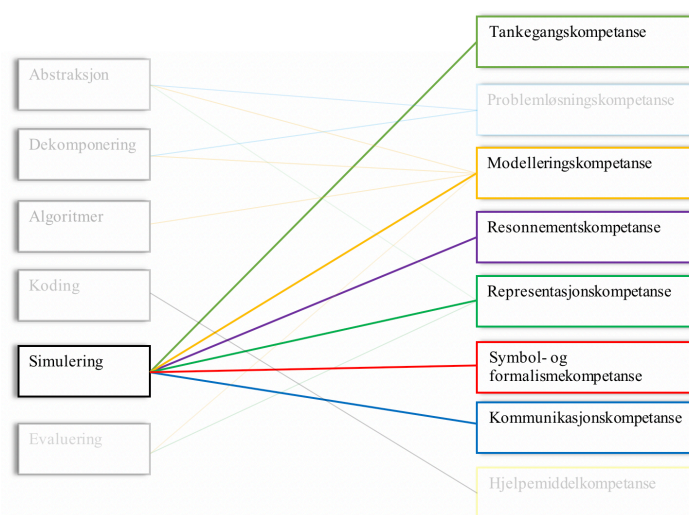
I dette kapittelet kommer jeg til å drøfte funnene presentert i forrige kapittel opp mot forskningsspørsmålene jeg har satt ut for å undersøke, samt se på dem i lys av den tidligere forskningen som ble presentert i avsnitt 2.5. Forskningsspørsmålene som det skal drøftes rundt lyder slik:

1. Hvilke av Niss matematiske kompetanser arbeider elevene med når de tar i bruk programmering og algoritmisk tenkning som medierende artefakter i matematiske oppgaver knyttet til sannsynlighetsregning?
2. I hvor stor grad gjenspeiles stegene i algoritmisk-tenkningsprosess i elevenes arbeid med oppgaver som tar i bruk programmering i Python?

5.1 Simulering med ferdigskrevne programmer som medierende artefakt for å bygge tankegangskompetanse

I arbeidet med oppgave 1 tok elevene i bruk simulering uten å først måtte gjennomføre de andre stegene i algoritmisk tenkning. Oppgavens konstruksjon la heller ikke opp til at dette skulle bli nødvendig, men var ment for å se hvilke kompetanser elevene fikk vist og arbeidet med når de ble presentert for ferdige, eller tilnærmet lik ferdige, programmer som kunne tas i bruk som simuleringer uten videre arbeid med dem.

Som presentert i resultatkapittelet kan man se tegn til at simulering alene er et godt medierende artefakt som kan hjelpe elevene med å arbeide med flere kompetanser, som vist i figur 5.1. I dette underavsnittet vil fokus ligge på funnene som knytter simulering opp mot arbeid med tankegangskompetanse. De andre kompetansene som knyttes til simulering vil tas opp videre i kapitlet.



Figur 5.1: Sammenhengen mellom simulering og matematiske kompetanser

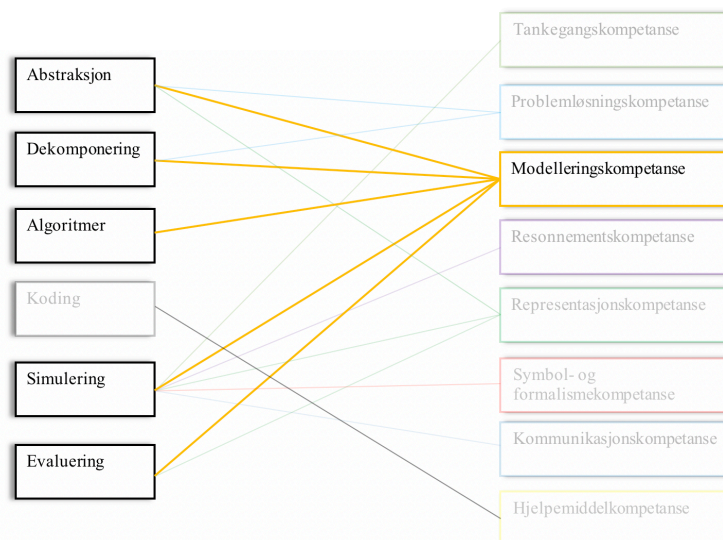
Det kom frem i alle gruppene at å se simuleringen av ulike antall myntkast ga elevene en god forståelse av sannsynlighetsbegrepet, og lot dem se både begrensninger og muligheter bruken av regning med stokastiske variabler gir. Dette viser at simulering kan være godt egnet for å introdusere og arbeide videre med dette temaet, da det gjør det enklere for elever å få et visuelt bilde av situasjonene de introduseres for. Dette kan spesielt være gunstig tidlig i undervisningsforløpet så elevene får en visuell forståelse av temaet i tillegg til det teoretiske de har fått presentert av læreren på tavla.

Det å påstå at tankegangskompetansen til elevene bygges underveis i arbeidet støttes av at flere av elevene som beskrevet i resultatkapittelet svarte på spørsmål knyttet til teori med mer og mer selvtillit etter flere simuleringer. Funnene viste også at elevene i dette arbeidet fremmet egne argumenter rundt matematisk teori, noe som medfører at de selv må ta aktivt stillingen til teorien og skape en selvstendig forståelse av den.

Dette er en kompetanse tidligere forskning jeg har lest ikke vektlegger, selv om det fra denne studien virker som å være en av kompetansene det tydeligst arbeides med når elevene tar i bruk simulering. En mulig forklaring på dette er at andre forskere ikke har tatt i bruk Niss kompetanser som teoretisk rammeverk, men jeg har heller ikke lest forskning som vektlegger denne kompetansen med andre ord eller navn heller. Det er for meg derfor ekstra viktig å poengtere elevenes arbeid med denne kompetansen, da det er en kompetanse jeg ser på som sentral når det kommer til elevers forståelse av matematikk og matematiske fenomener.

5.2 Algoritmisk tenkning som medierende artefakt for å gi elevene en helhetlig modelleringskompetanse

Etter å ha analysert resultatene av oppgavesettet og intervjuene med elevene kom det tydelig frem at elevene ved å lage koder fra bunnen av fikk arbeidet med modelleringskompetanse ved nesten alle stegene av algoritmisk tenkning, som demonstrert i figur 5.2.



Figur 5.2: Sammenhengen mellom algoritmisk tenkning og modelleringskompetanse

Som nevnt i drøftingen i avsnitt 5.1 viste funnene at elevenes bruk av simulering uten å lage kodene selv var et godt medierende artefakt for å gi elevene en god modelleringskompetanse i den forstand at de fikk arbeidet med evnen deres til å forstå og evaluere ferdige modeller, samt å kommunisere om styrker og svakheter ved de gitte modellene.

Niss (2002, s. 7) nevner derimot at det finnes flere viktige aspekter innenfor modelleringskompetanse. Blant disse nevnes det å selv aktivt lage modeller gjennom å løse eventuelle problem som oppstår i konstruksjonen, validere, evaluere og å ta del i hele modelleringsprosessen. Dette er aspekter ved modellering som ikke kom frem i elevenes arbeid med ferdige koder i oppgave 1 i dette prosjektet. Elevene fikk derimot arbeidet med disse aspektene i oppgave 2, gjennom ulike steg i algoritmisk tenkning.

I både abstraksjon og dekomponering fikk elevene arbeidet med å løse problemer som oppstod i konstruksjonen av oppgavene, samtidig som de fikk tatt aktivt del i disse delene av modelleringsprosessen. Mer om dette kommer i avsnitt 5.3 som vil gå mer i dybden på abstraksjon og dekomponering.

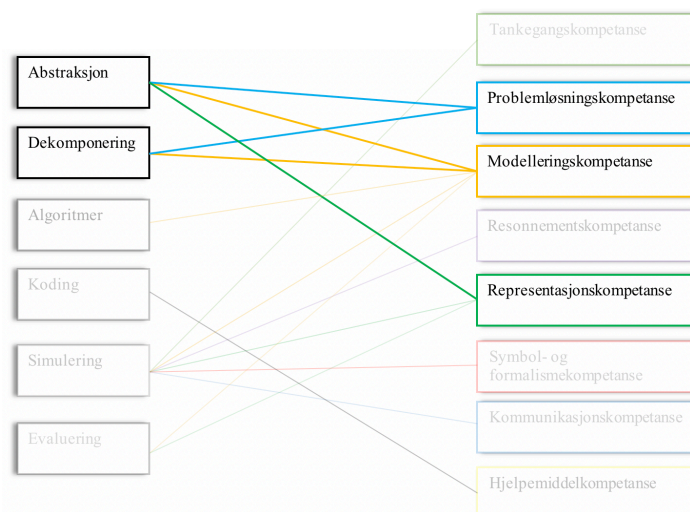
Det å konstruere algoritmer ga elevene mulighet til å bygge sin modelleringskompetanse gjennom å aktivt ta del i hele modelleringsprosessen, og selv være dem som tok valg knyttet til hvordan modellene skulle fungere. Det som derimot kan sees i denne delen av modelleringsprosessen er at elevene veldig sjeldent lager hele algoritmer før de går i gang med kodingen, som også ble funnet i studien til Hansen og Hadjerrouit (2021). Dette viste seg å ikke være problematisk i enkle oppgaver som dem elevene ble presentert her, men det å ha et bevisst forhold til å lage algoritmer som passer til modellen man skal lage er gunstig for elevenes modelleringskompetanse. Dette bevisste forholdet kan de til en viss grad miste om de aldri setter opp komplette algoritmer i løpet av arbeidet med programmet.

Elevene fikk en viss mulighet til å evaluere koder i arbeidet med ferdige koder, noe som var med på å styrke deres modelleringskompetanse. Det som derimot virker som en positiv side av å først konstruere koden selv før man evaluerer, er at elevene da virket til å ha en bedre forståelse av hva de ønsket at koden faktisk skulle gjøre. Dette kan sees gjennom at elevene i flere av gruppene i oppgave 2 først ikke så ut til å like programmene de hadde laget, da de ved første øyekast ikke så ut til å få opp resultatene som terninger. Dette peker mot at evalueringen av egenkonstruerte koder ga elevene rom til å gjennomføre grundigere evalueringer av modellene sine, noe som igjen er med på å styrke deres modelleringskompetanse.

Disse funnene stemmer godt overens med funnene til Lie et al. (2017) som pekte mot at elevene fikk en bedre modelleringskompetanse gjennom å arbeide med åpne programmeringsoppgaver. For å knytte disse funnene til mine funn bør det legges til at om modelleringskompetansen skal være mest mulig komplett er det fornuftig å lage oppgaver som tar i bruk alle stegene i algoritmisk tenkning, eller lære elevene opp til å gjennomføre disse stegene intuitivt.

5.3 Abstraksjon og dekomponering som medierende artefakter for å arbeide med matematisk kompetanse

I samtlige grupper viste elevene at det å skulle gjennomføre abstraksjon og dekomponering i starten av oppgave 2 ikke var noe de gjorde intuitivt uten å bli spurt av intervjuer. Det som derimot ble tydelig var at de innehadde kompetanser nok til å gjennomføre prosessene da de ble spurt. Dette kan bety at om man hadde introdusert algoritmisk tenkning for elevene tidligere, kunne det vært steg de kunne gjennomført selv, uten å bruke tid på å lære selve prosessene.



Figur 5.3: Sammenhengen mellom abstraksjon og dekomponering og matematiske kompetanser

Av sammenhenger mellom abstraksjon, dekomponering og matematiske kompetanser ble det tydelig at begge de to prosessene fremmer elevenes modelleringskompetanse, som i avsnitt 5.2 ble presisert at også involverer at elevene aktivt tar del i modelleringsprosessen. Dette er

aspekter ved modelleringskompetanse som ikke kommer frem i elevens arbeid med oppgave 1, noe som peker mot at elevene ikke gis mulighet til å arbeide med disse delene av kompetansen om de bare får presentert ferdige koder uten å måtte lage dem selv.

Bruken av abstraksjon og dekomponering som medierende artefakter virket også å styrke elevenes problemløsningskompetanse da prosessene ga elevene nye strategier til å løse store og komplekse problemer, som de kanskje ikke hadde klart å løse uten først å gjennomføre en abstraksjon og dekomponering.

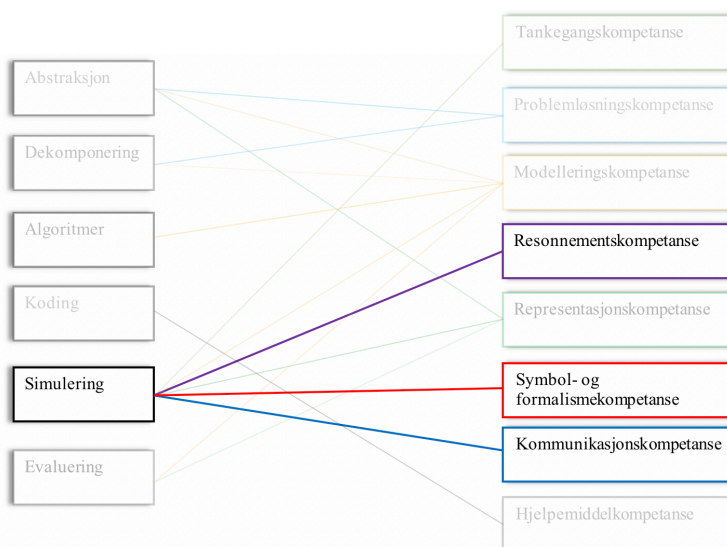
Resultatene tydet i tillegg på at abstraksjon krever at elevene aktivt må ta valg om hvilke av aspektene med situasjonen de ønsker å simulere som faktisk er utslagsgivende for situasjonen de ønsker å få representert. Dette peker mot at bruken av abstraksjon gir elevene mulighet til å arbeide med representasjonskompetanse på et dypere og mer aktivt plan enn det de fikk i oppgave 1.

Disse funnene sett sammen viser at å få elevene til å aktivt gjennomføre abstraksjon og dekomponering i oppgaver som handler om å konstruere koder er et godt medierende artefakt, som gir elevene muligheter til å arbeide med kompetanser de ellers ikke hadde fått arbeidet med om de fikk en ferdig kode eller bare fulgte en ferdigskrevet mal. I mitt oppgavesett er dette veldig tydelig i forskjellen på kompetansene elevene får vist og arbeidet med i oppgave 1 sett opp mot det de får vist og arbeidet med i oppgave 2.

Disse funnene stemmer godt med flere av artiklene jeg trakk fram fra tidligere forskning som la vekt på at programmering var et godt medierende artefakt for å styrke elevenes problemløsningskompetanse og modelleringskompetanse (Costa et al., 2017; Forsström & Kaufmann, 2018; Lie et al., 2017).

5.4 Python som medierende artefakt for å bedre matematisk kommunikasjon og resonnering

Fra analysen av arbeidet til elevene kom det frem at det å ta i bruk simuleringer krever at elevene selv kommuniserer godt rundt det de finner i simuleringene, samt også å resonnerer seg frem til hva funnene deres betyr for deres forståelse av matematikk.



Figur 5.4: Sammenhengen mellom simulering og flere av de matematiske kompetansene

En av grunnene til at disse kompetansene var nødvendige i arbeidet med oppgavesettet jeg konstruerte var at de gjennom å selv gjennomføre simuleringene ikke får betydningen av resultatene forklart av noen andre, noe som medfører at de selv må finne ut av dette. Det at de

selv måtte presentere hva funnene deres innebar gjorde at de ble tvunget til å finne en måte å kommunisere disse funnene på som var forståelig for alle involverte parter, noe som styrket både deres kommunikasjonskompetanse og symbol- og formalismekompetanse. Det bør her også poengteres at mye av kommunikasjonen som kan kategoriseres som matematisk foregikk etter at elevene hadde blitt spurt om det enten fra oppgaven eller intervjuer. Dette peker mot at elevene fremmer kommunikasjonskompetansen sin enda mer om oppgaven legger opp til at elevene må formulere funnene sine på en matematisk korrekt måte. Disse funnene stemmer godt overens med funnene til Lie et al. (2017) som også fant at bruk av programmering førte til en bedring av elevenes evner innen kommunikasjon.

Når resultatene først var forstått og kommunisert innad i gruppa og videre til en tredjepart (intervjuer) ble de spurt om å knytte dette opp mot teorien de allerede visste om sannsynlighet. Dette førte til at de fikk arbeidet med resonnementskompetanse gjennom å skulle argumentere rundt matematisk teori med sine egne funn som basis for argumentene de fremmet. Det at elevene aktivt måtte finne argumenter fra eget arbeid viser at simulering er et godt egnet medierende artefakt også for å fremme elevenes resonnementskompetanse. Dette stemmer godt overens med funnene til Psycharis og Kallia (2017) og Lie et al. (2017) som begge pekte mot at undervisning som inneholdt programmering ga økt kompetanse innenfor resonnering.

5.5 Programmering som medierende artefakt for å bygge elevers hjelpemiddelkompetanse

Et annet funn som er interessant for å svare på forskningsspørsmålene er funn knyttet til elevenes hjelpemiddelkompetanse. Elevene uttrykket gode refleksjoner og tanker rundt egen bruk av Python, som demonstrert i utdragene under:

1	Intervjuer	Har dere hatt noe programmering før?																	
2	1A	Altså, vi har jo hatt det her [I mattetimene]																	
3	1B	Men aldri lært det liksom																	HJE

Utdrag fra åpningen av intervjuet med gruppe 1 (Vedlegg 8.5.1)

207	4B	Ja, for nå skriver jeg bare mye av, og så satser jeg på det beste																	HJE
208	4A	Da lærer man ikke så mye av det																	HJE

Utdrag fra samtale med gruppe 4 etter at elevene hadde gjennomført oppgavesettet (Vedlegg 8.8.1)

199	Intervjuer	ja, det er ofte det man må gjøre egentlig. Er dette noe dere kunne tenke dere å ha mer av i mattetimene?																	
200	3B	Så lenge det ikke er for vanskelig. Noe av det kan man jo tenke seg til																	HJE
201	3A	Så er det jo sikkert noe som blir veldig sentralt i fremtiden																	HJE
202	Intervjuer	Ja, det kommer jo godt med i mange yrker																	
203	3A	mhm. Så å få litt grunnlag skader jo ikke det																	HJE

Utdrag fra samtale med gruppe 3 etter at alle oppgavene var ferdige (Vedlegg 8.7.1)

Elevene viser her at de klarer å tenke kritisk til hjelpemidlene de tar i bruk, og hvordan deres egen bruk av hjelpemidlene ikke er optimal. De viser også at de kan se de positive sidene ved å ta hjelpemiddelet i bruk og mulighetene det kan gi å mestre disse verktøyene. Dette viser at denne arbeidsmåten har vært et godt medierende artefakt for å få elevene til å arbeide med hjelpemiddelkompetanse, og å få elevene til å kritisk tenke over egen bruk av hjelpemidler i matematikk.

Disse utdragene kan derimot ikke knyttes direkte opp mot noen av stegene i algoritmisk tenkning, og kan heller sees på som et resultat av arbeidet med hele prosessen. Disse funnene vil derfor forsvinne om man bare ser på hvilke kompetanser enkeltsteg i algoritmisk tenkning kan hjelpe med å bygge. Det er derimot en god mulighet for elevene å bygge videre på sin hjelpemiddelkompetanse, og å tenke kritisk og helhetlig rundt egen bruk av hjelpemidler. Dette er en positiv konsekvens av arbeidet med programmering og bør derfor poengteres og løftes frem i lyset.

5.6 Lærer og/eller oppgavedesign i rollen som proksimal utviklingszone

Fra elevens arbeid med oppgave 2 ble det tydelig at om man som lærer ønsker at elevene skal gjennomføre alle stegene i algoritmisk tenkning på vei mot ferdige koder, må de orienteres i denne retningen av oppgaven eller læreren, frem til stegene eventuelt er så godt innarbeidet at de skjer intuitivt hos elevene. Dette ble spesielt tydelig ved abstraksjon og dekomponering i denne studien, hvor spesielt abstraksjon virket fjernt første gang elevene ble spurt om å gjennomføre en abstraksjon i oppgave 2.

Det kom tydelig frem fra analysen av datamaterialet at om man ønsker at elevene skal få arbeidet med flest mulig kompetanser må lærer eller oppgaveformuleringen legge opp til at elevene blant annet må dele og kommunisere funnene de har fått fra arbeidet. Det ga i disse oppgavene gode resultater i form av økt tankegangskompetanse, kommunikasjonskompetanse og resonnementskompetanse. Det at disse kompetanser først kom til syne når elevene ble spurt om det fra lærer eller oppgaven viser at det å orientere elevene i denne retningen er viktig for å få utnyttet potensialet arbeid med Python har for å fremme disse kompetansene.

Resultatene fra gjennomføringen av oppgavesettet, i tillegg til samtalen med klassens faglærer på forhånd, peker mot at elever i møte med Python sliter med å intuitivt overføre ideer til kommandoer datamaskinen kan forstå. Dette eksemplifiseres godt av elev 1B som underveis i oppgavesettet spør om man bare skal vite hvordan kommandoer i Python ser ut. Dette står i kontrast til funnene til Lie et al. (2017) som fant at elever intuitivt klarte å dette i møte med blokkprogrammering. Dette peker mot at lærere må gi elevene gode strategier for å finne aktuelle koder om elevene skal ta i bruk språk som krever at de skriver egne kommandoer.

Det at denne rollen som proksimal utviklingszone må dekkes er viktig å nevne da algoritmisk tenkning i oppgaver med Python ut ifra denne studien ikke virker som noen snarvei til å bedre kompetansene hos elevene om det ikke blir brukt «riktig». Flere av kompetansene som har blitt vist hos elevene har kommet etter at de har blitt plassert i en slik proksimal utviklingszone av intervjuer og oppgaveformuleringen, noe som viser at dette er noe vi som lærere må ha i bakhodet når elevene skal arbeide med slike oppgaver. Dette stemmer godt overens med funnene til Costa et al. (2017) som pekte mot at det å lage oppgaver som legger opp til at elever gjennomfører stegene i algoritmisk tenkning gir positive utslag i deres evne til å løse problemer.

5.7 Simulering som mulig løsning på problemer med tid

Flere av artiklene nevnt i teoridelen poengterer at det å innføre programmering i matematikkfaget er en tidkrevende prosess, spesielt om innføringen ikke også skjer i andre fag (Humble et al., 2020; Sanne et al., 2016).

Dette stemmer godt overens med erfaringen læreren jeg har vært i kontakt med har hatt med å innføre programmering i sin S1-klasse. I den klassen løste læreren dette problemet ved å gi elevene maler de kunne følge for å lage kodene, og koder som inneholdt små feil elevene måtte rette på. Som presentert i resultatene av denne forskningen har denne måten å undervise programmering på gitt elevene en viss kompetanse innen programmering, og en grunnforståelse for hvordan man kan ta i bruk programmering som hjelpemiddel.

Det som derimot kom frem i funnene fra mine intervjuer var at elevene etter en slik introduksjon ikke hadde fått arbeidet tilstrekkelig med algoritmisk tenkning til at dette var prosesser de intuitivt gjennomførte i arbeidet frem mot et ferdig program. Dette stemmer godt overens med funnene til Sanne et al. (2016) som pekte mot at det ikke er nok å bare innføre programmering i mattefaget, da dette vil gi for lite tid for å lære programmering og algoritmisk tenkning.

Resultatene fra elevenes arbeid med oppgavesettet viste som presentert i kapittel 4 at det å bare ta i bruk simulering, uten å gjennomføre alle stegene i algoritmisk tenkning på vei mot egenproduserte koder, ga elevene mulighet til å styrke flere kompetanser. Dette betyr at selv om elevene gjennom å bare lære programmering i matematikkfaget ikke har tid til å få tilstrekkelig kjennskap til algoritmisk tenkning, allikevel kan ha godt utbytte av å ta i bruk programmer i Python for å gjennomføre simuleringer av matematiske hendelser.

Å gå for denne løsningen vil derimot gå på bekostning av elevenes mulighet til å arbeide med flere kompetanser, som denne oppgaven har presentert at fremmes gjennom alle stegene i algoritmisk tenkning. Det som derimot må tas i betraktning her er at selv om simulering lar elevene arbeide med færre matematiske kompetanser enn om de hadde tatt i bruk algoritmisk tenkning, viser denne studien at å ta i bruk simulering i seg selv kan gi elevene mye god innsikt i matematisk teori. Dette er gunstig for elevenes læring på tross av om de har mulighet til å arbeide med flere kompetanser eller ikke, og svekker derfor i den forstand ikke simuleringens plass som et godt hjelpemiddel i matematikkundervisningen.

6 Avslutning

6.1 Konklusjon

I starten av denne masteroppgaven satte jeg ut for å undersøke hvilke muligheter innføringen av programmering i Python i læreplanen kunne ha for utviklingen av kompetanser hos elever i et matematikklasserom. Derfor ble forskningsspørsmålene denne oppgaven er bygget opp for å undersøke seende slik ut:

- 1. Hvilke av Niss matematiske kompetanser arbeider elevene med når de tar i bruk programmering og algoritmisk tenkning som medierende artefakter i matematiske oppgaver knyttet til sannsynlighetsregning?**
- 2. I hvor stor grad gjenspeiles stegene i algoritmisk-tenkningsprosess i elevenes arbeid med oppgaver som tar i bruk programmering i Python?**

Etter å ha analysert og drøftet funnene jeg fikk fra gjennomføringen av oppgavesettet og intervjuene, ble det tydelig at bruk av algoritmisk tenkning i programmeringsoppgaver ga rom for at elevene kunne arbeide med alle de matematiske kompetansene til Niss, så lenge oppgavene eller læreren la opp til det. Funnene peker mot at om elevene bare får presentert en ferdig kode som de får i oppgave å ta i bruk som et simuleringsverktøy kan dette være et godt medierende artefakt for å arbeide med tankegangskompetanse, kommunikasjonskompetanse, resonnementskompetanse, symbol- og formalismekompetanse, representasjonskompetanse og modelleringskompetanse.

Det ble derimot klart at for å maksimere elevenes bruk og styrkning av disse kompetansene kreves det at enten oppgavedesignet eller lærer aktivt legger til rette for det. Dette ble spesielt klart ved kompetansene som omhandler kommunikasjon og resonnering. Elevene fikk vist hvilke ferdigheter de hadde innenfor disse kompetansene først når de ble spurt om å oppsummere sine funn, noe en tredjepart ikke hadde fått innsikt i om elevene ikke hadde blitt spurt direkte om det. Dette betyr for lærere at vi i oppgaver som tar i bruk programmering må være bevisste på å skape en størst mulig proksimal utviklingszone for elevene gjennom vår og oppgavens formulering av spørsmål.

Funnene peker også mot at det å ta i bruk alle stegene i algoritmisk tenkning er et bedre medierende artefakt enn simulering med ferdige koder, når målet er å gi elevene mulighet til å bygge en mest mulig komplett modelleringskompetanse, da de ulike stegene i prosessen gir innblikk i hver sin del av modelleringsprosessen. Det å ta i bruk dekomponering og abstraksjon ga også elevene mulighet til å arbeide med problemløsningskompetanse, da det ga dem mulighet til å arbeide med hvordan man kan angripe store og mer komplekse matematiske problemer.

Når det kommer til det andre forskningsspørsmålet ble det tydelig i samtlige grupper at i elevenes selvstendige arbeid kom flere av stegene i algoritmisk tenkning i liten grad til syne. Dette gjaldt spesielt abstraksjon og dekomponering, som det virket som var nye prosesser for elevene. Det ble derimot tydelig at elevene hadde nok kunnskap til å gjennomføre disse stegene når de ble spurt om å gjennomføre dem, noe som peker mot at de ikke har god kjennskap til algoritmisk tenkning som prosess og ferdighet. Elevene viste derimot at de i stor grad fikk til å tenke i algoritmer og overføre dem til koder, samt å gjennomføre evalueringer av kodene de fikk presentert.

6.2 Studiens begrensninger

En åpenbar begrensning ved denne studien er tiden jeg har hatt til rådighet til å gjennomføre prosjektet. Dette har vært en masteroppgave med en tidshorisont på et semester, noe som har

begrenset hvor stort datagrunnlag jeg har kunnet analysere grundig. Begrensningen i tid har blant annet medført at jeg bare har fått gjennomført oppgaver knyttet til ett matematisk tema. Dette har ikke påvirket min evne til å besvare mine forskningsspørsmål, men har påvirket i hvor stor grad funnene mine kan generaliseres til å gjelde andre matematiske områder. Det hadde derfor vært interessant å se hvordan funn med en annen elevgruppe og innenfor et annet matematisk tema hadde sett ut i forhold til mine funn.

En annen potensielt begrensende faktor som er verdt å nevne for fremtidig relevans for S1-klasser er at de nye læreplanene på tidspunktet denne forskningen ble gjennomført fortsatt var relativt nye. Dette kan medføre at elevene i de kommende årene møter programmering i lavere klassetrinn, noe som betyr at de kan komme til å ha et høyere nivå innen algoritmisk tenkning og programmering enn det som ble funnet i denne studien.

Studien har i tillegg til de nevnte faktorene også blitt påvirket av mine egne valg knyttet til hvordan jeg gjennomførte studien. I avsnitt 3.11 nevnes blant annet at mitt valg av metode har ført til at studiens generaliserbarhet har blitt svekket, noe som begrenser i hvor stor grad andre lærere kan ta i bruk mine funn uten å se dem i lys av sine egne klasserom. Analysen har også blitt gjennomført av meg alene, som betyr at mine subjektive tanker og ideer kan ha spilt inn på måten datamaterialet har blitt analysert. Dette vil også være en begrensende faktor som kan påvirke konklusjonenes gyldighet, på tross av at jeg har forsøkt å ta valg underveis for å luke bort mine egne subjektive tolkninger.

Mitt valg av teoretisk rammeverk vil også kunne være med på å begrense studiens utforsking av intervjuene, da jeg analyserte datamaterialet etter kompetansene beskrevet av Niss. Det er ikke garantert at dette er den eneste korrekte måten å se på matematiske kompetanser, og det er en viss mulighet for at deler av matematikken faller mellom kategoriseringene gjort av Niss. Dette vil igjen medføre at det kan hende at enkelte aspekter av kompetanse kan ha dukket opp i mitt datagrunnlag uten at det har blitt vektlagt i min analyse og konklusjon. Jeg har derfor lagt ved alle transkripsjoner og besvarelser så forskere med eventuelt andre syn på kompetanse kan tolke datamaterialet i lys av sitt syn.

6.3 Implikasjoner og videre forskning

Om man velger å akseptere funnene fra denne studien vil det medføre flere implikasjoner for oss som lærere. Den første av disse er at oppgaver basert på algoritmisk tenkning i programmeringsoppgaver har vist seg å være en god arena for å gi elevene mulighet til å arbeide med flere matematiske kompetanser på en gang. Dette medfører at det er et godt redskap for oss lærere å ha tilgjengelig for å skape en variert undervisning som samtidig er gunstig for å styrke elevenes matematiske kompetanser. Det at det tillater arbeid med så mange kompetanser betyr også at det kan virke som at oppgaver som krever konstruksjon av programmer i Python kan være en god arena for å se i hvor stor grad elevene innehar alle de gitte kompetansene. Hvordan dette kan gjøres på best mulig måte har jeg ikke svar på fra denne oppgaven, men kunne vært interessant å forske videre på i et annet prosjekt.

Funnene fra denne studien støtter tidligere forskning som peker mot at elevene ikke får tilstrekkelig kompetanse til å konstruere komplekse koder selv om programmering bare innføres i mattefaget på videregående. Det at programmering nå også innføres i lavere klassetrinn kan være med på å bedre dette problemet, men som nevnt i avsnittet om tidligere forskning på temaet kan det se ut som om et eget programmeringsfag på skolen et sted underveis i studieløpet kan være fornuftig. Det å bruke så mye tid på å lære elevene å programmere kan forsvares ved at denne studien, og flere andre som er nevnt tidligere, viser til at programmering er et godt medierende artefakt for å bygge elever kompetanser, samtidig som skoleverket da vil utruste elevene for et moderne arbeidsmarked.

Arbeidet elevene har gjort i denne studien viser også at det å først gi elevene en tilnærmet ferdig kode som de videre modifierer, for så å skape egne lignende koder kan være en gunstig fremgangsmåte for å lære elevene programmering. Dette minner mye om fremgangsmåten Use-modify-create (bruk-modifiser-konstruer, min egen oversettelse) som er presentert av Franklin et al. (2020), som også baserer seg på å først gi elevene en ferdig kode som de kan modifisere, før de forsøker å lage egne koder. Det å forske videre på denne måten å introdusere programmering på kan derfor virke som en meget gunstig måte å få et rammeverk for implementeringen av programmering i klasserommet.

6.4 Eget utbytte

Etter at arbeidet med denne oppgaven er ferdig sitter jeg igjen med mye ny kunnskap som er relevant for min videre karriere som lærer. En av de meste åpenbare, og en av grunnene til at jeg var interessert i å undersøke nettopp dette temaet, er at jeg har tvunget meg selv til å lære mer om programmering, både å utvikle og skrive kode i Python og å implementere det i klasserommet. Dette ser jeg på som en stor fordel da programmering nå er en del av læreplanen, og det at jeg har økt min egen kunnskap i dette gjør at jeg nå kan bedre elevenes muligheter til å lære det læreplanen ønsker at de skal lære.

Jeg har også fått gode innblikk i hvordan man kan legge til rette for å fremme ulike kompetanser hos elevene i arbeidet med programmering. Dette ser jeg på som spesielt positivt av flere grunner. For det første gir det meg mulighet til å hjelpe elevene med å utvikle disse kompetansene. En annen positiv side med dette er at det har gitt meg ideer til hvordan jeg kan finne ut hvilke kompetanser elevene innehar, noe som kan gjøre at jeg får et mer helhetlig bilde av dem som matematikere.

7 Referansliste

- Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads*, 2(1), 48-54.
- Bell, J. & Waters, S. (2018). *Doing your research project : a guide for first-time researchers* (7. utg.). McGraw-Hill Education.
- Bryman, A. (2016). *Social research methods* (5. utg.). Oxford University Press.
- Costa, E. J. F., Campos, L. M. R. S. & Guerrero, D. D. S. (2017). Computational thinking in mathematics education: A joint approach to encourage problem-solving ability. I *Proceedings of 2017 IEEE Frontiers in Education Conference (FIE)* (s. 1-8). Indianapolis, USA. <https://doi.org/10.1109/FIE.2017.8190655>.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. & Woollard, J. (2015). Computational thinking-A guide for teachers. https://eprints.soton.ac.uk/424545/1/150818_Computational_Thinking_1_.pdf
- Forsström, S. E. & Kaufmann, O. T. (2018). A Literature Review Exploring the use of Programming in Mathematics Education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18-32. <https://doi.org/https://doi.org/10.26803/ijlter.17.12.2>
- Franklin, D., Coenraad, M., Palmer, J., EATINGER, D., Zipp, A., Anaya, M., White, M., Pham, H., Gökdemir, O. & Weintrop, D. (2020). An Analysis of Use-Modify-Create Pedagogical Approach's Success in Balancing Structure and Student Agency. I *Proceedings of the 2020 ACM Conference on International Computing Education Research* (s. 14-24). Virtuall konferanse, New Zealand. <https://doi.org/http://dx.doi.org/10.1145/3372782.3406256>
- Goldin, G. A. (2000). A scientific perspective on structured, task-based interviews in mathematics education research. I A. E. Kelly & R. A. Lesh (Red.), *Handbook of research design in mathematics and science education* (s. 517-545). Lawrence Erlbaum Associates.
- Hansen, N. K., & Hadjerrouit, S. (2021). Exploring Students' Computational Thinking for Mathematical Problem-Solving: A Case Study. I D. D.G. Sampson; D. Ifenthaler, & Isaías, P. (Red.), *Proceedings of the 18th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA2021)* (s. 251-260). IADIS Press
- Humble, N., Mozelius, P. & Sällvin, L. (2020). The Introduction of Programming in K-12 Technology and Mathematics: Teacher choice of programming tools and their perceptions of challenges and opportunities. I M. Carmo (Red.), *Education Applications & Developments V: Advances in Education and Educational Trends* (s. 117-126). Science Press.
- Imsen, G. (2014). *Elevens verden : innføring i pedagogisk psykologi* (5. utg.). Universitetsforlaget.
- Lie, J., Hauge, I. & Meaney, T. (2017). Computer programming in the lower secondary classroom: learning mathematics. *Italian Journal of Educational Technology*, 25(2), 27-35.
- Maher, C. A. & Sigley, R. (2020). Task-based interviews in mathematics education. I S. Lerman (Red.), *Encyclopedia of mathematics education* (s. 821-824). Springer.
- Mészárosóvá, E. (2015). Is python an appropriate programming language for teaching programming in secondary schools. *International Journal of Information and Communication Technologies in Education*, 4(2), 5-14.

- Misfeldt, M. & Ejsing-Duun, S. (2015). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. I K. Krainer & N. Vondrova (Red.), *Proceedings of CERME 9* (s. 2524-2530). Praha, Tsjekkia.
- Mladenović, M., Krpan, D. & Mladenović, S. (2016). Introducing programming to elementary students novices by using game development in Python and Scratch. *EDULEARN16 Proceedings*, 1622-1629.
- Niss, M. (2003). Mathematical competencies and the learning of mathematics: The Danish KOM project. I A. Gagatsis & S. Papastavridis (Red.), *3rd Mediterranean conference on mathematical education* (s. 115-124). Aten, Hellas.
- Niss, M. & Jensen, T. H. (2002). *Kompetencer og matematiklæring : ideer og inspiration til udvikling af matematikundervisning i Danmark* (Bd. nr 18 - 2002). Undervisningsministeriet.
- NOU 2013:2. (2013). *Hindre for digital verdiskaping*. Kommunal- og distriktsdepartementet. <https://www.regjeringen.no/no/dokumenter/nou-2013-2/id711002/>
- Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas*. Basic Books.
- Psycharis, S. & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional science*, 45(5), 583-602. <https://doi.org/10.1007/s11251-017-9421-5>
- Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., Mørken, K. M., Svorkmo, A.-G. & Voll, L. O. (2016). *Teknologi og programmering for alle - En faggjennomgang med forslag til endringer i grunnopplæringen - august 2016*. Utdanningsdirektoratet. <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2414205/teknologi-og-programmering-for-alle.pdf?sequence=3>
- Senter for IKT i utdanningen. (2016). *Programmering i skolen*. https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Utdanningsdirektoratet. (2020a). *Hva er nytt i matematikk?* <https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/>
- Utdanningsdirektoratet. (2020b). *Matematikk S (MAT04-02) : Kompetansemål og vurdering*. <https://www.udir.no/lk20/mat04-02/kompetansemaal-og-vurdering/kv295>
- Utdanningsdirektoratet. (2021a). *Hvorfor har vi fått nye læreplaner*. <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvorfor-nye-lareplaner/>
- Utdanningsdirektoratet. (2021b). *Algoritmisk tenkning*. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2021c). *Eksempeloppgaver i matematikk S1*. <https://www.udir.no/eksamen-og-prover/eksamen/eksempeloppgaver/eksempeloppgaver-i-matematikk-s1/#a166333>
- Vansteenkiste, G. (1975). Simulation in mathematics education. *International journal of mathematical education in science and technology*, 6(1), 83-86. <https://doi.org/10.1080/0020739750060108>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>

Yelland, N. (1995). Mindstorms or a storm in a teacup? A review of research with Logo. *International journal of mathematical education in science and technology*, 26(6), 853-869.
<https://doi.org/10.1080/0020739950260607>

8 Vedlegg

8.1 Informasjonsskriv/ Samtykkeerklæring

Vil du delta i forskningsprosjektet

” Innføring av programmering i matematikkundervisningen ”?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å se på hvordan programmering kan innføres på en god måte i matematikkundervisning, og hvilke kompetanser det kan føre til. I dette skrevet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål

I den nye læreplanen som nå skal tas i bruk på skolene i Norge nevnes programmering eksplisitt i flere kompetansemål i matematikk. Dette gjør at vi som lærere må finne ut gode måter å implementere dette i våre klasserom, så det skal gi best mulig læring for elevene. Målet med dette prosjektet er å se på et matematisk tema og undersøke hvordan programmering kan brukes for å bedre læringen i temaet og samtidig gi læring innenfor programmering.

Problemområdet jeg vil jobbe med handler derfor om hvordan elever kan ta i bruk programmering for som et verktøy for å lære matematikk.

Prosjektet er en del av min masterutdanning, og resultatet fra forskningen vil danne grunnlaget for min masteroppgave.

Hvem er ansvarlig for forskningsprosjektet?

Fakultetet for teknologi og realfag, matematisk institutt ved UiA er ansvarlig for prosjektet.

Hvorfor får du spørsmål om å delta?

Grunnen til at akkurat dere blir spurt om å delta er fordi Vennesla VGS er en av skolene som allerede har kommet et stykke på å få programmering inn i skolen, og som derfor er en god arena å undersøke en del av læreplanen som få har rukket å implementere.

Hva innebærer det for deg å delta?

- *«Hvis du velger å delta i prosjektet, innebærer det at du deltar i gjennomføringen av et sett med oppgaver knyttet til programmering. Det vil ta deg ca. 60 minutter. Jeg vil ta opp lyd underveis i gjennomføringen og komme med spørsmål underveis om tankegang rundt gjennomføringen.»*

Dersom deltaker eller foresatte ønsker å se oppgavene eller en guide til spørsmålene jeg ønsker å stille underveis er det bare å ta kontakt.

Det er frivillig å delta

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

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger

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

- *Ved UiA vil bare veileder og masterstudent ha tilgang på resultatene fra undersøkelsen*
- *Navnet ditt vil ikke brukes i masteroppgaven, og vil bli erstattet med et navn eller tall som ikke kan spores tilbake til deg.*

- *Lyddopptak vil ikke bli publisert og vil slettes etter at en anonymisert transkripsjon er gjennomført.*

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?

Opplysningene anonymiseres og all datamateriale utenfor oppgaven vil slettes når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er 16 mai 2022. Da vil alt av lydopptak bli slettet og det vil ikke finnes noe liste over deltakere i prosjektet.

Dine rettigheter

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

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

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra Fakultetet for teknologi og realfag, matematisk institutt ved UiA har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?

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

- Ved spørsmål knyttet til gjennomføring eller ved ønske om å lese gjennom oppgaven kan du ta kontakt med Markus Tveit på epost (m.tveit.98@gmail.com) eller på telefon: 41362814.
- Om du ønsker å ta kontakt med veileder på masteroppgaven kan du ta kontakt med Said Hadjerrouit på mail: said.hadjerrouit@uia.no
- Vårt personvernombud: Personvernombud@uia.no

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

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

Med vennlig hilsen

Said Hadjerrouit
(Forsker/veileder)

Markus Tveit
(Masterstudent)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet ” *Innføring av programmering i matematikkundervisningen* ” og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i gjennomføringen av et sett med oppgaver som blir observert med mulighet for spørsmål underveis
- at anonymiserte svar/resultater fra gjennomføringen kan tas i bruk i masteroppgaven knyttet til prosjektet
- Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

(Signert av prosjektdeltaker, dato)

(Signert av foresatte, dato)

8.2 Godkjenning fra NSD

Meldeskjema for behandling av personopplysninger

15.11.2021, 12.49



NSD sin vurdering

Prosjekttittel

Masteroppgave om innføring av programmering i matematikkundervisning

Referansenummer

579479

Registrert

09.11.2021 av Markus Tveit - markt17@student.uia.no

Behandlingsansvarlig institusjon

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

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Said Hadjerrouit, said.hadjerrouit@uia.no, tlf: +4738141793

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Markus Tveit, m.tveit.98@gmail.com, tlf: 41362814

Prosjektperiode

01.01.2022 - 16.05.2022

Status

15.11.2021 - Vurdert

Vurdering (1)

15.11.2021 - Vurdert

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet

den 15.11.2021 med vedlegg, samt i meldingsdialogen mellom innmelder og NSD.

Behandlingen kan starte.

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:

<https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/melde-endringer-i-meldeskjema>

Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET

Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 16.05.2022.

LOVLIG GRUNNLAG

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

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

PERSONVERNPRINSIPPER

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

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke 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

NSD vurderer at informasjonen om behandlingen som de registrerte og deres 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 sine/barnets rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER

NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1 f) og sikkerhet (art. 32).

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

OPPFØLGING AV PROSJEKTET

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

Lykke til med prosjektet!

Kontaktperson hos NSD: Gry Henriksen
Tlf. Personverntjenester: 55 58 21 17 (tast 1)

8.3 Oppgavesettet

Simulering av stokastiske hendelser

Oppgave 1: Myntkast

I mattetimen får dere beskjed om å lage et program som skal simulere det å kaste en mynt. Programmet skal vise om dere får kron eller mynt, og dette resultatet skal være helt tilfeldig. Programmet dere lager blir sende slik ut:

```
1 import random
2
3 mynt = ["Kron", "Mynt"]
4 kast = random.choice(mynt)
5
6 print(kast)
7
```

- Er det noen fordeler med å bruke simulering i slike forsøk?
- Hvordan ville dere gått frem for å få programmet til å vise to myntkast?

Etter å ha kjørt dette programmet et par ganger ser dere at vi ikke har fått akkurat like mange kron som mynt, noe man kanskje skulle tro fra det vi vet om sannsynlighet. For å teste dette videre bestemmer dere dere for å lage et program som simulerer 50 myntkast for å se om det da blir et jevnere utfall. Programmet blir sende slik ut, men fungerer ikke:

```
1 import random
2
3                                     # Presiserer at alle
4 Kron = 0                             #variabler starter på 0
5 Mynt = 0
6
7 while kast < 50:                       #Dette forteller hvor mange
8     kast += 1                           # kast som skal gjennomføres
9
10    utfall = random.randint(1, 2)       #random.randint(1,2) får
11    if utfall == 1:                     #programmet til å velge et
12        Kron += 1                       #heltall fra og med 1 til og
13    else:                                #med 2
14        Mynt += 1
15
16 print("Vi fikk til sammen,", Kron, "kron og", Mynt, "mynt etter", kast, "kast")
17
```

- Find feilen i koden, og endre den så simuleringen fungerer.
- Kjør simuleringen flere ganger, og se på resultatet. Vil du basert på det du ser si at det å regne med sannsynlighet er en fornuftig måte å anslå virkelige hendelser på?
- Hvordan kan koden endres dersom du vil at simuleringen skal demonstrere at det faktisk stemmer at det er 50% sannsynlighet for begge utfallene?

Oppgave 2: Terningkast

Dere er på hyttetur og har nettopp satt dere ned for å spille yatzy, men finner fort ut at ingen av terningene er der de skal. Det dere derimot har er en pc, og får en ide om at dere bare kan lage et program som kan simulere terningkastene.

- f) Lag et program som simulerer terningkastene i yatzy (Vanlig yatzy har fem terninger som kastes samtidig)

Hint: Når man jobber med programmering er et lurt skritt å ta det vi kaller dekomponering, som betyr å dele problemet opp i mindre deler som hver for seg er lettere å løse. Det kan være lurt å prøve på her også.

8.4 Intervjuguide

Intervjuguide til gjennomføring av masterforskning

HUSK Å SKRU PÅ LYDOPPTAK OG TILBAKESTILL KODER

Vær åpen for at det kan dukke opp interessante spørsmål underveis. Vær obs på om en av deltakerne er mer dominerende enn den andre

Starte med å takke for at elevene er med på opplegget, og fortelle om målet med masteroppgaven.

Spør om elevenes bakgrunn i programmering, og om de har brukt Python mye før

Oppgave 1

- Spiller det noen rolle her om dere fysisk kaster mynten eller om dere simulerer kastet?
Kan dere se noen tydelige fordeler med å bruke et program til å løse slike oppgaver?
- Hint:* Er det mulig å få printet to kast i print-funksjonen til programmet?
- Har dere en strategi når dere går i gang med å evaluere en kode som denne?
- Hvor mange ganger føler dere at dere må kjøre simuleringen før dere er sikre på at dere har et godt bilde av resultatet?
Hvordan vil dere argumentere for svaret deres?
- Hint:* Hva skjer om dere endrer antallet kast simuleringen gjennomfører?
Hvordan vil dere skrevet ned funnene deres om det skulle leses av en matematiker?
Hvordan påvirker dette resultatet svaret deres fra den forrige oppgaven? (Hvis det er endret)

Oppgave 2

Hvordan tenker dere å angripe denne oppgaven?

Kan man få bruk for noen matematiske temaer i yatzy? Hvilke da?

Om de sliter spør om hva de ønsker å få koden til å gjøre, for å se om det stopper på koding eller forståelse:

Kan dere bryte dette problemet ned i mindre problemer? F.eks ett terningkast?

Hvilke detaljer fra et fysisk kast er viktige å få med i programmet dere ønsker å lage?

Hva trenger datamaskinen å vite for å simulere det du ønsker den skal simulere?

Har du definert alt du trenger å definere?

Fikk dere koden til å vise det dere ønsket?

Hva kunne dere lagt til for å videreutvikle denne koden?

Stoppet dere opp og tenkte gjennom hvordan dere ønsket at koden skulle se ut før dere begynte å kode?

Til slutt: Igjen takke elevene for samarbeidet og informere at de kan ta kontakt om de vil trekke seg fra undersøkelsen eller få innsyn i hva som blir skrevet.

HUSK Å SKRU AV LYDOPPTAK OG TA SKJERMBILDER AV KODENE

8.5 Datamateriale gruppe 1

8.5.1 Transkribering og analyse av lydopptak

1	Intervjuer	Har dere hatt noe programmering før?									
2	1A	Altså, vi har jo hatt det her [I mattetimene]									
3	1B	Men aldri lært det liksom								HJE	
4	Intervjuer	For det er det vi skal prøve å gjøre det idag, lære det litt mer fra bunnen av									
5		Så da kan dere egentlig bare starte med å lese den første oppgaven									
6		(elevene tester den første koden)									
7	Intervjuer	Så ser du der nede at du nå fikk mynt. Så kan dere prøve en gang til bare å se om vi får begge alternativene									
8	1B	Så du får egentlig bare et vanlig myntkast da?			MOD	REP					
9	1A	Ja									
10	Intervjuer	Klarer dere å forstå hva hver av disse linjene gjør?									
11	1A	ehhh, ja skal vi se									
12		...									
13	1A	Jeg vet at print betyr at det skal skrives inn.. Nei, at skal skrives ut mener jeg									
14	Intervjuer	Ja									
15		...									
16	1B	Men sånn som det der skjønner ikke jeg (peker på random.choice)									
17	Intervjuer	Det er ikke så rart. Det er en kommando som ligger inne i programmer som velger et tilfeldig alternativ innenfor den lista vi skriver inni parantesen. Slike kommandoer må man									

		lære eller hente fra en liste.									
18	1A	Altså det mellom de klammene?									
19	Intervjuer	Ja, av det som finnes inni mynt									
20	1A	Ja									
21	1B	Ja, sånn ja									
22	Intervjuer	Så var det forståelig									
23	1A og 1B	Ja ja									
24	Intervjuer	Så da er jo spørsmålet, er det egentlig noen fordel med å bruke simulering her istedenfor å bare kaste en mynt?									
25		...									
26	Intervjuer	Føler dere det er noe poeng når dere ser på denne koden?									
27	1A	Når det bare er ett myntkast mener jeg man bare kunne gjort det, men hvis man skulle sett på sannsynlighet med mange kast hadde det vært lurt å gjøre sånn [simulering]			MOD		REP			HJE	
28	Intervjuer	Det er jo fornuftig. Men tror dere at jeg kunne øvd inn en teknikk så kunne få hvilket utfall jeg ville i virkeligheten? Så jeg kunne blitt svindler									
29	1A	Ja, kanskje									
30	1B	Ja, men det kan du jo ikke på den [simuleringen]			MOD		REP			HJE	
31	Intervjuer	Ja, så simuleringen gjør det helt tilfeldig								HJE	
32	1A	Ja, da er jeg med									
33		(elevene leser neste oppgave)									
34	1A	okei									

35	1B	Der mister du meg med en gang										
36	Intervjuer	Det er forståelig. Men hvis tenker tilbake på den forrige. Hva gjør linjene?										
37	1B	Den første viser hva man kan velge mellom										
38	Intervjuer	Ja										
39	1A	Og den nester kaster mynten										
40	Intervjuer	Riktig, så kunne dere da definert et til kast?										
41	1B	Kan du da skrive samme linja om igjen, eller må du skrive noe nytt på samme linja?			MOD							Algoritme
42	Intervjuer	Dere kan prøve, og se hva som skjer										
43		(elevene skriver inn en kode)										Koding
44	1B	Så må jeg ta command I [kommando for å kjøre koden]										
45	1A	Det funket jo ikke										Evaluering
46	Intervjuer	Du er veldig inne på noe, men nå printer den fortsatt ett kast										
47	1A	ja...										
48	1B	1A, kom igjen										
49	Intervjuer	Hva er det Pcen har fått beskjed om å gjøre her?										
50		...										
51	Intervjuer	Her har jo pcen bare fått beskjed å om gjøre akkurat samme ting to ganger, og printe den ene										
52	1A	Kan vi kalle den kron?										
53		(taster inn kode)										Koding
54	1B	Sånn liksom?										
55	1B	Nei, det var feil.. For det står jo ikke kron der (peker på forrige kode)										Evaluering
56	Intervjuer	Klarer pcen å forstå forskjellen på de to										

		kastene dere har definert?									
57	1A	Nei, eller vi må finne måte å gjøre det på									
58	Intervjuer	Kunne dere for eksempel kalt det kast 2?									
59	1B	Ja (skriver inn ny kode)									Koding
60	1A	Da fikk vi kron									
61	Intervjuer	Vi må huske at pcen ikke gjør mer enn den blir fortalt									
62	1B	Ja, også kanskje print også kast 2									
63	Intervjuer	Prøv det									
64		...									
65	Intervjuer	Ja, så bra									
66	1A	Wow, imponert									
67	1B	ja, nå imponerer vi her									
68	Intervjuer	Så dere skjønner forskjellen?									
69	1A	Ja									
70	1B	Ja, for den skriver jo bare akkurat det kastet der									
71	Intervjuer	Ja, da kan dere gå videre									
72		...									
73	1A	Så vi skal se hva som er feil i den, for den sjekker femti kast den?									
74	Intervjuer	Ja, for med femti kast er den forrige metoden veldig tungvinn, så nå begynner det å bli litt vanskeligere									
75	1B	Var det noe feil i den?									
76	Intervjuer	Ja, dere kan prøve å kjøre koden og se									
77		(Tester koden)									
78	1A	ja, okei									
79	Intervjuer	Vi kan jo først prøve å forstå hva alle delene av koden gjør, kan dere se hva som er målet med alle linjene									
80		...									

81	1B	Hvorfor står det +=?										
82	Intervjuer	Det betyr at så lenge antall kast er under femti så kaster vi en gang til										
83	1A	okei										
84	1A	Når det står while kast, er det bare en løkke som går da?										
85	Intervjuer	Ja, så lenge kast er under femti										
86	1B	Men, while hva betyr det?										
87	Intervjuer	så lenge. Så lenge kast er under femti så gjør vi ett til										
88	1B	Ja, sånn ja										
89	1A	mhm										
90	1B	Men hvordan vet du at du skal skrive random.randint?										
91	Intervjuer	Det er en kode som ligger inne i programmet, så det er noe man får finne i en database og som man lærer etterhvert										
92	1B	Ja, men jeg vet ikke hva som er feil										
93	Intervjuer	Men har dere en strategi for å prøve å finne ut av det?										
94	1A	Det står hvilken linje det er feil på, linje 8		PRO						HJE		Evaluering
95	1B	Den med while kast, så det er jo her da. Men jeg har ikke noe strategi		PRO								Evaluering
96	1A	Men kast er jo ikke definert										Evaluering
97	1B	Nei det var det. Det står jo ikke kast der										Evaluering
98	1A	Ja, for den vet jo ikke hva kast er										Evaluering
99	Intervjuer	Så hva kan vi gjøre for å fikse det?										
100	1B	Definere kast										Algoritme
101	1A	Hvis vi prøver å begynne da										
102		(Skriver inn kode)										Koding

103	1A	Hva skal vi definere det som?									
104	1B	Må vi ha de klammene?									
105	1A	Hvis vi skriver kron komma mynt									
106	Intervjuer	Når dere definerer kast, har dere da tenkt på hva dere vil at kast skal være?									
107	1A	Nei									
108		...									
109	Intervjuer	Hva er det egentlig kast betyr i vår kode?									
110	1B	Det er jo hvor mange ganger den skal, så er det fra 1 til 50?			MOD						Algoritme
111	Intervjuer	Så hva er da lurt å starte med kast som									
112	1B	en eller null?			MOD				HJE		Algoritme
113		(skriver inn kode med kast som 1)							HJE		Koding
114	Intervjuer	Hvor mange kast har vi fått nå?									
115	1A	49.									
116	1B	så da er det null			MOD						Evaluering
117		(Endrer koden)							HJE		Koding
118	1B	Oi, det var jo 50-50 også det									
119	Intervjuer	Bra. Så dere var hele tiden inne på noe. Vi har jo definert hvor den skal slutte, men aldri hvor den skulle starte									
120	1A	Ja									
121	Intervjuer	Så dere kan kjøre den koden et par ganger og se hva dere får									
122	1A	Det ble veldig nærme ja									
123	1B	Nå er vi litt lengre unna									
124	1A	Men vi får oftest 24 eller 25									
125	1B	Men som regel så er det sånn 50-50									
126	Intervjuer	Så hvor mye stoler vi på at det er 50-50 nå?									
127	1A	litt									

128	Intervjuer	Med det dere fikk fra denne koden, vil dere si at sannsynlighet er en fornuftig måte å beregne virkelige hendelser på?									
129	1B	Den treffer jo sånn cirka, men det er jo ikke hver gang	TAN		MOD	RES				KOM	
130	Intervjuer	Men hadde dere turt å vedde penger på det									
131	1B	Nei, det hadde ikke jeg veddet noe på									
132	1A	Men vi får det jo en del									
133	1B	Men kan ikke stole helt på det	TAN								
134	1A	Det tar oss vel til neste det?									
135	Intervjuer	Det tar oss til neste det ja									
136		...									
137	1B	Jeg ville jo ha økt antall kast		PRO							
138	Intervjuer	Hvorfor kan det være lurt?									
139	1B	ehhh, følte læreren sa det i timen									
140	1A	For jo mer vi tok, jo nærmere og nærmere kom vi									
141	Intervjuer	Ja, da kan dere prøve det									
142	1B	Skal vi prøve hundre først									
143	1A	Ja, prøv det									
144	1B	Det gikk bedre i stad									
145	1B	Oi, der var vi nærme!									
146	1B	Hvis vi prøver mer da									
147	1A	Prøve tusen?									
148	1B	Ja									
149	1B	Det var nærme da!									
150	1B	ti tusen da!									
151	1A	Det er veldig nærme									
152	1A	Prøv en gang til med en til null									
153	Intervjuer	Nå har det begynt å sentrere seg litt?									
154	1B	Ja, nå ligger den alltid på 50 eller 49 og noe									

155	Intervjuer	Ville dere vært villig til å vedde mer penger på at den havner på mellom 49 og 51 nå?									
156	1B	Ja									
157	1A	mhm									
158	1B	For den lander alltid på 49,8 og oppover									
159	Intervjuer	Kunne dere laget en setning som oppsummerte hva vi hadde funnet ut om sannsynlighet ved å bruke denne koden?									
160	1A	Det er jo 50-50 hver gang, men jo flere ganger du kaster jo mer sentrerer du til sammen for å likt resultat mellom kron og mynt	TAN			RES		SYM	KOM		
161	Intervjuer	Ja, det er veldig fint									
162	1B	Jeg er enig, men har ikke en setning akkurat nå									
163	Intervjuer	Da går vi videre til oppgave 2. Her skal dere prøve å gjøre litt mer selv, noe jeg har forstått at dere ikke er så vant med fra timene									
164	1A	Nei									
165	1B	Nei, for jeg tenkte i stad at jeg ikke skjønner hvordan man finner while og randint									
166	Intervjuer	Det er noe man bare må lære seg. Man kan enten finne det på google, eller i et register. Det blir litt samme som å lære et nytt språk									
167	1B	okei, det er sånn ja									
168	Intervjuer	Så jeg forventer ikke at du skal kunne dem uten å ha sett dem									
169	Intervjuer	Men dere skal ikke lage en så vanskelig kode altså. Vi skal bare øve litt på å starte fra bunnen av									

170		(elevene leser oppgaveteksten)									
171	Intervjuer	Når man skal starte med en slik oppgave finnes det noen gode tips. Hvis du for eksempel tenker deg et fysisk kast av en terning. Hvor mye av den fysiske situasjonen er viktig å få med seg?									
172	1B	Hva sa du nå?									
173	Intervjuer	Når man kaster en terning kaster man jo terningen med en viss fart fra en viss høyde før den treffer et bord før det viser hva man får. Hvor mye av dette er viktig å få med seg i programmet?									
174	1A	Nei									
175	Intervjuer	Hva er det essensielle å få med									
176	1B	Altså vi vil jo bare ha med tallet		PRO	MOD		REP				Abstraksjon
177	Intervjuer	Det å ta bort sånn unødvendig informasjon kaller vi abstraksjon i programmering									
178	Intervjuer	Da kan neste steg være å prøve å gjøre problemet litt enklere for oss selv. Å hoppe rett inn i et program med fem terninger kan være litt vanskelig. Kan dere prøve å gjøre det problemet litt mindre?									
179	1B	Vi kan begynne med en terning kanskje?		PRO	MOD						Dekomponering
180	1A	Så kan vi heller legge på flere kast		PRO	MOD						Dekomponering
181	Intervjuer	Det kan være lurt ja. Så ser dere at dere får det til før dere prøver på det store problemer									
182	Intervjuer	Så nå har dere de kodene fra forrige oppgave til hjelp, så kan dere prøve å lage koden til ett terningkast									

183	1A	Det eneste nå er at nå er det terningkast og ikke myntkast										
184	1B	Ja ja										
185		...										
186	1A	Hva hvis vi hadde tatt terningkast= også sånn tilfeldig tall?										Koding
187		...										
188	1A	Hvis vi hadde begynt med terningkast= (1,6) Da er jo alle tallene med. Vi prøver sånn da										Koding
189	1B	Men hvis du ser på den så hadde vi med sånne hermetegn, burde vi ha med det?(Peker på koden som definerte mynten tidligere)										Koding
190	1A	Vi kan prøve										
191	1B	Eller er det liksom definert der nå?										Algoritme
192	1A	Prøv da, så ser vi		PRO								
193		(Skriver inn kode)										Koding
194	1B	Men da burde vi jo egentlig definere kast										Koding
195		..										
196	1B	Random choice										Koding
197	1A	Ja, ut i fra den (peker på terningen de har definert)										Algoritme
198		(legger til random.choice til koden)										
199	1A	Var det sånn?										
200	Intervjuer	Og hva mangler nå for å få vist det?										
201	1B	Print										Evaluering
202	1B	Hva skal vi skrive										
203	1A	Print. Ehh. Print. Hva skal vi printe for noe										Koding
204	Intervjuer	Hva i den koden er det egentlig du vil vise?										
205	1B	Vi trenger jo bare et tall										Algoritme
206	Intervjuer	Ja										
207	1B	Men føler jeg burde skrive en setning										

208	Intervjuer	Du velger selv. Du skriver bare akkurat det du vil at pcen skal vise									
209	1A	kast									
210	Intervjuer	Ja, for eksempel									
211		(Tester koden, men det kommer ikke opp noe)									
212	1B	Nå kom det jo ingenting, men det er jo ikke feil									Evaluering
213		...									
214	Intervjuer	Men hva er det pcen velger mellom?									
215	1B	Må vi ha hermetegn rundt en, to, tre, fire, fem, seks?									Evaluering
216	Intervjuer	Prøv det ja, for nå definerte vi jo bare terningkast som 1,6									
217	1A	ahhhhh									
218		(Skriver nye kode)									Koding
219	1B	Dette var mye for meg nå									
220		(Kjører den nye koden)									
221	1A	Kast, ja det var jo det vi ville den skulle printe									Evaluering
222	1B	Ja, der									
223		(tester koden flere ganger)									
224	Intervjuer	Gjør koden det dere ville den skulle gjøre?									
225	1A	Ja, da har vi ett kast, har vi ikke det?			MOD		REP				Evaluering
226	1B	Ja, ett kast ja. Så er det bare å få flere kast									
227	1B	Må vi liksom sette opp flere kast? Liksom kast 2, kast 3?			MOD						Algoritme
228	Intervjuer	Dere kan jo prøve på det. Er du enig i den fremgangsmåten 1A?									
229	1A	Ja									
230		(Lager den nye koden ved å bruke kopier og lim funksjonen)			MOD						Koding
231	1B	Var det fem vi skulle ha?									
232	1A	Ja									

233	1A	men nå må vi jo endre noe på den. (Peker på print)										Koding
234	1B	Ja, vi må endre noe på print. Men hva? Må vi liksom skrive print på alle nå?										
235		(Legger alle kastene inn i printfunksjonen)										Koding
236		(elevene tester koden)										
237	1B	Dette ble helt feil			MOD							Evaluering
238	1A	Ja										Evaluering
239	Intervjuer	Hvorfor mener dere det?										
240	1B	Nei nei, det ble jo ikke det. Den viser alle tallene			MOD		REP					Evaluering
241		(Tester koden)										
242	Intervjuer	Er dere fornøyd med det dere har nå eller vil dere pynte litt på det?										
243	1B	Hadde det vært bedre å ha det under hverandre?										
244	Intervjuer	Enten det eller så kan dere legge inn litt tekst										
245	1B	Jeg er egentlig veldig fornøyd										Evaluering
246	Intervjuer	Ja, det er jo en fin kode										
247	1B	ja, du skjønner jo at det er de tallene. Nå er det jo egentlig bare pirk for å få det til å se bra ut										Evaluering
248	Intervjuer	Ja, da er det veldig lov å si seg fornøyd. Koden viser jo det den var ment til										
249	1A	Ja										
250	Intervjuer	Så bra. Bra jobbet. Håper dere har lært noe om hvordan man kan lage en kode fra bunnen av										
251	1A	Ja det var gøy å prøve										
252	1B	Jeg synes egentlig det var litt lærerikt										
253	Intervjuer	Så bra. Takk for at dere ville være med										

8.5.2 Kode fra oppgave 2

```

1 import random
2
3 terning = ["1","2","3","4","5","6"]
4 kast = random.choice (terning)
5 kast2 = random.choice (terning)
6 kast3 = random.choice (terning)
7 kast4 = random.choice (terning)
8 kast5 = random.choice (terning)
9
10 print(kast+kast2+kast3+kast4+kast5)
11
12

```

8.6 Datamateriale gruppe 2

8.6.1 Transkribering og analyse av lydopptak

1	Intervjuer	Da skal vi prøve å programmere litt, har dere gjort mye av det før?											
2	2B	2A har gjort det litt før											
3	2A	Ja, jeg gjorde det litt i TOF, men vi gjør ikke sånn programmering der [Python]											
4	Intervjuer	Nei, det er micro:bit der?											
5	2A	Ja											
6	Intervjuer	Men da skal vi se om får lært noe python											
7	2B	Ja											
8	Intervjuer	Da kan dere bare starte med å lese den første oppgaven											
9		...											
10	Intervjuer	Vi kan jo starte med om dere skjønner hva de forskjellige linjene betyr											
11	2B	Den første gjør liksom at mynt har to ting			MOD		REP						Evaluering
12	Intervjuer	Ja, det stemmer. Skjønner dere hva den neste linja gjør?											
13	2A	Den velger et tilfeldig alternativ?			MOD		REP						Evaluering
14	Intervjuer	Ja, den velger et tilfeldig alternativ fra mynten. Så dere kan jo prøve å kjøre koden et par ganger for å se hva som skjer											

15	2B	Var det den? [command I]									
16	Intervjuer	Ja									
17	2A	Mynt									
18	Intervjuer	Så da er jo det viktige spørsmålet: Er det noe fordel med å bruke programmering i dette eksempelet her?									
19	2B	Nei, det har vel ikke så mye å si			MOD		REP				
20	2A	Det går jo mye fortere å bare kaste			MOD		REP			HJE	
21	Intervjuer	Ja, så det vi skal prøve på videre nå er å utvikle den til noe som er en fornuftig kode å bruke, så dere kan jo gå videre til neste oppgave									
22		...									
23	Intervjuer	Har dere noen tanker her?									
24	2B	Vi gjorde jo noe av det samme i forrige time, men da hadde vi jo femten linjer eller noe									
25	Intervjuer	Ja, vi kommer dit etterhvert. Men nå tar vi det litt fra bunnen av for å lære å bygge opp koden									
26	2A	Veldig lurt, for det kan jeg ikke fra før av								HJE	
27	Intervjuer	Ja, det er jo det som er målet her. Å se litt hva som ligger bak [kodene]									
28		..									
29	Intervjuer	Hvis man ser på den linja som definerer mynten, trenger vi å lage to av den?									
30	2B	Hva mener du?									
31	Intervjuer	Hvis vi ser på begge linjene i koden, må vi lage dobbelt opp av alle?									
32	2A	Var det det der med 1,5?									
33	2B	I den forrige koden hadde vi 1,5 eller noe greier i en parantes									

34	Intervjuer	Vi prøver å gjøre det litt lettere for oss selv. Kan dere klare å lage ett til kast?									
35	2A	Ja det kunne vi jo									Algoritme
36		(Skriver inn ny linje i koden)									Koding
37	2A	Er det sånn at man må ha mellomrom imellom?									
38	Intervjuer	Neida, det bare ser litt ryddigere ut									
39		...									
40	2B	Nei, det funket ikke det									
41	Intervjuer	Nei, for hva er det nå dere hatt gitt beskjed til pcen om å gjøre									
42	2A	ehhm, ja. Vi må ta print også kast 2 også									Evaluering
43	Intervjuer	Det høre ut som en god ide									
44		(skriver inn ny linje i koden)									Koding
45	2A	sånn ja									
46	2B	Der									
47	Intervjuer	Så programmet vårt fungerer, men vi får jo ikke alltid 50-50									
48	2B	Nei									
49	Intervjuer	Men da kan dere gå videre til neste oppgave									
50		...									
51	Intervjuer	Nå skal vi se hva som skjer om vi legger til litt flere kast									
52	Intervjuer	Har dere en strategi på hvordan dere kan finne feilen i en slik kode?									
53	2A	Se hvor det er feil. Det står jo ofte hvilke linje det er på		PRO						HJE	Evaluering
54	2B	Det er den der femti, den der while									Evaluering
55	2A	Linje sju ja									Evaluering
56	Intervjuer	Vi kan jo se om vi skjønner hva alle linjene gjør. Skjønner hva de øverste der gjør									
57	2A	De gjør sånn at kron og mynt starter på null			MOD						

58	Intervjuer	ja, stemmer. Den neste da?											
59		...											
60	Intervjuer	Skjønner at den kanskje er ny. Det den gjør er at så lenge vi har mindre enn femti kast, så skal vi kaste en gang til											
61	2B	okei											
62	Intervjuer	Når vi da har nådd femti, så skal vi ikke kaste flere ganger											
63	Intervjuer	men vi har altså fått opp at vi sliter litt på den linja her											
64	2B	Må man ikke definere kast?											Algoritme
65	Intervjuer	Det høres lurt ut, for har vi egentlig gjort det?											
66	2A	Nei											
67	2A	Er det samme som sist da? Må vi definere det med <code>random.choice</code>											Koding
68	2B	Må vi ikke skrive kast her oppe?											Koding
69	Intervjuer	Ja, vi har jo sagt at vi skal stoppe på femti, men har vi sagt hvor den skal starte?											
70	2B	Nei											
71	2A	Må det ikke være <code>kast= 0</code> da?											Koding
72	Intervjuer	Dere kan jo prøve											
73		(Skriver inn ny linje i koden)											Koding
74		...											
75	Intervjuer	Så det dere kan gjøre om dere hadde fått en sånn oppgave, er å gå gjennom hver variabel hver for seg, og så se om dere er fornøyd med det den gjør med hver av dem											
76	2B	mhm											
77	Intervjuer	men dere kan jo kjøre denne et par ganger og se hva dere får											
78		...											
79	2A	30 og 20											
80		..											
81	2A	24 og 26											

82	Intervjuer	Så der er vi jo inne på noe?									
83	2B	Ja, blir mye det samme									
84	Intervjuer	Men av og til er vi litt fra 50-50. Men synes dere allikevel at sannsynlighet er en fornuftig måte å anslå virkelige hendelser på?									
85	2B	Nei det er jo det jo ikke, det er jo ikke sikkert	TAN		MOD	RES				KOM	
86	2A	Det vil jo ikke være helt nøyaktig	TAN		MOD	RES				KOM	
87	Intervjuer	Nei, dere ville ikke veddet penger på at etter 50 kast havnet dere på 25 hver									
88	2A	Nei									
89	2B	Nei									
90	Intervjuer	Men så sier vi jo allikevel i matte at det er 50% sannsynlighet for begge. Så kunne vi gjort noe med koden for å vise at matten ikke er helt gal									
91	2B	Øke hvor mange kast	TAN		MOD						
92	Intervjuer	Har du noen tanker på hvorfor det er lurt?									
93	2B	Nei altså, jo flere kast man har, altså så vil det ha mindre å si om det er en forskjell eller ti forskjell	TAN		MOD	RES		SYM		KOM	
94	Intervjuer	Ja, høres fornuftig ut. Så da kan dere jo prøve med å endre koden									
95	2A	Skal vi øke til 500 for eksempel?									
96	2B	Ja									
97	2A	Eller var det litt mye?									
98	2B	Nei									
99	Intervjuer	Det er en pc, så det er ingen som skal sitte og gjøre dette									
100	2B	49.									
101	Intervjuer	Dere kan jo kjøre den et par ganger til									
102	2B	51.									
103	2A	Skal vi prøve å øke litt til?									
104	2B	Ja									
105	Intervjuer	Ja, vi er jo litt bedre enn sist									

106	2A	Til tusen!									
107		...									
108	2A	Nå ble det jo veldig mye jevnere									
109	Intervjuer	Dere kan jo legge til enda noen nuller									
110	2B	50,05.									
111	Intervjuer	så vi ser at ideen med å legge til flere kast var veldig god. Men kunne dere oppsummert det koden har vist oss til nå med en setning?									
112	2A	Det er veldig tungvint å kaste terningen 100 000 ganger, så dette er et godt verktøy til det			MOD		REP			HJE	
113	2B	Og at vi må kaste mange ganger før den havner på 50 [%]	TAN		MOD	RES		SYM	KOM		
114	Intervjuer	Ja, det er jo veldig fint, men her har vi jo fått litt hjelp, så nå tenkte jeg at vi skulle prøve å lage en kode fra bunnen av. Har dere gjort det mye?									
115	2B	Nei									
116	2A	Ikke uten en mal å se på								HJE	
117	Intervjuer	Men da skal vi prøve å lære mer om det, så da kan dere lese gjennom den neste oppgaven									
118		...									
119	Intervjuer	Skjønner dere hva målet er?									
120	2A	Ja									
121	Intervjuer	Så da starter vi litt fra scratch. I et virkelig terningkast så er det ganske mye som skjer. Du skal kaste en terning med en viss fart, før den treffer et bord der det kanskje står andre ting også, før du skal se hvilken siden av terningen som peker opp, hvor mye av disse tingene fra kastet er viktig å få med i koden?									

122	2B	Ingenting		PRO	MOD							Abstraksjon
123	Intervjuer	For målet med koden										
124	2A	Er jo bare utfallet		PRO	MOD							Abstraksjon
125	Intervjuer	Ja, så det kan være lurt å ta med når dere skal lage koder. Det er ofte mye man kanskje har lyst til å ta men som egentlig ikke spiller noen rolle										
126	2B	Ja										
127	Intervjuer	Så er ofte neste steg å bryte ned oppgaven. For mange programmeringsoppgaver kan være veldig store, så det kan være lurt å bryte ned problemet til noe som er litt enklere å løse og som vi heller kan bygge videre på. Kan dere gjøre det her?										
128	2A	Vi kan jo begynne med 1		PRO	MOD							Dekomponering
129	Intervjuer	Ja, dere kan jo prøve på det.										
130	2A	Da kan vi jo begynne med å definere dem da		PRO	MOD							Algoritme
131	2B	Ja										
132	2A	De ulike tingene										
133	2B	Men der står det jo ikke noe om hvor mange kast [i koden fra 1A]										
134	2A	Men da kan jo bare ta flere			MOD							Algoritme
135	2B	Kan jo bare ta fem kast ja										Algoritme
136		..										
137	2B	Definerer vi sånn 1, 2, 3, 4, 5, 6?										Algoritme/koding
138	Intervjuer	Dere kan jo bare teste										
139	2A	Skal jeg skrive eller du?										
140	2B	Du kan gjøre det du										
141	2A	Okei										
142		...										
143	2A	Hva var det vi begynte med										
144	2B	Mynt... Eller terning										
145	2A	Den må jo inneholde seks forskjellige sider										Algoritme
146	2B	Måtte man ikke ha 1,6 da?										

147	Intervjuer	Dere kan se på de andre kodene om dere trenger litt inspirasjon										
148		...										
149	2B	Skal vi prøve å skrive sånn 1, 2, 3,4,5										Koding
150	2A	Ja, det kan vi										
151	Intervjuer	Så må dere bare huske ha dem inne i hermetegn										
152		(definerer terningen i koden)										Koding
153	2B	Må det være komma imellom?										Koding
154	Intervjuer	Ja, så koden skjønner hva den skal skille mellom										
155		...										
156	2A	Så skal vi gå ned på kast										
157	2B	random.. Jeg husker ikke										Koding
158	Intervjuer	Dere kan se på den forrige koden, jeg forventer ikke at dere skal forstå hva dere skal velge med en gang										
159	2B	Random.choice										koding
160	2A	men kan vi bare skrive ett av tallene der?										Koding
161		..										
162	2A	Nei, bare glem det										
163	2B	Også mynt, nei terning mener jeg										
164		...										
165	2B	Bli det riktig?										
166	Intervjuer	Dere kan jo teste og se hva som skjer										
167	2A	Nei										
168	2B	Der skjedde det ikke mye										Evaluering
169	Intervjuer	Nei, for hva har vi gitt pcen beskjed om å gjøre nå?										
170	2A	ja, vi må jo trykke print				MOD						Evaluering
171	2B	åja										
172	2A	Også kast blir det jo da										
173		(Legger til en printfunksjon i koden)										koding
174	2A	Ja										

175	Intervjuer	Det ser jo bra ut. Så vi klarte å lage en terning. Da kan vi gå videre til det originale problemet									
176	2B	ja, fordi vi skal jo ha fem kast									
177	2A	Men det må jo finnes en lettere måte enn å skrive kast fem ganger			MOD					HJE	Koding
178	2B	Ja, det var det jeg tenkte på									
179	2A	Men skal vi prøve å skrive kast fem ganger først? Og se om vi ser noe annet etterpå		PRO	MOD					HJE	Algoritme
180		(Kopierer kastfunksjonen fire ganger)									Koding
181	2B	Skal vi prøve?									
182	2A	Ja, så må vi også ha at den printer ut alle									Koding
183	2B	Ja, okei									
184		(skriver inn flere kast i printfunksjonen)									Koding
185	2A	ja, det er vel ikke sånn side på noen terninger?									Evaluering
186	Intervjuer	Hvis du kjører den en gang til									
187	2A	åja, åja, sånn ja									
188	Intervjuer	Er dere fornøyd									
189	2B	Jeg er veldig fornøyd			MOD						Evaluering
190	2A	Det går jo an å forstå det liksom, det så jo bare litt knotete ut			MOD						Evaluering
191	Intervjuer	Det går jo an å pynte litt på den. Dere kan jo skrive mer inn i printfunksjone. Om dere vil skrive noe tekst er det bare å ha det inni hermetegn									
192	2A	Må vi skrive sånn print1 print2 print3									
193	Intervjuer	Dere kan det, eller for eksempel skille med en bindestrek mellom tallene									
194	2A	Ja									
195	2B	Ja									
196	2A	Var det hermetegn du sa?									

197	Intervjuer	Ja, så må dere bare skille med komme mellom som i den andre koden									
198		(Legger inn bindestrek mellom hvert av kastene)									Koding
199	2A	Der ja, det var veldig mye penere									
200	Intervjuer	Så det å pynte på koden er jo noe man alltid kan gjøre om man vil når man har fått koden til å fungere									
201	2A	Ja									
202	Intervjuer	Har dere nå fått koden til å gjøre det dere vil den skal gjøre?									
203	2A	Ja			MOD						Evaluering
204	2B	Ja			MOD						Evaluering
205	Intervjuer	Og lærte dere noe om prosessen også?									
206	2B	Ja, noen av stegene før									
207	Intervjuer	Så føler dere koding er brukbart i matte									
208	2A	Ja, sannsynlighet er jo matte, og vi kan jo se på det								HJE	
209	2B	Så kan vi også lage koder til å se for eksempel den halverte								HJE	
210	2A	Det har vi gjort									
211	Intervjuer	Så bra. Jeg synes dere gjorde en god jobb. Takk for at dere ville være med									

8.6.2 Kode fra oppgave 2

```

1 import random
2 terning=["1","2","3","4","5","6"]
3 kast=random.choice(terning)
4 kast2=random.choice(terning)
5 kast3=random.choice(terning)
6 kast4=random.choice(terning)
7 kast5=random.choice(terning)
8
9 print(kast,"-",kast2,"-",kast3,"-",kast4,"-",kast5)
10

```

8.7 Datamateriale gruppe 3

8.7.1 Transkribering og analyse av lydopptak

1	Intervjuer	Har dere hatt mye programmering før?									
2	3A	Nei									
3	3B	Ikke så mye									
4	Intervjuer	Så det er fortsatt litt nytt									
5	3B	Ja									
6	Intervjuer	Da kan vi jo bare starte med første oppgave, så har jeg samme kode der på pcen å dere kan teste den									
7		..									
8	Intervjuer	Vi kan jo starte med å se om dere vet hva de ulike linjene betyr. Skjøner dere for eksempel hva den første her gjør?									
9	3B	Der definerer du bare kron og mynt			MOD		REP				Evaluering
10	3A	Du definerer variablene på den måten			MOD		REP				
11	Intervjuer	Ja, det er helt riktig. Og den neste da? Der har dere kanskje ikke brukt kommandoen før, men dere ser kanskje hva det går i?									
12	3A	Den velger enten kron eller mynt			MOD		REP				Evaluering
13	Intervjuer	Ja, helt riktig. Så dette er jo en ganske enkel kode. Spørsmålet er jo da om det egentlig er noe poeng i å bruke programmering her?									
14	3B	Det tar jo mindre tid å bare kaste en mynt			MOD		REP			HJE	
15	Intervjuer	Da er det kanskje enklere å bare kaste?			MOD		REP			HJE	
16	3A	mhm									

17	Intervjuer	Målet med resten av oppgaven er derfor å lage en kode der det er et poeng i å ta ibruk simulering. Så da kan dere jo gå videre til neste oppgave									
18		...									
19	3B	Var det sånn å lage kast1 og kast2?		PRO	MOD						Algoritme
20	Intervjuer	Det kan dere jo prøve på									
21	3B	Skal vi bare gjøre det her [På pcen]									
22	Intervjuer	Ja, for målet med opplegget er jo at dere skal prøve å lage litt koder selv									
23	3B	Skal vi ta sånn kast1 kast2 og se?									Algoritme
24	3A	Ja									
25	3B	Så må vi vel ha noe sum også kanskje?									Koding
26		(skriver inn et nytt kast i koden)									Koding
27	Intervjuer	Nå funket det jo ikke. Hva er det egentlig dere har fortalt pcen å gjøre?									
28	3A	At den skal kaste flere ganger									Evaluering
29	Intervjuer	Ja, og pcer gjør jo akkurat som vi sier									
30	3A	Ja									
31	3B	Var det den printgreia?		PRO							Koding
32	Intervjuer	Ja									
33	3A	Bli det kast1+kast2?									Koding
34	Intervjuer	Dere kan jo prøve på det									
35		...									
36	3A	Sånn ja									Evaluering
37	Intervjuer	Da kan dere jo kjøre koden en par ganger og se hva dere får									
38		...									

39	Intervjuer	Vi fikk jo nå ikke like mange kron som mynt, men vi vet jo fra matten at det skal være like stor sannsynlighet fra begge									
40	3A	Ja									
41	Intervjuer	Så da kan dere bare lese den neste oppgaven									
42		...									
43	3A	mhm									
44	Intervjuer	Dere kan jo prøve koden og se at den ikke virker									
45	3B	Det er linje sju står det								HJE	Evaluering
46	Intervjuer	Ja, det er der det står at det er feil, men vi kan jo starte med å se om vi forstår koden. Klarer dere å forstå hva linje 4 og 5 gjør?									
47	3A	Ja, kron og mynt. Den definerer bare dem liksom?			MOD						
48	Intervjuer	Ja, den forteller oss at kron og mynt tellingene starter på null. Klarer dere å forstå hva den neste er? Den er kanskje ny									
49	3B	Ja, tror vi bruker noe litt annet, men er det at den tar 50 kast liksom?			MOD		REP				
50	Intervjuer	Ja, så lenge vi har mindre enn 50 kast, så kaster vi en gang til									
51	3A	Ja									
52	Intervjuer	Den siste bolken da?									
53	3A	Nå velger den random tall, 1 eller 2?			MOD		REP				
54	Intervjuer	Ja, så legger den til en på kron eller mynt telleren vår. Så da skjønner dere koden?									
55	3B	Ja									
56	Intervjuer	Så da kan dere jo se om dere kan rette den.									

75	Intervjuer	Men vi ville ikke veddet penger på dette enda?									
76	3A	Nei									
77	3B	Nei									
78	Intervjuer	Da kan dere jo gå videre til neste oppgave. Kunne dere svart med en setning der?									
79		...									
80	3A	Hvis man tar det mange ganger får man jo en anelse på hva det skal være, men det er jo helt tilfeldig i kastene	TAN		MOD	RES		SYM	KOM		
81	Intervjuer	Så det er en veldig teoretisk greie med sannsynlighet									
82	3A	Ja									
83	Intervjuer	Da kan dere gå videre til neste									
84		...									
85	3B	Hvis vi tar mange flere kast									
86	Intervjuer	Har du noen tanker på hvorfor det er sånn?									
87	3B	Da er det jo større sjans for at man ikke bare ikke hadde flaks, at det blir mer riktig på en måte	TAN			RES		SYM	KOM		
88	Intervjuer	Så de små forskjellene har ikke så mye å si lengre?									
89	3A	Ja	TAN								
90	3B	Skal vi bytte [antall kast i koden]									Koding
91	Intervjuer	Det kan dere bare									
92	3A	Hvis vi tar 1000									
93	3B	499 og 501									
94	Intervjuer	Da begynner det jo å ligne på noe									
95		..									
96	Intervjuer	Nå får dere jo ganske store tall, kunne dere endret koden for å gjøre det lettere å se andelen dere får?									

97		...									
98	3B	Noe med at du deler dem på hverandre		PRO							
99	3A	Ja, du tar n/N. Var det delt på antall kast?	TAN	PRO							
100	Intervjuer	For hva er egentlig n og N her i vår kode?									
101	3A	Vi må jo definere det									
102	3B	Kast delt på ehh									
103	Intervjuer	Hvilken av dem vil man egentlig ha som kast?									
104	3B	Stor N									
105	3A	Stor N									
106	Intervjuer	Ja, og liten N er da?									
107	3B	Enten kron eller mynt									
108	Intervjuer	Ja, det er jo det vi kaller gunstig utfall.									
109	3A	Da kan vi jo bare skrive Kron/kast da									
110		(Skriver inn ny kode)									
111	3B	0.51 .. 0.51 ..									
112	Intervjuer	Dere kan jo også legge til noen flere kast, det er jo en pc tross alt									
113		(Endrer antall kast til 100 000)									
114	3B	Nå begynner det å bli ganske riktig med 50									
115	Intervjuer	Ja, nå er vi inne på noe									
116	3A	mhm									
117	Intervjuer	Ja, så hva vil dere nå si om det å bruke simulering som et verktøy?									
118	3A	Her ser man jo sannsynligheten mye lettere når man skal ta så mange kast					REP			HJE	
119	Intervjuer	Ja, det hadde jo tatt lang tid å kaste den 100 ganger. Men vil dere si at dere har lært noe matematikk her?									

120	3A	At det kanskje ikke blir riktig med bare ett eller to kast, at man må ha flere fordi sannsynligheten. Det er jo bare sannsynlighet. Ikke noe sikkert.	TAN			RES		SYM	KOM		
121	Intervjuer	Ja, er du enig i den påstanden?									
122	3B	Ja									
123	Intervjuer	Da kan vi gå videre til ned neste oppgaven, for jeg har forstått at dere ikke pleier å lage koder fra bunnen av?									
124	3B	Nei									
125	Intervjuer	Skjønner, for det er det vi skal prøve å lære litt mer om her.									
126		...									
127	Intervjuer	Det som kan være lurt å starte med når man lager en kode er å se hvor mye av den faktiske situasjonen man skal ha med i koden									
128	3A	Ja									
129	Intervjuer	Når man kaster en terning skal man for eksempel kaste den med en viss fart fra en viss høyde før den treffer et bord og ruller bortover og viser et tall. Hvor mye av denne situasjonen er det viktig å få med i simuleringen?									
130	3B	Tallet som kommer opp		PRO	MOD		REP				Abstraksjon
131	3A	mhm									
132	Intervjuer	Ja, er det noe annet vi vil ha med?									
133	3B	Nei		PRO	MOD						Abstraksjon
134	3A	Nok terninger		PRO	MOD		REP				Abstraksjon

135	Intervjuer	Ja, det er jo viktige ting å ta med. For alt det andre spiller jo ingen rolle, vi vil jo bare ha et tilfeldig tall. Da har vi jo der luket bort ganske mye vi ikke trenger å jobbe inn i koden									
136	3B	mhm									
137	Intervjuer	et fint neste steg kan være å bryte ned problemet litt til noe vi kan bygge videre på siden mange problemer kan være vanskelig å hoppe rett inn i									
138	3B	Ja									
139	Intervjuer	Har dere en tanke på hvordan dere kunne gjort det her?									
140	3A	Kan jo begynne med å bare ta en terning		PRO	MOD						Dekomponerin
141	Intervjuer	Ja, det er jo et godt eksempel. Da kan dere bare prøve, så har dere jo de forrige kodene som inspirasjon									
142	3B	Det blir vel noe av det samme bare at vi kan ha seks tall i stedet?		PRO	MOD						Algoritme
143	3A	Ja									
144		..									
145	3A	Skal vi definere den som 0 til 6 da									Koding
146	3B	Ja, eller 1 til 6									Koding
147	3A	Ja, 1 til 6									Koding
148	3B	Skal vi ta og definere terning da? For vi skal vel ikke ha noe annet. Eller vi må jo definere den som noe									Koding
149	3A	ja, ehmm vi kan jo komme tilbake til det									
150	Intervjuer	Hva er det egentlig dere vil at terningen skal være?									
151	3B	Bare tall egentlig									Algoritme

152		..									
153	3B	Skal vi skrive terning er lik 1, 2,3 ,4 5,6 liksom det forskjellige den kan bli?									Koding
154	3A	Ja, og random.choice 1-6									Koding
155		(Skriver inn kode)									Koding
156	Intervjuer	Så må dere få med de hermetegnene så pcen skjønner det									
157	3A	Ja									
158		(Fortsetter å skrive koden)									Koding
159	3B	Da blir det sånn da									
160	3A	Ja, vi skal jo bare ha ett kast nå									
161		...									
162	3B	Så må vi printe noe									Koding
163		(legger til en printfunksjon i koden)									Koding
164	3A	Tallet blir 4									
165	Intervjuer	Ja, da kan dere jo kjøre koden et par ganger for å se at vi får litt forskjellig									
166	3A	5. 2. 1. 3									
167	3B	Da har vi ett kast med en terning					REP				Evaluering
168	Intervjuer	Så koden gjør det dere ville?									
169	3A	Ja			MOD		REP				Evaluering
170	3B	Ja			MOD		REP				Evaluering
171	Intervjuer	Så bra, da har vi sett at vi klarer det. Så da kan vi bygge videre til det originale problemet									
172	3A	Da må vi vel ha fem terninger da?		PRO	MOD						Algoritme
173	3B	Skal vi bare skrive den om og om igjen da?		PRO						HJE	Algoritme
174	3A	Ja									
175		(Legger til flere kast i koden)								HJE	Koding
176	3A	Så må vi legge sammen til en sum? Eller									

177	3B	Vi er jo ute etter hver av dem									
178	3A	Ja									
179		(Legger inn alle kastene i printfunksjonen									Koding
180	3B	Skal vi se hva som skjer									
181		..									
182	3B	4 1 2 3 5									
183		..									
184	3B	Det funket jo veldig fint			MOD						Evaluering
185	3A	mhm									
186	Intervjuer	Gjør koden det dere vil den skal gjøre?									
187	3B	Ja, fem kast			MOD	REP					Evaluering
188	3A	ja			MOD	REP					Evaluering
189	Intervjuer	Det er jo veldig bra									
190	3B	Hvis vi vil ta vare på noen da?									
191	Intervjuer	Det er jo en veldig fin måte å bygge videre på herfra. Men den koden blir veldig komplisert, så den får vi ikke til å lage her nå									
192	3A	Okei									
193	Intervjuer	Men det hadde vært en veldig naturlig vei videre									
194	3B	Ja									
195	Intervjuer	Veldig bra. Hva synes dere egentlig om programmering som hjelpemiddel i matte?									
196	3A	Det er jo veldig kult, men veldig vanskelig å lære seg synes jeg. Ihvertfall de lange kodene								HJE	
197	Intervjuer	Ja. Men det er jo veldig vanlig å måtte søke opp kommandoer, så det er ikke forventet at dere må kunne alle altså.									
198	3A	mhm.									
199	3B	Så det er egentlig bare å søke seg litt opp									

200	Intervjuer	ja, det er ofte det man må gjøre egentlig. Er dette noe dere kunne tenke dere å ha mer av i mattetimene?									
201	3B	Så lenge det ikke er for vanskelig. Noe av det kan man jo tenke seg til								HJE	
202	3A	Så er det jo sikkert noe som blir veldig sentralt i fremtiden								HJE	
203	Intervjuer	Ja, det kommer jo godt med i mange yrker									
204	3A	mhm. Så å få litt grunnlag skader jo ikke det								HJE	
205	Intervjuer	Veldig bra. Da stopper jeg opptaket. Takk for at dere ville være med									

8.7.2 Kode fra oppgave 2

```

1 import random
2
3 terning = ["1","2","3","4","5","6"]
4
5 kast1 = random.choice(terning)
6 kast2 = random.choice(terning)
7 kast3 = random.choice(terning)
8 kast4 = random.choice(terning)
9 kast5 = random.choice(terning)
10
11
12
13 print ("Tallet ble", kast1, kast2, kast3, kast4, kast5)
14

```

8.8 Datamateriale gruppe 4

8.8.1 Transkribering og analyse av lydopptak

1	Intervjuer	Har dere hatt noe programmering fra før?									
2	4A	En god del									
3	4B	Vi har jo fra ToF da									
4	Intervjuer	men da har dere jo et godt grunnlag									
5	4B	Men vi er ikke noe gode i det									
6	4A	Nei									

7	4B	Vi hadde ikke Python i ToF									
8	Intervjuer	Stemmer, det er micro:bit der?									
9	4A	Men jeg har brukt Python en del i fysikk									
10	Intervjuer	Det høres jo spennende ut da									
11	4A	Men jeg husker ingenting fra det									
12	Intervjuer	Det er kanskje vanskelig å lage kodene selv i fysikk									
13	4A	Ja, det var veldig vanskelig									
14	Intervjuer	Men målet idag er ihvertfall å prøve å lage litt koder selv, og forstå litt hva som ligger bak. Så dere kan jo starte med å lese den første oppgaven									
15		...									
16	Intervjuer	Dere kan jo prøve koden, så ser dere at kastet kommer opp her nede									
17		(elevene tester koden)									
18	Intervjuer	Vi kan jo starte med å se om dere skjønner linjene i koden. Skjønner dere hva de forskjellige linjene betyr?									
19	4B	Ja									
20	4A	mhm									
21	Intervjuer	Forstår dere for eksempel linje 3?									
22	4B	Linje 3 viser jo bare mulighetene, og i linje 4 skal den jo velge en tilfeldig av dem			MOD	REP					Evaluerings

23	Intervjuer	Ja, så mye her var kanskje litt kjent. Så er jo spørsmålet videre om det egentlig er noe poeng i å lage en kode her?									
24	4B	Det hadde jo gått fortere å bare kaste den			MOD		REP			HJE	
25	4A	det er når man skal kaste flere at man trenger kode			MOD		REP			HJE	
26	Intervjuer	Så her er det enklere å bare flippe en mynt?									
27	4B	ja									
28	Intervjuer	Da er jo målet med oppgaven å til slutt få en kode dere mener er fornuftig. Så da kan dere gå videre til neste oppgave									
29		...									
30	4A	To myntkast ja									
31	Intervjuer	Ja, nå prøver vi å utvide den litt									
32	4B	he he, jaaaa									
33		..									
34	4B	At den skal kaste to mynter samtidig?									
35	Intervjuer	Ja, at den skal vise resultatet av to myntkast									
36	4A	ja, okei									
37		..									
38	4A	Må vi ikke bare legge til en til mynt?		PRO	MOD						Algoritme
39	4B	Kan jo bare lage en mynt 2 og gjøre det samme?		PRO	MOD						Algoritme
40	4A	Så kan man vel gjøre sånn at man velger hvor mange ganger programmet skal kjøre gjennom									

41	Intervjuer	Det kommer vi til etterhvert, men dere kan jo bygge videre på ideen med en til mynt									
42	4A	Ja, hvis vi skriver inn et kast 2									Koding
43	4B	Bare lage akkurat samme?									
44	4A	ehhm, ja.. Bare gjør det									
45		(Elevene legger inn et nytt kast)									Koding
46	4B	Men du må ha med kast2 i den print også									Koding
47	4A	Hvordan legger man inn den?									
48	4B	Du må bare skrive komma inni der, så kast 2									Koding
49	Intervjuer	Ja, så kan dere jo kjøre koden et par ganger og se at den virker									
50	4B	Kron-kron, kron-kron,kron-mynt, mynt-mynt									
51	4A	Ja									
52	Intervjuer	Ja, det ser jo fint ut. Men her ser vi jo at ikke alltid får en av hver. Hvordan stemmer det med det dere vet fra matematikken?									
53	4A	Det er jo egentlig like sannsynlig	TAN							KOM	
54	Intervjuer	Ja, men her får vi jo ikke alltid helt det, men det skal vi undersøke videre. Så dere kan jo lese den neste oppgaveteksten									
55		...									
56	4A	Ja									
57	Intervjuer	Har dere noen strategi når dere skal igang med en slik oppgave									

58	4A	Burde se på det man vet er riktig		PRO								
59	4B	Feilmeldingen hadde jo jeg gått etter om det hadde kommet noe		PRO						HJE		Evaluering
60	4A	Ja, det ja										
61	Intervjuer	Det høres jo fornuftig ut, dere kan jo prøve og se										
62	4A	Linje 7, while kast										Evaluering
63		...										
64	Intervjuer	Så den sliter der, men vi kan jo ta det fra bunnen av og se at vi skjønner koden										
65	4B	Ja										
66	Intervjuer	Skjønner dere hva de to første gjør?										
67	4A	De derfinerer dem										Evaluering
68	4B	mhm										
69	Intervjuer	Ja, de definerer hvor vi skal begynne. Skjønner dere hva 7 og 8 gjør?										
70	4A	Mens den viser noe som er under 50 så skal den legge på en til?			MOD							Evaluering
71	4B	Så kaster den en til										Evaluering
72	Intervjuer	Ja, så lenge vi har mindre enn 50 kast, så kaster vi en til. Skjønner dere da hva den neste bolken gjør?										
73		...										
74	4A	Den velger et tall mellom en og to, og hvis det blir en så .. kaster den en gang til??										Evaluering
75	Intevjuer	For hva er det egentlig den har fått beskjed om å endre?										
76	4B	Kron-tallet										Evaluering
77	4A	Sånn ja										
78	Intervjuer	Og hvis det ikke blir 1?										

79	4B	Så blir det på mynttallet										Evaluering
80	Intervjuer	Ja, så vi lager en liste med kron og mynt. Da kan dere fortsette med å finne ut hva som er feil										
81		...										
82	Intervjuer	Det var nok en gode ide det å se på feilmeldingen										
83	4B	Det er fordi kast ikke er noe										Evaluering
84	4A	Kast er ikke definert										Evaluering
85	Intervjuer	Ja, nå er dere inne på noe										
86	4A	Skal vi lage en som heter kast da... Jeg vet ikke										
87	Intervjuer	For hva er det dere vil at kast skal være?										
88	4B	Kast skal være 50.. Eller opp til 50										Algoritme
89	Intervjuer	Ja										
90	4B	Kast begynner jo på null da			MOD							
91	4A	Ja, null til?										
92	4B	Bare null			MOD							
93		(Elevene definerer kast i koden)										Koding
94	4B	Kanskje det fungerer nå?										
95	Intervjuer	Det er bare å prøve										
96	4B	Ja										
97	4A	Ja, der fungerte det										Evaluering
98	Intervjuer	Og nå har vi riktig antall kast?										Evaluering
99	4A	Ja										
100	Intervjuer	Dere kan jo kjøre programmet et par ganger, så kan vi se hva vi får										
101	4B	29 og 21, 35 og 15, 24 og 26, 28 og 23										

102	4A	Ser ut til å stemme dette										Evaluering
103	Intervjuer	Og vil dere nå si at dette er et godt verktøy istedenfor å kaste?										
104	4B	Ja, absolutt									HJE	
105	Intervjuer	Men stemmer noe bedre med det vi vet fra matematikken?										
106	4B	Nei, det er jo litt tilfeldig fra hver gang vi prøver	TAN		MOD	RES					KOM	
107	Intervjuer	Det stemmer jo kanskje ikke helt med det vi tenker. Så da kan dere gå videre til neste side										
108		...										
109	4A	Nei, det kommer an på hvor stor										
110	4B	Det kommer an på antall	TAN			RES						
111	4B	Hvis man har veldig lave tall vil det ikke alltid stemme	TAN			RES					KOM	
112	Intervjuer	Ja, lave antall kast?										
113	4A	Ja										
114	Intervjuer	Det høres fornuftig ut. Da kan vi hoppe videre til neste										
115		...										
116	4B	Kan man ikke bare øke [Antall kast]		PRO								
117	4A	Ja, øke til 500										
118		(Endrer koden til 500 kast)										Koding
119	Intervjuer	Nå er vi jo litt nærmere 50%										
120	4B	Ja										
121	Intervjuer	Så dere kan jo teste den et par ganger										
122		...										
123		(elevene øker til enda flere kast)										Koding

124	4A	Nå er det nærme									
125	Intervjuer	Ja, nå er vi jo veldig mye nærmere. Har dere noe tanker om hvorfor det er sånn?									
126	4A	Det er jo sånn at det blir mer riktig beregning. Det kan jo bare være flaks de få gangene	TAN		MOD	RES			KOM		
127	4B	Ja									
128	Intervjuer	Ja, de små forskjellene har ikke så mye å si lengre. Så hvis vi skulle oppsummert hva vi har funnet ut i denne oppgaven, hva ville dere sagt da?									
129	4A	Du må ha mange forsøk om det skal bli mest mulig nøyaktig [50%]	TAN		MOD	RES			KOM		
130	Intervjuer	Det er jo veldig fint oppsummert. Så nå tenkte jeg dere skulle få prøvd dere på å lage koder helt fra bunnen av, så dere kan gå videre til neste oppgave									
131		...									
132	Intervjuer	Så det som er en god plass å begynne når man skal lage en kode er å se på den virkelige situasjon vi vil simulere. I et terningkast skal man jo kaste terningen med en viss fart og fra en viss høyde før terningen skal treffe et bord, trille bortover og ende opp med en av sidene opp. Hvor mye av dette føler dere at det er nødvendig å ha med i koden?									
133	4A	Veldig lite		PRO	MOD		REP				Abstraksjon
134	4B	Bare tallet		PRO	MOD		REP				Abstraksjon

135	Intervjuer	For poenget med koden er jo?									
136	4B	Å få opp noen tall			MOD		REP				Abstraksjon
137	Intervjuer	Ja, vi ønsker jo bare noen tilfeldige tall. Så vi kan luke bort en god del									
138	4A	Ja									
139	Intervjuer	Noe annet som er greit å gjøre er å se om vi kan bryte ned problemet litt. For det kan jo være vanskelig å hoppe rett i det store problemet. Så spørsmålet her er jo om vi kan bryte problemet ned til noe vi heller kan bygge videre på.									
140	4A	Kan jo lage et program som vi heller kan kjøre flere ganger	PRO	MOD		REP					Dekomponering
141	4B	Ja, der vi kaster en terning	PRO	MOD		REP					Dekomponering
142	Intervjuer	Det er jo en god ide. Da har dere litt inspirasjon fra de forrige kodene. Så da kan dere jo se om dere får det til									
143	4A	Vi kan se på den her [Den første koden]									
144		...									
145	4B	Vi må definere dem									Algoritme
146	4A	Vi kan jo skrive terning også 1, 2, 3, 4			MOD					HJE	Algoritme
147		(Definerer terningen i koden)									Koding
148	4A	Så må vi gå tilbake igjen og se her									
149	4B	ehh, så må vi bare ha kast da									
150	4A	Så må vi bare ha kast1, kast2, kast3			MOD					HJE	Koding/algoritme
151	4B	Så må vi ha en while-løkke								HJE	Koding

152	4A	Men vi kan bare skrive kast1, kast2, kast3 også									HJE	Koding
153	4B	nedover her?										
154	4A	Ja										
155	Intervjuer	Ja, så kan dere jo prøve med en først bare for å se at det virker										
156	4B	Ja, bare ta en da. Også de der greiene som stod i den andre. Den med random choice										Koding
157	4A	Ja, random.choice										Koding
158	4B	Også (terning)										Koding
159	4A	Sånn. Så må vi printe										Koding
160		(Legger til kast og print i koden)										Koding
161	4A	Linje 5 er det noe galt										Evaluering
162	4B	Da må vi bare gå tilbake og se										Evaluering
163		...										
164	4A	er det en skrivefeil?										
165	4B	Print is not defined, men trenger vi å definere print da?										
166	4B	Ja, det er skrivefeil										
167		(retter opp fra pirint til print)										
168	4A	Det fungerer, så da må vi bare skrive flere kast nedover da?										
169	4B	Kan du ikke bare kopiere og lime inn?										
170	Intervjuer	Det høres ut som en veldig god ide										
171	4B	Du trenger bare å kopiere den linja der [kastfunksjonen]										
172	4A	Ikke print?										
173	4B	Nei										
174	4A	Ikke?										
175	4B	Nei, vi kan bare skrive dem i samme										

176		(Kopierer kastfunksjonen og lager fem kast)									
177	4B	Hvor mange har vi nå?									
178	4A	Vi trenger en til									
179	Intervjuer	Vi spiller bare vanlig yatzy nå									
180	4A	Åja, da trenger vi bare fem ja									
181	4B	Sånn, så må vi skrivde dem inn der [printfunksjonen]									
182	4A	Sånn bare liksom?									
183	4B	Ja, også bare kast2 og kast3.									
184	4B	Men blir det ikke her veldig vanskelig i yatzy, om man vil beholde noen av dem?			MOD						
185	Intervjuer	ja, vi kommer til det etterpå. Men koden for å få det til å skje er veldig lang og vanskelig, så det får vi nok ikke til idag									
186	4A	Der har vi det									
187	4B	men vi vil jo ikke beholde noen av dem, så da kaster vi bare igjen									
188	Intervjuer	Men koden fungerer jo veldig bra, er dere fornøyd med det koden viser?									
189	4A	Ja, og det enkleste er jo bare å ta et ark og skrive ned dem man vil beholde			MOD	REP		HJE		Evaluerings	
190	Intervjuer	Ja, så da hadde dere jo også hatt et naturlig steg til å bygge videre på koden om vi skulle jobbet med dette lenge									
191	4A	ja									

192	Intervjuer	Så kunne man jo også pyntet på koden og lagt inn noe tekst også om man vil									
193	4B	Ja									
194	Intervjuer	Men var det forståelig litt av prosessen bak hvordan man lager en kode?									
195	4A	Ja									
196	4B	Ja, men jeg må jo se på noe hvis jeg skal lage selv									
197	Intervjuer	Ja, det er nok veldig vanlig når man skal prøve å lære seg programmering. Men hva synes dere egentlig om programmering i mattetimene?									
198	4A	Jeg synes det er dritt									
199	4B	Jeg ser ikke helt poenget								HJE	
200	4A	Jeg liker best når matte er tall									
201	Intervjuer	Så dere følte ikke at det å bruke kode i den første koden for eksempel kunne gjort det lettere å visualisere sannsynlighet?									
202	4A	Ja, akkurat der ja								HJE	
203	4B	Ja, men i matte er det så mange formler som jeg bare må stole på, så da hadde jeg nok stolt på det også									
204	Intervjuer	Ja okei, så du er ikke helt overbevis om nyttigheten?									
205	4B	Nei, jeg er ikke helt for programmering									

206	Intervjuer	Men tror dere det hadde vært lettere om dere hadde begynt med programmering tidligere?									
207	4B	Ja, for nå skriver jeg bare mye av, og så satser jeg på det beste								HJE	
208	4A	Da lærer man ikke så mye av det								HJE	
209	4B	Så tar det så lang tid									
210	Intervjuer	Det skjønner jeg. Man lærer jo ikke så mye av å bare kopiere									
211	4B	Men jeg ser jo at vi trenger det i verden da, men det er ikke noe gøy allikevel								HJE	
212	Intervjuer	Ja, det kan jo være god trening for noen jobber									
213	4A	Ja									
214	Intervjuer	Men jeg skjønner hva dere mener med at det er vanskelig å få mye ut av det uten å kunne det skikkelig									
215	4A	Ja, vi har fått veldig lite forklaring og hva det er godt for og sånn								HJE	
216	4B	Ja, da synes jeg det er mye bedre med blokkprogrammering, det er mye lettere å forstå. Da skjønner jeg faktisk litt hva jeg gjør								HJE	
217	Intervjuer	Det skjønner jeg veldig godt. Men jeg håper at dere forsto litt mer etter disse oppgavene									
218	4A	Ja									
219	4B	Mhm									
220	Intervjuer	Så bra. Takk for at dere ville være med									

8.8.2 Kode fra oppgave 2

```
1 import random
2
3 terning = ["1","2","3","4","5","6"]
4 kast1 = random.choice(terning)
5 kast2 = random.choice(terning)
6 kast3 = random.choice(terning)
7 kast4 = random.choice(terning)
8 kast5 = random.choice(terning)
9 print (kast1,kast2,kast3,kast4,kast5)
10
```