

Ruben Hvatum Johnsen

Elevs opplevelse av programmering i fysikk i LK20

En mixed methods-studie om innføringen av algoritmisk tenkning og programmering i fysikkfaget i LK20.

Masteroppgave i MLREAL

Veileder: Trine Høyberg Andersen

Medveileder: Guri Sivertsen Korpås

Juni 2024



NTNU

Kunnskap for en bedre verden

Ruben Hvatum Johnsen

Elevers opplevelse av programmering i fysikk i LK20

En mixed methods-studie om innføringen av algoritmisk tenkning og programmering i fysikkfaget i LK20.

Masteroppgave i MLREAL
Veileder: Trine Højberg Andersen
Medveileder: Guri Sivertsen Korpås
Juni 2024

Norges teknisk-naturvitenskapelige universitet
Fakultet for naturvitenskap
Institutt for fysikk



Kunnskap for en bedre verden

Sammendrag

Med innføringen av ny læreplan i 2020 har numeriske metoder og digitale verktøy blitt en større del av fysikkfaget i videregående skole, og med dette har programmering blitt lagt til som en del av kompetansemålene til elevene. Denne programmeringen skal være av en utforskende natur, og både gi elevene et nytt verktøy for å gjøre problemløsning, og utvikle deres evne til å tenke algoritmisk. Men overgangen til ny læreplan har vært brå for mange, og innføringen av programmering i fysikkfaget har sine utfordringer.

Denne oppgaven ser på hvordan elever har opplevd innføringen av algoritmisk tenkning og programmering i fysikkfaget gjennom LK20. Dette gjøres gjennom å se på hvordan innføringen av LK20 har påvirket elevenes holdninger til programmering innen fysikk og ved å se på i hvilken grad undervisning i programmering i fysikk i videregående har blitt opplevd som tilstrekkelig for at elevene skal kunne bruke programmering som et verktøy. For å gjøre dette har jeg gjennomført en spørreundersøkelse på førsteårstudenter i fysikk på NTNU som nylig har fullført videregående skole, og gjennomført intervjuer med fem fysikklærere fra forskjellige skoler, og analysert funnene i lys av relevant teori.

Funnene fra dette tyder på at elever som har hatt ny læreplan, og derfor hatt programmering som del av fysikkfaget, rapporterer en liten økning i motivasjon for å lære programmering og i hvor nyttig de synes programmering er i fysikk, i forhold til elever som har hatt gammel læreplan. Men, det har også vært stor spredning i utbyttet elevene rapporterer å ha fått fra videregående, og der noen har fått et nyttig verktøy for problemløsning har andre opplevd at de ikke har fått lære programmering godt nok til å få noe gevinst fra det. Lærerne som har blitt intervjuet forteller om begrenset med tid i faget, stor spredning i elevenes kompetansemål, og dårlig forkursing som noen av årsakene til at det har vært vanskelig å få innført programmering inn i fysikkfaget på en god måte.

Konsekvensene av dette er at programmering ikke blir et verktøy men et hinder. Elevene får ikke utviklet sine problemløsningsstrategier, men får heller bare mer de må pugge for å komme seg gjennom fysikkfaget. For å unngå dette må programmeringen få nok fokus til at elevene får mulighet til å lære det godt nok til å kunne faktisk bruke det selv. Enkelte av lærerne rapporterer at de har endret på måten de underviser programmering i fysikk til å ha en slik jevn bruk av programmering, slik at det skal sitte bedre for elevene, men utviklingen virker til å gå sakte. Elevene rapporterer høy motivasjon for å lære programmering, men bare om det gjøres skikkelig. Det er derfor viktig at lærere følger opp på dette, og gir elevene muligheten til å tilegne seg et viktig nytt verktøy for en fremtid i fysikk.

Abstract

With the introduction of a new curriculum in 2020, numerical methods and digital tools have become a greater part of the subject of physics in high school, and programming is now part of the competence aims. This programming is meant to be explorative, give the pupils a new tool for problem solving, and develop their computational thinking. However, the transition to the new curriculum has been abrupt for many, and the introduction of programming in physics has had its challenges.

This thesis looks at how pupils have experienced the introduction of computational thinking and programming in high school physics through the new curriculum. This is done through seeing how the introduction of the new curriculum has affected the pupils attitudes to programming in physics, and by seeing to what degree education in programming in high school physics has been experienced as sufficient to let the pupils use programming as a tool. I have, to accomplish this, performed a survey of physics students at NTNU that have just finished high school, and by conducting interviews with five physics teachers from different schools, for then to analyse the findings through the lens of relevant theory.

The findings indicate that the pupils that have had the new curriculum, and thus had programming as part of their physics course, reports a small increase in motivation for learning programming and in their perceived usefulness of programming in physics, compared to pupils that had the old curriculum. However, there has been a wide variation in the benefits the pupils are reporting to have had from high school, and whilst some have gained a valuable new tool for problem solving, others have experienced that they haven't been able to learn programming well enough to get any use from it. The teachers that were interviewed tell about time constraints, pupils of varying skill level, and poor precourses as some reasons why introducing programming to the physics courses have been difficult to do well.

The consequences of this is that programming turns out not a tool, but an obstacle. The pupils are unable to develop their strategies for problem solving, and instead get more they have to cram to pass high school physics. In order to avoid this programming has to be given enough focus that the pupils are allowed to learn it well enough to be able to properly utilize it. Some of the teachers report that they have been developing their teaching strategies to have programming more evenly throughout the schoolyear to increase retention, but the development seems to be going slow. Pupils report high levels of motivation towards learning programming, but only if it is done properly. Therefore it is important that teachers follow up on this, and give the pupils the possibility to gain an important new tool for a future in physics.

Forord

Mot slutten av en femårig lektorgrad kan jeg se tilbake på et unikt studieløp på NTNU, alt jeg har fått oppleve, fra første uken som fadderbarn til å nå ferdigstille min helt egen masteroppgave. Disse årene på universitetet har gitt meg utrolig mye. NTNUs visjon er «kunnskap for en bedre verden». Jeg håper å kunne bruke den kunnskapen jeg har oppnådd til å gjøre mitt for å virkeliggjøre denne visjonen når jeg trer i jobb som lærer.

I den anledning ønsker jeg først og fremst å takke mine veiledere, Trine Høyberg Andersen og Guri Sivertsen Korpås. Deres veiledning, støtte, konstruktive tilbakemeldinger, og positive engasjement har vært absolutt nødvendig for meg for å få gjennomført dette siste steget av utdanningen min, og jeg er ekstremt takknemlig for alt de har gjort for å hjelpe meg.

Videre ønsker jeg å utgi en takk til venner og familie som har hjulpet meg med å se over oppgaven i de siste stegene. Det har vært lett å få tunnelblikk inn i eget arbeid når jeg har sittet med det daglig gjennom et helt semester, og å få eksterne øyne til å se over har avslørt flere formuleringer som har nytt godt av å bli omskrevet.

Til slutt ønsker jeg å takke LURDO!, lektorutdanningen i realfags brettspillklubb, for sitt betydelige bidrag til mitt sosiale velbehag gjennom studier, pandemi, praksis, eksamensperioder, masterskriving, og mye mer. En spesiell takk der til DnD-gruppen som jeg fra første studieår har møttes med mer eller mindre ukentlig for alle slags spennende eventyr.

Selv om det er deilig å endelig være ferdig kommer jeg alltid til å se tilbake til tiden på NTNU som en fantastisk reise, og jeg kommer til å savne studietiden og alt den har bydd på.

Ruben Hvatum Johnsen

Trondheim, juni 2024

Innholdsfortegnelse

| | |
|---|-----|
| Sammendrag..... | i |
| Abstract | ii |
| Forord | iii |
| 1. Introduksjon | 1 |
| 1.1 Motivasjon | 1 |
| 1.2 Bakgrunn..... | 2 |
| 1.3 Problemstilling og forskningsspørsmål | 3 |
| 2. Teori | 4 |
| 2.1 Innføringen av programmering i fysikk..... | 4 |
| 2.1.1 Læreplan, i teori og i praksis | 5 |
| 2.2 Computational literacy og computational thinking | 6 |
| 2.3 Algoritmiske tenkeren | 7 |
| 2.4 Holdninger til programmering i fysikk..... | 8 |
| 3. Metode..... | 10 |
| 3.1 Mixed methods..... | 10 |
| 3.1.1 Spørreundersøkelsens oppbygning og utforming..... | 10 |
| 3.1.2 Utvalg og innsamling for spørreundersøkelse | 13 |
| 3.1.3 Databehandling | 13 |
| 3.2 Intervju..... | 17 |
| 3.2.1 Kvalitativ analyse | 18 |
| 3.3 Ethiske betraktninger og personvern | 22 |
| 3.4 Validitet..... | 22 |
| 4. Resultater | 24 |
| 4.1 Spørreundersøkelse | 24 |
| 4.1.1 Motivasjon og mestringsforventning | 24 |
| 4.1.2 Holdninger og erfaringer | 34 |
| 4.2 Faglig overblikk..... | 41 |
| 5. Diskusjon | 45 |
| 5.1 Refleksjoner om teori og funn | 45 |
| 5.2 Forslag for videre forskning | 47 |
| 6. Konklusjon..... | 49 |
| Referanseliste:..... | 50 |

| | |
|---|----|
| Vedlegg | 1 |
| Vedlegg B: Utvalgte reponser på åpne spørsmål..... | 5 |
| Vedlegg C: Informasjonsskriv for informanter til intervju | 10 |

1. Introduksjon

Da det nye studieåret begynte i starten av høsten 2023, begynte også de første studentene som hadde gjennomført videregående skole (vgs.) etter den nye læreplanen på universiteter rundt i Norge. Denne nye læreplanen, kjent som LK20, har brakt med seg endringer til den norske skolen, deriblant introduksjonen av programmering i realfagene. Matematikk, fysikk, kjemi, og biologi har alle fått digitale ferdigheter inn i utdanningsdirektoratet sin liste over fagenes grunnleggende ferdigheter (Udir, 2019). I fysikk 1 innebærer et av kompetansemålene å kunne programmere en modell over ikke-konstant akselerasjon, mens fysikk 2 har det mer helhetlige målet at «[...] eleven skal kunne bruke numeriske metoder og programmering til å utforske og modellere fysiske fenomener» (Udir, 2019). Altså blir programmering et tema som kan dukke opp i hvilket som helst tema i fysikk 2. Dette krever at ikke bare elever men også lærere har en bred forståelse for programmering og hvordan det brukes i fysikkfaget.

1.1 Motivasjon

Som praksisstudent ved lektorutdanning i realfag ønsket jeg selv å få erfaring med å undervise programmering i matematikk og fysikk. Jeg holdt en slik time tidlig i min lærerpraksis, i matematikk 1T, som var både svært lærerik og viste meg at dette var noe elevene trengte å bruke tid på for å lære. I senere praksisforløp etterspurte jeg derfor å ha elevene i programmering for å få mer trening på dette, men fikk typisk som svar at det var noe som skulle bli gått gjennom på slutten av året, om det var tid. Jeg hørte med medstudenter som også var i praksis, og de fortalte gjerne at enten nedprioriterte lærerene programmering og prioriterte andre temaer, eller så var de svært takknemlige for å ha noen studenter som kunne ha de timene for dem. Kontakt med elever, både gjennom praksis og gjennom egen jobb som privatunderviser, gav inntrykket at det ikke var uvanlig at elever også nedprioriterer programmering og heller fokuserer på andre temaer de har mer tro de skal kunne mestre til prøver og eksamener.

Selv har jeg først og fremst lært programmering i relasjon til fag på universitetet. I fysikkfag ble vi ofte gitt oppgaver som krevde teknikker og ferdigheter som lå utenfor det vi lærte i grunnkurset i IT fra første semester, uten at fysikkfaget hadde undervisning eller ressurser for å lære disse ferdighetene. Vi studentene måtte dermed bruke mye tid på å lære oss programmering på egenhånd for å kunne løse oppgavene vi ble gitt. Derfor var det bekymringsverdig for meg å oppleve at det virket til å bli nedprioritert av både elever og lærere. Potensielt måtte elever som ønsket å henge med på programmeringen i faget, eller som ville stille forberedt til høyere utdanning, også lære seg dette på egenhånd til tross for at det er en av de grunnleggende ferdighetene nevnt av Utdanningsdirektoratet (Udir). I jobb som privatlærer har jeg møtt flere elever med høy måloppnåelse i både matematikk og fysikk som likevel ønsker at vi ikke bruker tid på programmering, men heller håper å gjøre opp for at de ikke kan dette med å gjøre det bra på andre temaer. For min masteroppgave ønsket jeg dermed å finne ut mer av hva elevene synes om programmering i fysikk, og hva de får ut av undervisningen i form av programmeringsferdigheter.

1.2 Bakgrunn

Som nevnt innledningsvis er det i skrivende stund kun ett kull som har fullført hele videregående skole med den nye læreplanen. Dermed er det vanskelig å finne studier som undersøker hvordan dette løpet har gått. Den teoretiske bakgrunnen for denne oppgaven er derfor i stor grad basert på utenlandsk forskning, der programmering har vært en del av fysikken også før LK20, og flere studier har blitt gjennomført. Like før LK20 ble implementert ble det skrevet en masteroppgave som konkluderer med at «programmering og algoritmisk tenking har en naturlig plass i det tjueførste århundrets fysikkundervisning.» (Nordby, 2019). Algoritmisk tenking sikter til en norsk oversettelse av termen «computational thinking». Udir beskriver essensen av algoritmisk tenking som «å bryte ned komplekse problem til mindre, mer håndterlige delproblemer som lar seg løse.» (Udir, 2019b).



Figur 1: Utdanningsdirektoratets 'Den algoritmiske tenkeren' (Udir, 2019c)

En undersøkelse av elevers mottakelse av programmering i fysikk finnes i Bott et al. (2019), der de i en studie fra USA undersøker holdninger («attitudes») til elever som har brukt programmering i fysikkoppgaver. Dette gjøres ved bruk av åpne spørsmål designet for å få en bredde av varierte responser. For data fra Norge er masteroppgaven til Henrik Storesund (2022) aktuell. Den ser på spørsmålet om programmering i fysikk fra perspektivet til fysikkstudenter ved bruk av lukkede spørsmål. I likhet med Norby diskuterer Storesund også algoritmisk tenkning, og ser på hvordan det brukes i programmering og utenfor.

I semesteret før masteroppgaven fulgte jeg et emne om forskningsmetoder, og dette metodefaget innebar å gjøre en prosjektoppgave i form av en selvvalgt studie. Jeg valgte å gjøre en studie som også ville hjelpe meg til masteroppgaven, ved å undersøke et relatert tema; hvordan lærere hadde blitt forberedt på innføringen av programmering med LK20, spesielt innenfor fysikk. Jeg valgte å undersøke dette via intervjuer med fysikklærere, og når jeg alt skulle gjøre slike intervjuer valgte jeg og også ha spørsmål rettet direkte til masteroppgaven for å bruke senere. Opptak av intervjuene ble transkribert og kodet i to bolker, en for prosjektoppgaven og en for masteroppgaven. Dette sparte tid for datainnsamling og gjorde at jeg hadde to datasett jeg kunne begynne å arbeide med så snart semesteret for masteroppgaven startet.

1.3 Problemstilling og forskningsspørsmål

Undersøkelse som til nå er nevnt ser på bruk av algoritmisk tenkning og holdninger til programmering i fysikkfaget. For å ta det videre i mitt forskningsprosjekt har problemstillingen for oppgaven blitt definert som følger:

Hvordan har elevene opplevd innføring av algoritmisk tenkning og programmering i fysikkfaget gjennom LK20?

Denne problemstillingen vil legge vekt på to hovedvinklinger; elevenes holdninger til- og elevenes mestring av programmering i fysikk. Disse vinklingene vil undersøkes henholdsvis gjennom to forskningsspørsmål:

Hvordan har innføringen av LK20 påvirket elevens holdninger til programmering innen fysikk?

I hvilken grad har undervisning i programmering i videregående skole blitt opplevd som tilstrekkelig for at elevene skal kunne bruke programmering som et verktøy i fysikk?

Formålet med forskningsspørsmålene er å få kartlagt elevenes subjektive opplevelse og deres faglige utbytte under LK20. Elevene fra LK06 har også holdninger til programmering til tross for at det ikke har vært en del av deres læreplan i matematikk og fysikk på vgs. Dermed kan det være nyttig å undersøke hvordan elevene som har fullført vgs. under LK20 tenker om programmering i fysikk sammenlignet med de som har hatt LK06. Har elevene blitt mer motivert for å bruke programmering etter å ha fått prøve det, eller har eksponeringen gitt dem mer negative assosiasjoner til programmering? Opplever de at programmering kan være nyttig for å jobbe med fysikk, eller har det blitt opplevd som mer tungvint enn hjelpsomt? Føler de at de får til å bruke det som et verktøy de mestrer, eller er det først og fremst en kilde til frustrasjon?

Intensjonen til denne studien er å kartlegge resultater fra første kullet som har gjennomført LK20, og se på hvordan den nye læreplanen har påvirket opplevelsen av fysikkfaget, spisset inn på programmering. Dette gjøres for å kunne tilpasse fagmiljøet, både på vgs. og på universitetene, slik at det best mulig svarer til elevenes nivå og forventninger og til å gjøre programmering til et verktøy en kan ta nytte av gjennom fysikkutdannelsen, heller enn et hinder man kommer seg over.

2. Teori

I dette kapitlet vil jeg legge frem undersøkelser og forskning som bygger grunnlaget for masteroppgave videre. Først presenteres læreplanverket fra Udir og hva som faktisk har vært endringen fra gammel læreplan, LK06, til ny læreplan, LK20. Deretter en utdypning av begrepene *computational literacy* og *computational thinking*, deres betydning i fysikksammenheng, og deres plass i den norske læreplanen. Til slutt presenteres undersøkelser om elever og studenters holdninger til fysikkprogrammering, som også har vært betydelige i å forme min egen metode for lignende undersøkelser.

2.1 Innføringen av programmering i fysikk

I LK06, som var gjeldende fra 2006 til 2020, beskriver Udir «digitale ferdigheter», en av flere punkt under listen over fysikkfagets «grunnleggende ferdigheter», som følger:

Å kunne bruke digitale verktøy i fysikk innebærer å utforske, måle, registrere, analysere, dokumentere og publisere digitalt. Det betyr å anvende animasjoner og bruke Internett til å hente inn fysikkfaglig informasjon. Å kunne bruke digitale verktøy i fysikk betyr å simulere fenomener og forsøk som det ellers er vanskelig å studere. (Udir, 2005)

Med simuleringen og digitale verktøy refereres det til å bruke programmer som Geogebra og/eller Tracker, for eksempel for å spore en radiobil gjennom en løype for å gjøre målinger, heller enn å skrive kode i språk som Python eller Scratch. I LK20 ser den tilsvarende teksten ut som følger (min utheving):

Digitale ferdigheter i fysikk innebærer å bruke digitale ressurser til å registrere, bearbeide, analysere, modellere og presentere data. Det innebærer også å innhente relevant informasjon for å studere fysikkfaglige fenomener og problemstillinger.

Videre innebærer det å bruke programmering og dynamiske verktøy til å utforske fysiske problemstillinger. (Udir, 2019b)

Altså har faget blitt endret fra å bruke digitale verktøy for å studere spesifikke fenomener og forsøkt til å bruke dem for å utforske fysiske problemstillinger. Den nye formuleringen legger altså vekt på utforskende arbeid. Det blir og spesifisert at programmeringen er lagt til for å være et verktøy som skal hjelpe elevene med å forstå fysikk, ikke programmering for programmeringens skyld.

Videre i kompetansemålene for fysikk 1 og fysikk 2 finner vi henholdsvis at eleven skal kunne «bruke numeriske metoder og programmering til å modellere og utforske bevegelse i situasjoner der akselerasjonen ikke er konstant» (Udir, 2019d) og «bruke numeriske metoder og programmering til å utforske og modellere fysiske fenomener» (Udir, 2019e). Det vil si at i fysikk 1 er det i hovedsak ment å bruke programmeringen kun i spesifikke temaer innenfor mekanisk fysikk, som typisk har sine kapitler tidlig i faget. Dette resulterer i at mye eller alt av programmering vil foregå på høstsemesteret, etterfulgt av en lang periode uten noe programmeringsrelatert annen enn prøver som dekker alt som er gjennomgått. I fysikk 2 er derimot programmering noe som kan høre til ethvert tema, og står læreren fritt til bruke det der de føler det passer. Med den friheten kommer også

ansvar, da læreplanen i fysikk 2 heller ikke gir noen pekepinn på hvor det faktisk passer, men gir denne avgjørelsen til lærerne. I likhet med under fagets grunnleggende ferdigheter (Udir, 2019) sier kompetansemålene at programmeringen skal være av en utforskende natur. Det vil si at også for eksempel temaet om ikke-konstant akselerasjon i fysikk 1 skal ikke programmeringen bare være et verktøy for å løse en oppgave. Programmeringen skal være utforskende.

2.1.1 Læreplan, i teori og i praksis

Det er viktig å være oppmerksom på at det er Udir som utvikler læreplan, men det er fremdeles lærerne som må overføre denne til klasserommet. «Læreplaner er ikke det samme som virkeligheten» står det i Lærerens Verden (Imsen, 2020, s. 293). For dette gis tre hovedgrunner:

- 1) Læreplanen er ikke altdekkende.
- 2) Elevene er ikke kun passive mottakere som læreren kan styre
- 3) Læreplanen vil bli tolket på forskjellige måter av forskjellige lærere

Læreplanen som gitt av Udir er det som kalles *den formelle læreplanen* (Goodlad et al., 1979, hentet fra Imsen et al., 2020, s. 294-296). Dette er andre nivå av Goodlads femsidig læreplan, som helhetlig er bygget opp som følgende:

Den ideologiske læreplanen: Idealplanen som styrer hvilke perspektiver og hovedvekker faget skal legge fokus på. Denne læreplanen omhandler hva som blir sett på som essensen av faget, men må vike for praktisk gjennomførbarhet og sosiopolitiske hensyn.

Den formelle læreplanen: Læreplanen slik den er skrevet og utgitt, for eksempel læreplanen vi finner på Udir sine nettsider. Denne læreplanen er først og fremst for å signalisere grunnidéene fra den ideologiske læreplanen, til den grad de kan konkretiseres.

Den oppfattede læreplanen: Læreplanen slik skoler, lærere, og foreldre tolker den. Denne læreplanen vil dermed være forskjellig fra klasse til klasse og underlagt subjektive tolkninger basert på hver enkelts lærerskjønn, og vil dermed være unik fra lærer til lærer.

Den gjennomførte læreplanen: Læreplanen slik den blir utført i klasserommet. Temaer som lærerens faglige interesser, skolens midler, og elevenes tilbakemeldinger er alle med på å gjøre hver gjennomførte læreplan unik fra klasse til klasse, selv med samme lærer.

Den erfarte læreplanen: Læreplanen slik elevene opplever det. Hva lærer de? Hva husker de? Hvis en elev en time var mer fokusert på å lese om sorte hull enn om Keplers lover vil det være det som preger deres erfarte læreplan for den timen.

I boken Fysikkdidaktikk (Angell et al., 2019) bygges det videre på dette konseptet, og dets betydninger for fysikkfaget. Forfatterne kommer med forslag til gjennomføring av multimediebasert undervisning, og tips til hvordan fysikklærere kan bruke digitale hjelpemidler, uten å overbelaste elevene (Angell et al., 2019, s254-260). Med dette bidrar de til hjelp med spørsmålet ovenfor om hvor og når det passer å bruke digitale hjelpemidler i fysikkundervisningen. Til tross for å være skrevet før LK20 kom ut blir programmering

beskrevet som en sentral ferdighet i fysikkfaget og som nyttig for forståelse for både forskning og fagets egenart. Videre knytter Angell et al. programmering opp mot kompetanse i *computational thinking*. For å kunne forstå hva som ligger i dette begrepet er det nyttig å først se nærmere på det overordnede begrepet *computational literacy*.

2.2 Computational literacy og computational thinking

Andrea diSessa beskriver at computational literacy «will impact all STEM disciplines at their very core, but most especially in terms of learning.» (2018, s. 2-3). Ordet 'literacy' beskriver han som mer enn evnen til å kunne lese, men også som evnen til å kunne bruke, utnytte, og forstå når det vil være nyttig. Med dette sammenligner han literacy i programmering med literacy i for eksempel algebra, og hvordan det har hatt stor påvirkning på hvordan vi gjør utregninger, men også som et konsept som har utviklet seg over tid. Som i algebra er forståelse og evne til å benytte programmering en bred egenskap langt større enn å kunne lese hvilke tegn og bokstaver som står i en likning/funksjon, men samtidig også som er en spesifikk egenskap man må vite når det passer og når det ikke passer. For å bruke diSessa sin egen allegori, så er algebra (og programmering) nyttig om man skal beregne klimaendringer, men ikke like nyttig for å skrive dikt. Literacy inkluderer å forstå nettopp dette.

I en tidligere bok har diSessa (2000) beskrevet computational literacy som bygget opp av tre grunnaspekter (foundational pillars): materielt, kognitivt, og sosialt. Den første av disse, det materielle aspektet, omhandler kjennskap til modeller og representasjoner og å kunne benytte seg av dem. Det vil være vanskelig å skrive en bok om man ikke forstår grammatikk, og tilsvarende vil det være vanskelig å programmere i Python om man ikke kan noen funksjoner. diSessa legger vekt på at det materielle aspektet er menneskeskapt, oppfunnet, og utviklet over tid. For eksempel vil det være lettere for en moderne student å gjøre matematiske bevis enn det var på Pythagoras' tid, fordi studenten har tilgang til blant annet algebra og kalkulus. Tilsvarende kan man i dag med lite trening gjøre simuleringer som ville vært umulig for tidligere dataingeniører eller fysikere, og det er ingen grunn til å tro at fremtiden ikke også vil muliggjøre det som nå er umulig for oss. «With all these new forms and more to come, it seems inconcievable our current material literacy basis could remain unaffected.» (diSessa, 2000, s 7).

Det neste aspektet omhandler den kognitive forståelsen, altså hvordan vi bruker det materielle til å forstå og skape. Det å ha kjennskap til syntax og funksjoner er ikke nok til å ha en helhetlig forståelse av programmering. Dersom man vil simulere planetbaner må man også skjønne hvordan alt henger sammen. Det kognitive aspektet handler om å kunne bruke den materielle kunnskapen i en større setting, men også om å anvende den for å få frem og forstå det store bildet (Odden et al., 2019)

Det siste aspektet diSessa beskriver er det sosiale, altså det større samfunnsmessige nettverket. diSessa forklarer dette gjennom å sammenligne Newtons og Leibnizs notasjon for kalkulus, og at Leibniz i stor grad fortjener æren for å ha gjort det forståelig og tilgjengelig. Samtidig påpeker han at både Newton og Leibniz var avhengig av systemer og informasjon som stammet fra andre mennesker for at de selv kunne bygge opp det de gjorde. Tilsvarende går programmering hverken fra eller til et vakuum, og et veldig kraftig program utretter lite om ingen forstår hvordan det skal brukes. Odden et al. (2019) nevner

å kommentere kode og å bygge dokumentasjon som eksempler på sosial computational literacy.

Odden et al. (2019) skriver videre om betydningene computational literacy har for fysikkdidaktikk. Forfatterne argumenterer at «computational literacy is a necessary precursor for a student to properly leverage the power of computation in their learning.» (Odden et al., 2019, s 18), og at det derfor er nødvendig å tenke gjennom hvordan pensum blir lagt opp for å lære elevene dette, og for lærere å være obs på elevene sine nivåer av computational literacy når de legger opp programmeringsoppgaver. I tillegg påpekes det at fysikkfaget er spesielt egnet for programmering, både for å bruke faget til å forstå programmering og for å bruke programmering for å forstå faget (Odden et al., s 1). Med dette knyttes computational literacy opp mot den relaterte termen *computational thinking*.

«Computational thinking generally refers to the ability to think in computational, procedural, or algorithmic terms for problem solving» (Odden et al., s 4). Computational thinking handler om den metodiske, algoritmiske tenkemåten for problemløsning uten å nødvendigvis relatere til å faktisk bruke digitale hjelpemidler, i motsetning til computational literacy sin direkte knytning til programmeringen. Computational thinking er dermed samtidig en underdel av computational literacy, det vil være vanskelig å mestre programmering uten, men samtidig en uavhengig ferdighet som også kan brukes når man arbeider med for eksempel penn og papir.

For en mer direkte analyse av computational thinking gir Wing (2006) en beskrivelse av trekk ved ferdigheten. «Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative.» (Wing, 2006, s 33-35) Computational thinking blir beskrevet først og fremst som en fremgangsmåte for problemløsning, å tenke analytisk og å dele opp større problemer i mindre biter, å kunne omformulere problemet til noe som er løsbart, å kunne benytte en logisk tilnærming på en kreativ måte. Som ved Odden et al. blir det eksplisitt separert fra programmering i at det er en fremgangsmåte som er «for everyone, everywhere.» (s.35), ikke bare for dataingeniører eller fysikere. Weintrop et al. (2016) ønsker å gi en mer praktisk rettet definisjon, og definerer algoritmisk tenkning gjennom praksisene (practices) databehandling, modellering og simulering, løsning av beregningsproblemer, og systemtenking.

2.3 Algoritmiske tenkeren

I det norske skoleverket brukes gjerne begrepet *algoritmisk tenkning* (Udir, 2019b). Udir forsøker seg ikke på noen konkret definisjon av algoritmisk tenkning, men forklarer at det innebærer en rekke ferdigheter som even til å bryte ned problemer, modellere, fjerne unødvendige detaljer, og å være systematisk og analytisk men samtidig eksperimenterende og åpen. En analytisk tenker, per Udir, har strategier for å identifisere og rette opp feil som blir gjort underveis og kommer seg frem ved å «fikle» og så feilsøke og dermed kontinuerlig tilpasse sin strategi underveis. I boken *Programmering i skolen* (Haraldsrud et al., 2021) beskriver den viktigste delen av algoritmisk tenkning som «problemløsning gjennom noen sentrale strategier som systematisering, analyse, gruppering, abstraksjon og evaluering» (s. 189).

I sin masteroppgave *Programmering og algoritmisk tenkning i fysikkundervisning* skriver Nordby (2019) at algoritmisk tenkning har «en naturlig plass i fysikkundervisningen» (s102.), men også at oversettelsen ikke nødvendigvis fungerer like bra som det engelske computational thinking, gitt at problemløsningen i fysikk er langt bredere enn bare algoritmer, og at uttrykket ofte misforstås av lærere. Han beskriver og at algoritmisk tenkning innebærer mer enn bare å programmere inn beregninger, og er en egen ferdighet som fortjener dedikert tid i undervisning. En annen masteroppgave, *Shaping programming in physics education* (Gjengset, 2022), støtter opp denne konklusjonen med å påpeke at algoritmisk tenkning allerede er et eksisterende begrep, distinkt fra computational thinking blant annet i at det er langt mer fokusert på, som navnet tilsier, algoritmer. Videre kritiserer Gjengset Udir for å gi uklare og selvmotsigende retningslinjer for hvordan programmering og algoritmisk tenkning skal inn i skoleverket. Dette er spesielt problematisk kombinert med at fysikklærere ofte ikke har tilstrekkelig erfaring i programmering til å kunne gjøre alle de nødvendige vurderingene.

2.4 Holdninger til programmering i fysikk

For å undersøke skoleverkets økte fokus på computational thinking og programmering i fysikk har Bott et al. (2019) fra universitetet i Michigan gjennomført en tematisk analyse av studentresponser på et spørreskjema. Spørreskjemaet bruker grunnstrukturen Colorado Learning Attitudes about Science Survey (CLASS) og den eksperiment-spesifikke varianten E-CLASS (Adams et al., 2018) som grunnlag til å designe en spørreundersøkelse med åpne spørsmål. I funnene sine fokuserer forskerne bak undersøkelsen på tre hovedtemaer som de fant gjennom kodingsarbeidet: anvendbarhet av fysikkidéer i koding, computational thinking-praksiser, og nytteverdien til programmering. Artikkelen fra Bott et al. finner at studentene ser stor nytteverdi i programmering og klarer å knytte det veldig direkte opp mot problemstillinger i fysikk, men også at det var et skille mellom studenter som syntes programmering var lett og de som syntes det var vanskelig. Det påpekes også at enkelte studenter rapporterte at det først var vanskelig, men at det ble enklere over tid. Avsluttende etterspør Bott et al. videre forskning på en større gruppe studenter, og anbefaler også som et neste steg å få tilbakemeldinger fra fysikklærere for videre analyse.

En relatert undersøkelse finner vi i masteroppgaven til Storesund (2022), *Beregningsorientering i fysikkutdanning*. Her undersøkes erfaringene til fysikkstudenter i de første tre studieårene, gjennom å se på de fire komponentene fra expectancy value-modellen (Eccles & Wigfield, 2002):

- Mestringsforventning: I hvilken grad studentene opplever å ha mestret programmeringsoppgaver i fysikk, og i hvilken grad de forventer å kunne løse fremtidige oppgaver.
- Interesseverdi: I hvilken grad studentene synes programmeringsaspektet ved fysikk er interessant.
- Nytteverdi: I hvilken grad studentene føler programmering er nyttig for fysikkfaget.
- Kostnad: I hvilken grad studentene synes det er krevende å jobbe med programmering i fysikk.

Undersøkelsen bruker mixed methods, med både åpne og lukkede spørsmål designet på egenhånd av Storesund (2022). Ett funn fra denne undersøkelsen, som delvis speiler

funnene til Bott et al. om en binær fordeling mellom at programmering var enten lett eller vanskelig, er at studentene på 2. årstrinn har markant høyere standardavvik på spørsmålet om mestringsforventning. Storesund tolker dette som at programmering i fysikk initielt er vanskeligere enn studentene først forventer, men at de enten klarer å overkomme dette og får mestringen når de er mer erfarne, eller at de bytter til andre studieprogram. Videre viser mestringsfølelsen korrelasjon til generell interesse for programmering i fysikk og også at den «[spiller] en stor rolle for studentens helhetlige inntrykk og motivasjon for oppgavene» (Storesund, 2022, s. 73). Også avdekket av undersøkelsen er at programmeringen i fysikkemner er stort sett mer avansert enn programmeringen i innføringsemnet i programmering, og krever ferdigheter som ikke dekkes av innføringsemnet. Dette betyr at de må bruke tid på å lære seg disse ferdighetene på egenhånd, men også her er det høyt standardavvik spesielt på 2. årstrinn. Ellers rapporterer studentene i Storesund sin undersøkelse at de ser høy nytteverdi i fysikkfaget for programmering, men også at å anse programmering som nyttig ikke er nok til å motivere studentene om de ellers har lav mestringsforventning.

3. Metode

I dette kapitlet skal jeg forklare fremgangsmetoden jeg har brukt for undersøkelsen, og forklare hvordan denne metoden har blitt formet. For å undersøke hvordan elevene har opplevd programmering i fysikk under LK20 har jeg samlet inn to datasett: En spørreundersøkelse gjennomført i en fysikkforelesning for 1. år fysikkstudenter på NTNU og en intervjuundersøkelse av fysikklærere fra forskjellige videregående skoler. Disse datasettene ble analysert gjennom henholdsvis mixed method (Robson & McCartan, 2016) og tematisk analyse (Braun & Clarke, 2006). Ved å bruke mixed method og kvalitativ data, og med både elevsyn og lærersyn for å få innsyn fra begge vinklinger, skal dette brukes til å få et helhetlig svar til forskningsspørsmålene presentert innledningsvis.

3.1 Mixed methods

Spørreundersøkelsen bruker en mixed method design, mer spesifikt *concurrent nested design* (Robson & McCartan, 2016). Det vil si at det er en primærmetode, i dette tilfellet kvantitativ metode, som blir supplementert med en sekundærmetode, i dette tilfellet kvalitativ metode. Denne fremgangsmåten styrker til analysen, blant annet at den gir mer helhetlig innsikt i resultatene og gjør det lettere å presentere data når den kan illustreres med både kvalitative og kvantitative eksempler (Robson & McCartan, 2016).

3.1.1 Spørreundersøkelsens oppbygning og utforming

Spørreundersøkelsen ble designet til å ha tre deler; info om respondenten, lukkede spørsmål, og åpne spørsmål (se vedlegg A). Gitt at respondentene skulle gjennomføre hele spørreundersøkelsen i en 15-minutters pause i en forelesning ble det lagt vekt på at undersøkelsen skulle være rask å gjennomføre og at det skulle være tydelig hva som ble spurt om. Derfor ble det lagt vekt på å unngå unødige spørsmål og forklaringer, og å minimere mengden ord. Spørreskjema ble laget på Nettskjema.no i henhold til retningslinjer fra NTNU (Kvalitetssystemet, NTNU) Spørreundersøkelsen var fullstendig anonym og ingen personopplysninger eller annen sensitiv informasjon ble innhentet i spørreundersøkelsen.

Den første delen, info om respondenten, var et sett med spørsmål som samlet inn grunnleggende informasjon som ble vurdert som potensielt nyttig for analyse: Årstall respondenten hadde fullført videregående skole, kjønn, hvilke fag de hadde hatt som inkluderte programmering, og hvilket studieprogram de gikk på. Med dette kunne respondentene sorteres slik at sammenhenger kunne undersøkes, primært for å sammenligne de som fullførte vgs. i 2023 mot de som hadde fullført tidligere. Respondenter som svarer at de fullførte videregående skole i 2023 (totalt 50 respondenter) ble klassifisert som LK20, mens resten ble klassifisert som LK06 (total 85 respondenter).

Den andre delen av spørreundersøkelsen, lukkede spørsmål, besto av 8 påstander der respondentene skulle svare med hvor enige de var i påstanden via en 5-steps Likertskala med svaralternativer «svært uenig», «uenig», «nøytral», «enig», «svært enig», i tråd med CLASS (Adams et al., 2018). Det ble i tillegg lagt til mulighet for å svare «ikke relevant» (se vedlegg A). Dette alternativet ble lagt til for at alle spørsmålene skulle kunne settes som obligatoriske, uten å skape forvirring for de som ikke hadde erfaring med situasjonen det ble spurt om eller av andre grunner ikke kunne eller ønsket å svare innenfor de 5 stegene.

Utformingen og påstandene brukt i seksjonen med lukkede spørsmål tok inspirasjon fra spørreundersøkelsen utviklet av Henrik Storesund (2022) og fra prosjektet Colorado Learning Attitudes about Science Survey (CLASS) (Adams et al., 2006). De lukkede spørsmålene har til hensikt å undersøke elevers motivasjon, trivsel, faglig utbytte, og opplevd nytteverdi i programmering i matematikk- og fysikkfagene.

Tabell 1: Påstander fra del 2, sortert etter konstrukter fra EV-modellen (Eccles & Wigfield, 2002).

| Konstrukt | Spørsmål |
|----------------------|--|
| Mestringsforventning | Programmeringen jeg har lært i fysikk har vært tilstrekkelig for å løse oppgaver i faget |
| Interesseverdi | Jeg er motivert for å lære programmering fremover Jeg har foretrukket å jobbe med programmeringsoppgaver i forhold til andre typer oppgaver |
| Nytteverdi | Programmering kan være nyttig for å løse problemer i fysikk Programmeringen jeg har lært i matematikk kan være nyttig for å arbeide med fysikk |
| Kostnad | Programmeringsoppgaver har vært mer krevende enn andre typer oppgaver |
| Forkunnskaper | Fysikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper Matematikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper |

Den tredje delen av spørreundersøkelsen, åpne spørsmål, besto av 5 spørsmål der respondentene hadde muligheten til å skrive et svar inn i en tekstboks (se vedlegg A), og ble oppfordret til å utdype svarene sine. Spørsmålene som ble brukt i seksjonen med åpne spørsmål er hentet fra en undersøkelse av Bott et al. (2019), som også bygger på CLASS. Spørsmålene ble oversatt fra engelsk til norsk, og ordlyden ble tilpasset for å passe bedre til prosjektet (se tabell 2). For eksempel ble spørsmålet «Why is computation useful for science? Please explain.» omskrevet til «Føler du at programmering er nyttig i fysikkfaget.» (Bott et al., 2019, s 59) for å unngå å stille et ledende spørsmål og for å spisse spørsmålet mer mot fysikk. Heller enn at det blir antatt at programmering er nyttig, og respondenten skal forklare hvorfor, blir respondenten spurt om de synes programmering er nyttig. Dette gir respondenter mulighet til også å gi svar som er uenige i at programmering er nyttig i fysikkfaget.

I tillegg til disse ble det gitt to åpne spørsmål som ikke er fra Bott et al.: Det første er om respondenten har lært programmering fra andre steder enn skolen. Det andre spørsmålet er om andre tanker om respondentenes forhold til programmering i fysikk, slik

at respondentene fikk muligheten til å utdype videre om tanker som ikke passet inn i noen av de andre spørsmålene. Kun disse to siste spørsmålene var ikke obligatoriske i spørreundersøkelsen. De åpne spørsmålene har til hensikt å undersøke respondentenes opplevelser og erfaringer med programmering i fysikkfaget.

Tabell 2: Originale spørsmål fra Bott et al. (2019, s 58) og min tilpasning

| Bott | Tilpasning ^a |
|--|--|
| 1. How do you feel about learning computation in your science class? Please explain. | 1. Hva tenker du om at programmering er en del av fysikkfaget? |
| 2. Why is computation useful for science? Please explain. | 2. Føler du at programmering er nyttig i fysikkfaget? |
| 3. Based on your experience, how easy/difficult has it been to learn coding in your science class? Please explain. | 3. Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk? |
| 4. When coding, do you think it is important to understand every line of the code before you can use it correctly? Why or why not? Please explain. | 4. Når du arbeider med programmering, syntes du det er viktig å forstå hver kodelinje? Hvorfor eller hvorfor ikke? |
| 5. When using code to solve a problem, how do your physics ideas apply to the problem? Please explain. | 5. ^b |
| 6. How does learning coding change your ideas about how physics works? Please explain. | 6. Hvordan har det å lære programmering påvirket din fysikkforståelse? |
| | 7. Har du lært programmering fra andre steder enn i skolen? Hvilke? ^c |
| | 8. Har du andre tanker om ditt forhold til programmering i fysikk? |

a) Ved del 3 ble det respondentene innledningsvis bedt om å vennligst utdype svarene sine.

b) Spørsmål 5 ble vurdert som for uklart og komplekst for elever i videregående, og ble fjernet.

c) Spørsmål 7 og 8 har ingen Bott-ekvivalent.

Spørreundersøkelsen ble gjort ved studiestart for at respondentene til spørreundersøkelsen skulle ha sine opplevelser og erfaringer fra videregående skole ferskt i minne, og også at de skulle ha blitt påvirket minst mulig av opplevelser og erfaringer fra universitetet. Spørsmålene jeg har brukt er inspirert av de lukkede spørsmålene fra Storesund (2022) og de åpne spørsmålene brukt i Bott et al. (2019). For de åpne spørsmålene har jeg i tillegg, i likhet med Bott et al., brukt CLASS (Adams, et al., 2018) til å forme spørreskjemaet.

3.1.2 Utvalg og innsamling for spørreundersøkelse

Av praktiske hensyn ble det også bestemt å gjennomføre spørreundersøkelsen med nyankomne fysikkstudenter på NTNU. Dette betyr at respondentene vil alle være personer som har valgt å fortsette med fysikk etter videregående skole, og dermed kan antas å være over gjennomsnittelig motivert for faget som helhet, men har også styrken at respondentene vil være fra forskjellige skoler over hele landet, og dermed unngås bias knyttet til enkeltskoler. Ettersom problemstillingen gjelder for videregående spørsmål ble det nevnt muntlig i forelesningen og også skrevet i spørreundersøkelsen at respondentene skulle bruke erfaringer fra videregående skole når de svarte på spørreundersøkelsen.

Innsamligen ble gjort ved å avtale med foreleser om å møte opp i en forelesning i mekanisk fysikk, et fag som er i 1. semester for samtlige studieprogram for fysikk, i 2. undervisningsuke. I pausen gikk jeg frem foran studentene, presenterte fort min oppgave og dens formål, og spurte om de kunne sette av litt tid til å svare på undersøkelsen. Det ble ikke gjort noen direkte optelling, men inntrykket var at nær samtlige studenter i salen valgte å svare på spørreundersøkelsen. Oversikt over respondenter finnes i tabell 1.

Tabell 3: Antall respondenter i de forskjellige kategoriene, i antall (ut av totalt 135) og prosent.

| | Svar | Antall | Prosent |
|------------------|---|--------|---------|
| Kjønn | Menn | 90 | 67 |
| | Kvinner | 45 | 33 |
| | Annet | 0 | 0 |
| | Ønsker ikke å oppgi | 0 | 0 |
| Studieprogram | Bachelor i fysikk (BFY) | 88 | 65 |
| | Fysikk og matematikk – masterstudium (MTFYMA) | 39 | 29 |
| | Lektor i realfag (MLREAL) | 7 | 5 |
| | Annet | 1 | 0.7 |
| År fullført vgs. | 2023 | 50 | 37 |
| | 2022 | 39 | 29 |
| | 2021 | 34 | 25 |
| | Før 2021 | 11 | 9 |

3.1.3 Databehandling

For å kunne analysere datasettene fra spørreundersøkelsen laget jeg selv kode i Python, via Jupyter Notebook, for å telle og sortere responser. På de lukkede spørsmålene var responsene faste, og programmet kunne derfor skrives til å sortere «Svært uenig» for seg, «Uenig» for seg, og så videre. Programmet kunne dermed settes opp til å lese gjennom de relevante kolonnene i .xlsx-filen som Nettskjema.no lager etter en gjennomført spørreundersøkelse. En funksjon ble laget for å gjøre dette arbeidet (se algoritme 1). Responsene ble i Excel delt opp i to filer, en med responser fra LK06 og en med responser fra LK20. Ved å da skrive inn for eksempel «mervanskelig06 = [listfromexcel](#)(LK06, 17,

len_LK06)» er resultatet en liste med antallet som har svart hvert alternativ på spørsmålet i kolonne 17, i prosent for å kunne sammenligne de to gruppene med ulikt antall respondenter. Nest siste element i listen ble satt som tomt for å skape mellomrom mellom uenig/enig-responsene og ikke relevant. Ved å sette LK06- og LK20-responsene for ett gitt spørsmål ved siden av hverandre i et plot via pyplot lages figurer som sammenligner kullene, som vist i for figurer 4-11 til i kapittel 4.1.1.

```
def listfromexcel(file, column, length):  
    a = [0,0,0,0,0,0,0]  
    for row in range(0, length):  
        for col in file.iter_cols(column, column):  
            if col[row].value == 'Ikke relevant':  
                a[6]+=1/length  
            elif col[row].value == 'Svært uenig':  
                a[0]+=1/length  
            elif col[row].value == 'Uenig':  
                a[1]+=1/length  
            elif col[row].value == 'Nøytral':  
                a[2]+=1/length  
            elif col[row].value == 'Enig':  
                a[3]+=1/length  
            elif col[row].value == 'Svært enig':  
                a[4]+=1/length  
            else:  
                print('Woops, bug')  
                break  
    return(a)
```

Algoritme 1: Funksjon for å telle opp responser fra lukkede spørsmål

Ettersom responsene på de åpne spørsmålene var gitt i form av langsvar var det nødvendig å først kvantifisere dem for å kunne telle opp svarene på en måte lik den brukt

på de lukkede spørsmålene. Dette ble gjort ved å gå gjennom hvert enkelt svar og sortere dem i forskjellige kategorier. Prosessen for dette blir vist i figur 2 og figur 3, der førstnevnte viser kategoriene med tilhørende farge og figur 3 viser et utdrag av datasettet der denne fargeinndelingen er blitt gjort. Disse kategoriene ble brukt som grunnlag, men tilpasset spørsmålene der formuleringen ikke passet. For eksempel var «positivt» at respondenten svarte at det var lett om spørsmålet omhandlet vanskelighetsgrad, og om de svarte helhetlig et ja der det var ja/nei-spørsmål. Ved gjennomgangen av disse spørsmålene ble og enkeltresponser valgt ut og lagret med hele responsen i et separat datasett, som utgjør den kvalitative delen av spørreundersøkelsen. Disse responsene ble valgt på grunnlag av å være gode eksempler på forskjellige trender for hvert spørsmål, ble sortert etter spørsmål, og det ble notert kjønn, år fullført videregående skole, relevante fag som IT 1 eller 2, programmering X (PX), eller teknologi og forskningslære (ToF), og linje på universitetet (se vedlegg B). Et eksempel fra spørsmålet om hvor lett/vanskelig det har vært å lære programmering i fysikk er fra Kvinne 2023, BFY: «Det har vore lett å lære den grunnleggande programmeringa, men vanskeligare å faktisk bruke programmeringa sjølv til å løyse større problem.». Dette sitatet ble tatt med fordi det er et godt eksempel på trenden der flere respondenter svarer at de synes det har gått greit å lære programmering, men at det ikke har føltes nyttig i faget, av forskjellige grunner. Dette eksemplet ble vurdert som «positivt, uten forklaring», fordi det svarer at det har vært lett, men ikke gir noen begrunnelse for hvorfor det har vært lett.



Figur 2: Fargekoder for åpne spørsmål

| | 25 | 26 | 27 |
|----|---|---|--|
| 1 | Hva tenker du om at programmering er del av | Føler du at programmering er nyttig i fysikk | Basert på din erfaring, hvor lett/vans |
| 2 | Har ikke hatt programmering på vgs, så det er | Det er kanskje nyttig, men vanskelig å ta stillin | Kan ikke ta stilling til spørsmålet |
| 3 | Jeg tenker at det er veldig viktig. Programmeri | Ja. Jeg har forklart hvorfor i det forrige spør | Programmering i fysikk er vanskelige |
| 4 | Bra, arbeidsrelevant | Ja | Har ikke hatt det enda |
| 5 | Hele verden kjører på dataprogrammer så det | Hittil har det ikke vært så mye, men det er | Har ikke lært noe programmering i f |
| 6 | Programmering er et nyttig verktøy for å løse | Absolutt | Har ikke lært programmering i fysikk |
| 7 | Jeg har ikke hatt programmering i det hele tatt | Jeg føler det er veldig nyttig fordi det er ma | Har ikke lært noe programmering i fy |
| 8 | Det er positivt. | Ja. | Vanskelighetsgrad 7/10 |
| 9 | Jeg tenker det er nyttig for faget, så det er gan | Jeg tenker at det er veldig nyttig for å løse | Det er har vært krevende å lære prog |
| 10 | Positivt, kan være en nyttig ressurs i fysikken. | Ja | Ingen erfaring |
| 11 | Det er bra fordi det er kanskje mange som har | Ja, til ein viss grad. Sånn eg opplevde det v | Eg synst det var vasnkeleg, fordi det |
| 12 | Bra! I arbeidslivet virker det som programmer | Absolutt! Kanskje ikke så mye med tanke p | Har ikke lært programmering i fysikk |
| 13 | Jeg tror det er interessant for mange og nyttig | Ja jeg tror det kan være nyttig til å løse eks | Jeg har ingen tidligere erfaring med p |
| 14 | På videregående så tenker jeg det er viktigere | Ja, i form av at man kan lage programmer | Ikke hatt programmering i fysikk |
| 15 | Tror det er nødvendig, blir mer og mer releva | Det har jeg inntrykk av ja | Har aldri hatt det i sammenheng me |

Figur 3: Eksempel på fargekoding

For å lage grafiske representasjoner ble det laget en kopi av hvert av excel-arkene, der responsene ble erstattet med enkeltbokstaver, slik at det skulle være mulig å telle gjennom med et enkelt Python-program. For denne fordelingen ble de positive kategoriene slått sammen, og det samme med de negative. For den kvantitative analysen var hovedfokuset om hvorvidt responsene var positive eller negative, og det gav bedre lesbarhet med færre kategorier. Så ved å bytte ut de positive responsene med 'a', negative med 'b', og så videre, kunne en kode tilsvarende den som ble brukt for lukkede spørsmål brukes for å også kategorisere de åpne spørsmålene. Fra dette kunne en funksjon liknende den fra de lukkede spørsmålene brukes til å telle de forskjellige responsene (se algoritme 2) og sette verdiene inn i lister.

```

def listfromexcel(file, column, length):
    a = np.array([0,0,0,0,0])
    for row in range(0, length):
        for col in file.iter_cols(column, column):
            if col[row].value == 'a':
                a[0]+=1
            elif col[row].value == 'b':
                a[1]+=1
            elif col[row].value == 'c':
                a[2]+=1
            elif col[row].value == 'd':
                a[3]+=1
            elif col[row].value == 'e':
                a[4]+=1
    return(a)

```

Algoritme 2: Funksjon for å telle opp responser fra åpne spørsmål

Disse listene ble brukt for å lage kakediagram, og det var derfor ikke nødvendig å få ut svarene i prosent. Det var heller ingen 'ikke relevant'-kategori, eller tilsvarende. Derfor blir listen en ren opptelling, med der tallene representerer antallet som har svart innenfor hver kategori. Kakediagrammene kunne dermed sidestilles for å sammenligne LK20 og LK06, på samme måte som for de lukkede spørsmålene. Kakediagrammene laget fra denne koden er figurer 12-17 i kapittel 4.1.2.

3.2 Intervju

Intervjuene ble gjennomført en-til-en, og fulgte en semi-strukturert fremgangsmåte. Det vil si at det finnes en intervjuguide, men den trengs ikke å følges til punkt og prikke, og brukes mer som en sjekkliste med temaer man vil innom (Robson & McCartan, 2016). Dette gir fordelen at man har mulighet til å følge interessante sidespor og spørre dypere på temaer som skulle dukke opp underveis i intervjuet, men at man også har en plan for hvor man skal gå videre når man er ferdig med et tema,

Spørsmålene i disse intervjuene var designet for å være åpne og brede, og inviterte læreren til å utdype om egne erfaringer, som også åpnet opp for nye spørsmål tilknyttet disse refleksjonene underveis i intervjuet. Spørsmålene var av typen generelt om situasjonen med programmering i fysikkfaget, om struktur, kultur, og elevrespons. Problemstillingen omhandler elevene, så hensikten med intervjuet er å få syn fra en andrepert som har oversikt over pensum, over kompetansemål, og som har fått observert elevenes opplevelser utenfra.

Det ble gjennomført intervju av 5 fysikklærere, hver fra forskjellige videregående skoler og fra totalt 3 forskjellige fylker. Intervjuene ble også gjennomført som en del av et annet fag der jeg skrev en oppgave i emnet RFEL3100 om hvordan lærere har blitt kurset og forberedt på å undervise i programmering i fysikk i forkant av implementeringen av LK20. Hvert av intervjuene har derfor vært todelt, der første del av intervjuet var rettet mot den oppgaven og andre del av intervjuet var rettet mot masteroppgaven.

Lærerne hadde fra 2 til 34 års erfaring, alle underviste fysikk og matematikk, det var også en av lærerne som underviste i teknologi- og forskningslære, og en som også underviste i naturfag (se tabell 4). 3 av intervjuene ble holdt digitalt, de resterende 2 ble holdt fysisk. Varigheten på intervjuene var fra 16 til 26 minutter. For alle intervjuene ble det gjort lydopptak via ekstern lydopptaker, som ble behandlet som fortrolig data og håndert etter NTNU sine retningslinjer for konfidensialitet (NTNU, Klassifisering av informasjonsverdier; NTNU, Lagringsguide). Lydopptakene ble deretter transkribert manuelt over i Word-filer. Transkripsjonen ble gjort i to arbeidssett. Først ble delene av hvert intervju relevant for metodefaget på høsten gjennomført og analysert, høsten 2023. Dette ble gjort i forbindelse med en innlevering i faget, og ble gjort separat fra masteroppgaven. Vår 2024, semesteret der jeg skriver master, ble tilsvarende gjort for delene relevante til masteroppgaven. Transkripsjonen ble forsøkt gjort så direkte som mulig, slik at pauser, latter, gjentakende ord, og tilsvarende ble inkludert i tekstfilene. Navn på skoler, fylker, og andre identifiserbare opplysninger ble erstattet i teksten med «<skole>», «<fylke>», og tilsvarende.

Tabell 4: Antall år med erfaring og relevante fag hos informanter

| | Antall år | Relevante fag |
|---------|-----------|---|
| Lærer A | 15 | Fysikk, Matematikk |
| Lærer B | 2 | Fysikk, Matematikk |
| Lærer C | 13 | Fysikk, Matematikk, Teknologi og Forskningslære (ToF) |
| Lærer D | 34 | Fysikk, Matematikk, Naturfag |
| Lærer E | 10 | Fysikk, Matematikk |

3.2.1 Kvalitativ analyse

Intervjuene ble transkribert ved at jeg hørte over lydfilene og skrev inn i Word. Transkriberingen av intervjuene ble så lastet inn i NVivo, et program som hjelper med å holde oversikt når man koder en tekst. Deretter ble intervjuene analysert ved bruk av tematisk analyse. Tematisk analyse ble valgt fordi det tilbyr en induktiv tilnærming med

fleksibilitet i forhold til datamaterialet (Braun & Clarke, 2006). Datamaterialet ble kodet etter fremgangsmåten beskrevet av Braun & Clarke (2006), og har fulgt deres steg-for-steg guide som listes slik:

1. Bli kjent med datasettet
2. Generer initielle koder
3. Søk etter temaer
4. Revider temaer
5. Definer og navngi temaer
6. Produser en rapport

Å bli kjent med datasettet ble gjort gjennom transkriberingen. Ved å gjøre dette uten noen automatisk generert lyd-til-tekst måtte jeg høre gjennom hver seksjon flere ganger for å få skrevet inn og rettet opp tekst, som også gav meg oversikt over innholdet i hvert intervju. Dette er også metoden Braun & Clarke anbefaler for datasett med verbal data (2006, s 87-88).

NVivo ble brukt til å holde kodingsarbeidet organisert, og arbeidet ble gjort ved at jeg gikk igjennom intervjuene hver for seg for å generere de initielle kodene. Gitt at jeg fra steg 1 var kjent med datasettet hadde jeg allerede en rekke koder jeg visste ville dukke opp, men var også bevisst på å gi lik oppmerksomhet til hver dataenhet, og jobbet derfor likevel systematisk fra begynnelse til slutt. NVivo sorterer automatisk koder både fra hvert enkelt intervju og fra alle. Heller enn et tematisk kart som vist i Braun & Clarke (2006) brukte jeg denne oversikten for å sortere koder. Med denne sorteringen var det lett å ha full oversikt over kodene, som var nyttig for å finne temaer. Etter revidering, definering, og navngiving endte jeg opp med temaene som vist i tabell 5.

Tabell 5: Temaer, koder, og eksempler på sitater

| Tema | Koder | Sitater |
|------------------|-------------------------------|---|
| Fysikk i sentrum | Programmere for fysikkens del | Men at mitt mål, eller min drøm var jo at jeg ikke skulle undervise programmering, jeg skulle -bruke- programmering. Og hvis jeg bare skulle bruke programmering så synes jeg egentlig at det er, det er null stress. Og det å faktisk bruke programmeringa til noe som er nyttig å bruke programmering til, det synes jeg er litt viktig. Så for eksempel å tegne grafer i Python synes jeg ikke er så viktig, for det gjør du lettere i Geogebra, med noen unntak selvfølgelig. Sånn datasett og sånn. Men bare, uhh, bruk det til det det på en måte er |

| | | |
|--------------------|--------------------------|---|
| | | hensiktsmessig å bruke det til. Da tror jeg og elevene synes at det er interessant. |
| Manglende grunnlag | Mangler fra ungdomsskole | De har tilnærmet ingenting fra ungdomsskolen, etter min opplevelse. Eh, så alt som blir gjort i 1T da, før de kommer til fysikk. |
| | | For det er så stor forskjell på ungdomsskoler, og veldig mange ungdomsskoler bortprioriterer programmering for dem har mer enn nok å, med tanke på elevmassen sin, å komme seg gjennom alle sammen på det som er, som man absolutt må. |
| | Mangler fra 1T | Men de fleste kan jo ikke så mye programmering når dem er ferdig med 1T, dem kan jo ikke så mye. |
| | | Nei, det handler vel litt om at det jeg trenger at dem skal kunne, i fysikken, er ikke nødvendigvis det mattelæreren trenger at dem skal kunne i matematikken. Sånn at det som er av programmeringsopplæring i matematikken fokuserer ikke på, ja, å ta inn datapunkter og plote dem, og å modelere grafer og, det, det skjønner jeg. Det er jo ganske kjedelig matematikk, funksjonsdrøfting liksom. Men, men det er det jeg har bruk for at dem kan, også har jeg ikke helt tid til å lære dem det sjølv så jeg har håpet på at noen andre skulle ta det. |
| Manglende tid | Fullt fag | Og 1T er stappa fullt med ting så det blir litt sånn liten ekstrating, og så er det mye man må gjøre i VG2 da |
| | | Og det stemmer nok til for det er, vi har for lite tid satt av til å lære noe veldig stort egentlig. |
| | Tid på det grunnleggende | Så jeg føler litt at det ikke blir gitt noe mer tid til det. Det er liksom enda en ting man må lære seg, og da blir det litt sånn ofte, |

det blir sånn oppsamling at man, jeg vet at flere lærere gjør, at det blir sånn at man må samle opp programmeringsbiten

| | | |
|---------------------------|-----------------------------|--|
| Elevenes nivå | Stor spredning | <p>Så det er også en sånn kjempekløft i klasserommet på at det er ingen som, det hanger avgårde på sånn passe greit for noen. Det går helt skitt eller helt amazing. Sånn at det er litt sånn vanskelig pedagogisk utfordring, at du snakker til to grupper som er så -veldig- langt fra hverandre.</p> <p>Men det er sånn, ja det varierer jo helt hva slags lærer dem har, hva slags skole dem går på. Så store forskjeller ja. Og de synes det er vanskelig og annerledes. Og det stemmer nok til for det er, vi har for lite tid satt av til å lære noe veldig stort egentlig.</p> |
| | Vanskelig og annerledes | <p>Men så er jo arbeidet da, for jeg har jo lyst til å programmere for fysikken sin del, at det bare skal være et verktøy, at det skal være like lett å si at «Ok, men det her håndterer vi med en løkke.» som å si «Ok, men det her plotter dere i Geogebra.», og elevene er ikke der. I det hele tatt.</p> |
| Utvikling av undervisning | Utvikling av undervisning | <p>Men når jeg har hatt det et par ganger til og får den here basisen så har man mer kapasitet til å ta inn og forbedre, ta inn mer ting på de her, men det krever litt tid. Det er liksom, det er hovedbøygen for mange da. At liksom ny læreplan, nye kompetansemål, vi skal jobbe med teorien på en litt annen måte, litt andre vinklinger, så den så har man nok med det de første to-tre årene.</p> |
| | Utvikling fra ungdomsskolen | <p>Ja, nå er det to ganger jeg har hatt 1T da. Første gangen da hadde jeg en, eh, studieforberedende klasse med KD og MK, og musikk, dans, drama. Og dem var jo helt blanke. Dem hadde ikke hatt noe programmering fra før. Så det var jo, det var jo liksom å gå gjennom det viktigste.</p> |

Nå er det litt bedre, nå blir det litt mer oppgaver de har løst nå innen matematikk da og kommet et steg videre.

Uhm... om det er noen forskjell? Ikke så veldig stor forskjell, faktisk. Eh, den burde ha vært mye større. For bare når man begynner med løkker så er det mange som er et stort spørsmålstegn. Og det er vanskelig å drive med programmering uten å komme til løkker og, i sånn rimelig, du har selvfølgelig kontrolstrukturer og if-setninger og sånt før det, men ehh.. Nei, jeg føler ikke basisen er så veldig sterk nei.

Fra sitatene i tabell 5 kommer det frem at det er noe overlapp mellom kodene. For eksempel under '*Stor spredning*' snakkes det om ferdigheter fra ungdomsskolen, mer spesifikt at spredning i kunnskapsnivå grunnes delvis at det er store forskjeller fra ungdomsskolen, som overlapper med temaet om manglende grunnlag. Braun & Clarke nevner i sin sjekkliste (2006, s 96) at temaer skal være distinkte. Likevel har jeg valgt å ha disse som separate temaer, gitt at begge er relevante også hver for seg for videre diskusjon. Utenom overlapp ble kriteriene sett over og sjekklisten ble brukt til å evaluere når jeg skulle si meg ferdig med analysen.

3.3 Ethiske betraktninger og personvern

Før intervjuprosessen begynte ble prosjektet meldt inn til Sikt, som godkjente opplegget. Siden intervjuene inneholder lydopptak, som gjør det mulig å gjenkjenne informantens stemme, ble lydopptakene fra intervjuene behandlet som fortrolig data og håndtert etter NTNUs retningslinjer. Alt av stedsnavn, navn på skoler, eller tilsvarende ble anonymisert slik at det ikke skal kunne gå an å identifisere noen skoler, klasser, eller individer. I forkant av hvert intervju ble informanten tilsendt et informasjonsskriv (se vedlegg C), og et samtykkeskjema som de måtte gi skriftlig tilbakemelding på at de godkjente. Spørreundersøkelsen ble gjennomført via Nettskjema.no, i tråd med NTNUs retningslinjer. Ingen personopplysninger ble samlet inn, og undersøkelsen var derfor ikke søknadspliktig.

3.4 Validitet

Spørreundersøkelsen har som mål å undersøke fysikkelevers opplevelser, men respondentene er hentet fra en forelesning på NTNU om mekanisk fysikk. Dette betyr at resultatene som er samlet inn ikke kan anses som representative for en fysikk-klasse i videregående skole, siden ikke alle fysikkelevne velger å gå videre med faget ved universitetet. Men en forelesning i mekanisk fysikk garanterer at respondentene har erfaring med fysikkfaget, gitt at alle studieprogrammene som har emnet også har krav om fysikk for å søke opptak, og også kommer til å bruke programmering fremover, som gir relevans til spørreskjemaet for respondentene. En annen fordel med dette utvalget er at en

forelesningssal inneholder individer med bakgrunn fra forskjellige skoler fra hele Norge, som betyr at undersøkelsen når et langt bredere spenn av erfaringer enn om for eksempel jeg hadde undersøkt en håndfull videregående skoler i Trøndelag. Samtidig bygges mixed methods-delen på å sammenligne kull fra LK20 med kull fra LK06, som gir en viss grad av utjevning av skjevheter ettersom kontrollgruppen også består av fysikkstudenter. I tillegg kan lærerintervjuene brukes til å komplimentere datasettet, og sørge for at forskningsspørsmålene blir besvart med videregående skole i fokus.

Siden jeg selv hadde lite erfaring med å holde et intervju var jeg til å begynne med usikker på egen teknikk under de første intervjuene. Etter hvert som jeg holdt flere merket jeg at jeg utviklet min egen teknikk og at å holde dem ble en læringsprosess. Dette betyr videre betyr at de senere intervjuene ble holdt på en annen måte enn de tidligere intervjuene. I tillegg så kom det frem underveis i intervjuene nye temaer jeg ønsket å undersøke nærmere. Også så kan min egen mangel på erfaring ha gjort at min kroppsholdning eller formulering kan ha påvirket intervjuobjektene en vei eller annen (Robston & McCartan, 2016). Dermed kan ikke intervjuene nødvendigvis stilles perfekt opp mot hverandre.

Mye av teorien brukt i oppgaven har gjort en grundig analyse ved å følge spesifikke modeller, for eksempel slik Storesund (2022) følger EV-modellen (Eccles & Wigfield, 2002) og analyserer sin data opp mot hver av komponentene i denne. For min oppgave har målet vært å få et overblikk over en situasjon som er i utvikling, heller enn fullstendig statistisk analyse, ettersom LK20 er implementert såpass nylig i den norske skolen og at både lærere og elever fremdeles er i prosessen med å tilpasse seg ny læreplan. Derfor brukes modeller som EV-modellen og Bott et al. sine temaer mer som veiledning og inspirasjon.

4. Resultater

Kapitlet om resultater vil ta for seg innsamlet data sortert etter type; Spørreundersøkelse, delt inn i lukkede spørsmål og åpne spørsmål, og intervju. Det vil bli presentert en detaljert gjennomgang av eksempelspørsmål, for å vise hvordan datasettet ser ut, mens de resterende blir oppsummert med en tabell. Siste del av kapitlet er en beskrivelse av temaer og koder fra intervjuet.

4.1 Spørreundersøkelse

Svarene fra spørreundersøkelsen brukes til å vurdere hvordan respondentene har opplevd programmering innen fysikk. Ved å sammenligne LK20-kullet til de andre kullene kan det første forskningsspørsmålet, om innføringen av LK20s påvirkelse av elevers holdning til programmering innen fysikk, besvares. Resultatene fra undersøkelsen gjennomgås henholdsvis etter de lukkede spørsmålene, inspirert av Storesund (2022) og EV-modellen (Eccles & Wigfield, 2002), og så de åpne spørsmålene, inspirert av Bott et al. (2019).

Første del av spørreundersøkelsen viser at 100% av respondentene fra LK20 har hatt matematikk R2, og 99% av LK06 har. Den ene respondenten som ikke har hatt R2 svarer at den har hatt 1T og S1. Basert på dette antas det at alle respondentene har vært gjennom 1T og dermed blitt undervist for i digitale hjelpemidler relevant for fysikk, som presentert i 4.1.1.

4.1.1 Motivasjon og mestringsforventning

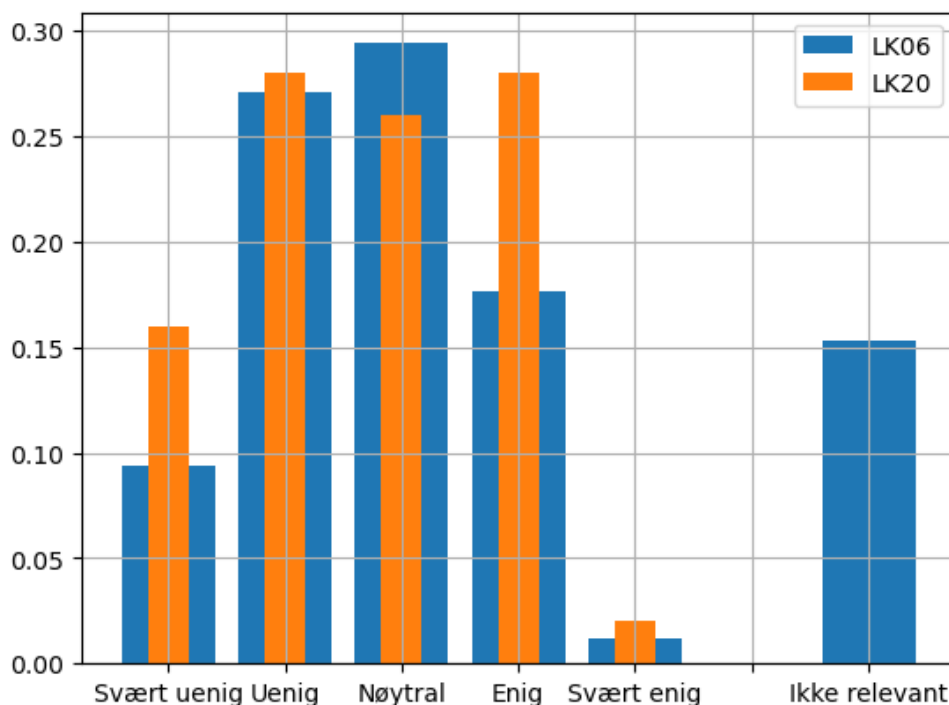
Tabell 6: Responser på lukkede spørsmål, i prosent

| Spørsmål | Kull | Svært uenig | Uenig | Nøytral | Enig | Svært enig | Ikke relevant |
|---|------|-------------|-------|---------|------|------------|---------------|
| Programmeringsoppgaver har vært mer krevende enn andre typer oppgaver. | LK20 | 16 | 28 | 26 | 28 | 2 | 0 |
| | LK06 | 9 | 27 | 29 | 18 | 1 | 15 |
| Jeg har foretrukket å jobbe med programmeringsoppgaver i forhold til andre oppgaver. | LK20 | 18 | 24 | 38 | 14 | 6 | 0 |
| | LK06 | 14 | 16 | 34 | 16 | 2 | 16 |
| Programmeringen jeg har lært i fysikk har vært tilstrekkelig for å løse oppgaver i faget. | LK20 | 6 | 20 | 22 | 52 | 0 | 0 |
| | LK06 | 17 | 6 | 16 | 0 | 0 | 60 |
| Programmering kan være nyttig for å løse problemer i fysikk. | LK20 | 0 | 0 | 2 | 22 | 76 | 0 |
| | LK06 | 2 | 1 | 9 | 41 | 42 | 4 |

| | | | | | | | |
|--|------|----|----|-----|----|----|-----|
| Jeg er motivert for å lære programmering fremover. | LK20 | 0 | 4 | 12 | 42 | 42 | 0 |
| | LK06 | 0 | 2 | 11 | 52 | 35 | 0 |
| Fysikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper. | LK20 | 16 | 16 | 8.0 | 42 | 16 | 2.0 |
| | LK06 | 64 | 5 | 6 | 1 | 0 | 24 |
| Matematikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper. | LK20 | 12 | 10 | 12 | 39 | 23 | 4 |
| | LK06 | 56 | 8 | 5 | 8 | 1 | 21 |
| Programmeringen jeg har lært i matematikk kan være nyttig for å arbeide med fysikk. | LK20 | 2 | 6 | 8 | 48 | 28 | 8 |
| | LK06 | 18 | 1 | 7 | 14 | 5 | 52 |

Tabell 6 viser responsene fra LK20 og LK06 til hver av påstandene i andre del av spørreundersøkelsen, lukkede spørsmål. Enkelte av spørsmålene har liten forskjell mellom læreplanene, mens for andre er det veldig ulike responser mellom LK20 og LK06. Responsene blir fremover presentert i mer detalj, og med en tilhørende graf (laget gjennom programmet beskrevet i kapittel 3.1.3 og algoritme 1) over responsene til hvert spørsmål.

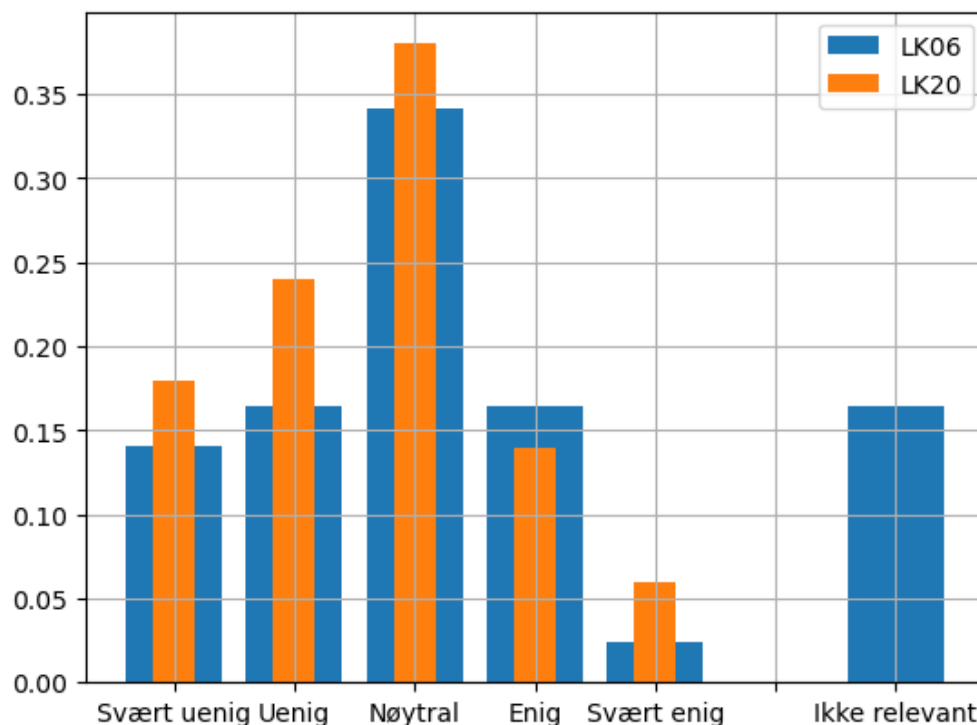
Programmeringsoppgaver har vært mer krevende enn andre typer oppgaver.



Figur 4: Responser til påstand om kostnad

Denne påstanden ser på konstruert kostnad, og er uttrykt slik at de som er mer enige i påstanden vurderer programmering til å ha høyere kostnad. På dette spørsmålet svarer LK20 mer på begge ytterkanter, altså mer at de er svært uenige og også at de er svært enige. Ingen fra LK20 svarer at ikke synes spørsmålet er relevant, mot 15% fra LK06. De fleste fra LK06 har altså en eksplisitt formening om hvorvidt de synes programmering har vært mer krevende. Dersom vi ser bort i fra de 15% som har svart ikke relevant har LK06 en prosentmessig høyere andel som svarer nøytral (35%, mot LK20 sin 26%) og uenig (32% mot LK20 sin 28%). Altså vurderer LK20 programmering til å ha høyere kostnad enn LK06, men har også en andel som vurderer kostnaden som langt lavere. Dette tyder på en splitt, der LK20 både har flere som synes at programmeringsoppgaver er krevende men også flere som synes de er svært enkle. Likevel kommer det og frem fra resultatene at en stor majoritet, hos både LK06 og LK20, ikke bruker ytterpunktene for å svare. Bare 2% og 1% svarer svært enig, for henholdsvis LK20 og LK06. Tilsvarende er det en større gruppe men fremdeles en minoritet som svarer at de er svært uenige, 16% og 9%, henholdsvis. Altså for de aller fleste respondenter, om de tilhører LK20 eller LK06, er det ikke en veldig stor forskjell på programmeringsoppgaver og andre typer oppgaver i kostnad.

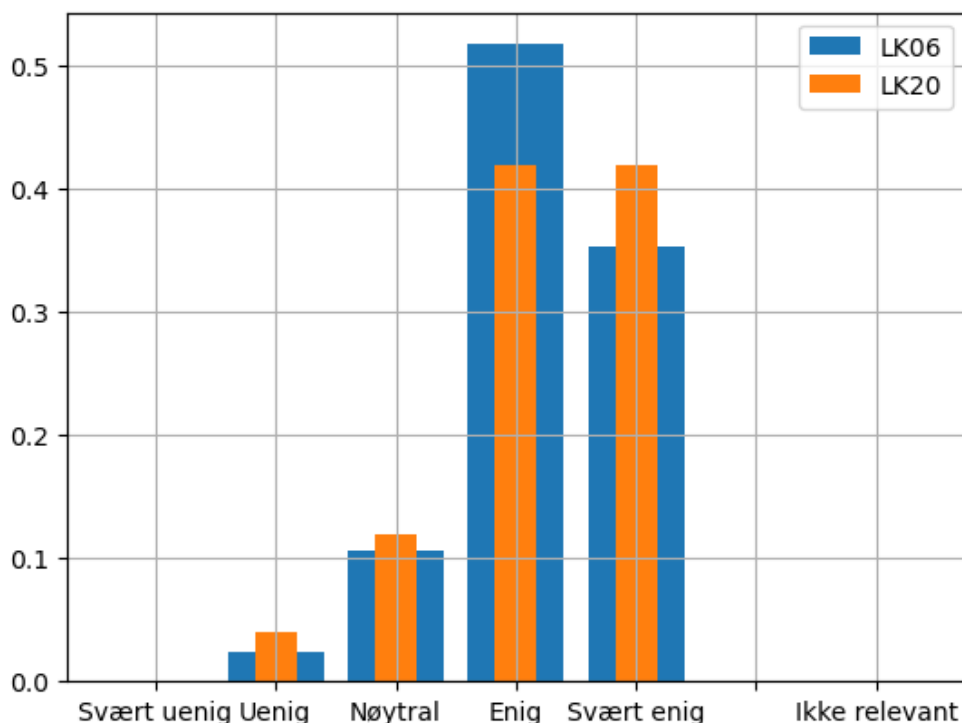
Jeg har foretrukket å jobbe med programmeringsoppgaver i forhold til andre typer oppgaver.



Figur 5: Responser til første påstand om interesseverdi

Dette spørsmålet er et av to som ser på konstruert interesseverdi, formulert slik at mer enig betyr høyere grad av interesse. Som på forrige spørsmål svarer LK20 mer på på både svært uenig (18%) og svært enig (6%) enn LK06 (henholdsvis 14% og 2%). Dermed kommer det frem mer tegn på en slik splittelse som kom frem ved forrige påstand, der en liten gruppe virker til å synes programmering er veldig greit å jobbe med mens en litt større gruppe synes liker det langt mindre enn andre typer oppgaver. Igjen svarer majoriteten moderat (66% uenig, nøytral, eller enig for LK06, 76% for LK20) men svarene viser at det er flere som svarer uenig eller svært uenig enn det er som svarer enig eller svært enig. Altså svarer respondentene helhetlig her at de ikke foretrekker å jobbe med programmeringsoppgaver.

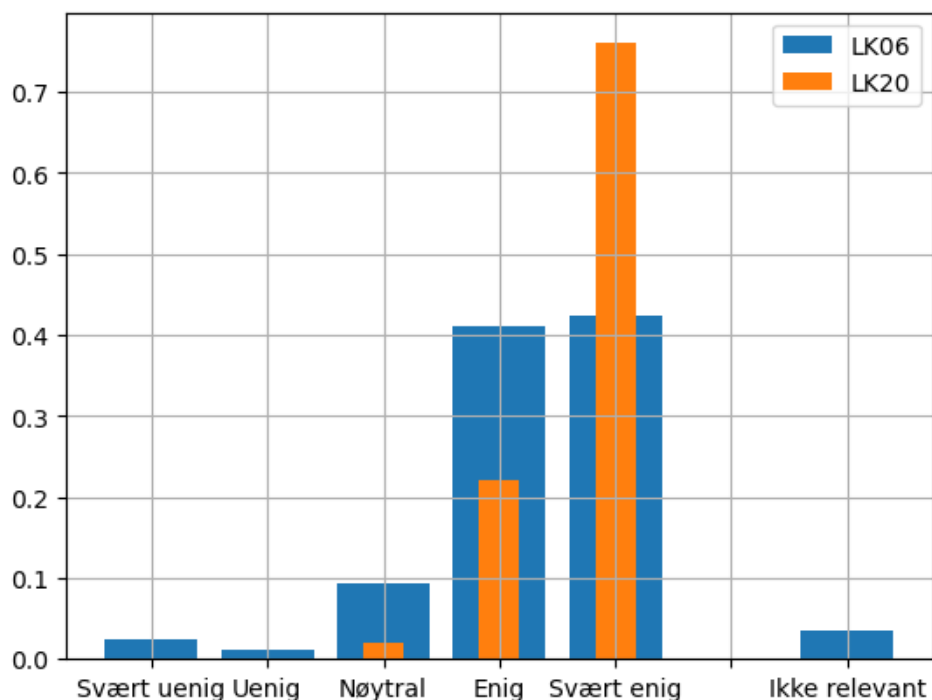
Jeg er motivert for å lære programmering fremover.



Figur 6: Responser til andre påstand om interesseverdi

Responsene på denne påstanden viser for både LK20 og LK06 er opplevd motivasjon høy. Det er også ingen som svarer at dette spørsmålet ikke er relevant, som kan tyde på at samtlige respondenter seg for seg å lære mer programmering fremover. Dette kan ha en sammenheng med at respondentene er førsteårs-studenter, der det er planlagt å bruke mer programmering i fag fremover, og det er tenkelig at en klasse fra fysikk 2 på videregående ville gitt et annet resultat. Det står også i kontrast med responsene til den første påstanden om interesseverdi, der det helhetlige bildet var noe negativt. Respondentene har foretrukket «vanlig» oppgaveløsning fremfor å jobbe med programmeringsoppgaver, men en stor majoritet er enig eller svært enig i at de er motiverte for å lære programmering fremover. Det vil komme frem tydeligere senere i oppgaven at dette kan være fordi det ikke nødvendigvis var selve programmeringen de ikke likte å jobbe med, men hvordan programmeringsdelen ble gjennomført i timene.

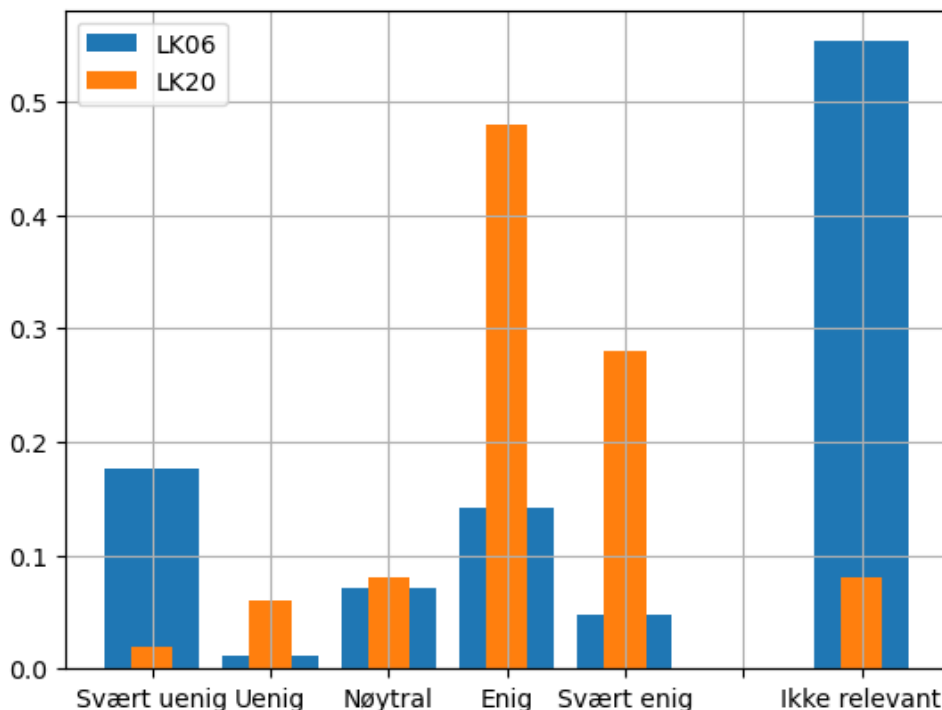
Programmering kan være nyttig for å løse problemer i fysikk



Figur 7: Responser på første påstand om nytteverdi

I likhet med forrige påstand viser responsene på denne at både LK06 og LK20 vurderer programmering til å ha høy nytteverdi, men LK20 gir en langt mer positiv respons. En stor majoritet svarer svært enig (76%) og ingen svarer svært uenig eller uenig. Dette gir videre indikasjon på respondentene ser på programmering som positivt selv om de ikke nødvendigvis har hatt gode erfaringer med det selv på videregående skole. Videre kommer det frem at selv om de som har hatt programmering som fysikkpensum sier i større grad at de ikke har foretrukket å jobbe med programmering mener de likevel i større grad at programmering kan være nyttig innenfor fysikk. Dette tyder på at respondentene klarer å se programmeringen i et større lys enn bare det de selv har hatt i undervisning, og kan derfor vurdere en større nytteverdig helhetlig enn hva de selv har fått ut av det. Dette kommes tilbake til under presenteringen av åpne spørsmål, kapittel 4.1.2, der kvalitativ data kan gi mer utfyllende informasjon om denne opplevelsen.

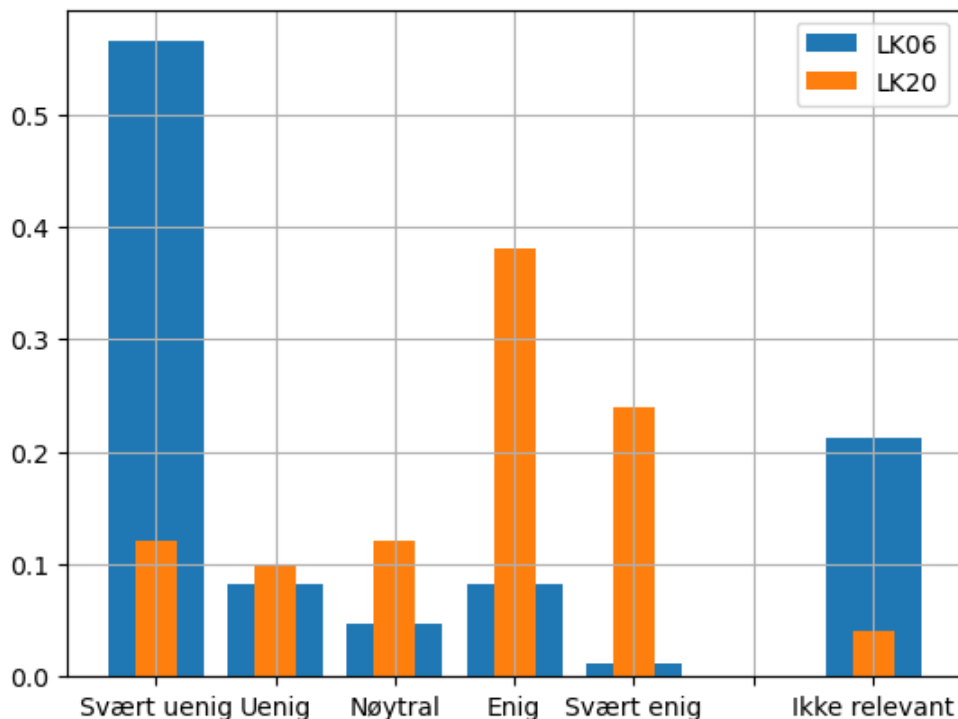
Programmeringen jeg har lært i matematikk kan være nyttig for å arbeide med fysikk



Figur 8: Responser på andre påstand om nytteverdi

Denne påstanden henger tett sammen med temaet om fysikk i sentrum, beskrevet i kapittel 4.2. Slik løpet er lagt opp på videregående skal 1T gi elevene en innføring i tekstprogrammering. Fysikkfaget skal bruke det elevene har lært i 1T for å gi dem fysikkoppgaver der fysikken, ikke programmeringen, skal stå i sentrum. De fleste respondenter fra LK20 er enig (48%) eller svært enig (28%) at programmeringen i matematikk kan være nyttig, men det er også en andel som svarer uenig (6%) eller svært uenig (2%). Det er med det til sammen 8% som mener at programmeringen de har lært i matematikken ikke kan være nyttig for fysikk, og enda 8% som vurderer spørsmålet som ikke relevant. Dette kan tyde på at selv om de fleste har fått den opplæringen de trenger er det enkelte som har måttet møte programmering i fysikkfaget uten det faglige grunnlaget som de trenger for å kunne gjøre oppgavene. LK06 svarer i hovedsak ikke relevant (55%) eller svært uenig (18%), men det er også enkelte som svarer enig (14%) eller svært enig (5%). Det er ikke mulig å finne forklaringer på hva de har ment med dette fra de åpne spørsmålene. De fra LK06 som har svart enig eller svært enig på denne påstanden har ellers svart på de åpne spørsmålene at de ikke har hatt programmering, men det kan spekuleres om at det siktes til å bruke programmer som Geogebra eller Excel.

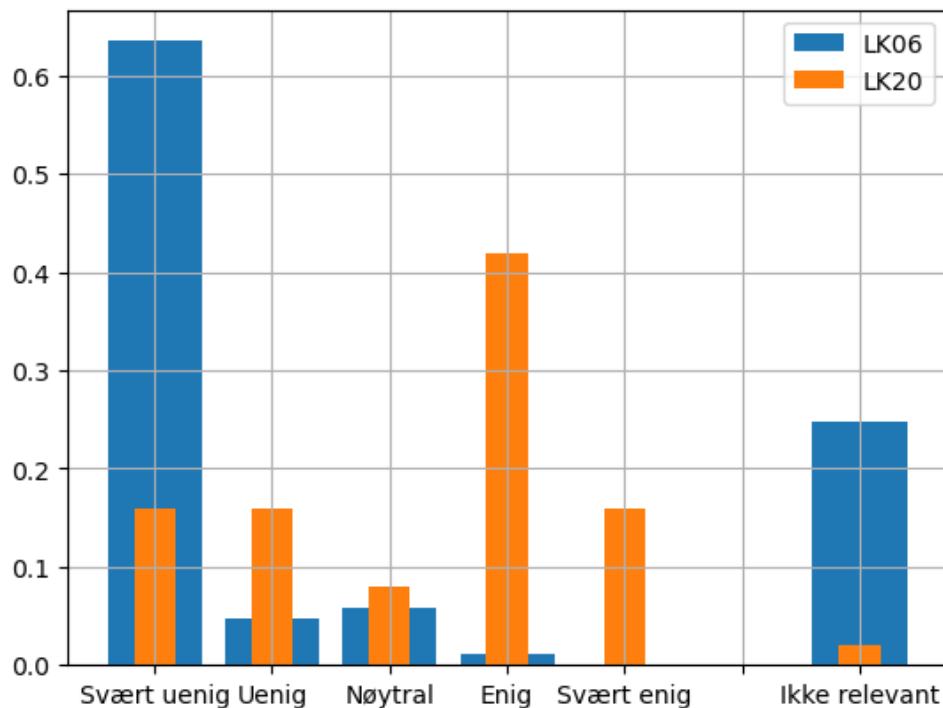
Matematikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper



Figur 9: Responser på første påstand om forkunnskaper

Responsene for denne påstanden viser at selv om en majoritet av LK20 er enig (38% eller svært enig (24%) i at matematikkfaget har bidratt til deres samlede programmeringskunnskaper er det og en betydelig andel der dette ikke har vært tilfellet, som svarer seg svært uenig (12%), uenig (10%) eller ikke relevant (4%). Dette kan tolkes som at opp til 26% av respondentene fra LK20 opplever at de ikke har hatt skikkelig opplæring i programmering fra 1T. Interessant nok er det flere som opplever at programmeringsopplæringen fra matematikk har vært nyttig for fysikk enn som opplever at matematikkfaget på videregående har bidratt til deres programmeringskunnskaper. Dette kan bety at de har tatt med programmering de har lært utenom skolen, men igjen er det ingen tydelig begrunnelse som kan finnes i de åpne spørsmålene, og flere som har svart forskjellig på disse påstandene har og svart at de ikke har lært programmering utenom vgs. Sammenlignet med LK06 ser vi likevel at den nye læreplanen har hatt en effekt på programmeringen i matematikk. LK06 svarer overveldende uenig (56%) eller ikke relevant (21%).

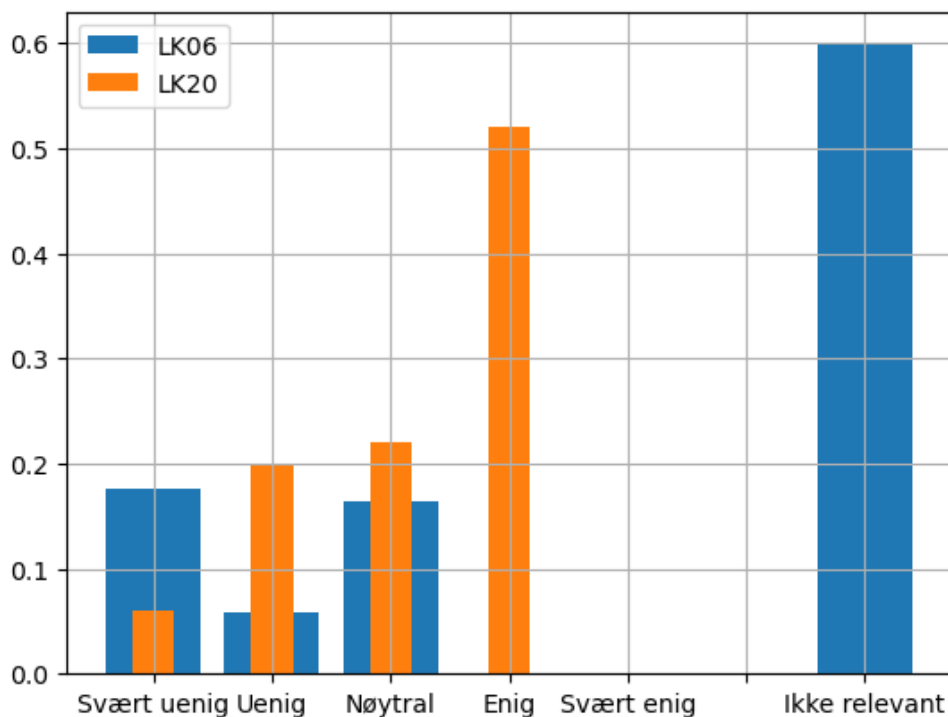
Fysikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper



Figur 10: Responser på andre påstand om forkunnskaper

Denne påstanden har i hovedsak veldig like responser som første påstand om forkunnskaper. Litt flere fra LK20 svarer uenig (16%) eller svært uenig (16%), og litt færre svarer svært enig (16%) men flere svarer enig (42%). Selv om målet fra lærernes side har vært at programmeringen i fysikk skal være et verktøy de alt har fra matematikken tydes det på at elevene har opplevd fagene som omtrent like når det kommer til å lære dem programmering. For LK06 er responsen her mer negativ enn for tilsvarende påstand om matematikkfaget, som kan tyde på at endringen mot mer programmering har vært opplevd som større for elevene innen fysikk enn matematikk.

Programmeringen jeg har lært i fysikk har vært tilstrekkelig for å løse oppgaver i faget



Figur 11: Responser på påstand om mestringsforventning

Påstanden om mestringsforventning viser store forskjeller mellom LK06 og LK20. Ingen er svært enig i påstanden, men der også ingen fra LK06 er enige er like over halvparten av respondenter fra LK20 svarer seg enig (52%). Etter dette kommer nøytral (22%), uenig (20%) og svært uenig (6%) i sykende antall svar. For LK06 svarer en majoritet at påstanden ikke er relevant for dem (60%). Av de resterende svarer flest svært uenig (17%), etterfulgt av nøytral (16%) og uenig (6%). Gitt at programmering, slik det er definert i spørreundersøkelsen og ble beskrevet for respondentene før de begynte den, ikke har vært pensum i fysikk for LK06 er det kanskje ikke overraskende at mange av dem svarer ikke relevant eller svært uenig. For LK20 er det totalt en moderat positiv respons, men det er ingen som er svært enige og det er enkelte som er svært uenige. Denne responsen tyder på at det stort sett har gått, men det er ingen som rapporterer at fysikkfaget har vært med på å få dem til å føle at de skikkelig mestrer programmering, mens enkelte føler at fysikkfaget ikke i det hele tatt har vært tilstrekkelig for å løse oppgavene. Dette temaet tas opp igjen i neste delkapittel, 4.1.2.

Oppsummering lukkede spørsmål

Helhetlig kommer det frem at det er stort samsvar i spørsmålene som omhandler interesseverdi, og kostnad, og store forskjeller i spørsmålene som omhandler forkunnskaper og mestringsforventning. For nytteverdi avhenger det av spørsmålet. Nyttverdi generelt er det generell enighet om, men nytteverdien fra det de har lært på videregående skole er det langt mer positivt fra LK20-kullet. Altså er en tydelig forskjell i opplevelser fra videregående, mens en mindre forskjell på motivasjonsfaktorene. Likevel kommer det frem

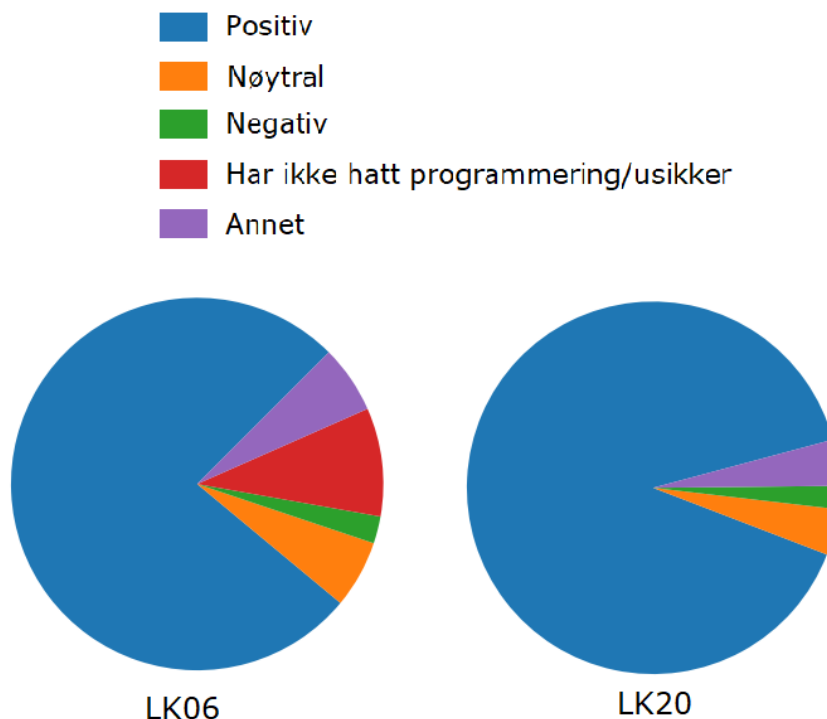
at selv om LK20 vurderer programmering til å ha noe høyere kostnad enn LK06 gjør, rapporterer de likevel en noe høyere opplevd motivasjon og opplevd nytteverdi for programmering. Eksponering for programmering har tilsynelatende gjort at de innser at det krever mye arbeid, men også at de innser at det er både spennende og nyttig.

4.1.2 Holdninger og erfaringer

Tabell 7: Åpne spørsmål og kvantifiserte responser, i prosent

| Spørsmål | Kull | Positiv | Nøytral | Negativ | Har ikke hatt/usikker | Annet |
|--|------|---------|-----------|-----------------|-----------------------|-------|
| Hva tenker du om at programmering er en del av fysikkfaget? | LK20 | 45 | 2 | 1 | 0 | 2 |
| | LK06 | 65 | 5 | 2 | 8 | 5 |
| Føler du at programmering er nyttig i fysikkfaget? | LK20 | 48 | 1 | 0 | 0 | 1 |
| | LK06 | 71 | 4 | 1 | 8 | 1 |
| Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk? | LK20 | 22 | 8 | 17 | 3 | 0 |
| | LK06 | 8 | 2 | 11 | 63 | 1 |
| Hvordan har det å lære programmering påvirket din fysikkforståelse? | LK20 | 9 | 7 | 31 | 3 | 9 |
| | LK06 | 4 | 1 | 19 | 59 | 2 |
| Når du arbeider med programmering, syntes du det er viktig å forstå hver kodeline? Hvorfor eller hvorfor ikke? | LK20 | 41 | 2 | 5 | 1 | 1 |
| | LK06 | 65 | 2 | 6 | 11 | 1 |
| | | Sosialt | Internett | Annen utdanning | Ja, uspesifisert | Nei |
| Har du lært programmering fra andre steder enn i skolen? Hvilke? | LK20 | 8 | 6 | 0 | 6 | 20 |
| | LK06 | 3 | 14 | 5 | 10 | 42 |

Hva tenker du om at programmering er del av fysikkfaget?



Figur 12: Responser til 1. åpne spørsmål

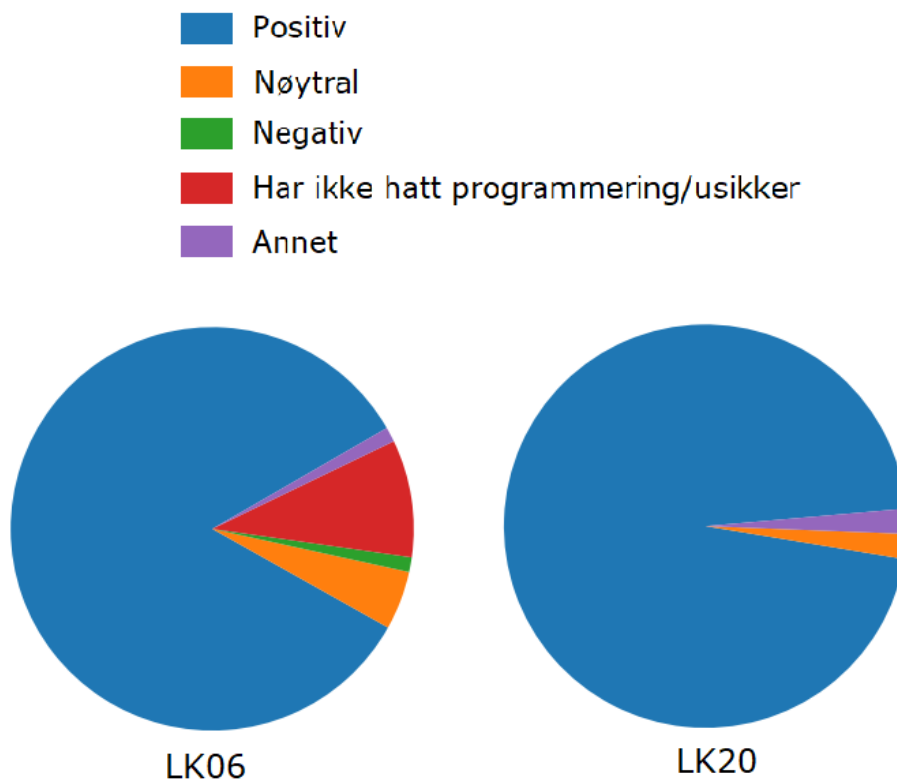
Generelt viser også de åpne spørsmålene at for generelt programmering i fysikk er respondentene overveldende positive. Ved å gå inn i kommentarene kvalitativt kommer det frem mange som priser programmering for nytteverdi i at den hjelper med mye av den mindre engasjerende delen av fysikk, som å regne ut store mengder data og iterativt arbeid. En kvinne fra 2023 kullet som går på BFY, svarer at «Det er et nyttig verktøy som gjør at man kan bla. løse problemer raskere enn ved hoderegning, penn og papir. Det er vanskelig i begynnelsen og må holdes vedlike, men veldig nyttig når man kan det.» En av respondentene nevner at ikke bare kan det brukes til å løse problemer, det påvirker hvordan man tenker om fysikk.

Jeg tenker at programmering er en nyttig problemløsningsmetode og introduserer en annenderledes tenkemåte. Denne tenkemåten er nyttig å lære tidlig og gir en dypere forståelse av faget. Derfor mener jeg det er stort sett en bra ting. Likevel er det litt for ofte bare kopiering av overkompliserte koder for syns skyld fremfor ekte problemløsning. (Mann, 2023, BFY)

Det er få andre responser som går inn på samme tema, men dette tyder på at computational thinking kan komme inn via programmeringen i fysikk, men det krever at det får fokus. Samtidig er det og mange som har kritikk med i ellers positive kommentarer, spesielt rettet mot gjennomføringen i videregående. En annen kvinne, også 2023 på BFY, svarer at «Jeg tenker at det er nyttig og relevant, men at måten det blir gjennomgått på av læreren er alt for dårlig og at det er for lite fokus på å forstå programmene.» Som nevnt

under påstanden 'Jeg er motivert for å lære programmering fremover.' så virker respondentene til å være positive til programmering i fysikk, men flere er ikke positive til måten programmering ble undervist om på videregående skole. I forhold til LK20 er hovedforskjellen at LK06 har en andel som svarer at de ikke har hatt programmering, men også her blant de som er positive har de ofte kritikk til gjennomføringen. «Vanskelig, men lurt. Man bør lære det. Men da bør også lærerne kunne det godt nok til å lære det bort på en god måte.» (Kvinne 2022 m/ToF, MTFYMA).

Føler du at programmering er nyttig i fysikkfaget?



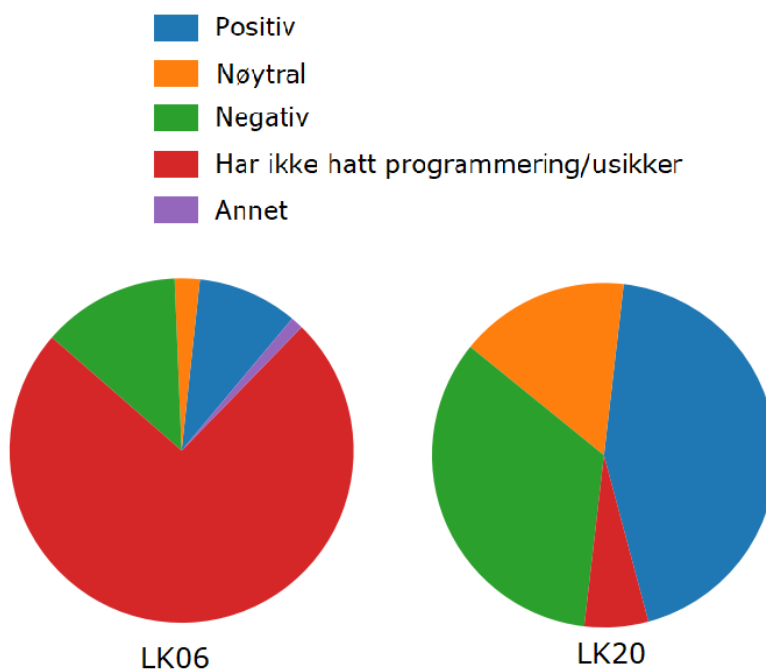
Figur 13: Responser til 2. åpne spørsmål

Som på forrige spørsmål er respondentene i all hovedsak positive. Av de som svarer nøytralt eller negativt er det i hovedsak fordi det tar bort tid som kan brukes på andre ting. «Halvveis, ja, men også tar det mye tid man heller kunne brukt på å forstå ting.», svarer en kvinne, 2022, fra BFY. Ellers, blant de positive svarene, er det også her litt kritikk til gjennomføringen på videregående skole. Mann, 2023 m/PX på BFY svarer «Absolutt! Kanskje ikke så mye med tanke på oppgaver vi får i faget generelt, men når man faktisk skal jobbe med fysikk er der bra å kunne! (Tror jeg)». En kvinne, 2023 fra BFY reflekterer rundt spørsmålet med svaret

Ja, til ein viss grad. Sånn eg opplevde det var programmeringsnivået i fysikk ganske lavt (det var ikkje forventa masse) og då er det berre nokon få oppgåve som ein får nytte av programmering i. Men til bedre i programmering man er til meir nyttig er det.

Dette beskriver godt en trend som flere respondenter skriver om. Nivået av programmering elever ligger på i videregående er ofte en god del lavere enn nivået de har for andre problemløsningsstrategier. Dermed opplever de at å måtte bruke programmering for å løse oppgaver de lett kunne ha løst på annet vis gjør at programmeringen blir et hinder fordi det begrenser hvilke måter de kan jobbe på, heller enn et verktøy som gir flere innfallsvinkler.

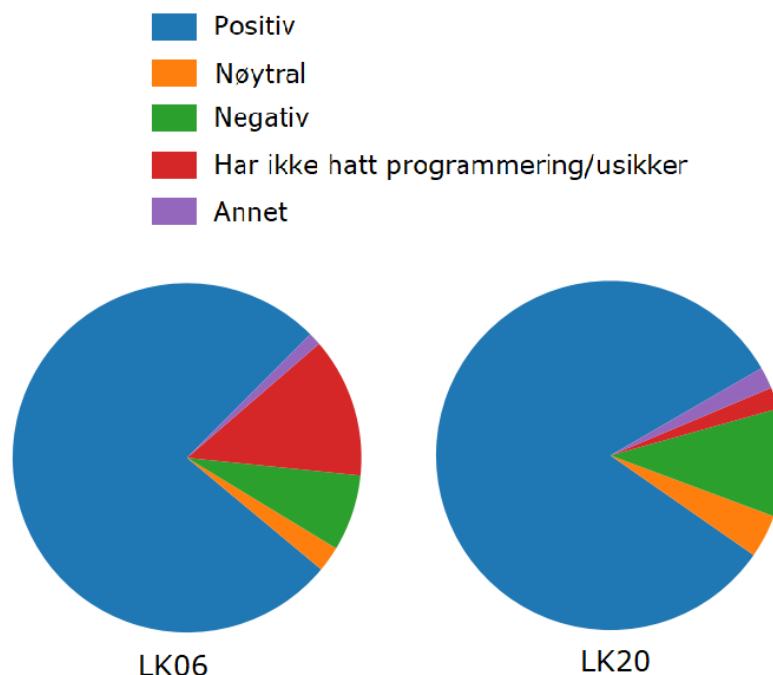
Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk?



Figur 14: Responser til 3. åpne spørsmål

På dette spørsmålet har responsene en klar forskjell mellom LK06 og LK20. Relativt sett er det en betydelig andel fra LK06 som svarer de ikke har hatt programmering/er usikre, men det er også for LK06 flere som svarer negativt enn positivt, altså flere som har erfart at det har vært vanskelig å lære programmering i fysikk. For LK20 er andelen større for positive og nøytrale i forhold til negative. Likevel er andelen negative svar også her betydelig, da 34% av LK20 svarer negativt. Et trekk som går igjen i svarene er at de som svarer positivt har også lært programmering fra et annet sted, som for eksempel et IT-fag. For eksempel svarer en kvinne, 2023, m/TOF på MTFYMA at «Jeg synes ikke det har vært spesielt utfordrende, i og med at vi også har hatt programmering i matte og andre fag. Du lærer mye av det samme, og kunnskapen kan brukes i alle de ulike fagene.» Blant de som svarer det har vært vanskelig er begrunnelsen ofte hvordan det er strukturert, for eksempel at oppgavene ikke samsvarer med hva de opplever å ha lært, eller at de føler de ikke får brukt det over tid slik at kunnskapen setter seg. En kvinne, 2023, på BFY gir svaret «Det har vært vanskelig, fordi programmene ikke har blitt gjennomgått av lærer og fordi de inneholder element som vi ikke har lært om» mens en mann, 2022 m/ToF, på BFY svarer «Litt vanskelig, lærer programmering raskt, men glemmer mye av det jeg lærer like raskt».

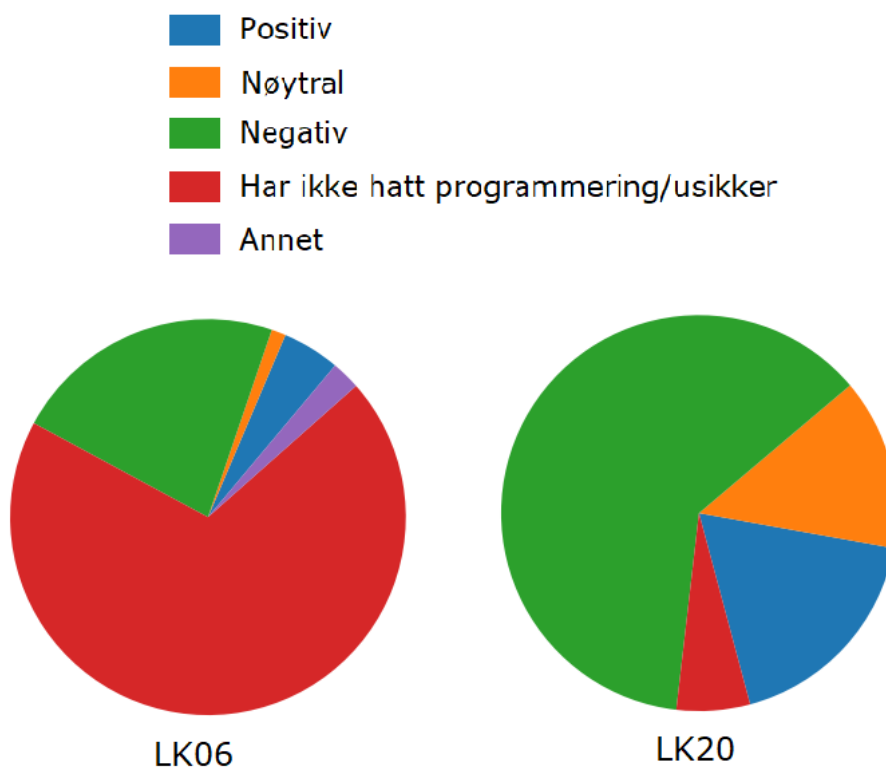
Når du arbeider med programmering, syntes du det er viktig å forstå hver kodelinje? Hvorfor eller hvorfor ikke?



Figur 15: Responser til 4. åpne spørsmål

Dette spørsmålet er eneste av de åpne spørsmålene der LK20 har en større andel som svarer negativt i forhold til positivt enn LK06 har, det vil si som mener at det ikke er viktig å forstå hver kodelinje. En majoritet er likevel fremdeles positiv, og de som svarer dette beskriver i hovedsak at å forstå hver linje er viktig for å forstå helheten. En del påpeker at det å forstå hver linje er ekstra viktig når man er en nybegynner. En del av dem svarer og at de synes det er mer motiverende og engasjerende når man ser hvordan hele koden er satt sammen og forstår hele systemet. Mann, 2022 m/PX, på BFY har som svar at «Ja! Når man driver med basics vil jeg gjerne forstå alt sånn at jeg har den grunnleggende forståelsen til senere. Det blir mer motiverende når man forstår». Kvinne, 2023, på MTFYMA gir en liknende respons: «Ja for da kan jeg lettere huske og skjønne hvordan jeg skal skrive en ny kode neste gang, og finne feil neste gang jeg skriver kode.» Blant de som svarer negativt er holdningen i hovedsak at det som er viktig er å forstå sammenhengene, selv om man ikke nødvendigvis forstår hver enkelt linje. Mann, 2021 m/PX, på MTFYMA svarer at «Nei så lenge du forstår hvert trinn i prosessen er ikke hver kodelinje nødvendigvis relevant», og Kvinne, 2023 m/IT1, på MTFYMA svarer «Ikkje nødvendigvis kvar linje men for det meste forstå kvifor programmet fungerer». Enkelte av responsene sikter og til at å forstå hver linje gir bedre eierskap til koden, at de føler det blir mer sitt eget verk, og at dette er mer motiverende enn om de har hentet inn kode de ikke forstår.

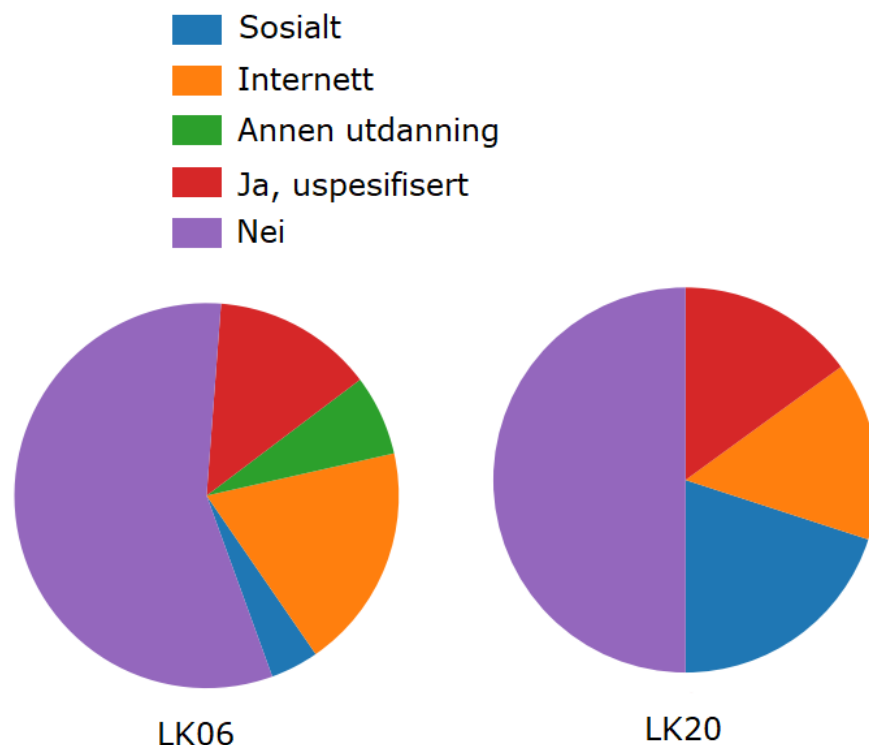
Hvordan har det å lære programmering påvirket din fysikkforståelse?



Figur 16: Responser til 5. åpne spørsmål

På dette spørsmålet svarer majoriteten av de som har erfaring med programmering at det ikke har påvirket deres fysikkforståelse. Som ble nevnt under responser til 'Føler du at programmering er nyttig i fysikkfaget?' opplever mange at nivået på programmeringen ligger såpass under nivået for fysikk at det de kan løse med programmering ofte ikke har kunnet bidra til å gjøre noe særlig nytt og utforskende. Kvinne, 2023, MTFYMA forklarer det som følger: «Foreløpig ingenting siden vi ikke har hatt det her. På videregående var det mer et tillegg enn noe som bidro til videre forståelse». En del av respondentene som svarer slik resonnerer likevel at selv om det ikke har bidratt noe nevneverdig til fysikkforståelsen enda, vil det antakeligvis kunne gjøre det fremover etter hvert som de blir flinkere og kan gjøre mer avanserte oppgaver med det. Kvinne, 2023, på BFY svarer at «Programmeringen jeg selv hadde i fysikk på videregående har ikke hjulpet meg noe spesielt, men jeg tror at den kunne ha gjort det, om den fikk litt mer fokus og tid». Kvinne, 2021, på MTFYMA gir en god oppsummering. «Ikke så mye foreløpig, men forhåpentligvis framover!». Blant de som svarer positivt tar de gjerne frem eksempler på oppgaver der programmering har hjulpet dem. Flere nevner ikke-konstant akselerasjon, et tema som og blir nevnt i kompetansemålet om programmering i fysikk 1. For eksempel svarer Mann, 2023, på BFY «Det har gitt med en mye bedre forståelse av integral og tilfeller med varierende akselerasjon som luftmotstand og verdensrommet». Andre nevner Eulers metode, eller at det har gitt dem trening i å tenke logisk.

Har du lært programmering fra andre steder enn i skolen? Hvilke?



Figur 17: Responser på 6. åpne spørsmål

Dette spørsmålet er ikke del av spørsmålene som er inspirert av Bott, har mer en oppsummerende funksjon, og har ikke vært obligatorisk å svare på. Likevel kan det være nyttig for å få et bedre overblikk over respondentene, og hvordan klimaet er rundt å lære programmering utenfor skoleverket. Halvparten av LK20 svarer at de har lært programmering fra andre steder, litt færre for LK06. Av de som oppgir andre steder nevner mange internett, spesielt Khan Academy eller Youtube. Flere nevner at de har lært seg det fordi det var gøy, mens enkelte svarer de gjorde det fordi de ikke følte de lærte nok fra skolen. Kvinne, 2023, på MTFYMA svarer «Prøvd å lære på nettet på egen hånd fordi opplæringen på skolen ikke var tilstrekkelig til å svare på de vanskelige programmeringsoppgavene i fysikk 2.» De sosiale svarene er i hovedsak venner eller familie, typisk da eldre søsken eller foreldre som driver med programmering. De som nevner annen utdanning nevner forkurs, folkehøyskole, eller annen høyere utdanning enn den de holder på med nå.

Har du andre tanker om ditt forhold til programmering i fysikk

Også dette spørsmålet var ment som oppsummerende, slik at tanker og meninger som ikke fikk kommet frem i de andre spørsmålene skulle kunne bli nevnt. Dette spørsmålet var heller ikke obligatorisk. Responsene for dette spørsmålet bygger ofte videre på tanker alt nevnt, og flere forklarer at de tror programmering kan bli mer spennende når de lærer mer av det. Mann, 2018, på MTFYMA svarer at «Jeg anser det som noe jeg må gjennom, men har troen på at det vil bli kjekt proposjonalt med økende kompetanse.» Kvinne, 2022

m/ToF, MTFYMA, svarer at «Gleder meg til jeg føler at jeg mestrer det og kan bruke det som hjelpemiddel». Enkelte svar påpeker at det er vanskelig å bruke når man ikke har erfaring med programmering, men helhetlig er de fleste av svarene på dette spørsmålet enten om at hvordan de ser for seg at programmering kan hjelpe dem fremover, eller at de ser frem til å kunne det skikkelig.

Oppsummering åpne spørsmål

De åpne spørsmålene viser videre at de fleste respondentene er svært positive til programmering i fysikk og motiverte for å lære det fremover, men er kritiske til hvordan undervisningen har blitt gjort på videregående og flere føler de ikke har fått mye utbytte i fysikk av å ha programmering med i faget. Disse tankene kommer frem fra både LK06 og LK20, men LK20 viser enda større grad av motivasjon og også er enda mer kritisk til videregående sin gjennomføring. Mange mener programmeringen på videregående ikke har vært knyttet godt nok til faget, og at de burde hatt mer undervisning i det på forhånd før det ble tatt opp i fysikken slik at de kunne brukt programmeringen sidestilt med andre typer problemløsning. Likevel ser de frem til å lære mer programmering på universitetet, og er tydelige på at de synes det virker som et veldig nyttig verktøy, om bare de får tid til å lære seg det skikkelig.

4.2 Faglig overblikk

Seksjonen om intervjuene fokuserer på temaene fra den tematiske analysen, som beskrevet i 3.2. Her settes de fem intervjuene, delt inn i informanter A, B, C, D, og E, opp ved siden av hverandre og temaene og kodene forklares nærmere. Temaene, kodene, og hvilke intervjuer de dukker opp i er vist i tabell 8. Eksempler på data som er kodet til de forskjellige kodene finnes i tabell 5 i kapittel 3.2.1.

Tabell 8: Temaer, koder, og informanter som nevner de market med X

| Tema | Kode | A | B | C | D | E |
|---------------------------|-------------------------------|---|---|---|---|---|
| Fysikk i sentrum | Programmere for fysikkens del | | X | X | X | X |
| Manglende grunnlag | Mangler fra ungdomsskole | X | X | | X | X |
| | Mangler fra 1T | X | | X | X | X |
| Manglende tid | Fullt fag | | X | X | X | X |
| | Tid på det grunnleggende | X | X | X | | |
| Elevenes nivå | Stor spredning | X | X | X | X | X |
| | Vanskelig og annerledes | | X | X | | |
| Utvikling av undervisning | Utvikling av undervisning | | X | X | | X |
| | Utvikling fra ungdomsskolen | X | X | | X | X |

Fysikk i sentrum

Dette er et tema med kun en kode, 'programmere for fysikkens del'. Dette omhandler at informanten beskriver at de ønsker å bruke programmeringen ikke for å programmere men for å gjøre fysikk. I tillegg til å nevnes av lærerne beskrives det og slik i læreplanen (se kapittel 2.1). Programmering for fysikkens del blir brakt frem av flere som

en ideell situasjon, mens i praksis må de ofte bruke mye tid på grunnleggende programmering. Dette er ikke bare noe som gjelder for undervisning i skolen, men også for kursene for lærere. Informant B beskriver at kursene ofte alle startet veldig med helt grunnleggende programmering, og at han ikke følte noen progresjon. «Det blir ofte veldig sånn personlig men noen føler at de kanskje har hørt det samme mange ganger, andre ganger så kanskje man ikke får nok konkrete ting man kan holde på med, at det blir litt løst noen ganger.» Informant C forteller og at hun har hatt denne opplevelsen, og gir en historie om at hun meldte seg på et kurs som skulle være handle om programmeringsdidaktikk, men som endte med å bare være grunnleggende koding. «Sånn at når kurset setter seg inn som en sånn, nærmest programmeringsdidaktisk innfallsvinkel så var det jo sykt fristenes, og også syyykt skuffenes å atter en gang skrive 'Hello World'.» Informant E har, I motsetning til de andre, kurs der det blir sett på programmering om fysikk, deriblant oppgaver med luftmotstand slik som beskrevet i kompetansemålene. Informant E, ulikt informanter B og C, virker fornøyd med kursingen som ble gjort i forkant. Informant D nevner kursingen mest i form av at han føler det har vært for lite tilgang til etterutdanning på hans videregående skole, og at han gjerne skulle hatt mer. På hans skole ble enkeltlærere tatt ut for å etterutdannes i programmering, men bare noen av gangen og prosessen gikk i følge D for sakte.

Manglende grunnlag

'Temaet om manglende grunnlag innebærer at elevene i fysikk-klassen ikke har det grunnlaget som skal til for å programmere i fysikkfaget. Dette deles inn i to kodesett; et hvor informantene peker til at elevene ikke har nok fra ungdomsskolen og et for faget 1T. I følge informantene skal elevene skal i utgangspunktet lære programmering i 1T, og så bruke denne programmeringen i fysikk som beskrevet i forrige avsnitt. Men, per informantene, er ikke dette tilstrekkelig. Flere årsaker blir gitt. A forklarer at elevene i 1T mangler motivasjon til å lære programmering, og informant D forklarer videre at programmeringen i 1T er krevende å lære seg om man ikke har et grunnlag fra før. Informant C nevner at programmeringsopplæringen i 1T ikke nødvendigvis svarer til det man trenger for å løse oppgaver i fysikk. Informant E mener de får en god basis fra 1T, men at det de får fra ungdomsskolen er for mangelfullt. Informanter A, B, og D nevner alle og at elevene har lite programmeringskunnskap fra ungdomsskolen, og at det og varierer hvor mye de har så man må alltid forvente at man må starte med det helt grunnleggende.

Mangel på tid

Her er delt inn i kodene 'fullt fag' og 'tid på det grunnleggende'. 'Fullt fag' omhandler at pensumet til fysikk, og fra noen også 1T, allerede krever hele semesteret. Derfor er det ikke mulig å legge til programmering uten å ta ut noe annet, noe informantene ikke ønsker. 'Tid på det grunnleggende' sikter til at informantene forklarer at de må bruke den tiden de har til programmering til å lære elevene grunnleggende programmering i stedet for å lære dem å løse fysikk med programmering. Informant E snakker om tid ikke bare i form av at det ikke er tid i undervisningen, men også at det tar mye tid for lærerne å forberede. Informant D, som da intervjuet tok sted hadde jobbet som lærer i 34 år, snakker om hvordan fysikkfaget over tid har fått mer og mer lagt til som elevene skal gjøre, så fokuset blir på å kunne for mange forskjellige ting, heller enn å forstå faget. «Ja, hvis jeg skal være litt kritisk så får du ikke, altså, elevene blir putta inn i så mye. [...] . Det blir bare en sånn

evig jag etter å prøve å beherske ting.» D beskriver og hvordan det er nødvendig å ta seg god tid om man skal lære noe skikkelig, og også å gjøre det over tid. Rundt en nylig prøve forteller han at de har lært programmering til den, men at de ikke skal ha det nå på lenge. «Så dem får med seg litt, men så går det i glemmeboka sånn at nå blir det en periode uten noe særlig programmering, uten sånn på våren igjen. Hvis du skal på en måte klare at dem skal få noe sånn reelt utbytte av det, i forhold til investeringa, så må man på en måte ha jevnt og trutt.»

Elevenes nivå

Med 'elevenes nivå' siktes det til at informantene snakker om at nivået i klassen påvirker hvordan de gjør undervisningen. Dette er delt inn i to koder, 'stor spredning' og 'vanskelig og annerledes'. Sistnevnte nevnes bare av to informanter, men spesielt informant Informant C bruker mye tid på å beskrive at oppgaveløsning med programmering er veldig forskjellig fra hvordan elevene er vant til å jobbe, og at de som viser høy måloppnåelse i fysikk og andre fag ikke nødvendigvis er de samme som viser høy måloppnåelse i programmering. Den andre koden, 'stor spredning', er den eneste koden som blir nevnt av samtlige informanter. Dette omhandler at informantene ser en tendens til at elevene enten mestrer programmering eller sliter med å få det til, med få som ligger midt i mellom. Igjen legger C frem om dette i detalj, og beskriver interesse som en viktig faktor som skiller elevene:

Så det er liksom en gruppe som bare avskyr det litt og føler at det er et sånt uromoment som ikke passer inn i deres «Jeg har regna alle oppgavene og jeg har lest teksten»-instilling. Og så er det jo en gruppe som synes programmering er helt koko-kult. Det er de som programmerer alle planetene i solsystemet med på en måte komplett bane og påvirkning fra gravitasjon, ikke bare fra sola men også fra de andre planetene og, eh, de synes jo det er superstas og null stress.

Andre informanter beskriver andre årsaker. Informant A snakker om hvordan elevene skilles basert på om de har kodet før videregående eller ikke, og at enkelte har hatt for eksempel Scratch på ungdomsskolen, men også at det har mye å gjøre med hvor modne elevene er og hvor villig de er til å prøve noe nytt. Informant E snakker om forskjeller basert på hva de har gjort på ungdomsskolen, og påpeker også at enkelte som driver med koding på fritiden har en stor fordel. «Ja, de som er veldig rutta de har gjerne holdt på med det ved siden av skolen.» Disse forskjellene i nivå speiler resultatene beskrevet i 4.1.2, hvor det fra de åpne spørsmålene også kommer frem at programmering i fysikk ofte oppleves som enten enkelt eller vanskelig, sjeldent «sånn passe».

Utvikling av undervisning

Dette temaet handler om at informantene snakker om endringer fra årskull til årskull. Koden 'utvikling av undervisning' snakker de om sin egen undervisning. De tre informantene som snakker om dette, informanter B, C, og E, snakker alle om at de har bygget opp erfaringer og gjort endringer med opplegget sitt underveis. B beskriver å gjøre programmeringen mer jevnlig, heller enn å ha en stor programmeringsøkt og så være ferdig. Informant C snakker om nivået til elevene, og når hun tar opp forskjellige typer oppgaver basert på hvor mye elevene kan på den tiden av året. Informant E snakker om å bli mer sikker på temaet, og å bygge opp basis for undervisningen over flere

gjennomganger. Den andre koden, 'utvikling fra ungdomsskolen', handler om hvordan endringen har vært i kullene fra ungdomsskolen. De fleste informantene rapporterer at det er litt forskjell mellom årskullene, men mindre enn de hadde forventet. På spørsmålet om det har vært noen forskjell i nivå på de nyeste kullene i forhold til de første under LK20 svarer han «Uhm... om det er noen forskjell? Ikke så veldig stor forskjell, faktisk. Eh, den burde ha vært mye større.» Informant A beskriver samme opplevelse: «vi ser jo at nivået går opp, men det går ikke veldig fort opp. Det går egentlig ganske sakte opp. Enda så tror jeg, altså, det er veldig varierende, hva dem kommer inn på videregående med av forkunnskunnskaper.»

Oppsummering intervju

Fra informantene kommer det frem at det har vært flere hindringer i å få gjort fysikkundervisningen på den måten som fagplanen sier det skal gjøres. Fra tidsbegrensninger til å måtte tilpasse undervisningen til elever som ikke har det nødvendige kunnskapsgrunnlaget for fysikkprogrammering. Det er mye snakk om nivåforskjeller, som kan gå på tross av forskjellene mellom de som ellers har høy måloppnåelse i fysikk og de som har lav. Forskjellene går heller på de som synes programmering er interessant og de som ikke er motiverte, og de som lærer programmering fra andre steder som fra ungdomsskolen eller valgfag, og de som ikke. Likevel snakker alle informantene varmt om selve programmeringen, og virker til å være engasjerte for å få det til i undervisningen og for å lære bort om programmering til elevene sine.

5. Diskusjon

Disusjonskapitlet vil ta for seg presentert teorien og funnene fra mine undersøkelser, og undersøke hvilke funn som kommer frem. EV-modellen (Eccles & Wigfield, 2002) vil være sentral i vurderingen av elevenes holdninger.

5.1 Refleksjoner om teori og funn

Gjennom de lukkede spørsmålene kommer det frem at mestringsforventning og forkunnskaper utgjør den største forskjellen mellom LK06 og LK20. Det mest ekstreme eksemplet kommer fra påstanden «Fysikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper», der 58% av LK20 er enig eller svært enig, mens for LK06 er tilsvarende verdi 1%. Dette tyder på at kullet fra LK20 føler seg mer komfortable med å bruke programmering. For de andre komponentene svarer også LK20 noe mer enig, men forskjellene er langt mindre enn de er på mestringsforventning og på forkunnskaper. På ett av spørsmålene om nytteverdi er det en stor forskjell mellom kullene fra hver av læreplanene; «Programmeringen jeg har lært i matematikk kan være nyttig for å arbeide med fysikk». Her og svarer LK20 langt mer mot enig enn LK06, men denne påstanden handler også om hva de har hatt på videregående skole. Det er verdt å nevne at dette også gjelder for spørsmålet om kostnad: LK20 svarer at programmering er mer nyttig og interessant enn LK06 gjør, men svarer og at det er mer krevende.

Nytteverdi og kostnad er også med i de åpne spørsmålene. Nyttiverdi får liknende responser; Dersom vi ser bort i fra de som svarer har ikke hatt/usikker finner vi at LK20 svarer 98% positivt, og også 98% som er enige eller svært enige på det lukkede spørsmålet. For LK06 svarer tilsvarende 93% positivt på det åpne spørsmålet og 83% enig eller svært enige på det lukkede. På spørsmålet kostnad er det derimot en forskjell. På spørsmålet «Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk?» svarer 46% av LK20 positivt, altså at de synes det har vært enkelt, mens 30% av LK06 svarer seg enig eller svært enig i det lukkede spørsmålet om kostnad, altså at programmeringsoppgaver har vært mer krevende. For LK06 svarer 38% positivt på spørsmålet om kostnad, mens 19% svarer seg enig eller svært enig i det lukkede spørsmålet. Altså er det flere i LK20 som svarer at programmeringsoppgaver er mer krevende, men også flere som svarer at de synes programmering har vært lett å lære. Dette tilsynelatende avviket kan undersøkes nærmere ved å bruke den kvalitative delen av datasettet. Blant svarene til spørsmålet om hvor lett/vanskelig det har vært å lære programmering i fysikk finnes begrunnelsene som respondentene har gitt. Kvinne 2023, BFY, svarer at «Jeg begynte med programmering i vg1 og hadde det i alle matte- og fysikkfagene jeg tok. Jeg tror programmering krever mengdetrening og vedlikehold, noe jeg fikk mulighet til gjennom hele vgs så jeg syntes at det var helt greit å lære.» Mann, 2022 m/ToF, BFY, svarer at «Litt vanskelig, lærer programmering raskt, men glemmer mye av det jeg lærer like raskt». Mange svarer at selv om det gikk greit å lære programmering har det vært vanskelig å bruke det i praksis, for eksempel fordi at det var vanskelig å vedlikeholde kunnskapen når den sjeldent ble brukt, eller fordi å løse problemer krevde mer avansert bruk av programmering enn den man ble lært.

Jevnlig bruk av programmering, slik flere av respondentene etterspør i de åpne spørsmålene, blir også nevnt av to av faglærerne. Informant D mener, i likhet med flere respondenter, at jevn bruk av programmering er nødvendig for å lære det skikkelig. B snakker om at han prøver å bruke det mer jevnlig og gjennomgående. Etter Udirs kompetansemål er det bare fysikk 2 hvor faget er satt opp på denne måten, mens fysikk 1 har kun programmering i mekanisk fysikk (Udir, 2019d, 2019e). Både spørreundersøkelsen og intervjuene kommer og inn på temaet om fysikk i sentrum. Informantene forklarer at som lærere er de tvunget til å bruke tid på å lære bort programmering, fordi elevene de har ikke behersker det nok til å bruke det til problemløsning i fysikk, og at de da ikke har tid til å bruke programmering på fysikk. Respondentene forklarer tilsvarende at de som elever har følt at programmering i fysikk ikke har hjulpet dem med fysikkfaget, fordi oppgavene har vært for grunnleggende. Informanter og respondenter er begge tydelige på at de er motiverte for programmering i fysikk, og forklarer dette med at de ser det som et potensielt veldig nyttig verktøy, men begge opplever og at de ikke får realisert dette potensialet fordi nivået av programmering ikke er høyt nok i videregående skole. Dette er ikke noe som gjelder for alle, og mange av respondentene gir positive tilbakemeldinger, men det er likevel et gjennomgående tema både i spørreundersøkelse og intervjuer.

En av årsakene som kommer frem til at undervisning i programmering er krevende er at det er stor spredning i nivået til elever. Respondentene gir varierende responser, spesielt på spørsmål som omhandler deres erfaringer fra undervisning. Enkelte beskriver programmering som veldig lett og spennende, andre som noe som var vanskelig, eller som ikke ble gitt nok fokus til å lære skikkelig. Samtlige intervjuer har koden stor spredning, og det gis flere mulige årsaker til denne spredningen. En av disse årsakene som blir nevnt av både informanter og respondenter er om opplæringen som er gjort i forhånd av fysikkfaget. På spørsmålet 'Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk?' svarer flere at det har vært vanskelig fordi det de har lært på forhånd, i 1T, ikke nødvendigvis har svart til det de trenger i fysikk. Dette er også en opplevelse informantene sier, at forkursene de fikk i forberedelse til LK20 har vært for generelle og ikke gitt dem det de trenger spesifikt for fysikk. Begge grupper sier at det da krever tid av fritiden, noe ikke alle har kapasitet til, for å kunne få det som kreves til å programmere skikkelig i fysikk. Dette speiler funnene til Storesund, som i sin konklusjon skriver at «opplæringen studentene får i ITGK og i fysikkemner ikke forbereder dem godt nok til å løse programmeringsoppgavene i fysikk» (2022, s 75). Altså er dette et gjentakende problem i fysikkundervisning, som finnes på elev-, student-, og lærernivå.

På den ene siden vil problemet med manglende grunnopplæring i programmering trolig bli mindre med tid. Overgangen til LK20 har for programmering vært stor, og mange skoler og lærere er fremdeles i prosessen med å tilpasse seg. Informant D nevner at fysikklærerne på skolen han jobber på fremdeles blir kurset og så hjelper hverandre. Elever vil bli undervist i programmering på barne- og ungdomsskolen og vil med det trolig ha et bedre grunnlag for å lære programmeringen i 1T, og tilsvarende vil lærerne ha holdt kurset flere ganger og bli bedre kjent med innholdet. LK20-respondentene begynte videregående i 2020, og var derfor første kullet i hvert ledd av den nye læreplanen. Samtidig rapporterer informantene at utviklingen fra ett kull til neste har gått sakte, og i mellomtiden er det en stor elevmasse som ikke får tilgang på de ressursene de trenger for å komme seg gjennom fysikkfaget. Mange av respondentene fra spørreundersøkelsen, fra både negative og positive

svar, sier at de synes programmering virker mer spennende og mer nyttig når man kan det. Altså at det ikke fungerer bra som et mindre tema som bare kommer frem noen få ganger i året. Enten så må programmering brukes aktivt nok til at det sitter og elevene kan bruke det som en problemløsningsstrategi, eller så er det lite poeng og programmering blir bare enda et hinder elevene må over for å komme seg gjennom fysikkfaget.

Mange respondenter forteller om oppgaver som bare er å kopiere eller plugge inn i kode uten noe fokus på å forstå fysikken eller koden, eller at programmeringen har vært såpass grunnleggende at det ikke har hatt noen nytte for dem. Derfor er det bare noen få respondenter som kommer inn på temaer som kan knyttes til computational thinking på en slik måte som blir beskrevet av diSessa (2000, 2018) og Odden et al. (2019) som så viktig for utviklingen av fysikk fremover. Samtidig nevner flere at selv om ikke programmeringen har påvirket deres fysikkforståelse enda, vil den antakeligvis gjøre det når de får mer erfaring og forståelse. Enkelte nevner og at programmering påvirker deres tenkemåte, eller at de har fått en ny måte å se på spesifikke problemstillinger.

Til tross for dette svarer respondentene at de er motiverte for å lære programmering fremover. Både LK06 og LK20 svarer i all hovedsak positivt på spørsmål om interesseverdi og om nytteverdi. LK20, som har fått oppleve å bruke programmering i fysikk, svarer enda mer positivt enn LK06 på begge disse konstruktene. Altså tross kritikk og tross problemene som har blitt beskrevet med innføringen av programmering i fysikkfaget ser vi at motivasjonen har gått opp for de som har vært gjennom det.

5.2 Forslag for videre forskning

I skrivende stund er det kun ett kull som har fullført videregående under ny læreplan. Dette er et kull som hadde LK20 først på videregående, og ble derfor kastet inn i programmering uten å ha fått den systematiske innføringen i det i barne- og ungdomsskole som læreplanen er lagt opp etter. Fremtidlige kull vil møte Python og tekstprogrammering i 1T og fysikk først etter å ha hatt blokkprogrammering på barne- og ungdomsskolen. Selv lærte jeg blokkprogrammeringsspråket Scratch etter jeg hadde lært tekstprogrammeringsspråket Python, men jeg opplevde det som betydelig lettere å lære et nytt språk når jeg kunne et fra før. Mange av problemene som blir belyst i denne oppgaven stammer fra at elever ikke har nødvendig programmeringskompetanse innen de begynner med det i fysikk. Derfor tror jeg det vil kunne ha en betydelig påvirkning om de starter 1T med programmeringserfaring fra å også ha hatt blokkprogrammering tidligere i utdanningsløpet. Dette kan igjen få ringvirkninger i andre fag som bruker programmering, siden elevens nivå er høyt nok til å bruke programmeringen som et verktøy og som en tenkemåte. Derfor anbefales det å gjøre videre studier som ser på sammenhengen mellom programmeringserfaring fra ungdomsskolen og faglig utbytte fra programmeringsundervisning på videregående. Gjør de som har lært Scratch det bedre i Python? Går dette videre til høyere mestringsgrad i programmering i fysikk? Vil en mer problemløsningsorientert programmering i fysikken bidra til bedre utvikling av computational literacy hos elevene?

Noe annet som dukket opp underveis i studien min var flere som snakket om å ha andre programmeringsfag hjalp dem med å få den jevnlige programmeringen som andre snakket om at manglet fra fysikkfaget. Flere respondenter rapporterte at siden de hadde jevnlig programmering i andre fag, for eksempel programmering og modellering X eller teknologi og forskningslære, så gikk det greit å forstå programmeringen i fysikkfaget. Andre

rapporterte at de ikke hadde et problem med å lære programmering, men heller med å vedlikeholde det de hadde lært siden det ikke ble brukt jevnlig. En studie som ser nøyere på hvordan å ha forskjellige fag som har programmering påvirker faglig utbytte i hver av disse fagene kan gi verdifull kunnskap som kan brukes til å videre utvikle programmeringsundervisning i skolen.

6. Konklusjon

I denne oppgaven har jeg sett på elevers opplevelse av innføringen av algoritmisk tenkning og programmering i fysikkfaget, som kom med den nye læreplanen LK20. Dette har jeg forsøkt å svare på gjennom de to forskningsspørsmålene «Hvordan har innføringen av LK20 påvirket elevers holdninger til programmering innen fysikk?» og «I hvilken grad har undervisning i programmering i videregående skole blitt opplevd som tilstrekkelig for at elevene skal kunne bruke programmering som et verktøy i fysikk?». Studien jeg har gjennomført har vist at elevene som har vært gjennom innføringen av LK20 på videregående har fått økt sin overordnede motivasjon for programmering i fysikk. Til tross for at de, sammenlignet med kullene som har hatt LK06 på videregående, opplever programmering som noe mer krevende rapporterer de også en liten økning i motivasjon og en moderat økning i opplevd nytteverdi. Det er likevel høy varians i hvordan opplevelsen på videregående har vært, der noen respondenter rapporterer at programmering har blitt bortprioritert eller hatt mangelfull undervisning, mens andre rapporterer at det har vært et engasjerende tema og at det har hjulpet dem med å forstå faget. Tilsvarende rapporterer lærere at de og opplever stor forskjell på hvordan elevene opplever møtet med programmering i fysikk. Svarene på spørreundersøkelsen tyder altså på at programmeringsundervisningen på videregående for mange ikke har vært tilstrekkelig for at elevene skal kunne bruke programmering som et verktøy i fysikk, blant annet fordi faget ikke gir lærerne tid nok til å bruke programmering jevnlig, og fordi både lærere og elever ofte ikke blir gitt forkunnskapene og mengden tid som skal til for å gjøre undervisningen god.

Oppsummert kommer det frem at programmering i fysikk blir sett på som noe med høyt potensiale, både fra elever og lærere, men dette potensialet har ikke nådd frem slik gjennomføringen har vært så langt. Algoritmisk tenkning har gjerne blitt satt til side fordi programmeringen ikke har nådd et nivå der elevene har fått mulighet til å bruke slike strategier. Likevel virker det til å være vilje for å fortsette med det, da både elever og lærere ser et stort potensiale i programmeringen. Med det ønsker jeg å avslutte denne oppgaven med et sitat fra spørreundersøkelsen, som kort og enkelt oppsummerer det jeg ser som den samlede opplevelsen elever har til programmering i fysikk: «Jeg anser det som noe jeg må gjennom, men har troen på at det vil bli kjekt proposjonalt med økende kompetanse.» (Mann, 2018, MTFYMA).

Referanseliste:

- Adams, W. K., Perkins, K. K., Podolefsky, N. S., Dubson, M., Finklestein, N. D. & Wieman, C. E. (2018) *Colorado Learning Attitudes about Science Survey (CLASS)*. PhysPort. Hentet 10.04.2024 fra <https://www.physport.org/assessments/assessment.cfm?A=CLASS>
- Angell, C., Bungum, B., Henriksen, E. K., Kolstø, S. D., Persson, J. & Renstrøm, R. (2019). *Fysikkdidaktikk (2. utgave)*. Cappelen Damm Akademisk.
- Bott, T. E., Weller, D. P., Caballero, M. D. & Irving, P. W. (2019). *Student-identified themes around computation in high school physics*. 2019 Physics Education Research Conference, s. 57-62. <https://doi.org/10.1119/perc.2019.pr.Bott>
- Braun, V. & Clarke, V. (2006). *Using thematic analysis in psychology*. *Qualitative Research in Psychology*, s. 77-101. <https://doi.org/10.1191/1478088706qp063oa>
- diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Mit Press. <https://doi.org/10.7551/mitpress/1786.001.0001>
- diSessa, A. A. (2018). *Computational Literacy and «The Big Picture» Concerning Computers in Mathematics Education*. *Mathematical Thinking and Learning*. UC Berkely. <https://doi.org/10.1080/10986065.2018.1403544>
- Eccles, J. & Wigfield, A. (2002). *Motivational Beliefs, Values and Goals*. *Annual Review of Psychology*, s. 109-132. <https://doi.org/10.1146/annurev.psych.53.100901.135153>
- Gjengset, F. (2022). *Shaping programming in physics education: A study on how teachers' conceptualization of computation in high school physics influence what is taught*. [Masteroppgave, Universitetet i Oslo]
- Haraldsrud, A. D., Sveinsson, H. A. & Løvold, H. H. (2021). *Programmering i skolen (1. utgave)*. Universitetsforlaget.
- Imsen, G. (2020). *Lærerens verden (6. utgave)*. Universitetsforlaget.
- Nordby, S. T. (2019). *Programmering og algoritmisk tenkning i fysikkundervisning*. [Masteroppgave, Norges Teknisk-Naturvitenskapelige Universitet]
- NTNU. *Lage spørreskjema – kvalitetssystemet*. Hentet 10.04.2024 fra <https://i.ntnu.no/wiki/-/wiki/Norsk/Lage+sp%C3%B8rreskjema+-+kvalitetssystemet>
- NTNU. *Lagringsguide*. Hentet 10.04.2024 fra <https://i.ntnu.no/wiki/-/wiki/Norsk/Lagringsguide>
- Odden, T. O. B., Lockwood, E. & Caballero, M. D. (2019). *Physics computational literacy: An exploratory case study using computational essays*. *Physical Review Physics Education Research*. <https://doi.org/10.1103/PhysRevPhysEducRes.15.020152>
- Robson, C. & McCartan, K. (2016). *Real World Research*, 4th Edition. Wiley
- Storesund, H. (2022). *Beregningsorientering i fysikkutdanning*. [Masteroppgave, Norges Teknisk-Naturvitenskapelige Universitet]
- Udir (2005). *Læreplan i fysikk - programfag i utdanningsprogram for studiespesialisering (FYS1-01)*. Hentet 23.04.2024 fra https://www.udir.no/kl06/FYS1-01/Hele/Grunnleggende_ferdigheter
- Udir (2019). *Grunnleggende ferdigheter*. Hentet 23.04.2024 fra <https://www.udir.no/lk20/fys01-02/om-faget/grunnleggende-ferdigheter>
- Udir (2019b). *Algoritmisk tenkning*. Hentet 23.04.2024 fra <https://www.udir.no/kvalitet-og-kompetanse/digitalisering/algoritmisk-tenkning/>

- Udir (2019c). Hentet med åpen lisens (OGL) fra <https://www.udir.no/kvalitet-og-kompetanse/digitalisering/algoritmisk-tenkning>
- Udir (2019d). *Kompetansemål og vurdering*. Hentet 23.04.2024 fra <https://www.udir.no/lk20/fys01-02/kompetansemaal-og-vurdering/kv466>
- Udir (2019e). *Kompetansemål og vurdering*. Hentet 23.04.2024 fra <https://www.udir.no/lk20/fys01-02/kompetansemaal-og-vurdering/kv467>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). *Defining computational thinking for mathematics and science classrooms*. *Journal of science education and technology*, s. 127-147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. M. (2006). *Computational thinking*. *Communications of the ACM*, s. 33-35. <https://doi.org/10.1145/1118178.1118215>

Vedlegg

Vedlegg A: Spørreskjema



Forkunnskaper i fysikkprogrammering

Hensikten med den følgende spørreundersøkelsen er å undersøke din erfaring med programmering. Spørreundersøkelsen er anonym og kan ikke spores. Jeg er meget takknemlige for at du tar deg tid til å svare på denne undersøkelsen som inngår i min masteroppgave.

Årstall fullført videregående skole:

- 2023
- 2022
- 2021
- 2020
- 2019
- 2018
- 2017
- 2016
- 2015
- 2014
- 2013
- 2012
- 2011
- 2010
- Før 2010

Kjønn:

- Kvinne
- Mann
- Annet
- Ønsker ikke oppgi

Kryss av for fagene du har tatt på videregående:

- Fysikk 1
- Fysikk 2
- IT 1
- IT 2
- Programmering og modellering X
- Matematikk 1T
- Matematikk S1
- Matematikk S2
- Matematikk R1
- Matematikk R2
- Andre fag med programmering

Hvilke andre fag har du hatt på videregående med programmering?

Dette elementet vises kun dersom alternativet «Andre fag med programmering» er valgt i spørsmålet «Kryss av for fagene du har tatt på

videregående:»

Studieprogram:

- Bachelor Fysikk (BFY)
- Lektor Real FAG (MLREAL)
- Master i fysikk og matematikk (MTFYMA)
- Annet

Motivasjon i programmering

Følgende spørsmål vil handle om programmering i videregående. Med programmering menes å bruke et kodespråk som Scratch eller Python gjennom en editor som for eksempel Anaconda Spyder, Thonny, eller Mu.

I hvilken grad er du enig med følgende påstander?

Programmeringsoppgaver har vært mer krevende enn andre typer oppgaver.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Jeg har foretrukket å jobbe med programmeringsoppgaver i forhold til andre typer oppgaver.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Programmeringen jeg har lært i fysikk har vært tilstrekkelig for å løse oppgaver i faget.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Programmering kan være nyttig for å løse problemer i fysikk.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Jeg er motivert for å lære programmering fremover.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Fysikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Matematikkfaget på videregående har bidratt til mine samlede programmeringskunnskaper

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Programmeringen jeg har lært i matematikk kan være nyttig for å arbeide med fysikk.

- Ikke relevant
- Svært uenig
- Uenig
- Nøytral
- Enig
- Svært enig

Videre kommer noen spørsmål om programmering og fysikkfaget i videregående skole. Vennligst utdyp.

Hva tenker du om at programmering er del av fysikkfaget?

Føler du at programmering er nyttig i fysikkfaget?

Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk?

Når du arbeider med programmering, syntes du det er viktig å forstå hver kodelinje? Hvorfor eller hvorfor ikke?

Hvordan har det å lære programmering påvirket din fysikkforståelse?

Avsluttende er jeg interessert i å høre mer om ditt generelle forhold til programmering, og om du har

noen videre tanker og meninger du ikke har fått mulighet til å få frem gjennom spørsmålene så langt.

Har du lært programmering fra andre steder enn i skolen? Hvilke?

Har du andre tanker om ditt forhold til programmering i fysikk?

Vedlegg B: Utvalgte reponser på åpne spørsmål

Hva tenker du om at programmering er en del av fysikkfaget?

Kvinne 2023, BFY - Det er et nyttig verktøy som gjør at man kan bla. løse problemer raskere enn ved hoderegning, penn og papir. Det er vanskelig i begynnelsen og må holdes vedlike, men veldig nyttig når man kan det.

Mann 2023, MLREAL - Jeg tenker at det er veldig viktig. Programmering gir oss muligheter til å forstå fysikken bedre enn det vi kan til vanlig. For eksempel, så kan Eulers metode brukes for å lage differensialligninger. Det er ikke noe man gjør for hånd på VGS.

Mann 2023, BFY - Jeg tenker at programmering er en nyttig problemløsningsmetode og introduserer en annenderledes tenkemåte. Denne tenkemåten er nyttig å lære tidlig og gir en dypere forståelse av faget. Derfor mener jeg det er stort sett en bra ting. Likevel er det litt for ofte bare kopiering av overkompliserte koder for syns skyld fremfor ekte problemløsning.

Kvinne 2023, BFY - Jeg tenker at det er nyttig og relevant, men at måten det blir gjennomgått på av læreren er alt for dårlig og at det er for lite fokus på å forstå programmene.

Mann 2022, MTFYMA - Er sikkert greit, men trur det hadde vore best som eige fag på vgs

Kvinne 2023, BFY – sikkert lurt men samtidig kjedelig

Kvinne 2022 m/ToF, MTFYMA - Vanskelig, men lurt. Man bør lære det. Men da bør også lærerne kunne det godt nok til å lære det bort på en god måte.

Mann 2023, MTFYMA - Det vil hjelpe med å kunne modellere, beregne og håndtere store mengder data og utregninger

Mann 2021, MTFYMA - Det er smart å lære det tidligere enn universitetet

Kvinne 2023, MLREAL – Gjør vondt verre

Føler du at programmering er nyttig i fysikkfaget?

Kvinne 2023, BFY - Ja, til ein viss grad. Sånn eg opplevde det var programmeringsnivået i fysikk ganske lavt (det var ikkje forventa masse) og då er det berre nokon få oppgåve som ein får nytte av programmering i. Men til bedre i programmering man er til meir nyttig er det.

Mann 2023, BFY - Programmering er nødvendig for mange problemer og løsningsmetoder, det er også bra for forståelse, så ja det er nyttig.

Kvinne 2023 m/IT1, IT2, MTFYMA - Ja absolutt. Jeg tror man kan lære mye fysikk uten programmering, men tror på forskningssiden at det er veldig viktig?

Mann 2023, MTFYMA - Ja, men grunnkunnskapene burde læres bort i et annet fag

Mann 2022 m/PX, BFY - Absolutt! Kanskje ikke så mye med tanke på oppgaver vi får i faget generelt, men når man faktisk skal jobbe med fysikk er der bra å kunne! (Tror jeg)

Kvinne 2023, MTFYMA - ja til tider, men jeg foretrekker å løse oppgaver for hånd

Basert på din erfaring, hvor lett/vanskelig har det vært å lære programmering i fysikk?

Kvinne 2023, BFY - Jeg begynte med programmering i vg1 og hadde det i alle matte- og fysikkfagene jeg tok. Jeg tror programmering krever mengdetrening og vedlikehold, noe jeg fikk mulighet til gjennom hele vgs så jeg syntes at det var helt greit å lære.

Mann 2023, MTFYMA - Det er har vært krevende å lære programmering i fysikk, fordi vi lært egentlig ikke noe. Vi fikk bare oppgaver å løse, og eksempler å følge, men fikk aldri opplæring i programmering i faget.

Kvinne 2023 m/ToF, MTFYMA - Jeg synes ikke det har vært spesielt utfordrende, i og med at vi også har hatt programmering i matte og andre fag. Du lærer mye av det samme, og kunnskapen kan brukes i alle de ulike fagene.

Kvinne 2023, BFY - Det har vore lett å lære den grunnleggende programmeringa, men vanskeligere å faktisk bruke programmeringa sjølv til å løyse større problem.

Mann 2023, MTFYMA - Jeg kunne litt programmering fra før så det var "lett" i fysikk. Allikevel har jeg opplevd at andre jeg hadde fysikk med slet fordi programmeringen gikk fort opp i nivå fra 0 forkunnskaper.

Mann 2022 m/ToF, BFY- Litt vanskelig, lærer programmering raskt, men glemmer mye av det jeg lærer like raskt

Mann 2021 m/PX, MTFYMA - Ikke vanskelig. Jeg har hadde programmering og modellering X som ga gode grunnkunnskaper.

Kvinne 2023, BFY - Det har vært vanskelig, fordi programmene ikke har blitt gjennomgått av lærer og fordi de inneholder element som vi ikke har lært om

Kvinne 2023, MTFYMA - Det har vært vanskeligere enn i matte, fordi vi ofte har fått kompliserte programmer presentert, uten at vi har lært de grunnleggende elementene i de

Mann 2023 m/IT1, BFY - I fysikkundervisningen på vgs følte jeg ikke vi "lærte" så mye, heller bare at vi fikk tildelt et program og skulle leke oss med det, uten at vi egentlig vissta hva alt betydde/gjorde

Mann 2023, MLREAL - Veldig utfordrene i starten, men når du skjønner programmerings språket mer blir desidert lettere å lære

Mann 2023 m/PX, MTFYMA - Vanskelig, da fysikken gjerne har tatt tak i en del mer komplekse funksjoner og problemstillinger

Når du arbeider med programmering, syntes du det er viktig å forstå hver kodelinje? Hvorfor eller hvorfor ikke?

Mann 2023 m/PX, MLREAL - Jeg synes det er viktig å forstå hver kodelinje. Det er ikke så viktig å ha skrevet hver eneste kodelinje selv, men hvis man skal bruke en annens kode, eller lage en med KI, så er det viktig at man forstå hva den gjør.

Mann 2022 m/IT1, IT2, MLREAL - Jo mer du forstår, jo mer kan du bruke det til, men jo mer tid du bruker på hver linje, jo mindre tid bruker du på de andre 50. Jeg synes det er viktig å vite hva de forskjellige kommandene gjør i sammenhengen enn å vite hva hver enkelt komponent gjør

Mann 2022, BFY - Ja da får man en mye større forståelse for hva det er koden gjør og hvordan den fungerer. Da kan man også endre på koden for å løse andre typer problemer.

Mann 2022 m/PX, BFY - Ja! Når man driver med basics vil jeg gjerne forstå alt sånn at jeg har den grunnleggende forståelsen til senere. Det blir mer motiverende når man forstår

Kvinne 2023, MTFYMA - Jeg synes det er viktig å forstå hver kodelinje, for da forstår jeg bedre hva koden gjør. Men samtidig er det mulig å skape seg et helhetlig inntrykk og forståelse av koden og hva den gjør, uten å forstå hver eneste linje.

Mann 2021, BFY - Ja. Jeg er nybegynner og kan ingenting. Det er viktig å få med seg alle detaljer.

Kvinne 2023, MTFYMA - Ja for å få en helhetlig forståelse, men noen koder må man bare godta

Mann 2020, MTFYMA - Nja, noen koder må man bare godta

Kvinne 2023, MTFYMA - Ja for da kan jeg lettere huske og skjønne hvordan jeg skal skrive en ny kode neste gang, og finne feil neste gang jeg skriver kode.

Mann 2023, BFY - Å forstå sammenhengen mellom linjene er bedre enn å forstå hver linje

Kvinne 2022, MTFYMA - Nei, ikke i en oppgave hvor du trenger bare å bruket programmet. Men hvis du vil lære noe nytt du kan bruke i fremtiden, så ja.

Mann 2021 m/PX, MTFYMA - Nei så lenge du forstår hvert trinn i prosessen er ikke hver kodelinje nødvendigvis relevant

Mann 2022, MTFYMA - Ja, siden jeg ikke har noen forkunnskaper er jeg opptatt av å forstå alt grunnleggende

Mann 2022, MTFYMA - Det er viktig å ha en god forståelse av hva de forskjellige linjene gjør, og forstå hvorfor de er der. Du trenger ikke nødvendigvis forstå alt av hvordan innebygde funksjoner fungerer, men du bør forstå hva de gjør.

Kvinne 2023 m/IT1, MTFYMA - Ikkje nødvendigvis kvar linje men for det meste forstå kvifor programmet fungerer

Kvinne 2021 m/IT1, IT2, MTFYMA - Ikke viktig, så lenge koden fungerer

Kvinne 2023, MTFYMA - Jeg vil gjerne forstå hver kodelinje, fordi hvis ikke føler jeg ikke eierskap til programmet

Mann 2021, MTFYMA - Ja egt. Jeg lærer best av å forstå hva jeg driver med

Mann 2023 m/IT1, BFY - Jeg mener det er viktigere å forstå helheten enn individene linjer

Mann 2023, MLREAL - Ja, men det er viktigere å forstå helheten

Mann 2023, MTFYMA - Ja, man må forstå hver enkelt linje for å forstå helheten

Hvordan har det å lære programmering påvirket din fysikkforståelse?

Mann 2021 m/ToF, BFY, Khanacademy - Man ser en del paralleller (litt spooky faktisk) mellom code og fysikk, så hvis jeg ikke hadde vært god i koding tviler jeg på at jeg hadde vært god i fysikk

Kvinne 2023, BFY - At eg har lært litt programmering hadde ikkje så mykje å seie for fysikkforståelsen min. Vi hadde så mykje programmering i fysikk, mest i matte der vi lærte det grunnleggande.

Mann 2019, BFY - Det å finne svaret på en matteoppgave som oppstår i fysikk kan hjelpe med å forstå fysikk, men jeg tror det hadde vært like illustrativt om jeg bare ble gitt svaret

Kvinne 2023, MTFYMA - Foreløpig ingenting siden vi ikke har hatt det her. På videregående var det mer et tillegg enn noe som bidro til videre forståelse

Kvinne 2023, MTFYMA - Har ikke hatt lærere som har knyttet programmeringen opp til noe særlig annet enn utregninger, så føler programmering bare er en måte å skrive fysikk på, ikke at det har fått meg til å forstå den noe bedre.

Mann 2023, BFY - Det har gitt med en mye bedre forståelse av integral og tilfeller med varierende akselerasjon som luftmotstand og verdensrommet

Kvinne 2022 - Det har hjulpet å bruke den logiske måten å tenke på som man gjør når man programmerer.

Kvinne 2022 m/ToF, NIO, MTFYMA - Ikke relevant, programmeringen jeg lærte var uavhengig av skolen, og den jeg gjorde i Teknologi og forskningslære var valgfri, og tilfeldigvis valgte jeg et overlapp med fysikk. Forståelsen min ble bedre, men det var uavhengig av lærernes undervisning, og det jeg lærte lærte jeg meg selv.

Kvinne 2021, MTFYMA - Ikke så mye foreløpig, men forhåpentligvis framover!

Mann 2021 m/PX, MTFYMA - Minimalt. Men den har forbedret min matematikk forståelse.

Kvinne 2023, BFY - Programmeringen jeg selv hadde i fysikk på videregående har ikke hjulpet meg noe spesielt, men jeg tror at den kunne ha gjort det, om den fikk litt mer fokus og tid

Mann 2023, BFY - Har ikke blitt påvirket mye, men fysikkforståelsen min har påvirket programmeringen

Mann 2023, MLREAL - Ikke stort, ser på det mest som et veldig bra hjelpe verktøy, mens du må kunne en grei del av teorien selv fra før

Mann 2023, BFY - Hittil har jeg brukt programmering for å løse problemer jeg kunne ha løst analytisk (kanskje litt eulermetode), så den har ikke påvirket den så mye.

Kvinne 2022, MTFYMA - Jeg har ikke fått noe mer fysikkforståelse av programmering, men har fått et verktøy til å hjelpe med beregninger og lage grafer

Har du lært programmering fra andre steder enn i skolen? Hvilke?

Kvinne 2023, MTFYMA - Prøvd å lære på nettet på egen hånd fordi opplæringen på skolen ikke var tilstrekkelig til å svare på de vanskelige programmeringsoppgavene i fysikk 2.

Mann 2020, MTFYMA - Jeg lærte ikke på vgs da jeg gikk ut i 2020. Fikk derimot litt opplæring i det på realfagsforkurs, men synes ikke det var godt nok

Har du andre tanker om ditt forhold til programmering i fysikk?

Kvinne 2023, BFY - Veldig vanskeleg, å gjere sjølv, men greit å forstå en allerede eksisterande kode.

Mann 2019, BFY - Det var ganger jeg følte lærerne antok at vi forsto kode jeg følte vi ikke hadde lært enda. Da måtte jeg jobbe en del for å catche opp med hva kodingen betydde

Kvinne 2022, MTFYMA - Jeg tenker det er viktig at en lærer det grunnleggende i fysikk før en begynner med programmering i faget. Dette for at faget skal dreie seg mer om fysikk og ressurser man kan bruke for å regne på fysikk, enn om bare programmering i seg selv.

Mann 2023 m/TOF1, TOF2, MTFYMA - Programmering i fysikk er som oftest veldig simpelt, kunne vært kult å lage mer komplekse koder, som kunne gitt oss andre kunnskaper enn det et tegneprogram kunne gitt oss.

Mann 2018, MTFYMA - Jeg anser det som noe jeg må gjennom, men har troen på at det vil bli kjekt proposjonalt med økende kompetanse.

Mann 2023, MTFYMA - Programmering kan gjøre flere ting mye enklere og kan gjøre simulasjoner og kalkulasjoner som ikke hadde vært mulig uten programmering. Så programmering er et ekstremt viktig og hjelpsomt verktøy i fysikken.

Kvinne 2022 m/TOF, MTFYMA - Gleder meg til jeg føler at jeg mestrer det og kan bruke det som hjelpemiddel

Vedlegg C: Informasjonsskriv for informanter til intervju

Deltakelse i forskningsprosjekt om programmering i fysikkundervisning

Formålet med prosjektet

Dette er et spørsmål til deg om du vil delta i et forskningsprosjekt hvor formålet er å undersøke læreres oppfatning av muligheter gitt til å bruke programmering i fysikkundervisning i videregående skole. Dette gjøres som del av forberedning til en masteroppgave, og kan og bli brukt i selve masteroppgaven dersom det blir relevant.

Hvorfor får du spørsmål om å delta?

Du får denne forespørselen fordi du som fysikklærer har verdifull innsikt og erfaringer med fysikkundervisningen i videregående skole, hvordan denne undervisningen blir organisert, og hvilke muligheter som blir gitt til fysikklærere.

Hvem er ansvarlig for forskningsprosjektet?

Forskningsprosjektet blir utført via NTNUs institutt for fysikk, som også er behandlingsansvarlige for personopplysningene som inngår i prosjektet.

Det er frivillig å delta

Det er frivillig å delta i prosjektet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg. Du kan når som helst bed om å få personopplysninger slettet ved å kontakte rubenhjo@stud.ntnu.no.

Hva innebærer det for deg å delta?

Som deltaker vil du delta på et personintervju, som kommer til å vare ca. 15-30 minutter. Det vil bli gjort lydopptak av intervjuet via en opptaker, som så vil lagres kryptert på en ekstern harddisk.

Kort om personvern

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler personopplysningene konfidensielt og i samsvar med personvernregelverket. Du kan lese mer om personvern under.

Med vennlig hilsen

Berit Bugnum
(Forsker/veileder)

Ruben Hvatum Johnsen

Utdypende om personvern – hvordan vi oppbevarer og bruker dine opplysninger

Opplysningene vil kun være tilgjengelig for student og veileder. Lyddopptaket, som vil være eneste personopplysning som samles inn, vil krypteres og lagres på ekstern harddisk. Ingen personopplysninger vil publiseres.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra NTNU, Institutt for fysikk, har personverntjenestene ved Sikt – Kunnskapssektorens tjenesteleverandør, vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter

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

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

Vi vil gi deg en begrunnelse hvis vi mener at du ikke kan identifiseres, eller at rettighetene ikke kan utøves.

Hva skjer med personopplysningene dine når forskningsprosjektet avsluttes?

Prosjektet vil etter planen avsluttes omtrentlig 03.06.2024, når masteroppgaven leveres. Opplysningene vil da slettes.

Spørsmål

Hvis du har spørsmål eller vil utøve dine rettigheter, ta kontakt med Berit Bungum, berit.bungum@ntnu.no. Vårt personvernombud: Thomas Helgesen, thomas.helgesen@ntnu.no, 93079038

Hvis du har spørsmål knyttet til Sikts vurdering av prosjektet, kan du ta kontakt på e-post: personverntjenester@sikt.no, eller på telefon: 73 98 40 40.

Jeg har mottatt og forstått informasjon om prosjektet *Programmering i fysikkundervisning* og har fått anledning til å stille spørsmål. Jeg samtykker til:

- å delta i personintervju
- at mine opplysninger behandles frem til prosjektet er avsluttet

Underskrift: _____

