

Masteroppgåve

**Korleis programmeringsoppgåver kan knytast  
til kjerneelementa og programmering i  
matematikkundervisinga**

Utvikling av eit rammeverk for å vurdere  
programmeringsoppgåver etter LK20

Torgeir Ølnes

105 sider

Erfaringsbasert master i undervisning og læring,  
spes. i matematikk

2023



HØGSKULEN  
I VOLDA

## Samandrag

Denne masteroppgåva undersøker korleis programmeringsoppgåver i matematikklærebøker etter innføringa av LK20, varetek 5 sentrale kjerneelement, og implementerer programmering. Ein del av prosessen var å vidareutvikle to eksisterande rammeverk til eit nytt rammeverk til bruk i vurdering av programmeringsoppgåver. Temaet er valt på grunnlag av at lærarar i stor grad nyttar lærebøker i undervisninga (Mullis et al., 2012). Det er derfor viktig med eit analyseverktøy som kan hjelpe lærarar og anna undervisningspersonell med å gjere gode val når det kjem til lærebøker.

Studien har følgande problemstilling:

**«Korleis programmeringsoppgåver frå 3 matematikklæreverk er knytte opp mot kjerneelementa og innføring av programmering i Kunnskapsløftet 2020?»**

For å svare på problemstillinga er det nytta tre forskingsspørsmål:

- Kva kjenneteiknar den totale mengda programmeringsoppgåver i studien
- Kva er forskjellen mellom lærebøkene som er nytta i studien
- Kva er forskjellen på programmeringsoppgåver i matematikk og andre matematikkoppgåver

Læreverka som er analyserte i studien, er: Matemagisk 9 (Kongsnes & Wallace, 2020b), Matematikk 9 (Hjardar & Pedersen, 2021a) og Maximum 9 (Tofteberg et al., 2021).

Oppgåvene som er analyserte er henta frå kapitla om sannsyn. Rammeverket legg vekt på inkludering av kjerneelement og programmering i oppgåvene og nyttar kvantitativ innhaldsanalyse som metode. Analysen tek utgangspunkt i korleis læreverka presenter oppgåvene, og ikkje korleis lærarar kan nytte oppgåvene i klasserommet eller som ledd i eit undervisningsopplegg.

Rammeverket kan nyttast til å analysere andre og nye læreverk som er tilpassa LK20, og kan også nyttast til å analysere eigne oppgåver og undervisningsforløp, både med tanke på kjerneelementa og med tanke på programmeringsferdigheiter.

Funn frå studien viser at programmeringsoppgåver, totalt sett, ikkje inkluderer kvart kjerneelement like mykje, og at det er kjerneelementet *modellering og anvendingar* som er mest nytta. Det er relativt stor forskjell på korleis læreverka i studien inkluderer kjerneelementa i programmeringsoppgåvene. Det er også gjort funn som syner at

programmeringsoppgåver vektlegg andre kjerneelement enn oppgåver som ikkje er programmeringsoppgåver. Med tanke på programmering i oppgåvene var det også svært varierende resultat mellom lærebøkene. Det er ikkje mogleg å konkludere om programmeringsoppgåver generelt sett underbygger kompetanseutvikling eller om programmeringsoppgåvene trenar programmeringsferdigheiter.

## Abstract

This master's thesis investigates how programming tasks in mathematics textbooks after the introduction of LK20, take care of 5 central core elements, and implement programming. Part of this process was to further develop two existing frameworks into a new framework for use in the assessment of programming tasks. The topic was chosen on the basis that teachers largely use textbooks in teaching (Mullis et al., 2012). It is therefore important to have an analysis tool that can help teachers and other teaching staff to make good choices when it comes to textbooks.

The study will answer the following question:

**"How programming tasks from 3 mathematics textbooks are linked to the core elements and introduction of programming in Kunnskapsløftet 2020?"**

To answer the problem, three research questions are used:

- What characterizes the total amount of programming tasks in the study
- What is the difference between the textbooks used in the study
- What is the difference between programming-assignments in mathematics and other mathematics assignments

The textbooks analysed in the study, are: Mathematics 9 (Kongsnes & Wallace, 2020b), Mathematics 9 (Hjardar & Pedersen, 2021a) and Maximum 9 (Tofteberg et al., 2021). The tasks that have been analysed are taken from the chapters on probability. The framework emphasizes the inclusion of core elements and programming in the tasks and uses quantitative content analysis as a method. The analysis is based on how the textbooks presents the tasks, and not how teachers can use the tasks in the classroom or as part of a teaching plan.

The framework can be used to analyse other and new textbooks that have been adapted to LK20, and can also be used to analyse the teachers own tasks and teaching plans, both in terms of the core elements and in terms of programming skills.

Findings from the study show that programming tasks, overall, do not include each core element equally, and that it is the core element *modelling and applications* that is used the most. There is a relatively large difference in how the textbooks in the study includes the core elements in the programming tasks. Findings have also been made, which show that programming tasks emphasize different core elements than tasks that are not programming

tasks. Regarding programming in the assignments, there were also very variable results between the textbooks. It is not possible to conclude whether programming tasks generally underpin competence development or whether the programming tasks train programming skills.

## Forord

Eit studieløp på 3 semester er komen til sin ende. Ei lenger pause frå arbeidet som lærar har resultert i ei masteroppgåve, ny kunnskap og nyttige erfaringar. Eg føler meg privilegert som i eit halvt år har fått fordjupe meg i eit tema og eit fagfelt som fascinerer og underheld meg, både som lærar og privat. Takk til lærarutdanning ved Høgskulen i Volda som har gitt meg moglegheita til å ta ein erfaringsbasert master.

Rettleiaren min, Andreas Brandsæter, må få mykje av æra for at prosessen med å skrive masteroppgåva har vore ei fin oppleving. Han har vist engasjement i arbeidet mitt og kunnskap om temaet eg har valt. Han er tilgjengeleg og har oppmuntra meg når eg har trengt det. Takk!

Biblioteket ved Høgskulen i Volda fortener ein medalje for å svare på alle mine spørsmål om smått og stort. Dei har spart meg for svært mykje frustrasjon i arbeidet med masteroppgåva og er ei gullgruve for oss studentar.

Ein stor takk til Runar Lie Berge, Hilde Opsal, Frode Opsvik, Torbjørn Frantsen og Joakim Erik Gustav Olsson som arbeider ved Høgskulen i Volda, og som har hatt sine dører og e-postkasser opne for meg når eg har hatt behov tips eller avklaringar.

Takk til Stig Eriksen ved UIA, som er halvparten av duoen bak artikkelen som ga meg inspirasjon til tema for masteroppgåva, og som har svart meg kjapt og godt på spørsmål som har dukka opp i arbeidet mitt.

Eg vil rette ei ekstra stor takk til min sambuar og våre to born. De har vore tolmodige og støttande. No skal vi snart bygge trehytte og fiske.

Takk til foreldre og svigerforeldre for både barnevakt-tenester, innlosjering på undervisningsdagar og ein dytt i ryggen i ny og ne. Ei spesiell takk til mor mi for god hjelp i innspurten før levering.

Og ikkje minst: Takk til grønsakshagen vår. Den har gitt meg meir zen enn grønsaker denne våren, og det er heilt perfekt.

Ørskog, mai 2023

Torgeir Ølnes

## Innhald:

Samandrag .....	i
Abstract .....	iii
Forord .....	v
Innhald:.....	vi
Tabellar: .....	ix
Figurliste: .....	ix
1 Innleiing .....	1
1.1 Bakgrunn for studien .....	1
1.1.1 LK20 og programmering.....	2
1.1.2 LK20 og kjerneelementa .....	3
1.2 Problemstilling.....	4
1.3 Avgrensingar .....	5
1.4 Oppbygging av oppgåva.....	5
2 Kunnskapsgrunnlag.....	7
2.1 Læreplanen .....	7
2.2 Grunnlag for og utvikling av kjerneelementa.....	8
2.2.1 Matematisk forståing.....	9
2.2.2 Matematiske kompetansemodellar .....	10
2.2.3 Kjerneelementa for matematikk i LK20.....	12
2.3 Kva er programmering?.....	13
2.4 Programmering og kompetansar.....	15
3 Metode.....	17
3.1 Metoden .....	17
3.2 Forskingsdesign, forskingsmetode og metodeval.....	17
3.2.1 Kvantitativ innhaldsanalyse .....	18

3.3	Metoden sitt overordna design .....	21
3.4	Utvikling av analytisk rammeverk .....	22
3.4.1	Variablar knytt til kjerneelementa.....	23
3.4.2	Variablar knytt til programmering .....	25
3.4.3	Operasjonalisering av variablar (kodeinstruks).....	28
3.5	Utval av oppgåver.....	33
3.6	Presentasjon av lærebøker .....	35
3.6.1	Matemagisk 9 .....	36
3.6.2	Matematikk 9.....	36
3.6.3	Maximum 9 .....	37
3.7	Forskingskvalitet .....	38
3.7.1	Reliabilitet .....	38
3.7.2	Validitet.....	42
3.7.3	Forskingsetiske vurderingar .....	43
4	Presentasjon av datagrunnlaget og resultat .....	45
4.1	Resultat av filtrering av oppgåver .....	45
4.2	Resultat av oppgåveanalysane .....	48
4.2.1	Resultat knytt til samlevariablane .....	48
4.2.2	Resultat knytt til dei individuelle variablane.....	52
4.2.3	Resultat knytt til individuelle oppgåver .....	61
4.2.4	Resultat knytt til programmeringskonsept.....	64
5	Drøfting .....	68
5.1	Drøfting knytt til resultat .....	68
5.1.1	Drøftingar av hovudfunna .....	69
5.1.2	Drøftingar rundt PR-variabelen.....	73
5.2	Drøftingar knytte til rammeverket .....	75
5.2.1	Utval av oppgåver .....	76



5.2.2	Oppgåveanalyse .....	77
5.2.3	Konstruksjon av rammeverket .....	79
5.2.4	Kva er rammeverket ikkje ser på? .....	80
5.2.5	Oppsummering og praktisk nytte av rammeverket .....	81
6	Oppsummering og vegen vidare .....	83
6.1	Kommentarar om kjerneelementa .....	83
6.2	Kommentarar om programmering .....	84
6.3	Kommentar om rammeverket .....	85
6.4	Vegen vidare .....	87
	Kjeldeliste .....	88
	Vedlegg 1: Kodeskjema .....	1

## Tabellar:

Tabell 1 Kilpatrick et al. sin kompetansemodell.....	12
Tabell 2 Fullstendig oversikt over variablar sortert etter kjerneelement.....	24
Tabell 3 Oversikt over variablar innanfor programmering .....	28

## Figurliste:

Figur 1 Kompetanseomgrepet .....	10
Figur 2 Oversikt over ulike datatypar.....	18
Figur 3 Oppgåve frå eksamen 10. trinn.....	27
Figur 4 Oppgåve frå eksamen 1P .....	27
Figur 5 Eksempel på oppgåve som ikkje vert rekna som programmeringsoppgåve.....	34
Figur 6 Eksempel på programmeringsoppgåve.....	35
Figur 7 Eksempeloppgåve med rekneark som ikkje er ned utvalet.....	35
Figur 8 Gjennomsnittskåre for runde 1 og 2 av analysen .....	40
Figur 9 Antal oppgåver med ulik skåre i analyserunde 1 og 2.....	41
Figur 10 Reliabilitetsberekning pr. variabel.....	42
Figur 11 Oppgåver i utvalet prosentvis fordelt på lærebøker.....	46
Figur 12 Andel oppgåver i utvalet pr. lærebok .....	46
Figur 13 Plassering i kapittelet for oppgåvene i utvalet.....	47
Figur 14 Om oppgåver i utvalet er nemnt eksplisitt eller om dei står i eit programmeringskapittel .....	48
Figur 15 Gjennomsnitt for kjerneelementa og programmering, for oppgåver i utvalet .....	49
Figur 16 Gjennomsnitt samlevariablar for kjerneelementa, oppgåver i og utanfor utvalet .....	50
Figur 17 Gjennomsnitt for samlevariablar for oppgåver i utvalet, pr lærebok .....	51
Figur 18 Gjennomsnitt variablar for kjerneelement, i og utanfor utvalet, pr. lærebok .....	52
Figur 19 Gjennomsnittskåre for alle variablar for oppgåver i utvalet.....	53
Figur 20 Eksempeloppgåve .....	54
Figur 21 Eksempeloppgåve .....	56
Figur 22 Gjennomsnitt for alle variablane for oppgåver i utvalet, pr lærebok.....	57
Figur 23 Andel oppgåver i kat. 0, 1 eller 2 for alle variablar for oppgåver i utvalet .....	58
Figur 24 Andel oppgåver som skårar 1 eller 2, pr. variabel i Matemagisk 9 .....	59

Figur 25 Andel oppgåver som skårar 1 eller 2 pr. variabel i Matematikk 9 .....	59
Figur 26 Andel oppgåver som skårar 1 eller 2 pr. variabel i Maximum 9 .....	60
Figur 27 Andel oppgåver i utvalet med skåre 2 pr. variabel, pr lærebok.....	61
Figur 28 Gjennomsnittskåre pr. oppgåve i utvalet, sortert.....	62
Figur 29 Gjennomsnittskåre på oppgåver i utvalet, sortert kronologisk .....	63
Figur 30 Gjennomsnittskåre for variablar knytt til hvv. kjerneelementa og programmering, pr oppgåve i utvalet .....	64
Figur 31 Andel oppgåver i kat. 0, 1 eller 2 for PR 3-variablar, for oppgåver i utvalet.....	65
Figur 32 Gjennomsnittskåre variablar for programmeringskonsept, pr lærebok, oppgåver i utvalet.....	65
Figur 33 Antal oppgåver i utvalet som fekk skåren 2, pr variabel, pr. læreverk.....	66
Figur 34 Antal variablar som fekk 1 i skåre i oppgåver i utvalet, pr variabel, pr læreverk .....	67
Figur 35 Gjennomsnittskåre for variabelen PR for oppgåver i utvalet ,pr. lærebok og plassering i kapittelet.....	67
Figur 36 Gjennomsnittskåre pr. oppgåve i utvalet, nummerert og samanlikna med og utan PR 3-variabelen.....	74
Figur 37 Gjennomsnitt for samlevariablar for oppgåver i utvalet, pr lærebok .....	85

# 1 Innleiing

Denne masteroppgåva analyserer programmeringsoppgåver frå ulike lærebøker. Analysen ser på korleis desse oppgåvene er knytte opp mot dei forskjellige kjerneelementa i matematikk i Kunnskapsløftet 2020. I dette første kapittelet blir bakgrunn for val av tema presentert, deretter kjem problemstilling, avgrensingar og til slutt ein gjennomgang av oppbygginga av oppgåva.

## 1.1 Bakgrunn for studien

Eit av dei viktigaste styringsdokumenta for undervisinga i den norske skulen, er læreplanane (Imsen, 2020, s. 277; NOU 2014: 7, s. 96). Læreplanverket er eit politisk manifest (Engelsen, 2020), kjem med ujamne mellomrom og kan gi lærarane nye og andre fokusområde og måtar å arbeide på enn dei tidlegare har hatt. Læreplanverket er regjeringa sine styringsdokument for den norske skulen og består av forskrifter til opplæringslova (Utdanningsdirektoratet, u.å.-f), og kan dermed ikkje veljast vekk av lærarar eller skuleeigarar i den offentlege skulen (Opplæringslova, 1998, §2-3).

I 2020 kom det eit nytt læreplanverk for den norske skulen. Prosessen med å gå frå det gamle læreplanverket Kunnskapsløftet 06 (LK06) til det nye læreplanverket Kunnskapsløftet 2020 (LK20) kallast Fagfornyinga (Utdanningsdirektoratet, u.å.-b). Av dei endringane som skjedde i Fagfornyinga, er det to som er utgangspunktet for denne studien: Innføring av kjerneelement i alle fag og innføring av programmering i m.a. matematikkfaget for heile grunnskuleløpet. Dette betyr ikkje at omgrepet programmering har vore fråverande i den norske skulen tidlegare, m.a. har grunnskulen hatt moglegheit til å tilby eit valfag i programmering sidan 2016 (Utdanningsdirektoratet, u.å.-d).

Innføring av kjerneelementa i alle fag og programmering i nokre fag krev ny kompetanse hjå lærarar. Etter innføringa av LK20 er det komen forskning som viser at ei slik kompetanseheving kanskje ikkje kjem fort nok. Ei undersøking viser at 23 % av lærarar rapporterer liten eller ingen opplæring og 27 % rapporterer litt opplæring i forbindelse med innføringa av LK20 (Møller et al., 2023). Forsking viser at lærarar kan oppleve dårleg tilgang på undervisningsressursar (Strand, 2022, s. ii) og at lærarar er utrygge på å inkludere programmering i undervisinga (Hirsch, 2022, s. i). Spesifikt for innføring av programmering i matematikken er behovet for kompetanse framheva (Burner et al., 2022).

Generelt er oppgåver frå m.a. læremiddel framheva som ein viktig ressurs etter innføringa av LK20 (Burner et al., 2022). Men dersom lærarar delvis manglar kompetanse og generelt får berre litt opplæring i dei nye elementa frå LK20, korleis kan då lærarar avgjere om oppgåvene dei gir elevane er av god kvalitet og hensiktsmessige? Studien freistar å utvikle eit analyseverktøy som skal svare på denne utfordringa ved å analysere oppgåver som er karakterisert som programmeringsoppgåver opp mot kjerneelementa og elevane sine behov for programmeringsferdigheiter.

### 1.1.1 LK20 og programmering

I sjølve læreplanen for matematikk for grunnskulen (Utdanningsdirektoratet, 2019b) er ordet «programmering» eller «programmere» nemnt 10 plassar. Ordet «programmering» finst i den grunnleggjande ferdigheita «Digitale ferdigheiter» (Utdanningsdirektoratet, 2019b, s. 5). Ordet «programmering» er òg ein del av kompetansemåla eller undervegsvurderingsmåla for alle trinna f.o.m. 5. trinn t.o.m. 10. trinn (Utdanningsdirektoratet, 2019b, s. 8–14). Fleire kompetansemål som ikkje eksplisitt nemner programmering vil i tillegg vere sterkt knytt til programmering, slik som t.d. «lage og følgje reglar og trinnvise instruksjonar i leik og spel» (Utdanningsdirektoratet, 2019b, s. 6). I kapittel 2.3 i oppgåva vert omgrepet programmering definert tydelegare.

I over halve grunnskuleløpet er programmering ein del av pensum i matematikk, men dette har ikkje vore ein del av undervisninga i lærarstudiet før dei siste åra. Dette gjer at mange lærarar må undervise i eit fagelement i matematikken dei ikkje tidlegare har undervist i.

I matematikken er det lang tradisjon for at læreverket styrer den aktiviteten som skjer i undervisninga (Remillard, 2005). I TIMMS kjem det fram at 97 % av undervisninga læraren gir, er basert på læreboka (Mullis et al., 2012, s. 392). Omgrepet «lærebok» i TIMMS vert tolka som «analog lærebok», all den tid digitale læreverk ikkje var vanleg i 2011. Lærebøker vert også rapportert som støttande for lærarane i innføringa av LK20 (Burner et al., 2022).

Tidlegare har utviklinga av nye læreplanar vore årsak til at nye lærebøker vert utvikla (Gjone, 2003, s. 265). No er det komen eit nytt læreplanverk, og det kan tenkast at lærarar stiller seg annleis til bruk av lærebøker etter dette. Nye læreplanomgrep som t.d. «djupnelæring» gjer at lærarar og skuleeigarar kanskje tenkjer på nye måtar og går vekk frå ei undervisningsform der lærebøkene vert brukte til oppgåveløysingsføremål og i like stor grad styrte undervisninga.

Det kjem ei ny TIMMS-undersøking i 2023 med gjennomføring i løpet av mars og april og denne undersøkinga vil gje oppdatert data på bruk av lærebøker i undervisninga.

Tidlegare var det slik at nye læreverk måtte ha godkjenning av departementet, og at m.a. læreplanen låg til grunn for godkjenninga (Gjone, 2003, s. 265). Denne ordninga vart avslutta i juni 2000 (Bratholm, 2001), etter å ha eksistert i 111 år. Det er altså fritt fram å produsere læreverk med dagens reglement. Då er det opp til kvar skuleeigar å vurdere kvaliteten og relevansen til læreverka.

Arbeidet med å kvalitetssikre undervisning er viktig. Det finst didaktiske relasjonsmodellar (t.d. Imsen, 2020, s. 320; Lillejord et al., 2010, s. 67) som inneheld faktorar som er gjensidig avhengige av kvarande for å lage relasjonell undervisning. Desse faktorane er m.a. mål og læringsaktivitetar i undervisninga. I planlegginga av undervisning kan eit element være utval av oppgåver som elevane skal løyse. I eit slik arbeid, kanskje spesielt med tanke på nye fagelement som t.d. programmering, er det nyttig med eit analyseverktøy som ser på kva mål læringsaktivitetane i undervisninga dekkjer.

Det er mange omsyn å ta når læraren skal velje ut oppgåver ein skal nytte i undervisninga, men den grunnleggjande tanken er at det arbeidet som vert gjort i skulen skal svare til forventingane i læreplanverket. Med LK20 er det ikkje lenger berre kompetansemåla som inngår i ei vurdering av kva læraren skal undervise, no spelar også dei nye kjerneelementa ei viktig rolle.

### 1.1.2 LK20 og kjerneelementa

I tillegg til å innføre programmering i fleire fag, innførte også LK20 kjerneelement i alle fag. Kjerneelementa er omtala som «... det viktigste faglige innholdet elevane skal arbeide med i opplæringa» (Utdanningsdirektoratet, 2019d). Kjerneelementa er altså det *viktigaste* elevane skal hente ut av faget (Meld. St. 28 (2015–2016), s. 34). Dette kan i utgangspunktet sjå ut som ein konkurrent til kompetansemåla, men Utdanningsdirektoratet (2019d) seier at dei heng saman. Medan kjerneelementa kan nyttast til å fylle kompetansemåla med innhald, konkretiserer kompetansemåla eit eller fleire kjerneelement (Utdanningsdirektoratet, 2022b). Eksempel på dette kan ein finne i Utdanningsdirektoratet sitt interaktive planleggingsverktøy, som er ein del av kvar digitale læreplan.

## 1.2 Problemstilling

Studien skal ta føre seg matematikkoppgåver knytte til programmering. Basert på innhaldet i kapittelet over, vil masteroppgåva belyse denne problemstillinga:

«Korleis programmeringsoppgåver frå 3 matematikklæreverk er knytte opp mot kjerneelementa og innføring av programmering i Kunnskapsløftet 2020?»

Studien vil vidareutvikle eit eksisterande rammeverk for analyse av matematikkoppgåver og nytte dette analyseverktøyet for å gje svar på problemstillinga. I tillegg til at analyseverktøyet ser på korleis oppgåvene frå lærebøkene svarar til dei forskjellige kjerneelementa, vil verktøyet også vurdere oppgåvene opp mot trening av programmeringsferdigheiter i matematikken.

For å svare på problemstillinga er det utvikla tre forskingsspørsmål:

- Kva kjenneteiknar den totale mengda programmeringsoppgåver i studien

Ved å analysere ei større mengde oppgåver kan det være mogleg å identifisere korleis læreverka vektar kjerneelementa. Ved seinare analyser med utgangspunkt i rammeverket frå denne studien, kan ei slik oversikt danne grunnlaget for vurdering av resultatata i den nye analysen

- Kva er forskjellen mellom lærebøkene som er nytta i studien

Dette forskingsspørsmålet kan gje svar på om alle læreverka gjer det same, eller om det er forskjellige måtar å tenke rundt programmerings i matematikkundervisinga.

- Kva er forskjellen på programmeringsoppgåver i matematikk og andre matematikkoppgåver

Datamateriale som beskriv oppgåver som ikkje er kategorisert som programmeringsoppgåver kan gje indikatorar kva som kjenneteiknar programmeringsoppgåvene, og ein kontekst å vurdere programmeringsoppgåvene i.

Desse forskingsspørsmåla må sjåast i lys av problemstillinga og avgrensingar som er gjort i oppgåva.

Det må understrekast at lærebøkene ikkje vil bli rangerte, og inga lærebok vil bli kåra som den beste læreboka gjennom studien.

### 1.3 Avgrensingar

Det er læreverk for ungdomstrinnet i norsk skule, meir konkret fysiske lærebøker med supplement, og som er tilpassa LK20 som vert vurdert i studien. Læreverka vil vere knytt til matematikkundervisinga i 9. trinn, og utvalet lærebøker vil bestå av dei lærebøkene som er knytt til dette trinnet. Nokre læreverk har berre ei bok for 9. trinn, medan andre har fleire. Totalt er det 5 lærebøker, inkludert supplement, som vert vurdert. Ein detaljert gjennomgang av lærebøkene i utvalet finnast i kap. 3.6.

Studien fokuserer på oppgåver som kan kategoriserast som programmeringsoppgåver i lærebok-kapitla om sannsyn. Av kompetansemåla som eksplisitt omfattar programmering, er det kompetansemålet om sannsynsrekning i 9. trinn som er det mest konkrete og difor vart sannsyn valt ut som fagområde. I preanalysen av lærebøkene vart det også notert at alle bøkene hadde kapittel om sannsynsrekning eller tilsvarande.

I problemstillinga er det nytta omgrepet «programmeringsoppgåver», og i studien er dette tolka som oppgåver som eksplisitt legg til rette for programmering. Dette vert utdjupa meir i kap. 3.5 som omhandlar utval av oppgåver.

Det er fysiske lærebøker som vert vurdert i studien. Av erfaring viser det seg at nettbaserte oppgåver kan verte endra, erstatta eller forsvinne utan at det vert loggført. Dette gjere det vanskeleg å etterprøve resultata i studien. Moglegheita for dynamiske stiar elevane kan følge gjennom dei nettbaserte læreverka kan gjere det vanskeleg å vurdere t.d. progresjon i det matematiske temaet og kan føre til forvirring om kva oppgåve som til ein kvar tid er med i utvalet av oppgåver og ikkje.

I studien er det ikkje læraren sin bruk av oppgåvene som vert vurdert. Oppgåver i lærebøkene kan utvidast av lærarar på mange måtar, m.a. ved å gje tilleggsoppgåver eller oppfølgingsspørsmål, eller endre miljøet ved å t.d. setje elevane i grupper. Dette vil i så fall kunne endre oppgåva i sin heilheit, og analyseverktøyet vil ikkje fange opp dette. Det er derimot mogleg å nytte analyseverktøyet til å finne ut kva kjerneelement ei oppgåve ikkje har dekt, og slik velje å utvide oppgåva til å dekkje kjerneelement etter ynskje.

### 1.4 Oppbygging av oppgåva

Kapittel 2 er eit kunnskapsgrunnlag som tek føre seg omgrep, definisjonar og gjennomgang av teoriar. Kapittel 3 beskriv metoden og korleis analyseverktøyet vart utforma. Kapittel 4 presenterer datagrunnlaget og presenterer resultata som vert nytta i drøftinga i kapittel 5. I



kapittel 5 vert resultata drøfta, og utforminga og nytta av analyseverktøyet vert også drøfta.  
Til slutt kjem kapittel 6 med ei oppsummering, avslutning og tankar om vidare forskning.

## 2 Kunnskapsgrunnlag

### 2.1 Læreplanen

Læreplanen er, som vist i kapittel 1.1, m.a. eit styringsdokument frå myndigheitene. Å gje ei heilt korrekt skildring av kva ein læreplan er, er vanskeleg (Imsen, 2020, s. 278). Lawrence Stenhouse har gitt ein definisjon: «Ein læreplan er eit forsøk på å kommunisere dei viktigaste prinsippa og eigenskapane ved eit pedagogisk opplegg på ein slik måte at den er open for kritisk gransking og mogleg å effektivt overføre til praksis» (Stenhouse, 1975, s. 4, mi omsetjing). Om denne definisjonen skulle definere den norske læreplanen, vil den ikkje treffe like godt på den engelske versjonen av ordet, «curriculum». Det engelske ordet inneheld meir enn berre læreplanen (Gjone, 2003, s. 265; Imsen, 2020, s. 278). Imsen skriv at omgrepet «læreplanen» kan nyttast om intensjonen bak undervisinga, medan den engelske varianten dekkjer også det som faktisk skjer i undervisinga. Samtidig seier Niss (1996) at grunngevinga for matematikk i læreplanen ikkje er tydeleg, og at grunnane til å inkludere matematikken i læreplanane er implisitte, indirekte og vage. Sidan det er vanskeleg å identifisere måla og grunngevinga for matematikken i læreplanen (Niss, 1996) er kanskje Imsen sin definisjon av omgrepet læreplanen hensiktsmessig for dei som skal nytte læreplanen, men at dei mest grunnleggande intensjonane bak å inkludere programmering i læreplanen ikkje kjem heil til syne.

Utdanningsdirektoratet skriv at det er i hovudtrekk tre måtar å sjå læreplanane på: Innhaldsorienterte, kompetansestyrte (målstyrte) og prosessorienterte (NOU 2014: 7, s. 97). Utdanningsdirektoratet skriv vidare at dei norske læreplanane gjekk frå innhaldsorienterte, med rammer som t.d. tema og arbeidsmåtar, til målstyrte i LK06. Sjølv om læreplanen ikkje er basert på berre éin modell er det i LK06 at overgangen er tydelegast, skriv Utdanningsdirektoratet. Ein målstyrt læreplan opererer med uttrykk som «kompetansemål» «kjenneteikn på måloppnåing» og har mål for kva eleven skal meistre (NOU 2014: 7, s. 98; St.meld. nr. 30 (2003–2004), s. 25). Det kan nemnast at barnehagen i Noreg har ein innhaldsstyrt rammeplan (Utdanningsdirektoratet, 2017), til forskjell frå ein målstyrt læreplan slik den norske skulen har.

Vidare i denne oppgåva vil heile den nasjonale læreplanen refererast til som «læreplanverket». Utdanningsdirektoratet kallar sjølv læreplanen for læreplanverket

(Utdanningsdirektoratet, u.å.-f). Læreplanverket består av tre delar: Overordna del, fag- og timefordelinga og læreplanar for kvart fag (fagplanar).

Fag- og timefordelinga er ei oversikt over kva fag som vert tilbydd elevane i skulen, og kva undervisingstimal desse faga er sett opp med. Denne oversikta viser også kva fag som byggjer på kvarande i det vidaregåande løpet. Sjå Utdanningsdirektoratet (2022a) for utfyllande informasjon.

I den overordna delen av læreplanverket står «...verdier og prinsipper for grunnopplæringen» (Utdanningsdirektoratet, u.å.-g). Ei viktig presisering her er at grunnopplæringa gjeld heilt frå 1. klasse på barnetrinnet til elevane er ferdige med eit utdanningsprogram i den vidaregåande skulen. Grunnskulen, derimot, er avgrensa til den 10-årige grunnskuleopplæringa (Utdanningsdirektoratet, 2021, s. 28). Den overordna delen inneheld ikkje fagspesifikke detaljar, men lagar ein paraply som læreplanane for kvart fag ligg under. Denne paraplyen beskriv kva verdigrunnlag læreplanane er tufta på, kva prinsipp som ligg til grunn for den læringa som skal skje og rammer for skulens praksis i gjennomføring av utdanningsløpa. Saman med læreplanane for kvart fag gir overordna del eit meir komplett bilete av korleis undervisinga bør gjennomførast.

Det er i læreplanane for kvart fag at hovudmåla til undervisinga vert kommunisert. I denne oppgåva er det læreplanar for kvart fag som er vesentleg, og desse vert vidare kalla berre «læreplanar». Desse læreplanane inneheld det som er knytt opp til undervising i kvart fag, og har element som kompetansemål for faget, kva kjerneelement faget inneheld og korleis faget dekkjer dei grunnleggande ferdigheitene (Utdanningsdirektoratet, 2019b). I tillegg inneheld læreplanane informasjon om fagrelevans og vurdering, og korleis ein kan arbeide med faget i dei tverrfaglege tema som ligg inne i læreplanverket. Læreplanane har lik struktur, med eit unntak: matematikk 1-10 har fått kompetansemål for oppnådd kompetanse i alle årsklasser, bortsett frå første trinn.

## 2.2 Grunnlag for og utvikling av kjerneelementa

Kunnskapsdepartementet (2018a) skriv at dei fornyar faga i LK2020 og at ei endring i matematikkfaget vert at «Elevene skal jobbe mer med metoder og tenkemåter slik at de får større forståelse for faget». Stortingsmeldinga (Meld. St. 28 (2015–2016), s. 5) om Fagfornyninga, heiter «Fag – Fordypning – Forståelse». Frå læreplanen sitt avsnitt om matematikkfaget sin relevans, står det

Kritisk tenkning i matematikk omfatter kritisk vurdering av resonnementer og argumenter og kan ruste elevene til å gjøre egne valg og ta stilling til viktige spørsmål i sitt eget liv og i samfunnet. Når elevene får tid til å tenke, reflektere, resonnere matematisk, stille spørsmål og oppleve at faget er relevant, legger faget til rette for kreativitet og skapertrang. Matematikk skal bidra til at elevene utvikler evne til å jobbe selvstendig og samarbeide med andre gjennom utforskning og problemløsning, og kan bidra til at elevene blir mer bevisste på sin egen læring. Når elevene får mulighet til å løse problemer og mestre utfordringer på egen hånd, bidrar dette til å utvikle utholdenhet og selvstendighet. (Utdanningsdirektoratet, 2019b, s. 2)

Dette viderefører tankegangen bak LK06 og peiker på at elevane må lære seg å jobbe sjølvstendig med problemløysing og resonnering (Nosrati & Wæge, 2015, s. 2).

### 2.2.1 Matematisk forståing

Skemp (1976) skil mellom instrumentell og relasjonell forståing av matematikk. Dette vert i litteraturen ofte teken fram når det kjem til undervisningsmetodar (Nosrati & Wæge, 2015, s. 4). Skemp (1976) skriv sjølv at han fekk påpeikt at det var to versjonar av ordet «forståing» innan matematikken; ein som Skemp sjølv ikkje ville definert som undervising tidlegare og som han refererer til som «rules without reasons» (1976, s. 2), og ein anna versjon som innebar at elevane hadde forståing for den matematikken dei dreiv med. Ei samanlikning Skemp brukte til å synleggjera dette, er by-besøket (1976, s. 13): Skemp besøkte ein by for første gong. Han lærte seg fort ei rute mellom bustaden, arbeidsplassen og eit par ruter til. Når han hadde fritid, nytta han den til å gå rundt i byen og utforske området. Sjølv om det var umogleg for andre som observerte han å sjå, arbeidde han med å lage seg eit mentalt bilde av byen. Dersom det ein dag var vegarbeid på ein av rutene han hadde lært seg kunne han, takka være det mentale kartet han hadde konstruert seg, no finne ei anna rute mellom to punkt, sjølv utan å ha gått den ruta før. Dette er det Skemp refererer til som relasjonell forståing (1976, s. 14). Når elevane berre lærer faste ruter til målet, algoritmar, gir dette ei instrumentell forståing av matematikken og elevane er avhengig av ekstern hjelp for å lage seg nye ruter. Med relasjonell forståing tek ein, i prinsippet, vekk behovet for hjelp frå læraren til å forklare nye vegar frå eit problem til ei løysing (Skemp, 1976, s. 14–15). Dette skiljet mellom to forskjellige måtar å sjå matematisk forståing på ligg framleis i botn for korleis matematisk forståing vert definert i dag. For ein meir detaljert diskusjon om utviklinga av omgrepet

«matematisk forståing» sjå Eriksen og Bolme (2021, s. 5–6) og Nosrati og Wæge (2015, s. 4–5).

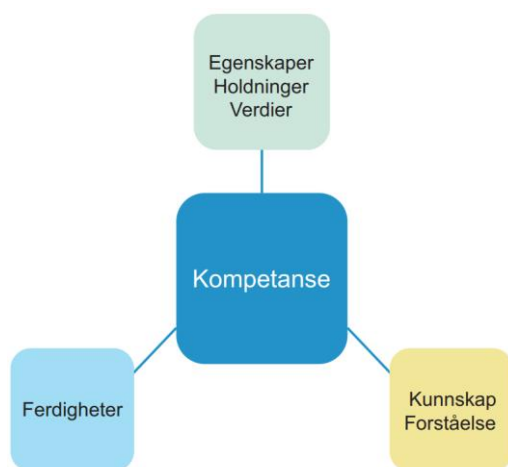
Medan den tradisjonelle matematikkundervisinga hadde fokus på fakta-pugging og innøvde algoritmar, m.a.o. den instrumentelle forståing av matematikken, har etter kvart omgrepet matematisk kompetanse fått fotfeste i utvikling av læreplanar, også i Noreg (Boesen et al., 2014, s. 74).

### 2.2.2 Matematiske kompetansemodellar

Kompetanseomgrepet i matematikken kan være utfordrande å definere (Kilpatrick, 2014, s. 85). I læreplanverket for LK20 (Utdanningsdirektoratet, u.å.-a) finst denne definisjonen av det generelle kompetanseomgrepet: «Kompetanse er å kunne tilegne seg og anvende kunnskaper og ferdigheter til å mestre utfordringer og løse oppgaver i kjente og ukjente sammenhenger og situasjoner. Kompetanse innebærer forståelse og evne til refleksjon og kritisk tenkning». Niss og Højgaard (2011, s. 49) har gitt denne definisjonen på matematisk kompetanse:

«mathematical competence comprises having knowledge of, understanding, doing, using and having an opinion about mathematics and mathematical activity in a variety of contexts where mathematics plays or can play a role». Innholdsmessig er det mykje likt i desse to definisjonane, det er vektlagt kunnskap, forståing, refleksjon og kritisk tenking (opinion), og ulike kontekstar. Dette er nært knytt til relasjonell forståing i matematikkfaget, og Nordbakke (2018, s. 39) skriv om utviklinga av kjerneelementa i Fagfornyinga der ho viser til m.a. Skemp.

Figur 1 Kompetanseomgrepet



Figur 1 Frå NOU 2018: 2 av Kunnskapsdepartementet s. 15 (<https://www.regjeringen.no/no/dokumenter/nou-2018-2/id2588070/>). Offentleg dokument

Her er det eit tydeleg skilje mellom kompetanse og ferdigheiter, der ferdigheiter inngår i kompetanseomgrepet. Dette er ulikt slik Kilpatrick et al. (2001) presenterer sin modell, der ein spesifikk type kompetanse inngår i omgrepet ferdigheit, sjå Tabell 1.

Niss og Højgaard (2011, s. 51) har gjennom det danske KOM-prosjektet også sett saman ein modell som inneheld det dei kallar matematiske kompetansar:

- Mathematical thinking
- Problem tackling
- Modelling  
Reasoning
- Representing
- Symbol and formalism
- Communicating
- Aids and tools

PISA (OECD, 2018) nyttar i sine undersøkingar m.a. eit rammeverk av matematiske kompetansar for å avgjere vanskegraden på matematiske spørsmål, og lista over dei matematiske kompetansane ser slik ut (Turner et al., 2013):

- Problem solving
- Modelling
- Reasoning and argumentation
- Representation
- Using symbolic formal and technical language and operations
- Communication

Kilpatrick et al (2001) har valt ein litt annan modell, og har laga ein kompetansemodell som i større grad syner at dei fem elementa modellen er oppbygd av, er samanvevde. Dette er også presisert for modellen til Niss og Højgaard (2011) og Turner et al. (2013), men ikkje like åpenlyst. Kilpatrick et al. (2001, s. 117) namngir sine fem element slik, med mine nynorskversjonar av Enge og Valenta (2022, s. 624–625) i kolonne to:

Tabell 1 Kilpatrick et al. sin kompetansemodell

Omgrep henta frå Kilpatrick et al. (2001, s. 117)	Mi nynorskskriving av Enge og Valenta (2022, s. 624–625) si omsetjing	Hovedinnhald i elementet, henta frå Kilpatrick et al. (2001, s. 117–133). Stikkordsform.
Adaptive reasoning	resonnerande tankegang	Think logically about the relationships among concepts and situations.
Strategic competence	strategisk tankegang	Formulate mathematical problems, represent them, and solve them. Problem solving.
Conceptual understanding	omgrepsmessig forståing	Grasp of mathematical ideas.
Productive disposition	produktiv tilbøyelegheit	Motivation to learn mathematics.
Procedural fluency	kunnskap om framgangsmåtar	Knowledge of procedures. Knowing when to use them.

Tabell 2: Kjelder står i tabellen.

Her er det litt andre omgrep enn i modellen til Niss og Højgaard (2011), men omgrepa både i den norske omsetjinga og stikkorda frå teksten peiker på fleire like punkt mellom Niss og Højgaard (2011), PISA (Turner et al., 2013) og Kilpatrick et al. (2001, s. 117).

### 2.2.3 Kjerneelementa for matematikk i LK20

I 2017 vart det sett ned ei kjerneelementgruppe med oppdrag om å utvikle kjerneelementa i matematikk (Nordbakke, 2018, s. 35). Nordbakke (2018) påpeiker at Utdanningsdirektoratet ga beskjed til kjerneelementgruppa om at prinsipp som sto i den overordna delen av læreplanen ikkje skulle inngå i kjerneelementa. Dette førte til at kjerneelementgruppa ikkje kunne nytte Kilpatrick et al. (2001) si skildring av matematisk kompetanse, då to av elementa i den skildringa allereie sto i den overordna delen (Nordbakke, 2018, s. 38). Nordbakke (2018) seier ikkje noko om at modellen til Kilpatrick et al. (2001) er ein betre modell, men resultatet vart at modellen til Kilpatrick et al. (2001) ikkje kunne nyttast som mal for kjerneelementa i læreplanen for matematikk.

Kjerneelementa fekk desse namna:

- Utforsking og problemløysing (UP)
- Modellering og anvendingar (MA)
- Resonnering og argumentasjon (RA)
- Representasjon og kommunikasjon (RK)
- Abstraksjon og generalisering (AG)
- Matematiske kunnskapsområde (MK)

Forkortinga i parenteser er laga av oppgåveforfattaren.

I kompetansemodellen til Niss og Højgaard (2011), PISA (Turner et al., 2013) og Kilpatrick et al. (2001, s. 117) er det mange samanfallande omgrep, både mellom modellane og med dei omgrepa som finst i kjerneelementa for matematikk i grunnskulen (Utdanningsdirektoratet, 2019b, s. 2): Problemløysing, modellering, resonnering, representasjonar og kommunikasjon. Nokre omgrep er heilt like, medan andre må tolkast litt. I kjerneelementa er det eit todelt kjerneelement, der den eine delen inneheld omgrepet «abstraksjon». I læreplanen handlar dette omgrepet om ei gravis utvikling av formalisering av «tankar, strategiar og matematisk språk» (Utdanningsdirektoratet, 2019b, s. 2) og utvikling av symbolspråk, og det er naturleg å plassere denne kompetansen i samanheng med «Symbol and formalism». Samtidig skriv Niss og Højgaard (2011, s. 52–53) at kompetansen «Mathematical thinking» heng saman med «abstraksjon og generalisering», så har overlappar kjerneelementa kvarande.

I samanlikninga av kompetansemodellar er det tydeleg samanheng mellom elementa til Niss og Højgaard (2011), PISA (Turner et al., 2013), læreplanen (Utdanningsdirektoratet, 2019b) og til dels også i modellen til Kilpatrick et al. (2001). Likevel nyttar ikkje læreplanen kjelder for opphavet til kjerneelementa. Men sjølv utan kjelder og forklaring, er det tydeleg at innhaldet i kjerneelementa er forankra i internasjonal forskning. I ei stortingsmelding (Meld. St. 28 (2015–2016), s. 34) viser kunnskapsdepartementet til m.a. internasjonal forskning på «big ideas», «core concepts» og «key concepts, og nyttar desse som grunnlaget for kjerneelementa.

### 2.3 Kva er programmering?

Sjølv om læreplanen i matematikk nemner programmering både i kompetansemål og i den digitale grunnleggjande ferdigheita, er det ikkje forklart noko plass i læreplanen for



matematikk kva omgrepet programmering faktisk rommar. Slik er også situasjonen for fleire andre land (Heintz et al., 2015).

Ei mogleg tolking av omgrepet «å programmere» er at det er verbet for å lage eit dataprogram (Sevik, 2016; Van den Brink & Kaufmann, 2022). Eit dataprogram er ei oppskrift, ei algoritme, som datamaskina nyttar til å utføre ei eller fleire oppgåver (t.d. Kongsnes & Wallace, 2020a, s. 184; Mannila, 2017, s. 19–20). Dette støttast også av Skeie (2016) som meiner at ordet programmering har fleire synonym, m.a. koding og utvikling. Mannila (2017, s. 20) skil i større grad mellom «programmering» og «koding», og poengterer at eit program er ei algoritme som er skriven inn i eit programmeringsspråk. Det er dei programmeringsspråk-spesifikke kommandoane som vert kalla kode, og det er desse kommandoane som nyttast i koding (Kaufmann et al., 2022, s. 547; Mannila, 2017, s. 20). Eit programmeringsspråk er eit språk som har eit sett med kommandoar, reglar og grammatikk (syntax) (Mannila, 2017, s. 71) som gjeld for det spesifikke programmeringsspråket.

Læreplanen nemner ikkje programmeringsspråk. Kjerneelementet «utforskning og problemløysing» spesifiserer at elevane skal nytte «algoritmisk tenking» (Utdanningsdirektoratet, 2019b, s. 2), og ei algoritme er i førre avsnitt definert som ei oppskrift. Ei algoritme treng ikkje være digital, medan eit program er ei algoritme som er koda i eit programmeringsspråk (Kaufmann et al., 2022, s. 547).

Ein kan t.d. løyse dette ved å nytte «pseudokode» (Kaufmann et al., 2022, s. 567–577; Mannila, 2017, s. 69). Dette er ei form for algoritme som i utgangspunktet ikkje nyttar standardiserte kommandoar eller syntax, men ofte har eit munnleg preg. Pseudokode treng heller ikkje være digital. Eit eksempel kan være at elevane går saman to og to, og den eine eleven skriv ned eit sett med instruksjonar som den andre skal nytte for å gå frå punkt A til punkt B. Her kan elevane nytte tekst eller symbol etter eige ynskje. Settet med instruksjonar elevparet kjem fram til, er ein algoritme. Eit anna eksempel på pseudokode er at ein elev som skriv ned framgangsmåten for å løyse eit problem.

Det er likevel vanskeleg å kome utanom å nytte eit programmeringsspråk i matematikkfaget. Frå kompetansemåla til 6. trinn står det «bruke variabler, løkker, vilkår og funksjoner i programmering til å utforske geometriske figurer og mønstre» (Utdanningsdirektoratet, 2019b, s. 10). I kompetansemåla for 8. trinn står det «utforske hvordan algoritmer kan skapes, testes og forbedres ved hjelp av programmering» (Utdanningsdirektoratet, 2019b, s. 12) og det er lett å sjå koplinga til eit programmeringsspråk her.

Det kan vere hensiktsmessig å dele inn programmeringsspråka i to grupper: blokkbasert og tekstbasert (Mannila, 2017, s. 71–72). Blokkbaserte programmeringsspråk har visuelle blokker, nesten som bitar i eit puslespel, som passar saman. Dette gjer at det vert mindre skrivefeil i koden (Mannila, 2017). Eit tekstbasert programmeringsspråk består av tekst og symbol, og gir større rom for feilstavingar av ord og kommandoar, og syntaxfeil. Oppgåvene i utvalet som er presentert med kode, og eksempla som er nytta i kapitla om sannsynsrekning, nyttar utelukkande tekstbaserte programmeringsspråk.

## 2.4 Programmering og kompetansar

Mange land har dei siste åra knytt arbeid med programmering til skulen, anten som eigne fag eller som ein del av undervisinga (Dolonen et al., 2019; Sevik, 2016; Waite, 2018). I hovudsak er det tre forskjellige måtar å innføre programmering i skulen på (Sevik, 2016, s. 7):

- Eit eige programmeringsfag
- Nytte programmering som ein del av kvart enkelt fag
- Sjå på programmering som ein kompetanseparaply for fleire fag

Skulen i Noreg har dei siste åra hatt moglegheit for eit valfag i programmering (Utdanningsdirektoratet, u.å.-d), men har ikkje eit eige obligatorisk programmeringsfag slik nokre andre land har. I læreplanen er programmering lagt inn under «digitale ferdigheiter» (Utdanningsdirektoratet, 2019b, s. 5) og er ikkje direkte representert i kompetansane skildra i kjerneelementa. Dermed er ikkje programmering rekna som ein kompetanseparaply, men som ei grunnleggjande, fagovergripande ferdigheit som i tillegg grip inn i nokre enkeltfag. Sjølv om alle fag i skulen skal arbeide med dei grunnleggjande ferdigheitene, kvilar det eit spesielt ansvar på desse enkeltfaga.

Å nytte programmering i undervisinga er ikkje ein ny idé, og tankegodset låg på teiknebrettet allereie på 1980 og -90-talet (Dolonen et al., 2019). I denne perioden vart det forska på programmering som verktøy for kognitiv trening, men det var ikkje funne resultat som viste at programmeringa hadde nokon effekt (Dolonen et al., 2019, s. 6). Det vart også forska på om programmering som ein problemløysingsøving med overføringsverdi til andre situasjonar og fag, men heller ikkje her vart det funne ein samanheng (Dolonen et al., 2019, s. 6).

At programmering no har fått ein renessanse kan i stor grad tilskrivas omgrepet «Kompetansar for det 21. århundret» («21. century skills») og behovet for algoritmisk tenking, i tillegg til ei utvikling i verktøya som nyttast til programmering (Dolonen et al.,

2019, s. 6; Kaufmann et al., 2022, s. 545–546; Sevik, 2016, s. 10–11). Tankar rundt programmering i skulen har vorte relansert under omgrepet «computational thinking» (Dolonen et al., 2019, s. 24). På norsk nyttast omsetjinga «algoritmisk tankegang» (Sevik, 2016, s. 13). Dette omgrepet ikkje har ei felles innhaldsforståing (Brennan & Resnick, 2012, s. 2; Dolonen et al., 2019, s. 24; Jacob & Warschauer, 2018, s. 1), og t.d. Utdanningsdirektoratet omtalar algoritmisk tenking som ein problemløysingsmetode (2019c, s. 1). I læreplanen står algoritmisk tenking under kjerneelementet *utforsking og problemløysing* (UP) (Utdanningsdirektoratet, 2019b, s. 2), og her det står at algoritmisk tenking «...inneber å bryte ned eit problem i delproblem...». Sevik (2016, s. 13) utvidar omgrepet ein del, men det er tydeleg programmering er eit middel og ikkje eit mål i den algoritmiske tankegangen. Dolonen et al. (2019, s. 24) foreslår derfor omgrepet «digital tenking» som ei breiare norsk omsetjing der også bruk av formålstenlege verktøy skal inkluderast. Her går Dolonen et al. (2019) vekk frå å sjå på computational thinking som eit omgrep som inneheld utelukkande kognitive ferdigheiter og arbeidsmåtar, og dermed element som allereie er skildra gjennom kjerneelementa (sjå t.d. punkta til Utdanningsdirektoratet (2019c)).

Binkley et al. (2012, s. 36) har laga eit rammeverk for «Kompetansar for det 21. århundret» og har delt 10 ferdigheiter inn i 4 grupper, og to av desse ferdigheitene er *utforsking* og *problemløysing*. Fokuset på utforsking og problemløysing i læreplanen vert framheva av prosessen med å lage kjerneelementa (Nordbakke, 2018), og sjølv om læreplanen ikkje opent vektlegg eit kjerneelement meir enn eit anna i matematikken, dukkar (spesielt) problemløysing opp fleire plasser som eit relevant omgrep knytt til programmering i matematikken (Kaufmann et al., 2022; Sevik, 2016).

## 3 Metode

### 3.1 Metoden

For å svare på problemstillinga vil studien nytte fleire ledd. Først vil det bli utvikla kriteria for kva oppgåver som skal inngå i studien. Fokuset er på programmeringsoppgåver i matematikk, og oppgåvene i dei aktuelle kapitla i lærebøkene vil filtrerast utifrå dette (sjå kap. 3.5). I neste omgang vil dei oppgåvene som omhandlar programmering, verte analyserte basert på rammeverket i kap. 3.4. Dette rammeverket er todelt: ein del tek utgangspunkt i, og er ein modifisert versjon av, studien til S. Eriksen og Vos (2022), og ser på korleis oppgåvene oppfyller kjerneelementa i læreplanen for matematikk. Den andre delen av rammeverket vurderer om oppgåvene er eigna til å gje elevane øving i programmering og er ein del av eit rammeverk frå Brennan og Resnick (2012). Dei to delane i rammeverket er sidestilte, og inngår i same analyse.

### 3.2 Forskingsdesign, forskingsmetode og metodeval

Studien sjå på eit utval oppgåver frå eit utval lærebøker. Som det kjem fram i kapittel 1.3 er studien avgrensa til å sjå på oppgåver henta frå kapittel om sannsyn for ungdomstrinnet. Dette er ei avgrensing som reduserer utvalet på oppgåver betydeleg. Likevel vert oppgåvemengda stor nok til at det gir mening å presentere resultat og samanlikningar statistisk. Basert på problemstillinga og avgrensinga i hvv kapittel 1.2 og 1.3, og utvalet skissert i kapittel 3.6, blir denne studien klassifiserast som ein Liten N-studie (Postholm & Jacobsen, 2018, s. 74–75). Studien ser på ein liten mengde matematikkoppgåver over fleire lærebøker, altså på tvers av ulike kontekster.

Metoden som vert nytta vil ta føre seg større mengder informasjon og analysere desse statistisk. Statistiske analyser av større mengder informasjon vil bevege metoden vekk frå vurderingar og tolkingar, og erstattast desse med målingar. Det vil føre til ein meir overførbar og generaliserbar kunnskap, og slike metodar har ei *positivistisk tilnærming* (Dalland, 2012, s. 52–55; Postholm & Jacobsen, 2018, s. 99; Thrane, 2018).

Det er i hovudsak to hovudtypar av metodar som vert nytta i forskingsaugemed: kvantitativ og kvalitativ (Dalland, 2012; Grønmo, 2016; Postholm & Jacobsen, 2018). I ein debatt om korleis hovudtypane skal tolkast, argumenterer Grønmo (2012, s. 1–2) for at *data* karakteriserast som kvantitative eller kvalitative, i staden for at metoden er karakterisert som

kvantitativ eller kvalitativ. Sjøvoll (2018, s. 23–24) operer også med kvantitative og kvalitative *data*.

Sett i lys av argumentet til Grønmo, vil det passe å belyse omgrepet kvantitative og kvalitative *tal* (Postholm & Jacobsen, 2018, s. 100–101). Her forklarar Postholm og Jacobsen omgrepa med at nokre tal har ei meining i seg sjølv, t.d. skonommer og alder, medan nokre tal er symbol som representerer noko anna informasjon, slik det vert gjort i denne oppgåva (sjå kapittel 3.4). Desse måtane å skilje tal på vert kalla hvv. kvantitative tal og kvalitative tal.

Men der Postholm og Jacobsen to-delar omgrepet, nyttar Grønmo ein figur for å illustrere ei meir nyansert tilnærming, der data kan plasserast på ei linje og vere ein plass *mellom* kvantitative og kvalitative data (sjå Figur 2). Her vert omgrepet «ikke-metriske data uttrykt ved tall» (Grønmo, 2016, s. 137–138) forklart som «kategoriske data». Dette kan m.a. være ein variabel på ordinalnivå der eit tal representerer meir informasjon, t.d. inntekt (lav[1], middels[2] eller høg[3]). I figuren til Grønmo vert slike data plassert nærmare kvantitative data enn kvalitative data, og dette er den typen data denne studien nyttar (sjå kapittel 3.4). Grønmo (2016, s. 137–138) forklarar også at «metriske data» er data som t.d. alder (målt i antal år) og metriske data kan, som ikkje-metriske data uttrykt ved tal, analyserast ved hjelp av statistiske metodar. «Data uttrykt ved andre mengdetermer» referer derimot til kvantitative forhold som stor-større-størst og kan ikkje analyserast statistisk, og det kan heller ikkje kategorien «data uttrykt ved tekst» (Grønmo, 2016, s. 137–138).

Figur 2 Oversikt over ulike datatypar

Kvantitative data		Kvalitative data	
Metriske data	Ikkje-metriske data uttrykt ved tall	Data uttrykt ved andre mengdetermer	Data uttrykt ved tekst

Figur 2 Laga av forfattaren som ei attgjeving av figuren frå *Samfunnsvitenskapelige metoder* (s.24) av S. Grønmo, 2016, Fagbokforlaget.

Som det kjem fram av kapittel 3.4 vil studien operasjonalisere omgrepa i dei matematiske kjerneelementa, og standardisere informasjonen om oppgåvene for å kunne samanlikne utfallet av datainnsamlinga. Ein slik forskingsmetode vert kalla kvantitativ (Grønmo, 2016; Postholm & Jacobsen, 2018, s. 166–170).

### 3.2.1 Kvantitativ innhaldsanalyse

I denne masteroppgåva vert deler av innhaldet i kvar lærebok vurdert. Ei slik metode kan kallast *kvantitativ innhaldsanalyse* (Bratberg, 2021, s. 99–125; Grønmo, 2016, s. 143). Innhaldsanalyse «... går også ut på å systematisere de deler av innholdet i ulike dokumenter som er relevante for gitte problemstillinger» (Grønmo, 2016, s. 143) og omgrepet viser til «... enhver analyse som systematisk sammenfatter tekstinnhold» (Bratberg, 2021, s. 101). Både Grønmo og Bratberg skriv at ei innhaldsanalyse kan være både kvalitativ og kvantitativ. Ei god skildring på korleis denne studien genererer data er å finne i Bratberg si tolking av kvantitativ innhaldsanalyse

I dette kapittelet skal vi imidlertid avgrense begrepet til *teknikker for sammenfatting og beskrivelse av innhold i tekst ved hjelp av kvantitative mål*. Innhaldsanalyse måler forekomsten av bestemte elementer i tekst gjennom å tallfeste, og disse skårene kan så legges til grunn for statistisk analyse. (Bratberg, 2021, s. 101)

Tekstmaterialet, eller spesifikt matematikkoppgåvene i denne studien, kan dermed nyttast i deskriptiv statistikk, slik at mønster i tekstmaterialet/oppgåvene kan påvisast.

Omgrepet «innhaldsanalyse» kan romme fleire tolkingar, og kan generelt sett nyttast for å skildre prosessen med å tolke, oppsummere og beskrive tydinga av innhaldet i eit datamateriale (Cohen et al., 2018). I den kvantitative innhaldsanalyse kan dette skje via eit kodeskjema. Kodeskjemaet består av variablar, og desse variablane har sitt utspring i problemstillinga (Bratberg, 2021, s. 106; Grønmo, 2016, s. 213–215). Generelt forklart er variablane det som blir målt i innhaldsanalyse. Variablane for denne studien vert spesifisert i kapittel 3.4, og består av operasjonaliserte kjerneelement frå matematikkfaget. Eit eksempel er variabelen *matematisere – lage ein modell* som er ein av fleire variablar frå kjerneelementet *modellering og anvendingar* (Utdanningsdirektoratet, 2019b). Når kodeskjemaet er ferdig, er det berre variablar i kodeskjemaet som skal vurderast, det er ingen opne variablar kodaren (den som fyller ut kodeskjemaet) kan nytte. For kvar variabel vert det fastsett kategoriar kodaren kan nytte for å skåre variabelen.

Saman med eit kodeskjema må/bør det følge med ein kodeinstruks (Grønmo, 2016, s. 213–219). Kodeinstruksen inneheld informasjon til kodar(ane) om korleis kodeskjemaet skal fyllast ut og korleis kodinga skal utførast, i tillegg til ei skildring av kvar variabel og kategori. Kodeinstruksen for denne oppgåva kan lesast i kap. 3.4.3.

Kombinasjonen av kodeskjema med faste variablar, kodeinstruks og predefinerte kategoriar for kvar variabel reduserer rommet for skjønn, og gjer at kvantitativ innhaldsanalyse ofte

skårar høgt på reliabilitet (Bratberg, 2021, s. 103,119-123; Grønmo, 2016, s. 220), sjå kapittel 3.7 i oppgåva for detaljar spesifikt for studien.

Eit poeng i den kvantitative innhaldsanalyse er at kodeskjema og kategoriane er ferdige før den *systematiske* informasjonsinnhentinga (Grønmo, 2016, s. 143). I dette poenget til Grønmo ligg det ein tanke om kronologi. Avhengig av den vitskaplege ståstaden til forskaren vil rekkjefølgja på nokre hovudelement i forskingsdesignet variere. I ein klassisk positivistisk tilnærming vil forskaren sjå på teori først, deretter utvikle ei hypotese for så å samle empiri for å sjå om hypotesen stemmer med verkelegheita. I eit konstruktivistisk perspektiv vil forskaren snu på modellen og først samle inn empiri utan å ha ein hypotese å bevise, deretter sortere og systematisere observasjonane og nytte dette som grunnlag for ei hypotese. Desse metodane kallast hvv. deduktiv og induktiv tilnærming (Kleven & Hjordemaal, 2018, s. 23–25; Kvale & Brinkmann, 2015, s. 224–225; Postholm & Jacobsen, 2018, s. 101–104; Tjora, 2018, s. 14–15). Postholm og Jacobsen ser dei tilnærmingane som er reint deduktive eller induktive som ytterpunkt på ein skala og meiner slike ytterpunkt er umogleg å gjennomføre. Ein forskar kan ikkje forhalda seg utelukkande til teori utan å ha observert noko som lagar grunnlag for teorien, og motsett er det ikkje mogleg å gå ut og observere verda utan å ha noko fram for fordømmar rundt det ein skal observere (Postholm & Jacobsen, 2018, s. 102).

Kvantitativ innhaldsanalyse er i analytisk forstand deduktiv (Bratberg, 2021, s. 119). Ramma for analysen er sett av kodeskjemaet, og kodeskjemaet er utarbeidd på grunnlag av teori. Samtidig er det tydeleg at det er involvert prosessar som ikkje er reint deduktive. Nokre innhaldsanalysar kan t.d. samanlikne bruk av like ord i fleire tekstar. Før den endelege lista over variablar er ferdig i slike studiar, er det anbefalt å gjere ei vurderinga av variablane «...i lys av de utvalgte tekstene» (Grønmo, 2016, s. 215). Det er i slike tilfelle, som Grønmo presiserer, lite poeng i å ha eit kodeskjema som inneheld variablar som ikkje finst i tekstane. Dersom ein studie derimot skal vurdere frekvensen av ord i ein tekst, er det ikkje naudsynt å «tekst-vaske» lista med variablar. Kodeinstruksjonen kan godt være eit resultat av ein pretest (Grønmo, 2016, s. 219) og er heller ikkje eit direkte resultat av rein teoretisk tilnærming.

Kvantitativ innhaldsanalyse nyttar seg av eitt kodeskjema pr. analyseining (sampling unit) (Bratberg, 2021, s. 105; Grønmo, 2016, s. 219). Ei *analyseining* er den teksten som vert analysert for å skåre variablane. I denne studien vil det være ei matematikkoppgåve, fordi kvar matematikkoppgåve skal vurderast mot kvar variabel i kodeskjemaet. Kvar analyseining er vidare delt opp ei *kodeining* (recording unit) og *konteksteining* (Bratberg,

2021, s. 105; Grønmo, 2016, s. 218). Kodeeininga er den delen av analyseeininga som skal kodast og kan ha mange former, som t.d. syntax (ord eller setningar) eller fysiske skilje (sider i ei avis og liknande).

Nokre studiar har analyseeiningar som er så små at dei er både analyseeining og kodeeining (Grønmo, 2016, s. 218). Så små analyseeiningar som matematikkoppgåver, kan være ei konteksteining i seg sjølv, men matematikkoppgåver står skjeldan åleine i eit vakuum og kan i aller høgste grad tolkast i lys av andre kontekstar. Ei preanalyse av utvalet lærebøker (sjå kapittel 3.6 for utval av lærebøker) synte at nokre forlag deler opp læreverket i ei teori-bok og ei rein oppgåvebok, medan andre ikkje gjer det. Dette fører til at omgrepet konteksteining i studien går utover definisjonen til Grønmo og er *utanfor* analyseeininga, i staden for å være ein del av den. Grunnen er at ei matematikkoppgåve kan i teoriboka sjåast i lys av teori i forkant av oppgåva, eller tidlegare i boka. I oppgåvebøker er oppgåvene stort sett grupperte i tema som også gir kontekst til oppgåvene.

I neste kapittel vil ein gjennomgang av studien sitt overordna design gi eit klarare bilde av metoden som er nytta i studien.

### 3.3 Metoden sitt overordna design

Dette kapittelet vil gje eit bilde av korleis metoden vart designa for å fylgje rammene til kvantitativ innhaldsanalyse og gje eit best mogleg svar på problemstillinga. Kapittelet er skreve semi-kronologisk då nokre prosessar vart gjennomførte samtidig.

Metoden starta med utvikling av rammeverket for analysing av programmeringsoppgåver (kap. 3.4). I forkant av dette arbeidet vart det gjennomført ein liten preanalyse av lærebokstoffet som skal nyttast i studien. I denne utviklinga av rammeverket vart det laga eit kodeskjema (kap. 3.4) og ein kodeinstruks (kap. 3.4.3). Begge desse vart piloterte og deretter justerte etter tilbakemeldingar. Når kodeskjema og kodeinstruksen vart sendt ut, vart det også lagt ved nokre oppgåver som skulle vurderast. I forkant av denne utsendinga vart det laga kriterium for kva oppgåver som inngår i omgrepet «programmeringsoppgåver» (kap. 3.5), og som dermed skulle være med i studien. Desse filtrerte oppgåvene vert kalla «utvalet» i studien.

Etter justeringane av kodeskjemaet og kodeinstruksen, vart rammeverket nytta til å analysere utvalet av oppgåver. Kodinga vart gjort av studieforfattaren. Data frå analysen vart deretter



behandla gjennom ein dataanalysemodell og resultatane frå denne dataanalysen ga grunnlaget for presentasjonen av funn og drøfting av desse (hvv. kap. 0 og 0).

### 3.4 Utvikling av analytisk rammeverk

Rammeverket studien nyttar for analyse av oppgåvene, tek utgangspunkt i ein studie av S. Eriksen og Vos (2022). Studien til S. Eriksen og Vos ser på Utdanningsdirektoratet sine eksempeloppgåver for matematikk P-faget (Utdanningsdirektoratet, 2019a) på vidaregåande skule i forbindelse med innføringa av LK20, og vurderer desse opp mot dei matematiske kjerneelementa i LK20. Desse eksempeloppgåvene inneheld oppgåver frå fleire matematiske tema, og S. Eriksen og Vos ser ikkje berre på programmeringsoppgåver slik denne studien gjer. Vidare har dei operasjonalisert kvart kjerneelement og definert kategoriar.

Merk at artikkelen til S. Eriksen og Vos ikkje nyttar heilt det same omgrepsapparatet som vert nytta i denne oppgåva. Der S. Eriksen og Vos nyttar omgrepa «kategoriar» og «kodar», nyttar denne studien hvv. «variablar» og «kategoriar», i tråd med modellen for innhaldsanalyse som vart presentert i kapittel 3.2.1.

Kategoriane til S. Eriksen og Vos skal seie noko om kor vidt intensjonen med kvart kjerneelementet er oppnådd i kvar oppgåve. Dei nyttar eit eksempel frå kjerneelementet representasjon og kommunikasjon, med definerte kategoriar i utheva tekst:

Representasjonar i matematikk P er måter å uttrykke matematiske begreper, sammenhenger og problemar på. Representasjonar kan vere konkrete, kontekstuelle, visuelle, verbale og symbolske. Kommunikasjon i matematikk P handler om at elevene **bruker matematisk språk** i samtaler, argumentasjon og resonnerer. Elevene må få mulighet til å **bruke matematiske representasjonar** i ulike sammenhenger gjennom egne erfaringar og matematiske samtaler. Elevene må få mulighet til å forklare og **begrunne val av representasjonsform**. Elevene må kunne oversette mellom matematiske representasjonar og dagligspråket og **veksle mellom ulike representasjonar**. (S. Eriksen & Vos, 2022, s. 46)

Utvalet kategoriar er grunna i måla for kjerneelementa: å skildre kva elevar må vere i stand til å *gjere* (S. Eriksen & Vos, 2022, s. 45). Kategoriane vart koda med 0 poeng dersom eleven ikkje hadde moglegheit til å vise dette målet i oppgåva, 1 poeng dersom oppgåva ga moglegheit til å vise målet og 2 poeng dersom oppgåva eksplisitt hadde målet som krav i løysinga av oppgåva (S. Eriksen & Vos, 2022, s. 46). På same måte vil denne studien nytte

dei operasjonalisere kjerneelementa som variablar og nytte same kategoriar,  $\{0, 1, 2\}$ , når variablane vert skåra.

I tillegg til at denne studien tek utgangspunkt i analysemodellen til S. Eriksen og Vos (2022), vil rammeverket i kap. 3.4.2 utvidast med eit mål på korleis kvar oppgåve skårar på variablar om programmering, også her innanfor kategoriskalaen  $\{0, 1, 2\}$ . Dette vert gjort for å lage eit utvida analyseverktøy som kan nyttast av dei som ynskjer å vurdere både kor godt programmeringsoppgåvene dekkjer måla i kjerneelementa og kor godt dei dekkjer mål om programmering.

### 3.4.1 Variablar knytt til kjerneelementa

Dette delkapittelet vil skildre prosessen med å operasjonalisere kjerneelementa i matematikk for ungdomstrinnet (Utdanningsdirektoratet, 2019b, s. 2). Studien nyttar S. Eriksen og Vos (2022) sine operasjonaliseringar, med unntak som er skildra lenger ned i delkappitlet. Det er nokre mindre forskjellar i ordlyden mellom matematikk P og matematikk 1-10 for kjerneelementet argumentasjon og resonnering. 1-10 skildrar ei utvikling mot formalisering av språket og resonnement, medan P vektlegg bruken av formelle symbolspråk og resonnement. I tillegg er det i same kjerneelement litt forskjellar i ordlyden rundt utforsking med formål om å uttrykkje resultat og samanhengar med algebra og formålstenlege representasjonar. Dette inneber litt omskriving av S. Eriksen og Vos sine kategoriar.

Tabell 2 Fullstendig oversikt over variablar sortert etter kjerneelement

Kjerneelement	#	Variabel
Utforskning og Problemløysing (UP)	1	Avdekke mønster og samanhengar
	2	Dele eit problem opp i delproblem
	3	Vurdere bruk av digitale hjelpemiddel eller ikkje
	4	Utvikle ei metode i ein ukjent situasjon
Modellering og anvendingar (MA)	1	Knyte til dagleglivet, arbeidslivet eller samfunnet
	2	Matematisere – lage ein modell
	3	Kritisk vurdere ein modell
	4	Bruke ein modell i ein annan kontekst.
Resonnering og argumentasjon (RA)	1	Utvikle ei tankerekkje
	2	Grunngje eller bevise
Resonnering og argumentasjon (RK)	1	Bruke matematisk språk
	2	Bruke matematiske representasjonar
	3	Grunngje valet av representasjonsform
	4	Byte mellom representasjonar
Abstraksjon og generalisering (AG)	1	Nytte formelt symbolspråk eller formelle resonnement
	2	Oppdage matematiske samanhengar og generalisere
	3	Utforske tal, utrekningar og figurar. Formaliser med algebra og formålstenlege representasjonar.

Oversikta er sortert etter kjerneelement og nummerert. Seinare i oppgåva er variablane refererte til i formatet [Forkorting av kjerneelement Nummer], t.d. MA 3 for den tredje variabelen i kjerneelementet *modellering og anvendingar*.

Denne studien vil, ulikt studien til S. Eriksen og Vos (2022), sjå vekk frå kjerneelementet *matematiske kunnskapsområde* (MK). I 2018 ga Kunnskapsdepartementet (2018b) føringar for utforming av læreplanar for fag i LK20 m.m., og her står det at dei fem første kjerneelementa slik dei er lista opp i læreplanen (Utdanningsdirektoratet, 2019b, s. 2–4) skildrar måtar å arbeide og tenkje på, og metodar i matematikken. Kjerneelementet MK skal skildre sentrale kunnskapsområde i matematikken, og elevane skal få kunnskap om desse sentrale områda gjennom dei andre fem kjerneelementa (Kunnskapsdepartementet, 2018b, s. 15).

Denne formuleringa kan tolkast slik at innhaldet i MK allereie er dekt inn i dei fem andre kjerneelementa, og at ei sidestilling av MK saman med dei andre kjerneelementa vil gje eit ukorrekt bilde av oppgåvene. Dersom ei oppgåve vert kategorisert med verdien 2 i variabelen [MA:3] så betyr det også at oppgåva treff inn i kjerneelementet MK. Ei slik poeng-setting kan føre til at oppgåvene skårar høgare ut på MK enn på dei andre kjerneelementa, og dette kan vere ein av grunnane til at MK skårar så høgt hjå S. Eriksen og Vos (2022, s. 48).

Kjerneelementet MK i matematikk 1-10 (Utdanningsdirektoratet, 2019b, s. 3–4) ber preg av

oppsummering og gir eit svært generelt bilete av kva matematikkfaget for 1-10 trinn skal innehalde i dei forskjellige kunnskapsområda. Kjerneelementet MK er annleis formulert for matematikk P (Utdanningsdirektoratet, 2019a), noko som kan vere grunnen til at S. Eriksen og Vos valte å ta MK med i sin studie.

### 3.4.2 Variablar knytt til programmering

Studien ser på programmeringsoppgåver og det ligg til grunn at alle oppgåvene som vert analyserte på handlar om programmering i ei eller annan form, sjå kap. 3.5 for ein meir detaljert gjennomgang av utval av oppgåver. Difor vil kodeskjemaet ikkje innehalde ein variabel som sjekkar om ei oppgåve er ei programmeringsoppgåve. I staden vert det i dette kapitlet lista opp grunnjevne variablar som er med på å avdekke om programmeringsoppgåvene oppfyller behova elevane har for læring i programmering. Ei avgrensing som er gjort i studien er at oppgåver som kan eller bør løysast i t.d. Excel eller i ein grafteiknar ikkje vert omfatta av definisjonen programmering (sjå kap. 3.4.2) og dette vert spegla i variablar knytt til programmering.

I kap. 2.3 vert omgrepet programmering drøfta, og kapitlet startar med å skildre at læreplanen i matematikk ikkje har eit klart læringsmål for programmeringsundervisinga. Programmering er nemnt i kompetansemåla (sjå kap. 1.1.1) og i dei digitale ferdigheitene. Likevel er det utydeleg kva elevane faktisk skal lære. Nordbakke (2018, s. 38) skriv om stor motstand mot programmering i matematikk, og at dette var medverkande til at programmering vart inkludert i læreplanen som ein arbeidsmåte for å lære matematikk, og ikkje som ei rein teknisk øving. Målet med dette delkapitlet er å identifisere kva kompetanse elevane må ha om programmering, og å spesifisere variablar som gjer at studien kan måle desse kompetansane i programmeringsoppgåvene.

Kap. 2.3 tek også føre seg programmering og koding, og litt om korleis omgrepa kan tolkast og forskjellane mellom omgrepa. I denne oppgåva er det sett ei avgrensing ved at utvalet oppgåver ikkje inkluderer oppgåver som kan løysast i t.d. eit rekneark. Rekneark-programma har vorte svært avanserte og kan t.d. programmerast med kontrollstrukturar, men i avsnittet om den digitale grunnleggande ferdigheita står det «Digitale ferdigheiter i matematikk inneber å kunne bruke grafteiknar, rekneark, CAS, dynamisk geometriprogram og programmering...» (Utdanningsdirektoratet, 2019b, s. 5 mi utheving). Ei mogleg tolking av dette, og som studien støttar seg på, er tolkinga om at programmering er sidestilt og forskjellig frå det å nytte grafteiknar, rekneark, CAS og dynamisk geometriprogram.

Brennan og Resnick (2012) laga eit rammeverk for å kople programmering saman med computational thinking, og utvida omgrepet til å dekkje tre undergrupper: computational concepts, computational practices og computational perspectives. Computational concepts er ifølge Brennan og Resnick (2012, s. 2–6) dei grunnleggjande programmeringskonseptane: sekvensar/framgangsmåtar, løkker, handlingar, parallellisme (eller multitasking), kontrollstrukturar, operatorar og data, sjå Brennan og Resnick for ein detaljert gjennomgang. Desse konseptane vert også utdjupa litt i kap. 3.4.3. Programmeringskonseptane vert ikkje fanga opp i kjerneelementa, men er likevel ein grunnleggjande kunnskap eleven må ha for å kunne programmere i eit programmeringsspråk, og delvis for pseudoprogrammering. Kvar av desse programmeringskonseptane vert vurdert individuelt, og vert til slutt samla i ein variabel som heiter «programmeringskonsept». Variabelen «programmeringskonsept» vert sett til gjennomsnittet av dei individuelle programmeringskonseptane.

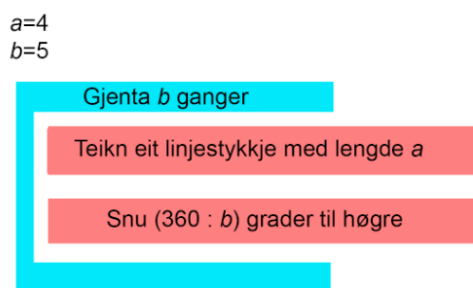
Computational practices er elementa i konstruksjonsprosessen til prosjektet som dei unge programmerarane held på med (Brennan & Resnick, 2012, s. 6–9). Elementa frå konstruksjonsprosessen har mykje til felles med kjerneelementa i matematikk, og med nøkkelomgrep og arbeidsmåtar i algoritmisk tenking (Utdanningsdirektoratet, 2019c): «being incremental and iterative» inngår i arbeidsmåten «fikling» og nøkkelomgrepet «evaluering», «testing and debugging» inngår i arbeidsmåten «feilsøke», «reusing and remixing» inngår til dels i arbeidsmåtene «skape» og «samarbeide», og «abstracting and modularizing» inngår i nøkkelomgrepa «abstraksjon» og «dekomponering». Når desse elementa er så like vert dei i også fanga opp av kjerneelementa, og det er ikkje naudsynt å vurdere desse i tillegg til kjerneelementa.

Computational perspectives handlar om eit skifte i perspektivet på teknologi (Brennan & Resnick, 2012, s. 10–11). Brennan og Resnick identifiserer tre forskjellige perspektiv, og dei to første er knytt spesifikt til programmeringsspråket og -plattforma Scratch (u.å.), og effekten av deling av kode og kommunikasjon innanfor den plattformen. Desse to perspektiva vil bli sett vekk ifrå i studien. Det tredje perspektivet «questioning», her omsett til «undring», handlar om korleis programmerarane undrar seg over korleis objekt som t.d. brødristarar, er programmerte. Dette perspektivet kjem i hovudsak til uttrykk gjennom at dei unge programmerarane føler seg trygge på teknologien som er rundt oss, og ser moglegheiter for forbetringar (Brennan & Resnick, 2012, s. 10–11). Forskinga til Brennan og Resnick tek utgangspunkt i unge programmerarar som har nok kunnskap i programmering til å utvikle eigne komplekse program som hobby, og dette er i all hovudsak eit stykke forbi den

kunnskapen det er lagt opp til at matematikkfaget 1-10 skal dekkje. Den norske skulen har valfaget «Programmering» (Utdanningsdirektoratet, u.å.-d) der alle desse tre perspektiva kan utviklast i større grad, og dette siste perspektivet vil heller ikkje bli ein del av studien. Perspektivet undring kan i tillegg knytast til forskjellige element innan algoritmisk tenking, m.a. nøkkelomgrepa logikk og mønstre, men omfattar likevel noko meir enn det som ligg desse omgrepa.

Noko som kan kaste lys over kva elevane treng å lære seg, er eksamen. Det er ikkje uvanleg å trene på det ei vurdering skal vurdere, og viktige vurderingar kan forsterke denne effekten (Niss & Jensen, 2002; Schoenfeld, 2007, s. 12). Eksamen er dessutan ein effektiv måte å innføre endringar i skulen på (Fried & Amit, 2015; Morgan & Sfard, 2016, s. 92; Wall, 2006). Ved å sjå på Utdanningsdirektoratet. sin sentralgitte eksamen for matematikk i 2022 kan det kaste meir lys over kva programmeringundervisinga i skulen bør innehalde. Eksamen for matematikk 10. trinn (sjå søkemotor frå Utdanningsdirektoratet (u.å.-c)) inneheld ingen programmeringsoppgåver knytt til eit spesifikt programmeringsspråk, medan eksamen for matematikk 1P inneheld ei oppgåve med syntax og språk som er identisk med programmeringsspråket Python.

Figur 3 Oppgåve frå eksamen 10. trinn



Figur laga av forfattaren som ei attgjeving av oppgåve frå eksamen 10. trinn vår 2022

Figur 4 Oppgåve frå eksamen 1P

```
1 startverdi=2000
2 verdi=startverdi
3 vekstfaktor=1.05
4 ar=0
5
6 while verdi < startverdi * 2:
7     verdi=verdi*vekstfaktor
8     ar=ar+1
9     print(verdi)
10    print(ar)
11
```

Figur laga av forfattaren som ei attgjeving av oppgåve frå eksamen i 1P. trinn vår 2022

I Figur 3 får eleven ei oppgåve som er laga i pseudokode, formatert som blokkprogrammering (sjå kap. 2.3 for detaljar om desse omgrepa). Når  $a$  og  $b$  er gitt, kan desse fyllast inn og oppgåva kan løysast. Oppgåva krev m.a. at eleven kan dekomponere.

Figur 4 syner ei oppgåve der eleven må ha kunnskap om kommandoen «while», saman med kva argument denne kommandoen tek. Denne kommandoen er spesifikk, men ikkje unik, for programmeringsspråket Python. Dette betyr at på vegen mot eksamen i matematikk 1P må elevane lære seg eit tekstbasert programmeringsspråk, anten Python eller eit anna tekstbasert programmeringsspråk som har syntax og språk likt som Python. Dette harmonerer med

resultatet av preanalysen til studien der Python viser seg å være det framherskande tekstbaserte programmeringsspråket (t.d. Kongsnes & Wallace, 2020a, s. 184). Med behov om kompetanse både innan pseudoprogrammering og eit programmeringsspråk vil studien vurdere om ei programmeringsoppgåve inneheld desse måtane å programmere på. Basert på informasjonen frå eksamen vil studien vurdere om eleven kan lese/tolke kode, og om eleven skal kode sjølv.

Som i Tabell 2 vil variablane for programmering (PR) i Tabell 3 få ei kode for seinare referanse. Desse variablane startar med PR og får kvar sitt nummer. Tabell 3 listar opp variablar henta frå Brennan og Resnick (2012) si oversikt over programmeringskonsept og oversett til norsk. I tillegg inneheld tabellen to variablar for å sjå om oppgåva krev eller gir moglegheit til å anten lese/tolke koding, eller utføre koding.

*Tabell 3 Oversikt over variablar innanfor programmering*

Programmering	#	Variabel
PR	1	Lese/tolke koding (både pseudo og programmeringsspråk)
PR	2	Utføre koding (både pseudo og programmeringsspråk)
PR	3	Programmeringskonsept
PR	3.1	Sekvensar/framgangsmåtar
PR	3.2	Løkker
PR	3.3	Handlingar
PR	3.4	Parallellisme (eller multitasking)
PR	3.5	Kontrollstrukturar
PR	3.6	Operatorar
PR	3.7	Data

Den samla oversikta over variablar for studien kan finnast i Vedlegg 1: Kodeskjema

### 3.4.3 Operasjonalisering av variablar (kodeinstruks)

Målet med å presentere kodeinstruksen med detaljar og refleksjonar, er å auka reliabiliteten til resultatata. Ei utfordring med rammeverket som vert nytta, er at S. Eriksen og Vos ikkje

forklarer i detalj kva dei la i operasjonaliseringane av kjerneelementa når dei gjennomførte analysen. Unntaket er ein gjennomgang av kjerneelementet *representasjon og kommunikasjon* (RK), med tilhøyrande kategoriar og eksempel på skåring (S. Eriksen & Vos, 2022, s. 46). Eksempellet gir ein peikepinn på korleis dei har tenkt, men mangel på ein komplett kodeinstruks frå S. Eriksen og Vos gir også eit svakare grunnlag for samanlikning av resultatata.

I studien vert koding og pilotering gjennomført av matematikklærarar. Gjennom utdanning og erfaring er tanken at matematikklærarar vurderer innhald i både kjerneelement og oppgåver likare enn andre utan bakgrunn som matematikklærar og dermed auke reliabiliteten til resultatata. Kodeinstruksen som studien nyttar vil gje føringar for korleis variablane skal vurderast, men vil også overlate noko av denne jobben til dei som skal kode.

#### Utforsking og problemløysing (UP) 1: Avdekke mønster og samanhengar

Dette er eit svært ope element, og kan dekkast på mange måtar. T.d. ved figuralt og talmønster. Kan også beskrive korleis noko heng saman. Uformelle samanhengar, t.d. samanhengen mellom sidekantane i ein trekant (t.d. Pytagoreiske Tripler), men ikkje korleis formlar for areal og volum heng saman (sjå AG 2). I sannsynsrekning kan samanhengen mellom store tal ved simulering og like stor sjanse for å få eit tal på ein terning, være ein samanheng i UP 1.

#### UP 2: Dele eit problem opp i delproblem

Dei fleste matematiske rekneoppgåver kan delast opp i delproblem (t.d.  $12 + 34 = (10 + 30) + (2 + 4) = 46$ ), men oppgåvene må vurderast i lys av kva trinn dei er gitt på, i dette tilfellet 9. trinn. Opnar metoden opp for å skåre all moglegheit for dekomponering, vil i praksis dei aller fleste oppgåver hamne i kategori 1 eller 2. Variabelen knyter seg til å bryte ned problem til delproblem, og ikkje berre innanfor eit reknestykke. Oppgåvene må vurderast utifrå konteksten dei står i, og kva læringsmål oppgåva har. Dette vert ofte definert av overskrifter eller teori i forkant av oppgåva. Å kode eit program del for del er ikkje det same som å dele eit problem opp i delproblem, men delproblema kan eventuelt løysast med programmering.

#### UP 3: Vurdere bruk av digitale hjelpemiddel eller ikkje

Dei aller fleste oppgåver *kan* verte løyste med programmering utan at det nødvendigvis er hensiktsmessig. T.d. kan oppgåva «Kva vert summen av heiltal mellom 1 og 6» løysast ved



programmering, sjølv om det er meir hensiktsmessig å løyse oppgåva på andre måtar. Dersom oppgåva nyttar summen mellom 1 og 1000 er programmering ein framgangsmåte. Men formel finnast også.

#### UP 4: Utvikle ei metode i ein ukjent situasjon

Det er noko uklart kva S. Eriksen og Vos (2022) legg i dette elementet, men i studien vert variabelen tolka til å handle om problemløysing, meir konkret analysing og omforming av problem, både kjente og ukjente (Utdanningsdirektoratet, 2019b, s. 2). Validering ved å vurdere løysingane er også ein del av dette.

#### Modellering og anvendingar (MA) 1: Knyte til dagleglivet, arbeidslivet eller samfunnet

Er oppgåva knytt til noko i dagleglivet, arbeidslivet eller samfunnet. T.d. terningkast og spel går ikkje under dette punktet, medan virusspreiing og sannsynsrekning i vêrmodellar inngår her.

#### MA 2: Matematisere – lage ein modell

Eleven må lage/konstruere ein matematisk modell av noko frå røynda. T.d. terningkast og spel.

#### MA 3: Kritisk vurdere ein modell

Eleven må vurdere modellen kritisk, ikkje berre vurdere om den fungerer eller ikkje. Dette kan vere t.d. modellens gyldigheitsområde eller å vurdere dei matematiserte variablane.

#### MA 4: Bruke ein modell i ein annan kontekst.

Ta ein modell ut av ein kontekst og inn i ein annan. Det kan t.d. være ein modell for terningkast og nytte same modellen for å simulere myntkast. Merk at modell er ikkje det same som dataprogram.

#### Resonnering og argumentasjon (RA) 1: Utvikle ei tankerekkje

Oppgåva etterspør uformelle resonnement

#### RA 2: Grunngje eller bevise

Oppgåva etterspør ei grunngjeving, gjerne uttrykt ved spørsmål som startar med «kvifor» eller «korleis». Inkluderer både formelle og uformelle bevis

#### Representasjon og kommunikasjon (RK) 1: Bruke matematisk språk

Det matematiske språket vert i studien definert som eit språk med teikn, symbol, matematiske omgrep og formlar (Hinna et al., 2011, s. 1059–1111). Oppgåva må gje eleven moglegheit til å nytte/vise, og ikkje berre lese, *uformelt* matematisk språk. Det matematiske språket må være knytt til det matematikkfagelege tema til oppgåva (t.d. sannsynsrekning).

Programmeringsspråk er i studien ikkje definert som det same som matematisk språk, og skal ikkje vurderast innanfor denne variabelen.

### RK 2: Bruke matematiske representasjonar

Matematiske objekt kan ikkje bli observerte direkte og tilgangen til matematiske objekt skjer gjennom eit system av representasjonar (Duval, 2006, s. 106–107; Kilpatrick et al., 2001, s. 94). Omgrepet matematisk objekt kan romme matematiske omgrep, idear og operasjonar (Svingen, 2018, s. 2). S. Eriksen og Vos (2022) skil i sitt rammeverk mellom å *lese* og å *tolke* matematiske representasjonar. Duval (2006) skil mellom «treatment» og «conversion» når det kjem til representasjonar, der «treatment» vert utført innanfor ein type representasjonar medan «conversion» beskriv overgang mellom ulike typar.

Denne variabelen er knytt til Duval (2006) sitt omgrep «treatment». Duval (2006) forklarar at dette handlar om handlingar innanfor eit register av representasjonar, og har som formål å endre ein representasjon innanfor eit register. Duval (2006) beskriv t.d. løysing av likningar og utrekningar innanfor same talsystem som «treatment». Vidare vert variabelen i denne studien vurdert i forhold til kva grad eleven har laga representasjonen sjølv og/eller brukt representasjonen *til* noko.

### RK 3: Grunnje valet av representasjonsform

Knytt til RK 2 og RK 4, og måler om oppgåva gir moglegheit til å grunnje val av representasjonsform nytta til å svare på oppgåva.

### RK 4: Byte mellom representasjonar

Ulike matematisk representasjon er når ei matematiske tilnærming erstattar ei anna og uttrykker det same (Duval, 2006, s. 112; Hinna et al., 2011, s. 1059; Svingen, 2018). Denne variabelen er knytt til definisjonen av RK 2, og måler om eleven kan byte mellom forskjellige representasjonar. Duval (2006, s. 112) skriv at omgrepet «conversion» innan matematiske representasjonar handlar om å skifte representasjonsregister utan å endre det matematiske objektet, og nyttar som eksempel overgangen frå algebraisk representasjon av ein funksjon til ein grafisk representasjon av det den same funksjonen. Funksjonen er uendra og representert

innanfor to ulike register. Byte mellom representasjonar føregår mellom forskjellige representasjonstypar: visuell, symbol, fysisk/konkret, kontekstuell og verbal (Leinwand et al., 2014, s. 25; Svingen, 2018, s. 3; Utdanningsdirektoratet, 2019b, s. 3).

#### Abstraksjon og generalisering (AG) 1: Nytte formelt symbolspråk eller formelle resonnement

Algebra er eit eksempel på eit formelt symbolspråk (Hinna et al., 2011, s. 196). I denne metoden vert omgrepet formelt symbolspråk tolka som eit internasjonalt system av teikn (Hinna et al., 2011, s. 1059). Formelt resonnement vert vurdert bevis/grunngeving gitt med eit formelt symbolspråk som t.d. algebra.

#### AG 2: Oppdage matematiske samanhengar og generalisere

Oppgåva presenterer ikkje eleven med ein ferdig løysingsmetode/algoritme. Eleven kjem fram til ein formel som er ein matematisk samanheng, generalisert. I sannsynsrekning kan samanhengen mellom store tal ved simulering og sannsynsrekning (t.d. side/antal sider) sannsyn for å få eit tal på ein terning) være ein

#### AG 3: Utforske tal, utrekningar og figurar. Formaliser ved algebra og formålstenlege representasjonar.

Dette elementet treng ingen instruks

#### Programmering (PR) 1: Lese/tolke koding (både pseudo og programmeringsspråk)

Oppgåva krev eller gir rom for lesing eller tolking av kode

#### PR 2: Utføre koding (både pseudo og programmeringsspråk)

Oppgåva krev eller gir rom for skiving av kode

#### PR 3: Programmeringskonsept

Oppgåva inneheld ein eller fleire av dei følgande programmeringskonsept henta frå Brennan og Resnick (2012, s. 3–6):

##### PR 3.1 Sekvensar/framgangsmåtar

At koden har ein oppbygging som er samanhengande, steg for steg.

##### PR 3.2 Løkker

Løkker kan brukast ofte i programmering, men det er ikkje alltid hensiktsmessig. I utgangspunktet kan ei løkke nyttast til å kode alt frå 2 like operasjonar til så mange ein

ynskjer utan ei praktisk øvre grense. Ein robot som skal programmerast til å gå frå punkt A og 7 skitt framover treng ikkje ei løkke for å få til operasjonen, bevegelsane kan kodast enkeltvis. Men dersom oppgåveformuleringa indikerer at roboten kanskje må gå lenger, er ei løkke smidigare å nytte og oppgåva vil skåre verdien 2 for denne variabelen. Kva skåre som er riktig å gje for denne variabelen må vurderast i kvar enkelt oppgåve. Simuleringsoppgåver i sannsynsrekning nyttar gjerne mange simuleringar og løkker vil ofte være hensiktsmessig.

### PR 3.3 Handlingar

Sjekkar programmet om handlingar er utført, for deretter å utføre kode basert på om ei handling er utført eller ikkje. T.d. trykke på mellomrom for å hoppe med ein figur.

### PR 3.4 Parallellisme (eller multitasking)

Programmet kan utføre fleire ting samtidig, t.d. via sekvensar eller via handlingar i programmet.

### PR 3.5 Kontrollstrukturar

Programmet sjekkar verdien til variablar og utfører kode basert på verdiane. T.d. hvis-kommando.

### PR 3.6 Operatorar

Matematiske, logiske og tekst-streng-operatorar. T.d. addisjon, større enn, sjekke om to variablar er like og sjekke lengda på ein tekststreng

### PR 3.7 Data

Programmet lagrar data i variablar eller lister

Kategoriar:

- 0 Oppgåva gir ikkje eleven moglegheit til å vise dette
- 1 Oppgåva kan gi moglegheit til å vise (litt av) dette
- 2 Oppgåva krev dette

## 3.5 Utval av oppgåver

Ein viktig del av metoden er å lage kriterium for kva oppgåver som skal inkluderast i studien. Mange oppgåver kan løysast med programmering, men det er ikkje nødvendigvis hensiktsmessig for alle oppgåver. Denne metoden legg difor til grunn at oppgåver skal løysast

med programmering når det er eit eksplisitt krav eller ligg i ein del av læreboka som handlar om programmering. Utan eit slikt skilje vil dei fleste oppgåve verte inkluderte i studien, og metoden hadde dermed ikkje vore eigna til å gje svar på problemstillinga.

Ei ulempe med eit slikt kriterium for utval er at det kan vere oppgåver som er eigna å løyse med programmering, men som ikkje kjem med i utvalet. Dersom ei oppgåve er formulert slik: «Forklar korleis du vil gå fram for å rekne ut summen av alle heiltal mellom 1 og 10 000», kan det tenkast at oppgåveforfattaren ynskjer at eleven skal/kan løyse denne med programmering. Ei slik oppgåve vil ikkje verte fanga opp i utvalet oppgåver med dei kriteria som ligg til grunn, sjølv om den absolutt kan løysast med programmering.

Likevel er det slik at studien har fokus på programmeringsoppgåver. For å sikre at det er programmeringsoppgåver som ligg til grunn for hovudanalysen, er det formålstenleg å lage utvalskriterier som sikrar at utvalet består av oppgåver som er tydelege definert som programmeringsoppgåver.

Metoden nyttar følgjande inklusjonskriterium for studien, der ei oppgåve må kvalifisere til eit eller begge av desse:

- Programmering er eksplisitt nemnt i oppgåva eller i konteksteininga til oppgåva
- Oppgåva ligg i eit kapittel eller eit delkapittel/seksjon som handlar om programmering.

Figur 5 viser ei oppgåve henta frå ei lærebok nytta i studien og viser ei oppgåve som kan løysast ved programmering. Likevel vil oppgåva, innanfor ramma til studien, ikkje bli inkludert i utvalet: Det er ikkje eksplisitt nemnt programmering i oppgåva eller i konteksteininga og oppgåva står ikkje i eit kapittel som omhandlar programmering.

*Figur 5 Eksempel på oppgåve som ikkje vert rekna som programmeringsoppgåve*

Oppgåve 4.1: Nejat har tre skjorter og tre bukser som ho vurderer å bruke på skulen. Kor mange ulike antrekk (skjorte og bukse) kan ho velje mellom?

Figur laga av forfattaren som ei attgjeving av oppgåve 4.1 frå læreboka til Hjardar og Pedersen (2021a, s. 242).

Figur 6 viser ei oppgåve, også henta frå ei lærebok som er nytta i studien, som vert rekna som ei programmeringsoppgåve i studien: Program er eksplisitt nemnt i oppgåva og oppgåva ligg i ein del av læreboka som handlar om programmering.

Figur 6 Eksempel på programmeringsoppgåve

Oppgåve 13.15, a: Lag eit program som simulerer 100 kast med ein mynt. Programmet skal rekne ut relativ frekvens for dei to utfalla.

Figur laga av forfattaren som ei attgjeving av oppgåve 13.15, del a frå læreboka til Kongsnes og Wallace (2020b, s. 96)

Under den endelege analysen av lærebøkene kom det til syne ei utfordring med måten metoden vel ut oppgåver til utvalet på som ikkje vart fanga opp i preanalysen. I kriteriet for oppgåveutval står det at oppgåver som ligg i eit kapittel eller eit delkapittel/seksjon som handlar om programmering, vart teken med i utvalet. Læreboka Matematikk 9 opplyser i seksjonen «Simulering» at når ei hending skal simulerast vert det nytta programmering *eller* eit rekneark (Hjardar & Pedersen, 2021a, s. 283). I kap. 3.4.2 definerer denne oppgåva rekneark som noko anna enn programmering, og det er dermed ikkje tydeleg om kapittelet om «Simulering» i Matematikk 9-boka oppfyller kriteriet knytt til oppgåveutvalet.

For å avgjere om oppgåvene i kapittelet «Simulering» skulle inkluderast i utvalet, vart oppgåvene i dette kapittelet vurderte. Oppgåvene i kapittelet nyttar utelukkande omgrepet program om kommandoar eller ei samling kommandoar knytt til bruk i rekneark, sjå Figur 7 for eksempel. Når det i kap. 3.4.2 vert definert ein skilnad mellom programmering og rekneark, vert reknearkoppgåver ekskludert frå utvalet i studien. Konklusjonen vart at oppgåver i kapittelet «Simulering» i Matematikk 9-boka ikkje innfridde nokon av dei to kriteria for å verte inkludert i oppgåveutvalet, og konsekvensen vart at kapittelet om sannsynsrekning i læreboka Matematikk 9 ikkje har nokon oppgåver som vert filtrert inn i oppgåveutvalet.

Figur 7 Eksempeloppgåve med rekneark som ikkje er ned utvalet

4.51: Lag eit program som simulerer eit kast med éin terning. Bruk formelen «=TILFELDIGMELLOMROM» når du lager programmet.

#### HUGS

Du kan bruke \$-teiknet framfor kolonnebokstav og radnummer for å låse cellereferansen.

Figur laga av forfattaren som ei attgjeving av oppgåve 4.51 frå læreboka til Hjardar og Pedersen (2021a, s. 268).

### 3.6 Presentasjon av lærebøker

I studien er det valt ut lærebøker frå tre av dei mest nytta forlaga. Desse lærebøkene er «Matemagisk 9» (Kongsnes & Wallace, 2020b) frå Aschehoug, «Maximum 9» (Tofteberg et

al., 2021) frå Gyldendal og totalt tre deler av læreverket Matematikk 9: «Matematikk 9 Grunnbok» (Hjardar & Pedersen, 2021a) og «Matematikk 9 oppgåvebok» (Hjardar & Pedersen, 2021b) og Programmering 8-10 (Hjardar, u.å.) frå Cappelen Damm. Bøkene frå Cappelen Damm vert behandla som ei lærebok i oppgåveanalysen. Fagbokforlaget har også ein serie med matematikklærebøker for ungdomstrinnet, men boka for 9. trinn er p.t. ikkje oppdatert for LK20 (Carlsson et al., 2007)

Vidare kjem ein presentasjon av kvar lærebok. For å vise forskjellen på lærebøkene vart det i preanalyen av læreboka undersøkt kva type struktur det er på oppgåvene, og andre ting som er relevant for boka si oppbygging.

### 3.6.1 Matemagisk 9

Matemagisk er eit læreverk som strekk seg gjennom heile grunnskulen, frå Matemagisk 1 til Matemagisk 10. I den vidaregåande skulen har forlaget skifta namn på bøkene, og desse heiter «Matematikk» og er kombinert med ein fag-kode, t.d. «R2».

Frå Aschehoug sine nettsider om Matemagisk 8-10 står det m.a. «*Matemagisk* utvikler algoritmisk tenking og elevene lærer programmere på fagets premisser» (sic) (Aschehoug, 2021). Læreverket for ungdomstrinnet inneheld grunnbok og elevhandbok, i tillegg til digitale bøker og nettressursar.

Boka har ei fellesløype som omfattar grunnleggande teori for emnet, saman med eksempel og oppgåver m.m. Denne løypa inkluderer programmering i delkapitla der det er relevant, ifølgje boka. Det er egne oppgåver for å snakke matematikk, og desse er merka med «Snakk matte». I slutten av kvart delkapittel er det «Følg stien»-oppgåver som dekkjer det mest sentrale faginnhaldet, og som trener på ein ting om gangen. Desse oppgåvene bygger vidare på innhaldet i fellesløypa. I slutten kvart delkapittel finst også «Terrengløypa»-oppgåver med samansette utfordringar frå fleire tema, samtidig. Alle desse oppgåvetypene vert tekne med i studien.

Boka inneheld ikkje eit eige programmeringskapittel.

### 3.6.2 Matematikk 9

«Matematikk»-læreverket strekker seg gjennom heile grunnskulen, frå 1. trinn til 10. trinn. På Vidaregåandetrinnet heiter læreverket frå Cappelen Damm «Sinus». Læreverket «Matematikk» for den vidaregåande skulen er laga av Aschehoug og er ikkje ei forlenging av

læreverket Matematikk 1-10. I tillegg til ei grunnbok har matematikk 9 ei oppgåvebok med treningsoppgåver frå dei ulike emna i grunnboka, Matematikk 9 oppgåvebok.

Kvart kapittel i grunnboka er delt i tre delar: Lærestoff og oppgåver, undervegsvurdering og oppgåve til tverrfaglege tema (Hjardar & Pedersen, 2021a). Undervegsvurderinga er eit sett med oppgåver samla bakarst i kapittelet saman med ei tverrfagleg oppgåve. I grunnboka nyttar forfattarane ei pil eller eit spørsmålsteikn på diskusjonsspørsmål som boka stiller til eleven individuelt eller til klassen. Nokre av oppgåve i grunnboka og alle oppgåvene i oppgåveboka har symbol for oppgåver som høyrer til nivå 1, nivå 2 og nivå 3. Dette vert ikkje forklart nærare i grunnboka, men det er underforstått at dette er ei nivåddifferensiering for elevane, og dette vert bekrefta i Matematikk 9 oppgåvebok. Studien tek ikkje omsyn til denne nivåddifferensieringa og vurderer kvar oppgåve som ei heilheit. Alle oppgåvene som er nemnde til no vert vurderte i studien.

Matematikk 9 oppgåvebok har tre fargekoda deler i kvart kapittel: Oppgåver, repetisjon og fordjupingsoppgåver. Alle oppgåvene i oppgåveboka er nivåddifferensierte, men også her ser studien på kvar oppgåve under eitt.

Grunnboka og oppgåveboka har ikkje eit eigne programmeringskapittel. Det som ikkje kjem fram i lærebøkene, men som kjem fram frå nettsida til læreverket, er at programmeringsoppgåver bevisst er halde utanfor lærebøkene (Cappelen Damm, u.å.). Programmeringsoppgåver- og opplæring kjem i eit eige hefte som heiter «Programmering 8-10» (Hjardar, u.å.).

Likevel er heftet frå Cappelen Damm (Hjardar, u.å.) teken med i lærebokutvalet til studien. Etter ei totalvurdering vart det inkludert av to grunnar. For det første er det eksplisitt nemnt i nettstaden til læreboka at heftet skal komplementere lærebøkene. For det andre er heftet inndelt i oppgåver for hvv. 8, 9 og 10 trinn, der oppgåvene innanfor kvart trinn er inndelt i seksjonar som korresponderer til seksjonsoverskrifter i lærebøkene, og som gjer at dette heftet fungerer som ei forlenging av lærebøkene. Seksjonen som vert inkludert i studien, er «Python: Simulering» (Hjardar, u.å., s. 46–47).

### 3.6.3 Maximum 9

Matematikklæreverket Maximum er eit læreverk for ungdomstrinnet. På barne- og mellomtrinnet heiter Gyldendal sitt læreverk «Multi». I den vidaregåande skulen har Gyldendal læreverket «Mønster» for matematikkundervising.



Maximum 9 (Tofteberg et al., 2021) er tilpassa LK20. Boka kjem anten i ei heildigital utgåve, eller i ein kombinasjon av analogt og digitalt. Dei digitale ressursane er «Maximum Smart Øving» og «Skolestudio». Smart Øving «... gir mulighet for mengdetrening og øving tilpasset elevens nivå» (Gyldendal, u.å.), medan Skolestudio presenterer m.a. innhald frå elevboka, fasitar, prøver og lærarretteiing. Skolestudio gir også tilgang til eit programmeringsverktøy.

I læreboka står det at den inneheld forskjellige typar oppgåver. Det er oppgåver til diskusjon og samarbeid, oppdrag og praktiske aktivitetar, og alle desse vert inkluderte i studien. I slutten av kapittelet har boka ein seksjon som heiter «Sjå samanhengar» og det står i boka at det er varierte oppgåver og aktivitetar utforma for djupnelæring og for å sjå samheng både i faget og mellom faga. Oppgåver under «Sjå samanhengar» vert også inkluderte i studien.

Boka har oppgåver med ulik vanskegrad, men studien tek ikkje omsyn til denne nivåddifferensieringa og vurderer kvar oppgåve som ein heilskap.

Boka har ikkje eit eige programmeringskapittel.

### 3.7 Forskingskvalitet

Resultata av ein slik studie skal i størst mogleg grad være pålitelege og gyldige. God kvalitet på datamaterialet og analysane er med på å sikre dette. Her tek oppgåva føre seg faktorar knytt til datakvalitet og forskingskvalitet.

#### 3.7.1 Reliabilitet

Reliabilitet kan også kallast pålitelegheit og er omgrepet som vert brukt til å skildre om datainnsamlinga gir pålitelege data (Grønmo, 2016, s. 240–241; Postholm & Jacobsen, 2018, s. 223). Reliabiliteten, som saman med validitet inngår i studiens totale truverdighet, er knytt opp til prosessen med å lage utvalet, metodevalet og moglegheita til å etterprøve resultata (Grønmo, 2016, s. 240–241; Postholm & Jacobsen, 2018, s. 223). Den høgaste form for reliabilitet er at resultata kan reproduserast identisk av andre forskarar (Grønmo, 2016, s. 240–241; Postholm & Jacobsen, 2018, s. 223). Omgrepet vert tolka litt forskjellig alt ettersom om studien har ein kvantitativ eller kvalitativ tilnærming (Cohen et al., 2018; Grønmo, 2016, s. 244; Postholm & Jacobsen, 2018), og Postholm og Jacobsen skriv at det vil vere svært vanskeleg å reprodusere like data i ein kvalitativ studie. Denne studien nyttar innhaldsanalyse som datainnsamlingsmetode, og ein styrke med denne metoden er etterprøvbarheit som eit resultat av lite rom for skjønn og eit relativt standardisert møte med det empiriske materialet (Bratberg, 2021, s. 103).

Det vert vanlegvis vektlagt to hovudtypar reliabilitet: stabilitet og ekvivalens (Grønmo, 2016, s. 242–244). Stabilitet seier noko om samsvar mellom data som er samla inn med same metode på same fenomen, men på ulike tidspunkt, medan ekvivalens seier noko om samsvar mellom data som er samla inn med same metode, på same tidspunkt, men t.d. er samla inn av forskjellige kodarar.

Studien ser på eit utval oppgåver henta frå tre trykte lærebøker som var produserte for å dekke endringane i LK20. Det er vanskeleg å nøyaktig anslå kor mange som nyttar desse lærebøkene. I TIMMS kjem det fram at 97% av undervisinga i den norske skulen er basert på lærebøker (Mullis et al., 2012, s. 392) er det truleg at lærebøkene i studien representerer store deler av det pensum-materialet som vert nytta i norske klasserom.

Å vurdere stabilitets-pålitelegheita til studien er ikkje interessant i denne samanheng. Studien har som mål å utvikle eit nytt og ubrukt rammeverk for å analysere programmeringsoppgåver, og data frå studien kan ikkje direkte samanliknast med tidlegare studiar. Stabiliteten til datamaterialet gir uttrykk for utviklingstendensar og endringar, og fram i tid kan metoden frå studien nyttast på andre, og nye læreverk for å sjå korleis reliabiliteten er mellom data i denne studien, og dei som vert innsamla seinare.

Ekvivalens-pålitelegheita er viktig for å sikre at datamaterialet har tilfredsstillande reliabilitet, også når datainnsamlinga blir utført av éin person (Grønmo, 2016, s. 244). Grønmo (2016) oppgir to metodar for å reliabilitetsteste ekvivalens: intersubjektivitet og halveringsmetoden. Intersubjektivitetsmetoden (Grønmo, 2016, s. 245–246) kan nyttast i kvantitativ innhaldsanalyse, men krev fleire kodarar og studien har berre ein kodar. Eit eksempel på ein intersubjektivitetsmetode som kan nyttast med to kodarar er Cohen's Kappa (Nagaraja & Choudhary, 2017, s. 289). Halveringsmetoden (Grønmo, 2016, s. 246–247) kan nyttast dersom datasettet inneheld fleire indikatorar som måler det same, og som kan delast i to grupper med indikatorar som kan målast opp mot kvarande. Studien ser på kjerneelementa i matematikk og desse overlappar i nokon grad, men skilja er uklåre og ikkje alle kjerneelementa overlappar tydeleg. Dette gjer at halveringsmetoden heller ikkje er hensiktsmessig å nytte seg av.

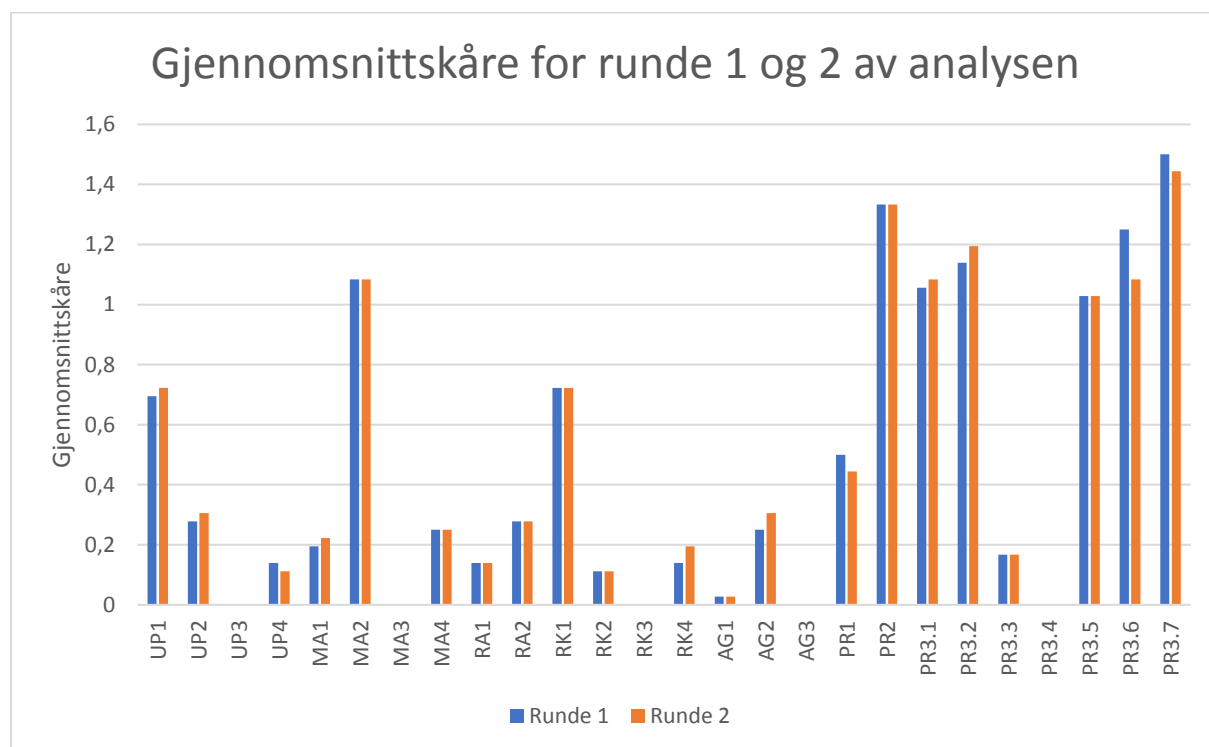
For å forbetre reliabiliteten til studien vart kodeskjema koda to gongar etter kvarandre. Det vart oppretta to kodeskjema, og etter to rundar med koding vart datamaterialet slått saman og resultatata samanlikna. Denne måten å samanlikne resultat på er ein variant av

intersubjektivitetsmetoden. Vidare vart oppgaver som fekk forskjellig skåre i ein eller fleire variablar, reanalyserte enno ein gong for å avgjere kva verdi variabelen skulle ende opp med.

I arbeidet med å samanlikne dei to rundane med koding vart det tydeleg at forståinga av kjerneelementa og operasjoniseringa av desse, hadde utvikla seg etter den første koderunda, sjå Figur 8. I kor stor grad kodaren kodar likt gjennom heile kodeskjemaet, vert kalla intrakoderreliabilitet (Turner et al., 2013) og i forkant av oppgåveanalysen vart det arbeid med å operasjonisere kjerneelement og utvikla kodeinstruks.

Måten å tolke oppgaver på hadde også endra seg mellom koderunde ein og to, og i den tredje analysen av oppgaver med forskjellig resultat i koderunde ein og to, vart det tydeleg at verdien frå den andre runda med koding vart ståande. Figur 8 gir ei oversikt over gjennomsnittsskåren for hvv. første og andre runde av analysen.

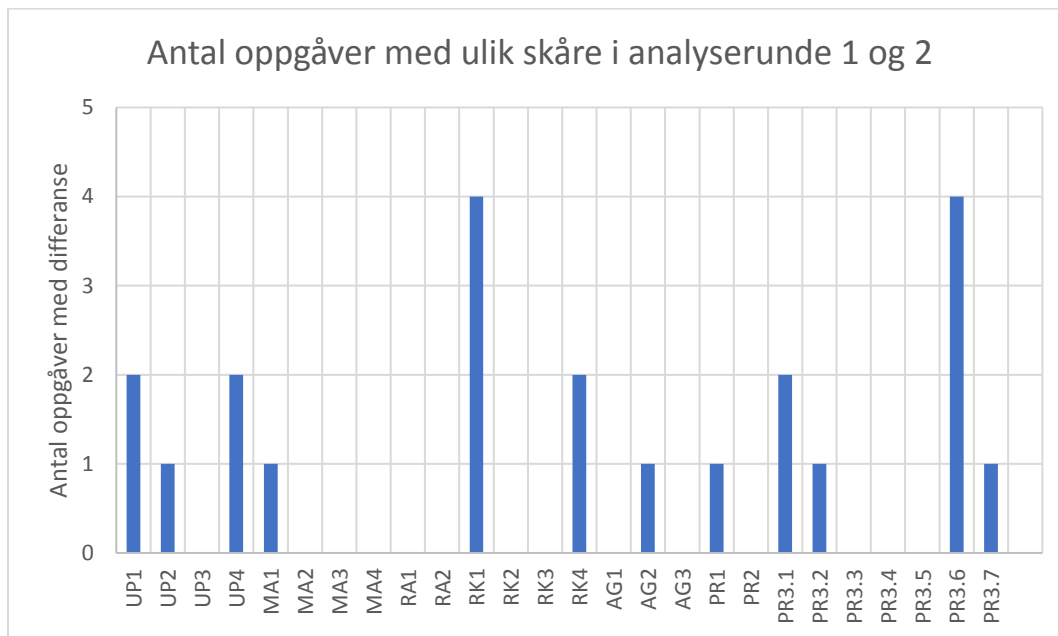
Figur 8 Gjennomsnittsskåre for runde 1 og 2 av analysen



Figur 8 gir kanskje ikkje eit godt nok bilete av skilnaden på analyserunde 1 og 2. Sjølv om søylene for analyserunde 1 og 2 for eit kjerneelement viser lik høgde, betyr ikkje det at alle oppgåvene vart likt koda for denne variabelen. Dersom den første halvparten av oppgåvene i utvalet vart koda med skåren 2 og andre halvparten med 0 i første runde, og omvendt i andre runde, ville søylene fått lik høgde. Figur 9 gir eit betre bilete av dei faktiske forskjellane i

skåring av oppgåver. Merk at samlevvariabelen *programmeringskonsept* (PR 3) er fjerna frå Figur 8 og Figur 9.

Figur 9 Antal oppgåver med ulik skåre i analyserunde 1 og 2

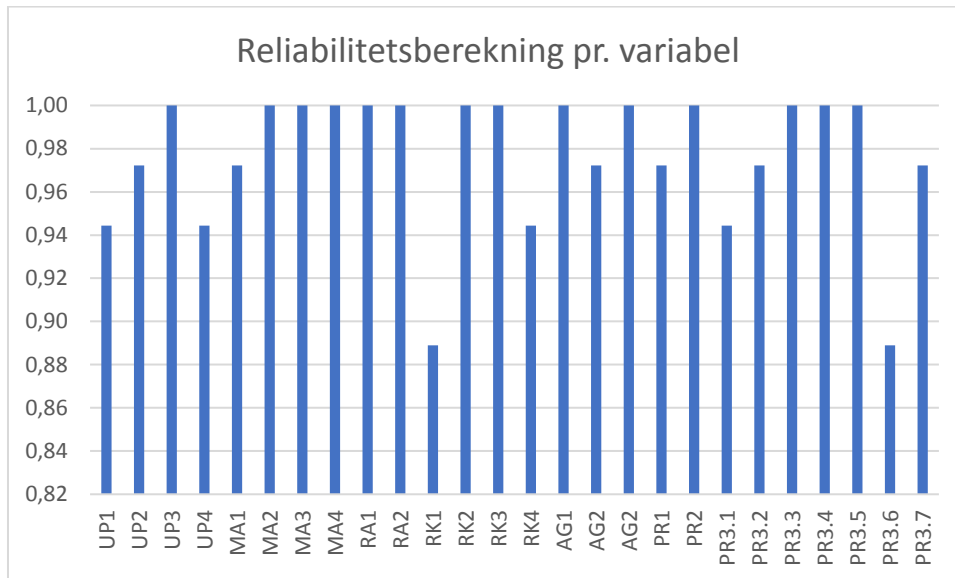


Forskjellen på søylene til runde 1 og runde 2 har større forskjellar for programmeringsvariablane (PR) enn for variablane knytt til kjerneelementa, med hhv. 2,6% og 2,1% avvik i variabelskåringane. Ein mogleg del av forklaringa er at kodaren har arbeidd som lærar og tidlegare har operasjonalisert kjerneelementa gjennom å undervise i matematikk. I arbeidet med å sikre høg intrakoderelibilitet og utvikling av kodeinstruks vart det også arbeidd meir med operasjonalisering av kjerneelementa enn med variablane for programmering. Rammeverket til S. Eriksen og Vos (2022) kjem ikkje med ein fullstendig kodeinstruks for variablar knytt til kjerneelementa, medan rammeverket til Brennan og Resnick (2012) hadde ei meir utfyllande skildring av sine variablar knytt til programmeringsvariablane. Dette førte til eit større behov for å arbeide med kjerneelementa, og ga kanskje ei betre internalisering av innhaldet i variablane til kjerneelementa.

I sum er det forskjellar mellom svara i 22 tilfelle av totalt 936 moglege. Ved å nytte ein enkel måte for å berekne reliabilitet (Grønmo, 2016, s. 247) gir dette eit samla reliabilitetsmål for datamaterialet på  $\frac{(936-22)}{936} = 0,98$ , noko som er eit uttrykk for høg reliabilitet. Kodinga vart berre gjennomført av éin kodar så reliabilitetsmålet vert ikkje korrigert for tilfeldigheter slik t.d. metoden Scotts Pi gjer (Grønmo, 2016, s. 247). Figur 10 syner reliabilitetsberekning for

kvar einiskild variabel i studien, og nyttar same metode for berekning som vart nytta for heile datamaterialet.

Figur 10 Reliabilitetsberekning pr. variabel



Etter to rundar med koding var det naturleg å stille spørsmålet: Kor mange rundar med koding er hensiktsmessig? Det er vanskeleg å svare på, men i studien vart det stopp etter to rundar, med ei tredje runde for å avgjere der det var differanse mellom rundane. Ei mogleg løysing på reliabilitetsspørsmålet i denne situasjonen kan være å fortsette rundar med koding av datamaterialet og samanlikne resultatata frå kvar runde med tidlegare rundar. Metoden kan undersøke om differansen i resultatet mellom rundane aukar eller minkar og sjå på utviklinga av differansen. Dersom det etter nokre rundar er liten eller ingen endring i differansen syner dette at kodaren er einig med seg sjølv.

### 3.7.2 Validitet

Validitet handlar om kor gyldige datamaterialet er som grunnlag for å finne svar på problemstillingane (Grønmo, 2016, s. 241–242). Postholm og Jacobsen omset omgrepet validitet med *gyldigheit* og skil mellom indre og ytre gyldigheit (2018, s. 222–235), men hvv. intern og ekstern validitet vert også nytta i litteraturen (Cohen et al., 2018; Grønmo, 2016). Den indre gyldigheita seier noko om kor vidt resultatata er gyldige for det som er studert og den ytre seier noko om kor vidt resultatata er overførbare til andre kontekster (Postholm & Jacobsen, 2018). For å oppnå høg validitet er det viktig at omgrepa er operasjonaliserte på ein slik måte det innhaldet i kvart omgrep kjem klart og tydeleg fram (Grønmo, 2016, s. 252;

Postholm & Jacobsen, 2018, s. 231). Dette gjeld både det teoretiske innhaldet i oppgåva og i utviklinga av kodeinstruksen i kap. 3.4.3.

I studien er det forsøkt å oppnå høg intern gyldigheit ved at det er nytta relevant teori, læreplanar og omgrepsdefinisjonar for å velje ut lærebokoppgåver som er relevante for å svare på problemstillinga mi. Desse lærebokoppgåvene vart deretter analyserte med eit analyseverktøy som vart utvikla i studien, og som vart sett saman av to andre rammeverk (Brennan & Resnick, 2012; S. Eriksen & Vos, 2022) og deretter modifisert. Sidan rammeverket vart konstruert for studien kan det ikkje seiast med stor grad av sikkerheit at analyseverktøyet måler det som faktisk skal målast i studien. Sjølv om rammeverka som var grunnlaget for utviklinga av analyseverktøyet brukt i studien, kvar for seg målar deler av det studien ynskjer å undersøke, er det ingen automatikk i at det er hensiktsmessig å kople dei saman, og modifiseringa av dei to rammeverka kan ha gitt effektar som gjer analyseverktøyet mindre eigna. Viser til ei lenger drøfting av konstruksjonen av rammeverket i kap. 5.2.3.

Ei utfordring knytt til gyldigheit i studien er formuleringa i problemstillinga og moglegheita til å seie noko om korleis metoden måler det oppgåva freistar å måle. Eit mål med studien var å utvikle rammeverket som er nytta i analysen av lærebokoppgåvene, og at dette rammeverket skulle kunne hjelpe lærarar og andre til å analysere deler av ei lærebok eller eit undervisningsopplegg. Målet var ikkje å rangere lærebøker på marknaden i dag og difor vart det i problemstillinga nytta omgrepet «korleis» i staden for «i kor stor grad». Studien gir derimot eit resultat som må vurderast opp i mot andre faktorar knytt til undervisning. Likevel indikerer funna i studien at analyseverktøyet gir svar som kan nyttast i undervisningssamanheng og gir svar på problemstillinga.

### 3.7.3 Forskingsetiske vurderingar

I ei oppgåve med kvantitativ innhaldsanalyse som metode er problemstillinga rundt etiske vurderingar litt annleis enn for t.d. ei oppgåve med ei kvalitativt tilnærming og intervju som metode. Der studiar med intervju som metode må søke om lov til å samle inn data, sikre anonymitet i datamaterialet og passe på at det er informantane sine autentiske fortellingar som kjem til syne, har den kvantitative innhaldsanalyse færre formelle krav.

Likevel finst det generelle forskningsetiske retningslinjer (De nasjonale forskningsetiske komiteene, 2019) som også den kvantitative innhaldsanalyse må forhalda seg til. I desse retningslinjene står det m.a. at forskaren skal freiste å få fram ei sanning, at forskinga skal ha

høg kvalitet og vere truverdig. Deler av retningslinjene vert dekkja av reliabilitet og validitet, men det er også viktig å legge merke til krava om at ein forskar ikkje må fabrikkere eller forfalske data, ei heller la vere å oppgje resultat som forskaren ikkje ynskjer å vise fram.

I denne oppgåva er det frå forfattaren si side freista å vise kva som har vorte gjort, og kva vurderingar som ligg til grunn. I t.d. kodeinstruksen i kap. 3.4.3 vert variablane som blir opplevde som diffuse, diskuterte og operasjonaliserte enno meir. Metoden er forklart i detalj og gjennomgangen av metoden sitt overordna design i kap. 3.3 er lagt inn i oppgåva for å gje eit bilete av forskingsprosessen som er nært opp til røynda. Målet er å vise forskning som er prega av transparens og truverdighet.

## 4 Presentasjon av datagrunnlaget og resultat

Resultata vert delte opp i to hovudgrupper: Resultat av filtrering av oppgåver, og resultat av oppgåveanalysar.

### 4.1 Resultat av filtrering av oppgåver

Desse resultata tek føre seg data i forbindelse med utval av oppgåver til analysedelen.

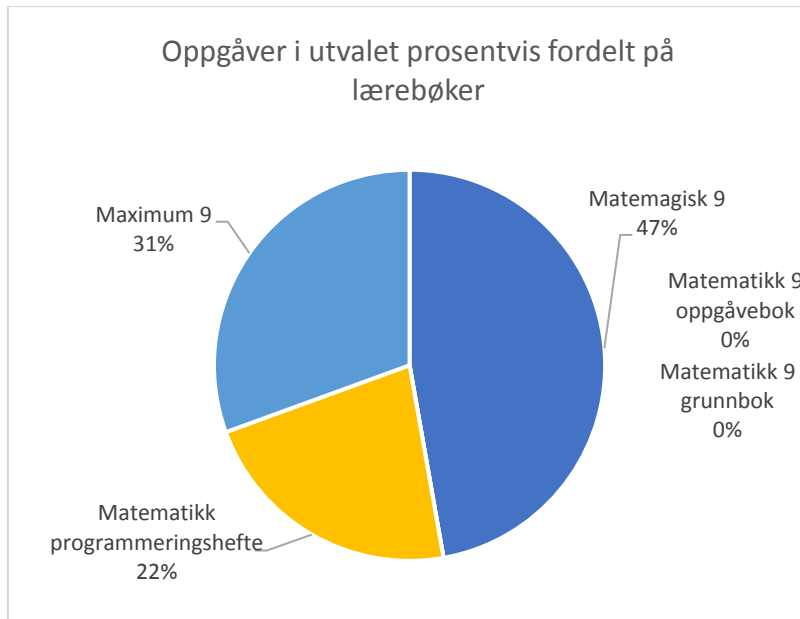
Totalt for alle lærebøkene var det 296 oppgåver som vart vurderte i filtreringsprosessen.

Oppgåvene var fordelte over fem lærebøker i tre læreverk. Med alle tre bøkene for «Matematikk»-læreverket slått saman, inneheld dette læreverket vesentleg fleire oppgåver enn dei andre læreverka; 149 oppgåver kan Matematikk 9-læreverket skilte med, medan Maximum hadde 87 og Matemagisk hadde 60.

Av alle oppgåvene i sannsynskapitla var det 36 oppgåver (12%) som passa kriteria for å verte inkluderte i utvalet oppgåver og 260 som ikkje gjorde det. Fordelinga av oppgåver inkluderte i utvalet var noko skeivfordelt på lærebøkene, og to bøker inneheldt ikkje oppgåver som fall innanfor programmeringskriteria i det heile. Bøker som inneheldt oppgåver inkluderte i utvalet kom frå alle tre læreverka (Figur 11). Matemagisk 9-læreboka hadde 17 oppgåver i utvalet, Matematikk 9-læreverket hadde 8 og Maximum 9-læreboka hadde 11. Matemagisk 9-læreboka har ein del fleire oppgåver representerte i utvalet enn dei andre lærebøkene, til tross for at dette er den læreboka med færrest oppgåver i kapittelet for sannsynsrekning, totalt sett. I Figur 11 er Matematikk 9 grunnbok, Matematikk 9 Oppgåvebok og programmeringsheftet som høyrer til Matematikk-læreverket skilt ut som eigne bøker for å illustrere kvar i lærebøkene oppgåvene finnast.

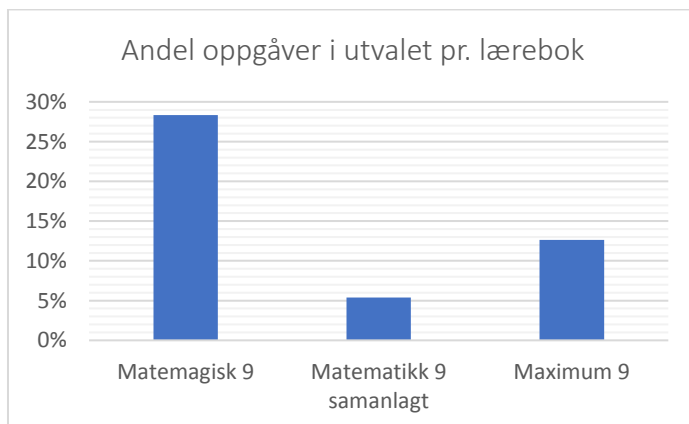


Figur 11 Oppgåver i utvalet prosentvis fordelt på lærebøker



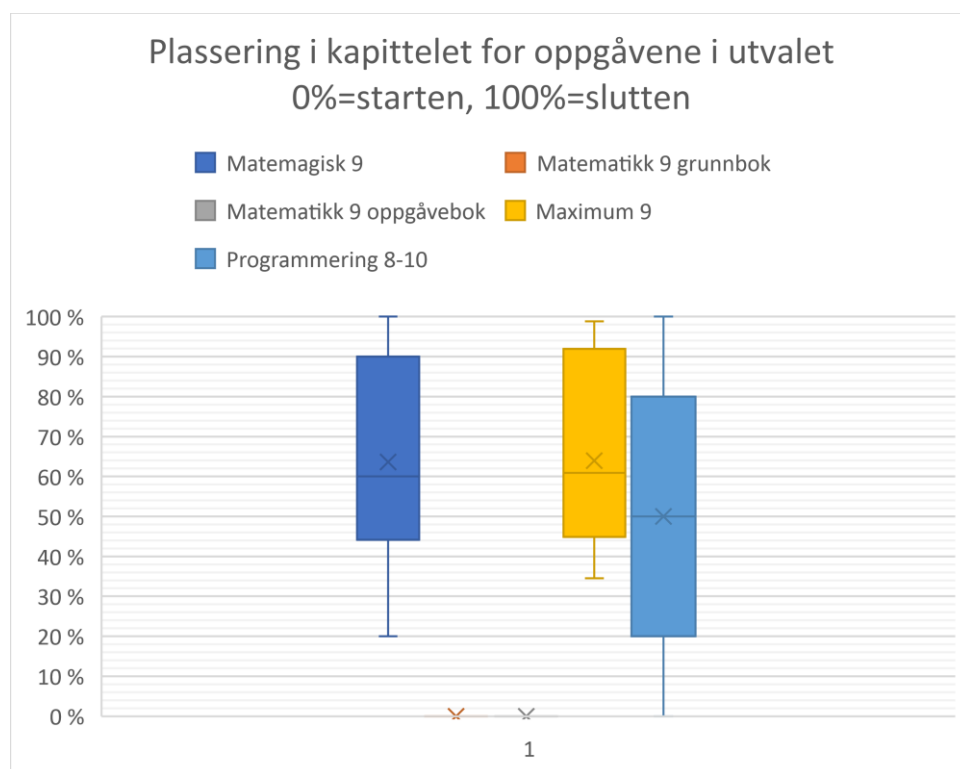
Det er stor variasjon i andel oppgåver i utvalet for kvart læreverk (Figur 12).

Figur 12 Andel oppgåver i utvalet pr. lærebok



Vidare kan ein hente ut informasjon om kvar i kapittelet programmeringsoppgåvene finst. Figur 13 viser dette sett opp i eit boksdiagram. Eit slikt diagram viser kvar hovudtyngda programmeringsoppgåver er samla, men også kor stort spenn det er i utbreiinga av programmeringsoppgåver i kapittelet.

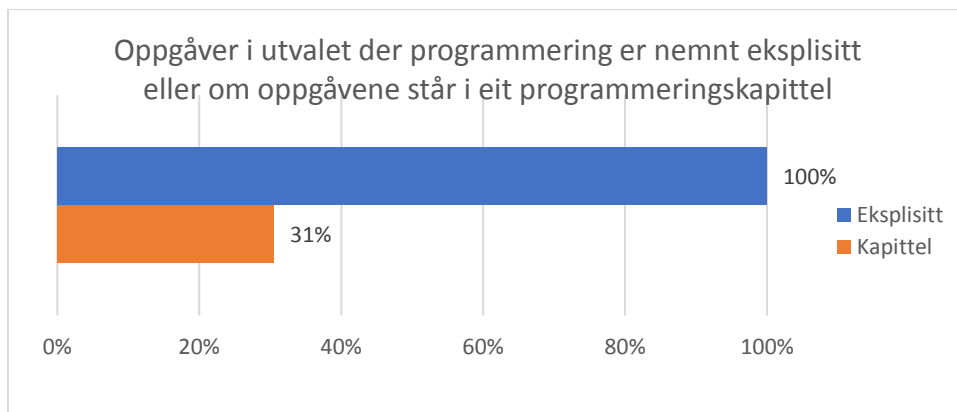
Figur 13 Plassering i kapittelet for oppgåvene i utvalet.



Plasseringa av programmeringsoppgåver i kapittelet kan vere ein nyttig statistikk t.d. for å tilpasse opplæring til den enkelte elev. Dersom matematikklæraren vel å tilpasse opplæringa ved å tilby eigne lærebøker eller oppgåver på ark (Botten et al., 2008, s. 24), for dei som ikkje klarar å halde læreboka sitt tempo, kan desse elevane gå glipp av programmeringsoppgåver som kjem seinare i kapittelet. Men dersom lærebøkene skal starte trening av programmeringsferdigheiter i starten av kapitla, kan det bli komplisert for elevane når dei i tillegg til programmering, skal starte på nytt fagstoff. Preanalysen av læreverka syner at alle kapitla om sannsyn er starta med oppgåver som ikkje er komplekse, og det er ikkje ein klar fasit på kvar i kapitla programmeringsoppgåvene bør starte.

I prosessen med å filtrere oppgåver vart det vurdert om programmering var spesifikt nemnt i oppgåva eller om oppgåva sto i eit kapittel, ein seksjon eller ein kontekst som bestemte at oppgåva skulle nyttast som ei programmeringsoppgåve. Filtringa viser at alle oppgåvene i utvalet spesifiserte at oppgåva handla om programmering, medan 31% står i ein del av lærebøkene som spesifiserte at hensikta med oppgåvene er programmering (Figur 14). Med dei lærebok-kapitla studien såg på, var det i praksis ikkje behov for to inklusjonskriterium. Det er derimot ikkje gitt at andre kapittel i lærebøkene, eller andre lærebøker, har lik oppgåveformulering som oppgåvene i utvalet til studien.

Figur 14 Om oppgåver i utvalet er nemnt eksplisitt eller om dei står i eit programmeringskapittel



Inkludert i desse tala er også oppgåver frå Matematikk 8-10 programmeringshefte, som vart vurdert slik at dei ligg i ein del av læreboka der hensikta er programmering.

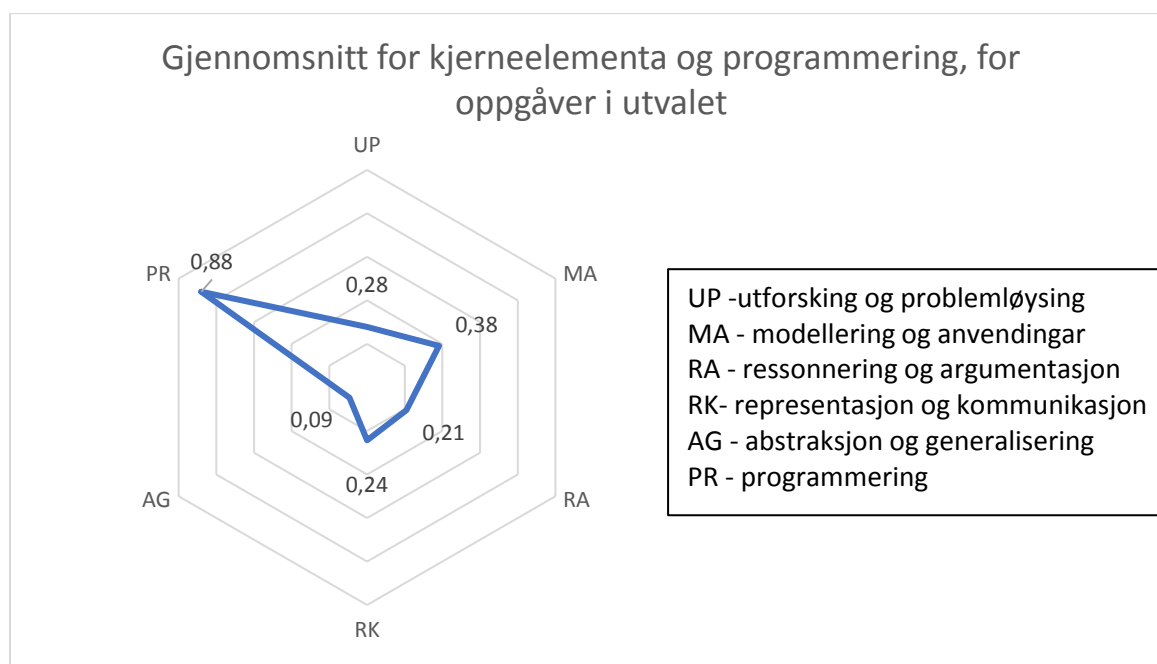
## 4.2 Resultat av oppgåveanalysane

Dette delkapitelet tek føre seg resultatata etter at oppgåvene i utvalet er ferdiganalyserte.

### 4.2.1 Resultat knytt til samlevvariablane

Figur 15 gir ei oversikt over gjennomsnittskåren for oppgåver i utvalet, samla pr kjerneelement og for programmeringsvariablane samla (PR). Her er skåringane for kvar individuelle variabel gruppert basert på kva kjerneelement dei tilhøyrer, og eit gjennomsnitt vart teken av kvart kjerneelement. Det same vart gjort med programmeringsvariablane PR 1, 2 og 3.

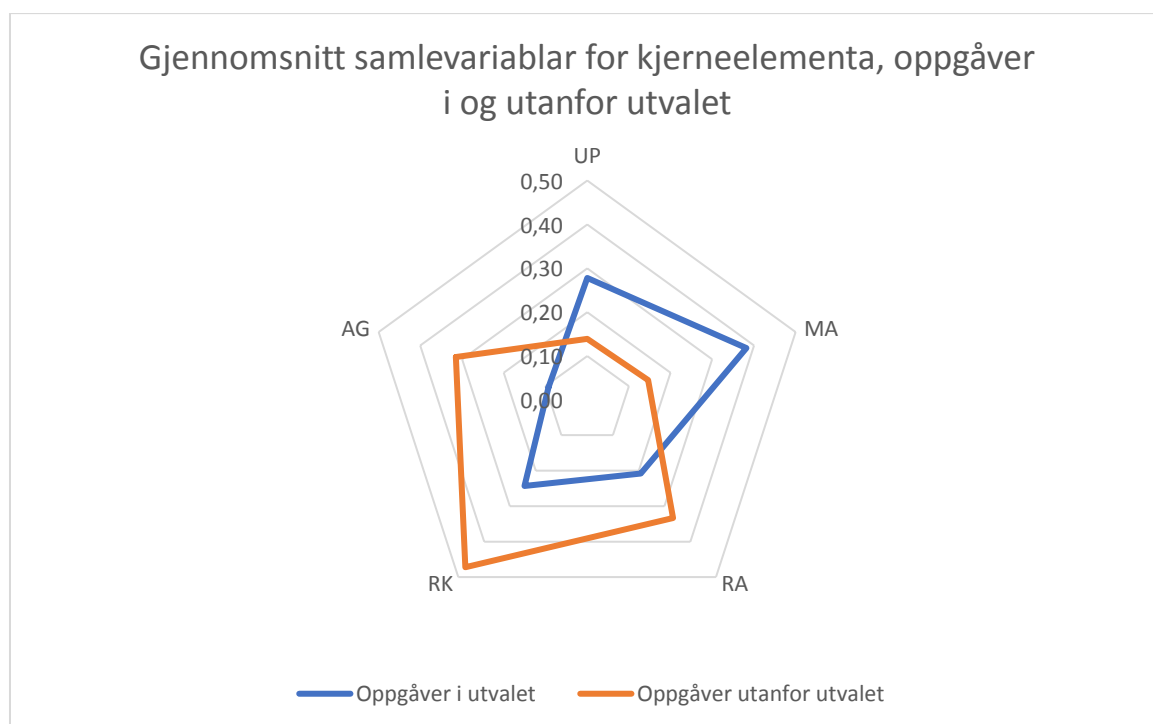
Figur 15 Gjennomsnitt for kjerneelementa og programmering, for oppgåver i utvalet



Her kjem det tydeleg fram at programmering har eit vesentleg høgare gjennomsnitt enn dei fem kjerneelementa. Dette er kanskje ikkje unaturleg når utvalet består av oppgåver som er kategoriserte som programmeringsoppgåver. Dette indikerer at oppgåvene som er inkluderte i utvalet i studien i, stor grad nyttast til å lære å lese/tolke eller kode programmeringskonsept som løkker, kontrollstrukturar osv. Det ser ut til at oppgåvene i mindre grad er tenkt å hjelpe elevane med utvikling av matematiske kompetansar slik dei er skildra i kjerneelementa.

For å gje ein kontekst til analysen av oppgåvene i utvalet, vart det i studien også analysert tilsvarende mange oppgåver (36) som ikkje vart tekne med i utvalet. Det vart plukka ut 12 tilfeldige oppgåver frå kvart læreverk. Oppgåvene vart plukka frå dei resterande oppgåve i kapitla om sannsyrrekning i dei 3 læreverka for 9. trinn og analyserte ved hjelp av det same rammeverket som oppgåvene i utvalet vart. Oppgåvene som vart tilfeldig valt ut, vert vidare i oppgåva vist til som «oppgåver utanfor utvalet». Ingen av oppgåvene utanfor utvalet skåra 1 eller 2 i programmeringsvariablane, og Figur 16 viser ei samanlikning av oppgåver i og utanfor utvalet utan å ta med variablane for programmering (PR). For å auke lesbarheita til figuren er aksene kutta, den har i praksis ein maksimal verdi på 2.

Figur 16 Gjennomsnitt samlevvariablar for kjerneelementa, oppgåver i og utanfor utvalet



Det generelle bildet for alle læreverka samla, er at dei oppgåvene som er kategoriserte som programmeringsoppgåver, skårar jamt over betre på *utforsking og problemløysing* (UP) og *modellering og anvendingar* (MA), men lågare på dei 3 andre kjerneelementa som var representert i studien.

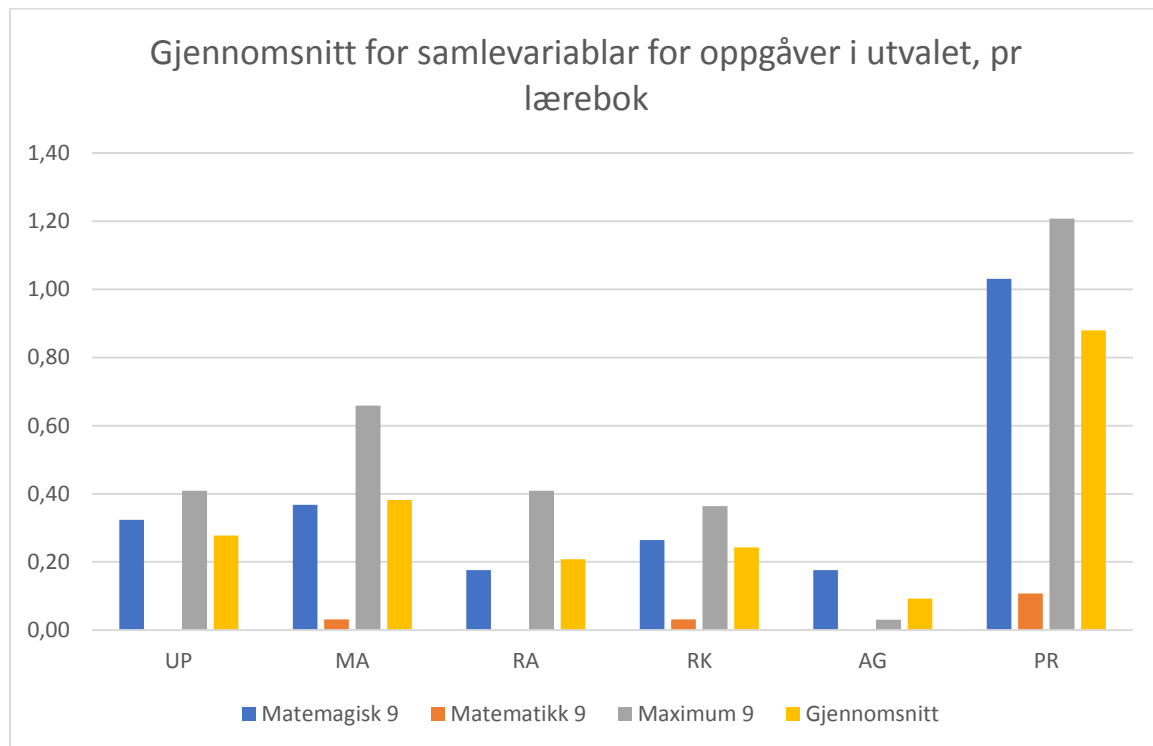
Sentralmålet *gjennomsnitt* er nytta for å seie noko om potensialet oppgåvene har for å dekke variabelen gjennomsnittsverdien er gitt for. Slik kategoriane for å skåre variablane er lagt opp, er det ikkje gitt at ein gjennomsnittskåre  $> 0$  gjer at eleven nyttar kompetansen innan ein variabel. Ei oppgåve med variablar som alle er skåra med kategorien 1, og der eleven *kan* vise kompetanse, får same gjennomsnittskåre som ei oppgåve der annankvar variabel er skåra med kategorien 2, der oppgåva *krav* at eleven syner kompetanse innanfor variabelen. Oppgåvene er svært ulike, sjølv med same gjennomsnitt.

Difor vert gjennomsnittstala kombinerte med ein annan metode for å syne resultat, og gje eit betre bilete av datamaterialet. I kap. 4.2.2 vert det synleggjort kor mange oppgåver som skårar 0, 1 og 2 i kvar variabel. Dette gir eit betre bilete av kva det vert kravd at eleven skal svare på, og kva eleven kan få moglegheit til å vise.

Figur 17 gir ei oversikt over korleis gjennomsnittskåren var pr. lærebok for oppgåver i utvalet, med totalgjennomsnittet som referanse. Maximum 9 skårar høgast på 5 av 6 variablar, inkludert programmering, og hevar totalgjennomsnittet i studien. Matematikk 9-lærebøkene

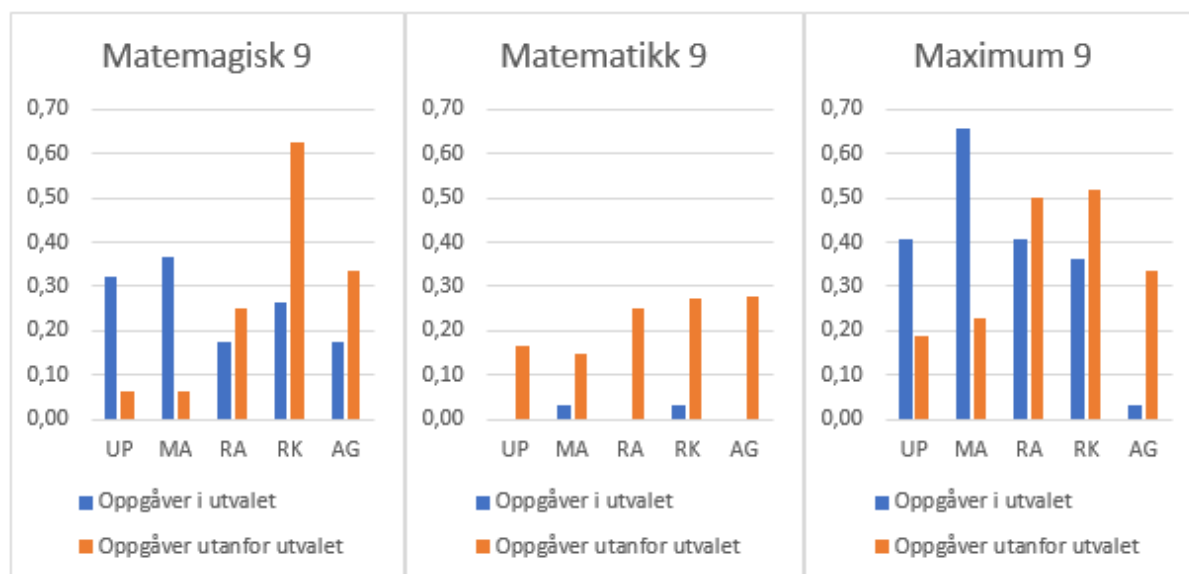
skårar lågt på alle variablane og dermed senkar totalgjennomsnittet. Matemagisk 9 ligg litt over gjennomsnittet.

Figur 17 Gjennomsnitt for samlevariablar for oppgåver i utvalet, pr lærebok



Utvalet inneheld 12 % av den totale oppgåvemengda i kapitla for sannsynsrekning. Ein samanlikning av gjennomsnittet for variablar knytt til kjerneelementa i oppgåver i og utanfor utvalet vil belyse korleis kvar enkelt lærebok skil mellom oppgåvene i arbeid med kompetansane. Figur 18 syner ei oversikt for kvart enkelt læreverk og utdjupar innhaldet i Figur 16.

Figur 18 Gjennomsnitt variablar for kjerneelement, i og utanfor utvalet, pr. lærebok

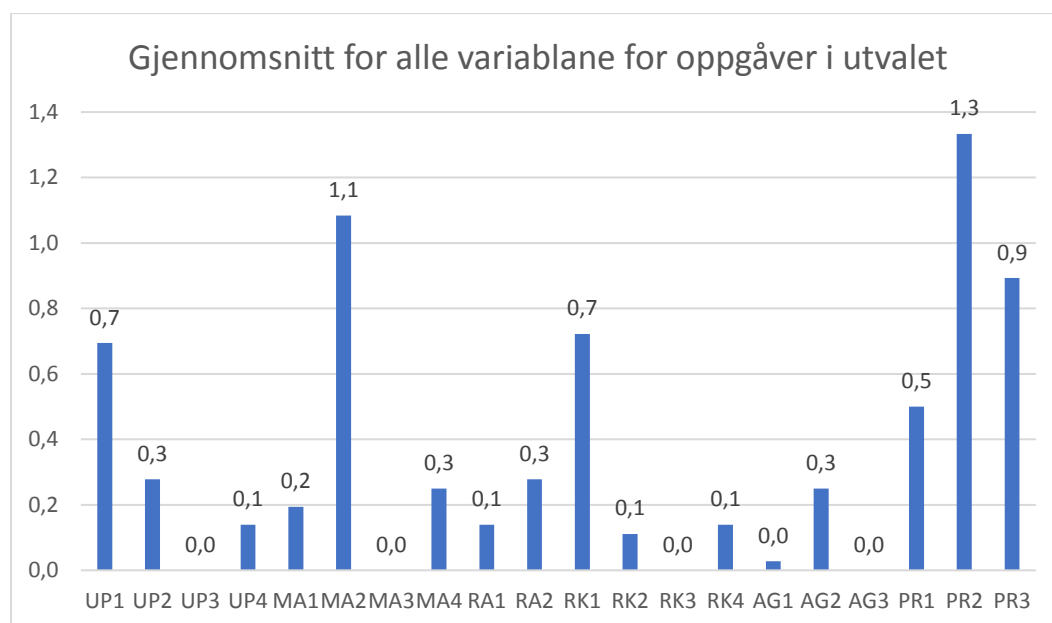


Spesielt Matemagisk 9 bidreg til å gje store utslag på differansen mellom oppgaver i og utanfor utvalet for kjerneelementa *utforsking og problemløysing* (UP) og *modellering og anvendingar* (MA). Gjennomsnittet av oppgåvene som er utanfor utvalet i Matematikk 9-lærebøkene er det lågaste av dei tre læreverka, med Matemagisk 9 litt over og Maximum 9 øvst. Gjennomsnittet av oppgåvene utanfor utvalet er vesentleg likare mellom læreverka enn gjennomsnittet av oppgåvene i utvalet, og Matematikk 9-læreverket det læreverket med jamnast fordeling av skåring av kjerneelementa. Til tross for låg gjennomsnittskår i oppgaver i utvalet vert det i Matematikk 9-lærebøkene likevel gitt oppgaver som fell innanfor *utforsking og problemløysing* (UP) og *modellering og anvendingar* (MA). Dette syner at sjølv utan oppgaver som fell inn under studiens kategorisering av programmeringsoppgaver, kan det arbeidast med kompetansar knytt til kjerneelementa UP og MA.

#### 4.2.2 Resultat knytt til dei individuelle variablane

Figur 19 gir ei oversikt over gjennomsnittskåren for kvar variabel for oppgaver i utvalet. Variabelen *programmeringskonsept* (PR 3) er eit gjennomsnitt av variablane PR 3.1-3.7, som dekkjer ulike konsept innan programmering. Variablane PR 3.1-3.7 får meir fokus i kap. 4.2.4.

Figur 19 Gjennomsnittsskåre for alle variablar for oppgåver i utvalet



Innanfor programmering er tydeleg at det er langt fleire oppgåver som etterspør koding frå eleven (PR 2) enn oppgåver som krev at eleven kan lese eller tolke kode (PR 1). Nokre oppgåver inneheld krav om å både lese/tolke programmering og koding, t.d. ved at eleven skal endre på koda han har lest. Desse oppgåvene fekk då skåren 2 i både PR 1 og PR 2.

Det er fire variablar som aldri vert vurderte til kategori 1 (eleven har moglegheit til å vise litt av innhaldet i variabelen) eller 2 (oppgåva krev at eleven nyttar seg av innhaldet i variabelen). Desse fire variablane er: *vurdere bruk av digitale hjelpemiddel eller ikkje* (UP 3), *kritisk vurdere ein modell* (MA 3), *grunnkje valet av representasjon* (RK 3) og *utforske tal, utrekningar og figurar og formaliser ved å bruke algebra og formålstenlege representasjonar* (AG 3). Variabelen *nytte formelt symbolspråk eller formelle resonnement* (AG 1) vart skåra i kategori 1 i ei oppgåve, men står oppført med 0,0 på grunn av avrundinga av skåren. Dette betyr at *ingen* oppgåver i utvalet krev at eleven nyttar nokon av desse variablane, og at det for dei fire variablane som fekk ein rein 0,0 heller ikkje vart vurdert slik at det var moglegheit for å vise kompetanse knytt til desse variablane.

For variabelen *kritisk vurdere ein modell* (MA 3) vil det i praksis ikkje vere mogleg å få ein skåre på 1 eller 2 med utvalet oppgåver studien nyttar. Slik det kom fram i Figur 14, har alle oppgåvene i utvalet eksplisitt nemnt at løysinga skal programmerast og eleven kan ikkje sjølv vurdere bruk av digitale hjelpemiddel eller ikkje. I nokre få oppgåver kan eleven sjølv velje mellom programmering og rekneark, men heller ikkje desse er vurderte slik at eleven kan velje mellom å nytte digitale hjelpemiddel *eller ikkje*. Figur 19 viser også meir detaljert kvifor



samlevariabelen til kjerneelementet *abstraksjon og generalisering* (AG) fekk så lav gjennomsnittsskåre: to av tre variablar knytt til AG fekk 0,0 i gjennomsnitt. Eksempeloppgåva i Figur 20 viser ei oppgåve som vart klassifisert som kategori 0 på m.a. dei fem omtale variablane; UP3, MA 3, RK 3 AG 3 og AG 1.

#### 4.2.2.1 Eksempeloppgåver og analyse knytt til nokre få variablar

Figur 20 Eksempeloppgåve

Oppg. 2.52 i Maximum 9:

Samarbeid to og to eller tre og tre. Lag eit dataprogram som kan gjere ei trekning. Den som bruker programmet, skal oppgi totalt antal tal det skal trekkast frå, kor mange tal som skal trekkast, og om trekkinga er med eller utan tilbakelegging. Spelet Lotto frå Norsk Tipping skjer utan tilbakelegging. Bruk dataprogrammet de har laga, til å trekke ut ei lottorekke.

Figur laga av forfattern som ei attgjeving av oppgåve 2.52 frå læreboka til Tofteberg et al. (2021, s. 120).

I oppgåva i Figur 20 vert det spesifisert at oppgåva skal løysast med eit dataprogram, og eleven får dermed ikkje sjølv avgjere om oppgåva skal løysast ved hjelp av digitale hjelpemiddel (UP 3). Oppgåva hadde som føremål å lage ein modell av eit tal-trekkjesystem og vart som dei fleste andre oppgåvene i studien skåra i kategori 2 for variabelen *matematisere – lage ein modell* (MA 2). Men i oppgåva vart det ikkje eksplisitt bedt om at elevane skulle vere kritiske til modellen dei laga, så dermed kunne ikkje oppgåva få skåren 2 for variabelen *Kritisk vurdere ein modell* (MA 3). Med dei føringane som er lagde til grunn for kategorien 1 og for variabelen MA 3, vert det i studien vurdert at denne oppgåva heller ikkje kvalifiserer til skåren 1.

Det er ikkje alltid lett å vurdere om elevane har høve til, i dette tilfelle, å kritisk vurdere ein modell, og dermed kunne skåre variabelen med kategori 1. I utvalet med programmeringsoppgåver er omgrepet «modellen» tolka som den matematiske modellen, medan dataprogrammet er modellen omsett til kode. Å feilsøke kode er i studien ikkje det same som å kritisk vurdere ein modell. Kritisk vurdering av ein modell er i studien tolka som den modellen som kan eksistere utanfor sjølve koda, og kritikken kan t.d. sjå på gyldigheitsområdet til modellen eller matematiseringsprosessen i overgangen frå eit ekstra-matematisk domene til eit matematisk domene (Niss & Blum, 2020).

Oppgåva i Figur 20 inneheld ikkje krav om representasjonar som elevane kan grunngje valet av, så variabelen *grunngje valet av representasjonar* (RK 3) kan ikkje skårast med kategorien 2. Det er i utgangspunktet heller ikkje ei tydeleg moglegheit for elevane å nytte seg av denne variabelen i denne oppgåva, då vert heller ikkje RK 3 skåra med kategorien 1. Det var heller ikkje tal å utforske eller noko å formalisere med algebra, så variabelen *Utforske tal, utrekningar og figurar og formaliser ved å bruke algebra og formålstenlege representasjon* (AG 3), vart skåra med kategorien 0. I eksempelet vert det ikkje nytta formelt symbolspråk, så variabelen *nytte formelt symbolspråk eller formelle resonnement* (AG 1) vert også sett til 0.

Dei tre variablane, sett vekk frå programmeringsvariablane (PR), som har fått høgast gjennomsnittskåre er *avdekke mønster og samanhengar* (UP 1), *matematisere – lage ein modell* (MA 2) og *bruke matematisk språk* (RK 1). MA 2 er den variabelen med desidert høgast snittskåre av desse 3: 1,1. Dei fleste dataprogramma som er i lærebøkene er nettopp modellar av røynda. Eksempel frå lærebøkene er m.a. terningkast og lykkehjul. Når eleven då lagar eit program sjølv, lagar også eleven ein modell og MA 2 vert skåra med kategorien 2. UP 1 er ein svært open variabel, og det var knytt ei utfordring til å skilje samanhengar i UP 1 frå matematiske samanhengar i variabel AG 2, *oppdage matematiske samanhengar og generalisere*. Skiljet i studien går i stor grad på at AG 2 også handlar om generalisering, medan UP 1 handlar om utforsking. I arbeidet med å utvikle kjerneelementa vart programmering plassert under problemløysing og utforsking (Nordbakke, 2018, s. 38), så det er ikkje overraskande at ein variabel innan *utforsking og problemløysing* (UP) får ein god skåre. Det er derimot meir overraskande at det samla gjennomsnittet for UP ikkje var høgare når studien ser på programmeringsoppgåver. Figur 21 er ei oppgåve som syner korleis desse tre variablane kjem til syne, henta får ei lærebok nytta i studien.

Oppg. 13.17: Vi trekkjer eitt kort frå ein vanleg kortstokk.

- a) Kva er sannsynet for å trekkje ein konge
- b) Lag eit program som simulerer at vi trekkjer eitt kort frå ein kortstokk 100 gonger, og tel kor mange gonger du får konge. Programmet skal deretter rekne ut den relative frekvensen for å trekkje konge
- c) La programmet simulere at vi trekkjer eitt kort frå ein kortstokk 10 000 gonger. Kva blir den relative frekvensen for å trekkje konge no?
- d) Korleis stemmer sannsynet frå oppgåve a med dei relative frekvensane du fekk i oppgåve b og c?

Tilleggsopplysning i boka: Det er 13 moglege «tal» i ein kortstokk.

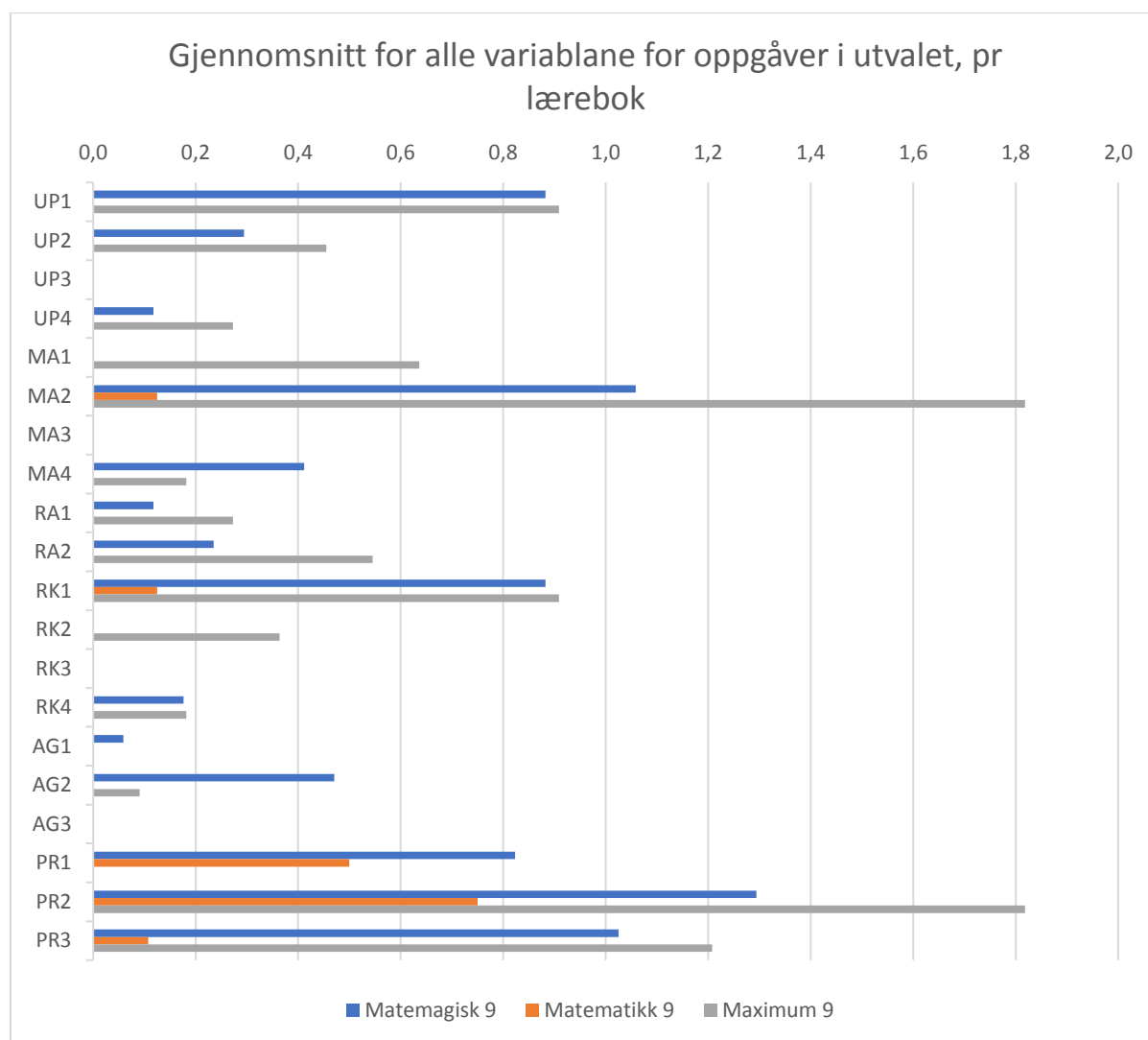
Figur laga av forfattaren som ei attgjeving av oppgåve 13.17 frå læreboka til Kongsnes og Wallace (2020b, s. 98)

Eleven må i denne oppgåva både rekne ut sannsyn og relativ frekvens, og viser at ho kan nytte seg av eit matematisk språk både ved å forstå oppgåveteksten med aktuell matematikkfagleg terminologi, nytte seg av innhaldet i omgrepa og formulere ei vurdering (RK 1). Oppgåva leier eleven til å sjå ein samanheng mellom simulering med store (nok) tal og sannsynet for eit utfall (UP 1). Eleven må i oppgåva lage ein modell av korttrekking frå ein vanleg kortstokk (MA 2)

#### 4.2.2.2 Samanlikning av lærebøkene

Figur 22 syner ei oversikt over korleis dei forskjellige læreverka skårar på dei individuelle variablane for oppgåver i utvalet. Det er til tider svært store differansar mellom lærebøkene. For samlevariabelen til programmeringskonsept (PR 3) er det overraskande stor forskjell på Matematikk 9-læreverket og dei to andre læreverka, med tanke på at det er oppgåver kategoriserte som programmeringsoppgåver, som vert analyserte. I analysen av Matematikk 9-bøkene vart det notert at det i oppgåvene i utvalet vert lagt opp til at dei skal løysast med blyant. Ein slik løysingsmetode kan indikere at Matematikk 9-læreverket i større grad legg opp til at eleven skal nytte pseudokode, men dette kjem ikkje tydeleg til uttrykk; nokre oppgåver vil ha elevane til å skrive svaret på ein svar-strek, andre i ein svarboks.

Figur 22 Gjennomsnitt for alle variablane for oppgåver i utvalet, pr lærebok



Eit punkt som er verdt å merke seg er at Maximum 9-boka ikkje har oppgåver i utvalet som skal lesast/tolkast med tanke på programmering (PR 1) medan dei aller fleste oppgåvene skal kodast (PR 2). I analysen av oppgåvene til læreboka vart det notert at det var lite kode å tolke, sjølv om det var oppgitt kode som skulle skrivast inn i eit Python-program. I mogleg forklaring er at Maximum 9-læreverket legg opp til at bygging av eigne kodeferdigheiter også skal gje eleven moglegheit til å tolke andre si kode.

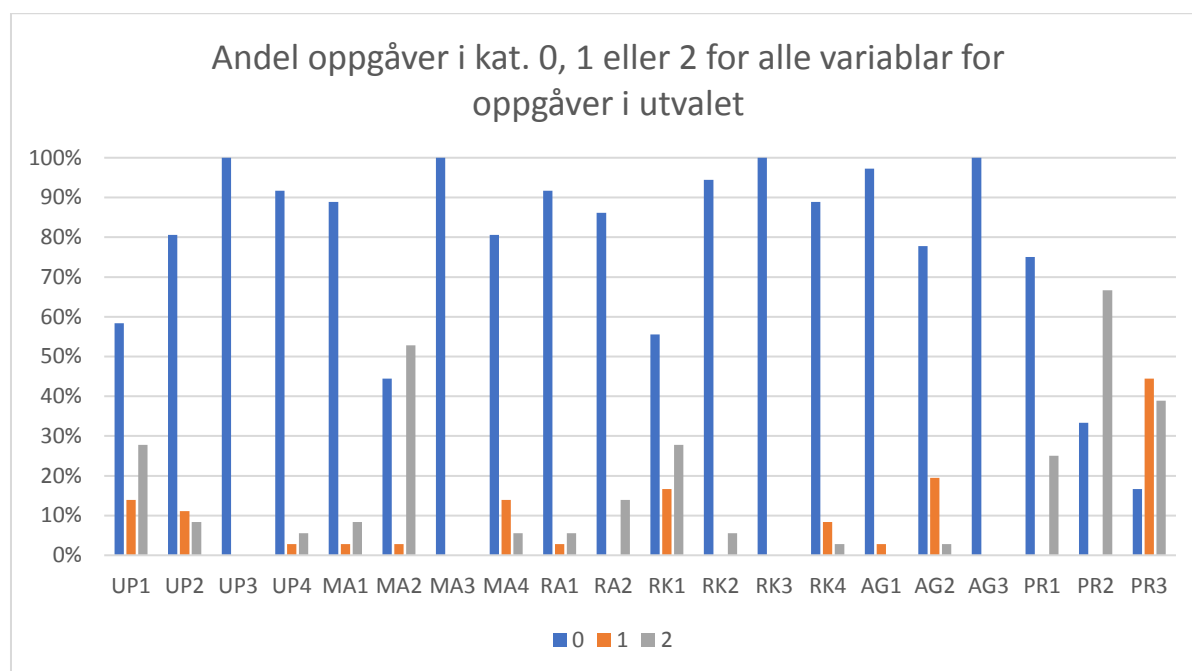
#### 4.2.2.3 Antal oppgåver i kat 0,1,2, for alle variablar

Figur 23 viser kor mange oppgåver som vart skåra med 0, 1 eller 2 innanfor dei forskjellige variablane i utvalet. For variabelen PR 3 vart det gjort ei tilpassing for dette diagrammet. Variabelen er ein gjennomsnittsskåre av dei individuelle PR 3-variablane, og dette fører til at PR 3-variabelen skjeldan treff akkurat på 0, 1 eller 2. Variabelen PR 3 vert plassert i kategori 0 når den har ein snittsskåre på 0, i kategori 1 når den har ein snittsskåre i intervallet  $<0,1]$  og i

kategori 2 når dei har ein snittskåre  $>1$ . Dette medfører at statistikken for PR 3 kan framstå ubalansert. T.d. er det i røynda berre 3 oppgåver i intervallet  $<1.43,2]$ . Men vert intervallet utvida med ein tiandedels desimal nedover er det 14 oppgåver som vert inkluderte, så det er tydeleg at det ligg mange oppgåver med ein gjennomsnittskåre på 1,42-1,43. I oppgåva vert likevel PR 3 presentert på er på ein slik måte, for å få fram kor mange oppgåver som fekk ein gjennomsnittskåre på 0, og som ikkje inneheldt noko programmeringskonsept.

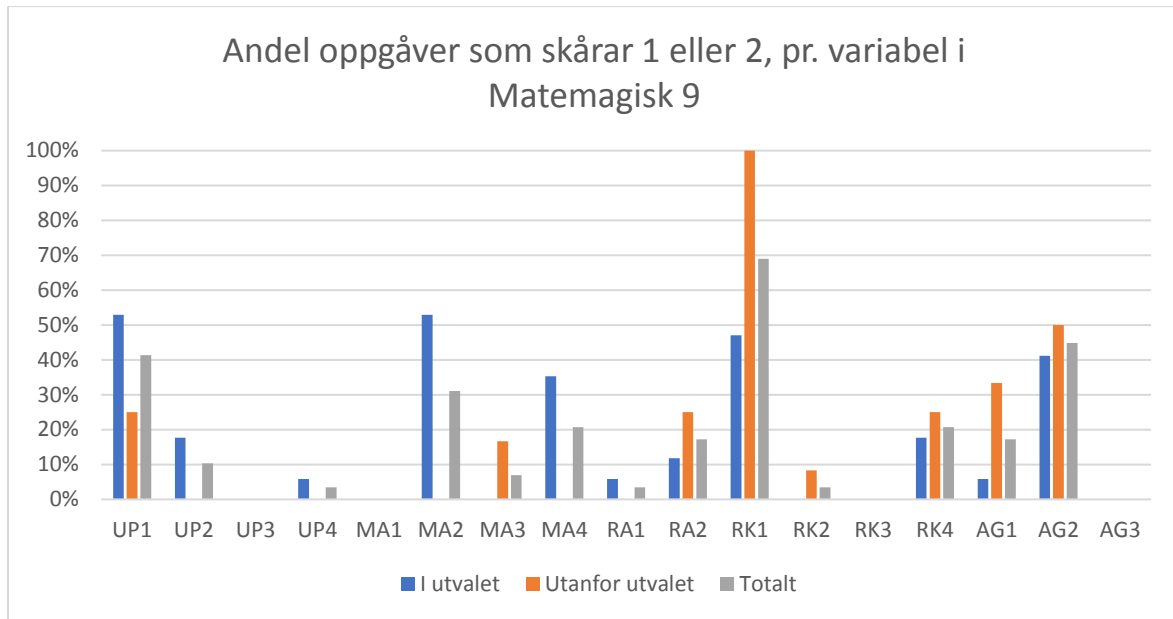
For variablane PR 1 og PR 2 stemmer tala i tabellen, men dersom andelen oppgåver som har fått skåren 2, vert lagt saman, gir det eit litt urett bilde. Oppgåver kan få skåren 2 i både PR 1 og PR2, og dette skjeddde i eit tilfelle i utvalet. Så antal oppgåver som fekk skåren 0 for begge variablane er 4, ikkje 3 slik det kan framstå av figuren.

Figur 23 Andel oppgåver i kat. 0, 1 eller 2 for alle variablar for oppgåver i utvalet

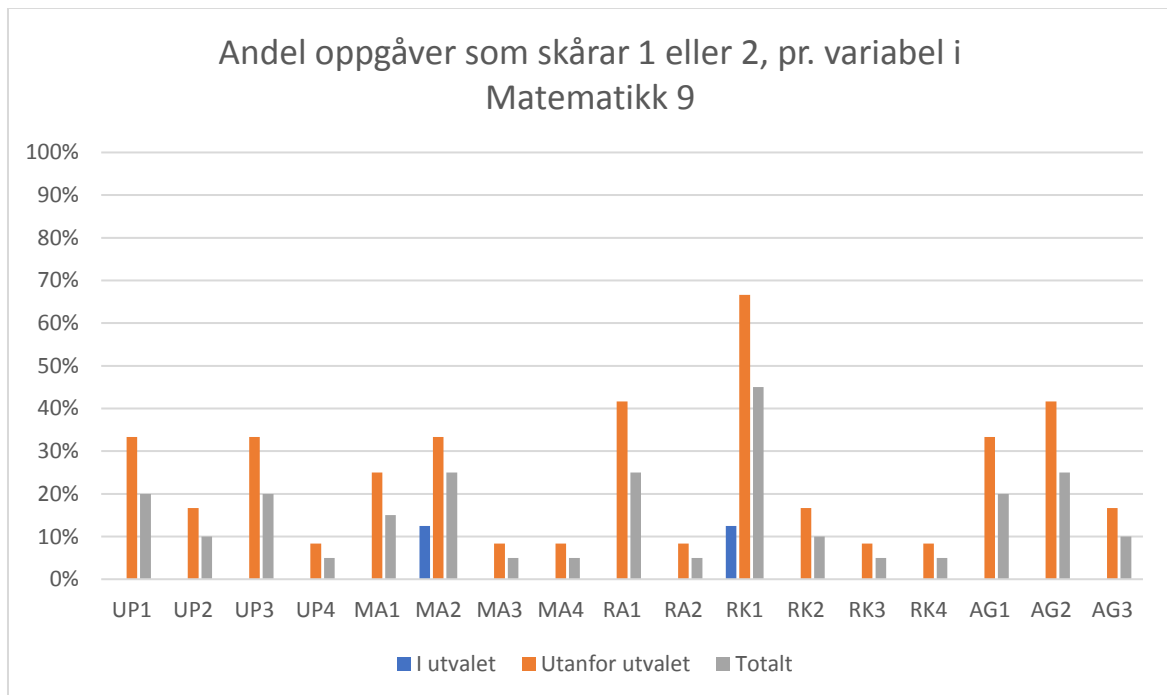


Medan Figur 23 kan kaste lys over kva variablar dei 3 læreverka samla vektlegg for oppgåver som er i utvalet, syner Figur 24, Figur 25 og Figur 26 ei oppdeling for kvar lærebok og inkluderer oppgåver utanfor utvalet for samanlikning. Oppgåver som er skåra med 0 er teken vekk frå figuren for å auke lesbarheita. Figurane syner oppgåver som har skåra 1 eller 2 innanfor kvar variabel knytt til kjerneelementa, og om oppgåva er i utvalet eller utanfor utvalet.

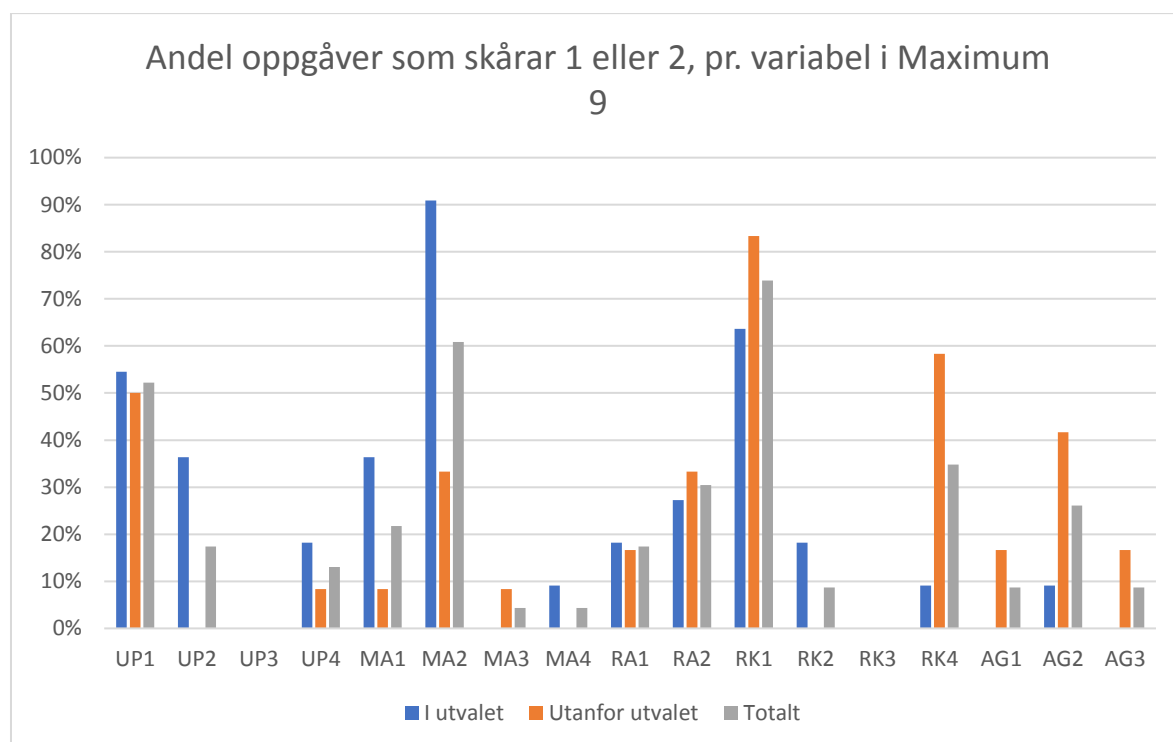
Figur 24 Andel oppg ver som sk r 1 eller 2, pr. variabel i Matemagisk 9



Figur 25 Andel oppg ver som sk r 1 eller 2 pr. variabel i Matematikk 9



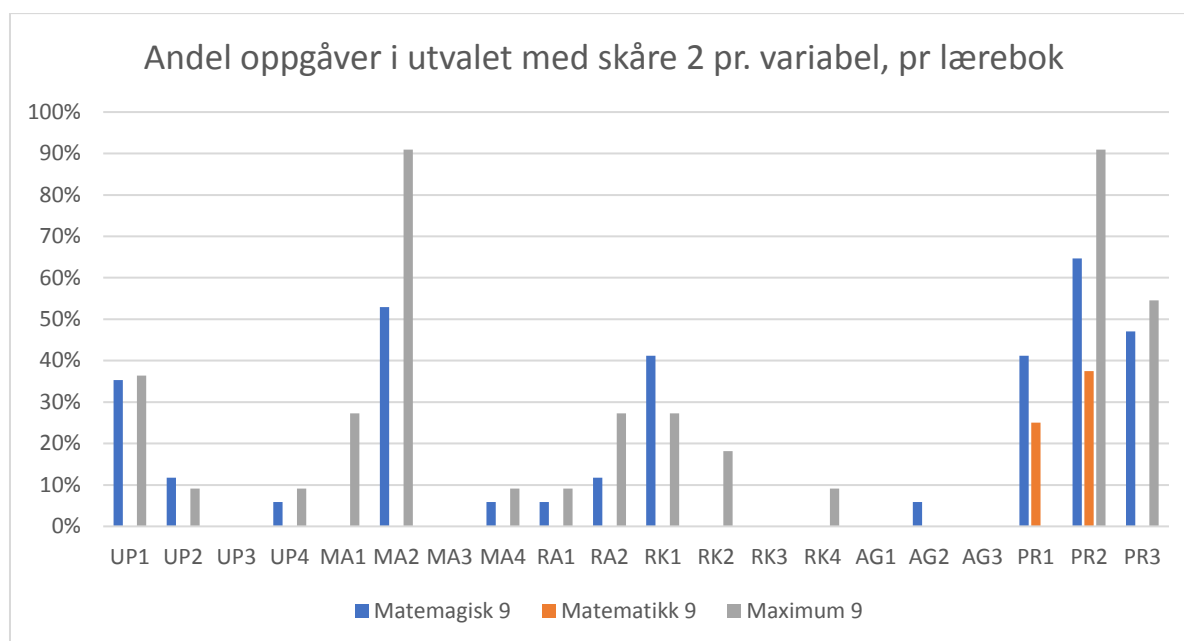
Figur 26 Andel oppg ver som sk r 1 eller 2 pr. variabel i Maximum 9



Det er knytt usikkerheit til kor godt analyseresultata fr  oppg vene utanfor utvalet, kan generaliserast. Det var ein liten andel oppg ver som vart analyserte, og resultatata b r vurderast som indikasjonar. Matematikk 9 er det einaste l reverket som sk r 1 eller 2 p  alle variablane, medan dei to andre l reverka har nokre variablar dei utelukkande sk r 0 p . Matemagisk 9-l reboka har ikkje berre fleire oppg ver i utvalet som sk r 1 eller 2 enn utanfor utvalet, men oppg vene i utvalet dekkjer ogs  fleire variablar enn oppg vene utanfor utvalet. Matemagisk 9-l reboka har ogs  fleire oppg ver fr  utvalet representerte i variablane *representasjon og kommunikasjon*(RK) og *abstraksjon og generalisering* (AG) enn Figur 18 skulle tilsei. Det er verdt   merke seg at Matemagisk 9-l reboka har fleire oppg ver i utvalet enn utanfor utvalet i studien. Maximum 9-l reboka har ca. like mengde oppg ver som fekk sk ren 1 eller 2 i utvalet, som utanfor utvalet.

Ei samansmelting av oppg vene som gir sk re 1 eller 2 er ikkje n dvendigvis hensiktsmessig, og gir i praksis same utfordringa som med gjennomsnitt; det er ikkje gitt at ein elev f r trening i ein kompetanse dersom ikkje oppg va krev det. Figur 27 syner ei oversikt over variablar som har f tt sk ren 2 p  oppg ver i utvalet, og knyter desse tala til l reverka.

Figur 27 Andel oppgaver i utvalet med skåre 2 pr. variabel, pr lærebok



Igjen så er det variablane *avdekke mønster og samanhengar* (UP 1), *matematisere - lage ein modell* (MA 2) og *bruke matematisk språk* (RK 1), saman med samlevvariablar for programmering, som skil seg ut. Oppgåvene frå både Matemagisk 9 og Maximum 9-lærebøkene krev at eleven syner kompetanse innanfor dei fleste variablar. Matemagisk 9 kjem ut med den høgaste andelen oppgåver i denne figuren, i hovudsak frå variabelen *lese/tolke koding* (PR 1).

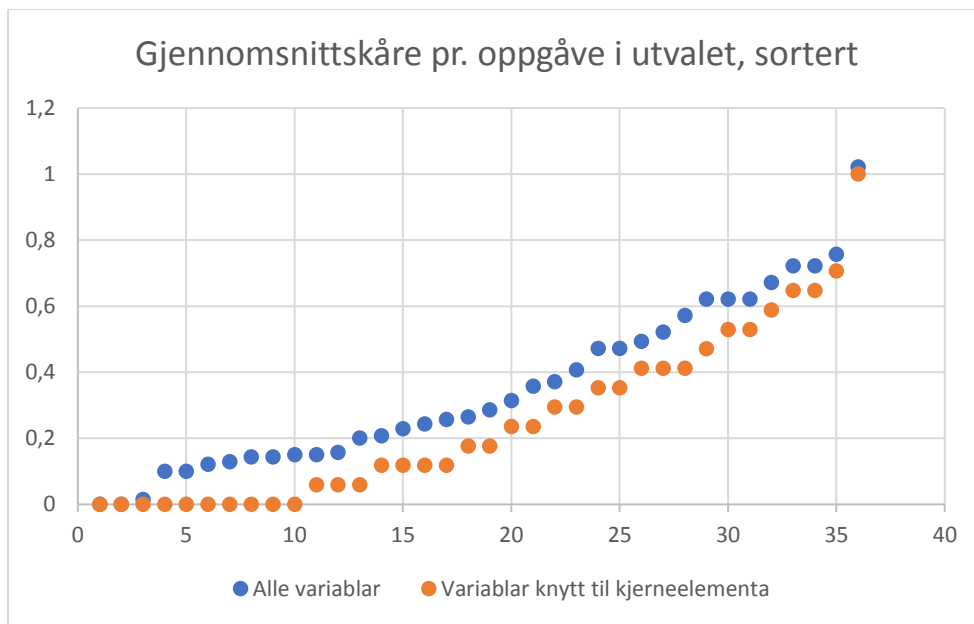
#### 4.2.3 Resultat knytt til individuelle oppgåver

Så langt har resultatata fokusert pr. kjerneelement eller pr. variabel, men det er også mogleg å hente ut resultat pr. oppgåve. Figur 28 viser ei oversikt over gjennomsnittskåren til kvar enkelt oppgåve i utvalet, både med og utan programmering inkludert. Desse blir presenterte i stigande rekkefølge basert på gjennomsnitt med programmeringsvariablane inkluderte. Kvart punktpar på same plass på førsteaksen tilhøyrrer same oppgåve. Målet med framstillinga er å synleggjere om det er oppgåver jamt representerte mellom dei med låg gjennomsnittskåre og dei med høg gjennomsnittskåre, og om programmeringsvariablane hevar gjennomsnitta på oppgåvene.

Som det det går fram av oversikta, er antal variablar som er inkluderte i oppgåvene varierte, og relativt jamt fordelte mellom oppgåvene med færrest og oppgåvene med flest variablar inkludert.



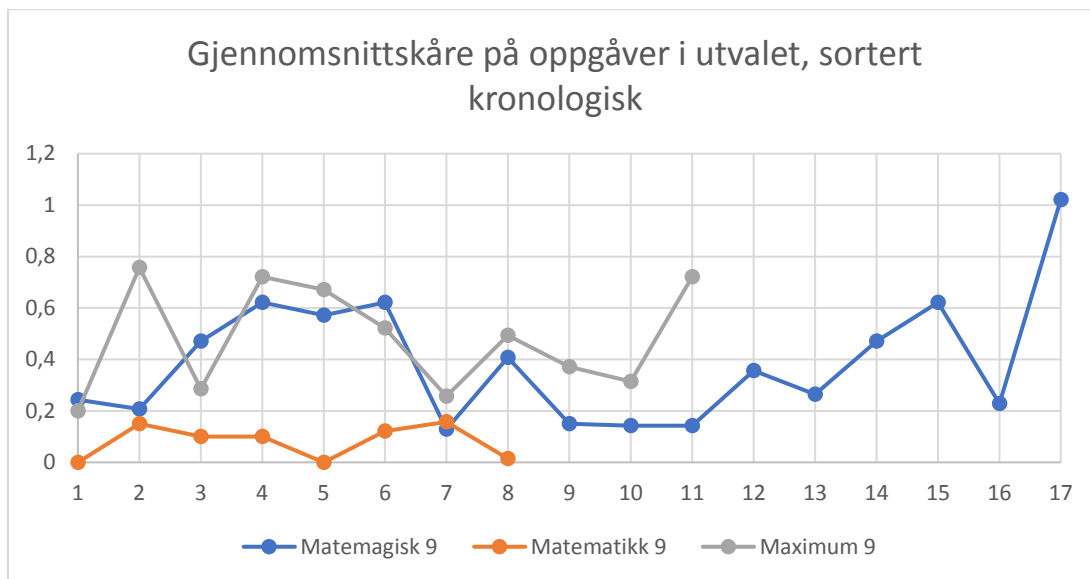
Figur 28 Gjennomsnittskåre pr. oppgåve i utvalet, sortert



I grafen over gjennomsnittskåre pr oppgåve i utvalet, har dei 10 oppgåvene med lågast gjennomsnitt ingen poeng i studien. Gjennomsnittet sank frå 0,35 til 0,25 når programmeringsvariablane vart fjerna frå gjennomsnittet. Det er spesielt Programmeringsheftet til Matematikk 8-10 (Hjardar, u.å.) som dreg ned gjennomsnittet. 6 av dei 10 oppgåvene som fekk 0 poeng i variablar knytt til kjerneelementa, kjem frå dette heftet. Dette indikerer at heftet har eit fokus på programmeringsopplæring.

Gjennomsnittskåren på oppgåvene i utvalet aukar ikkje utover i kapitla. Figur 29 viser at det i studien ikkje er ein samanheng mellom kor langt inn i kapitelet ei oppgåve er, og gjennomsnittskåren oppgåva får.

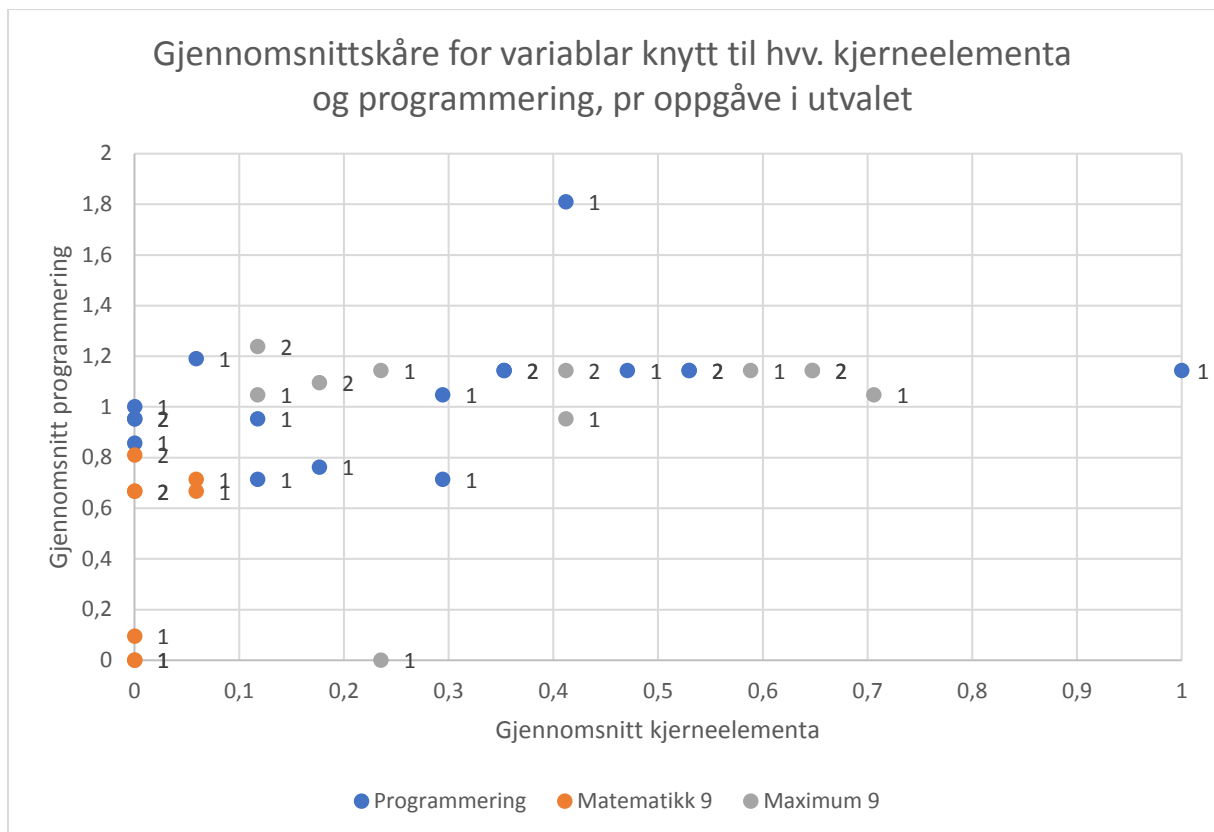
Figur 29 Gjennomsnittskåre på oppgåver i utvalet, sortert kronologisk



Ingen læreverk har ein tydeleg progresjon der oppgåvene inkluderer fleire variablar utover i kapitelet.

Figur 30 viser at gjennomsnitta for variablane knytte til programmering, stort sett ligg samla i området 0,7-1,2 for oppgåver i utvalet. Gjennomsnitta for variablar knytt til programmering viser ikkje tydeleg samvariasjon med verdien på gjennomsnitta av variablane knytte til kjerneelementa, korkje for kvar enkelt læreverk eller for datamaterialet i sin heilskap. Tal knytte til kvart punkt i figuren syner kor mange punkt som ligg oppå kvarande, og figuren gir dermed ikkje eit heilt rett bilde av kvart læreverk.

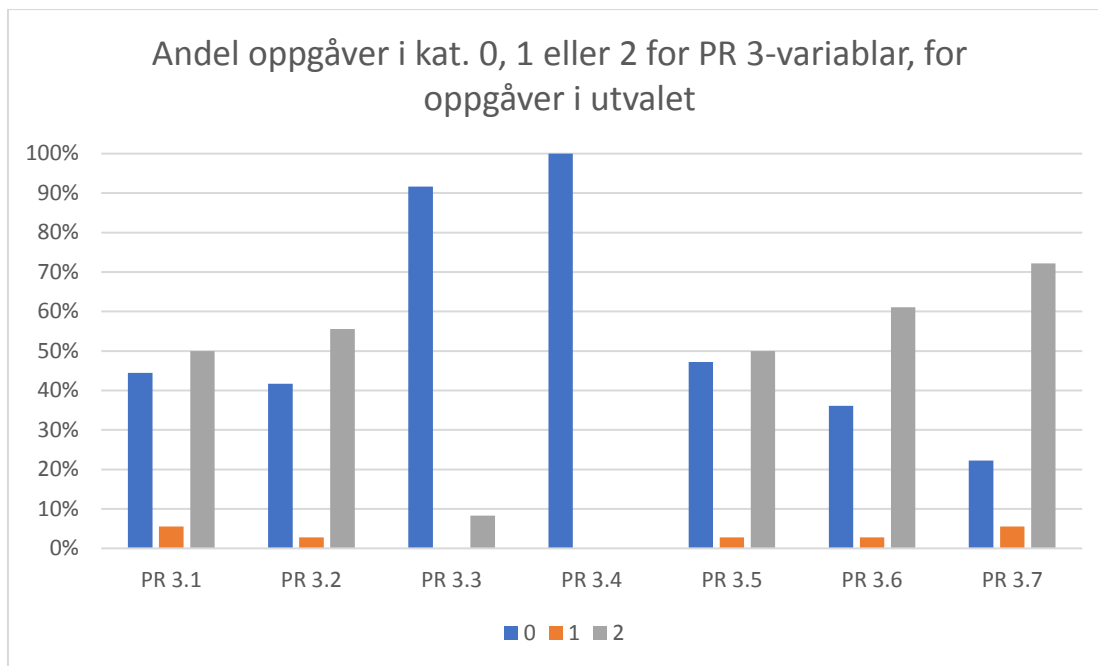
Figur 30 Gjennomsnittskåre for variablar knytt til hvv. kjerneelementa og programmering, pr oppgåve i utvalet



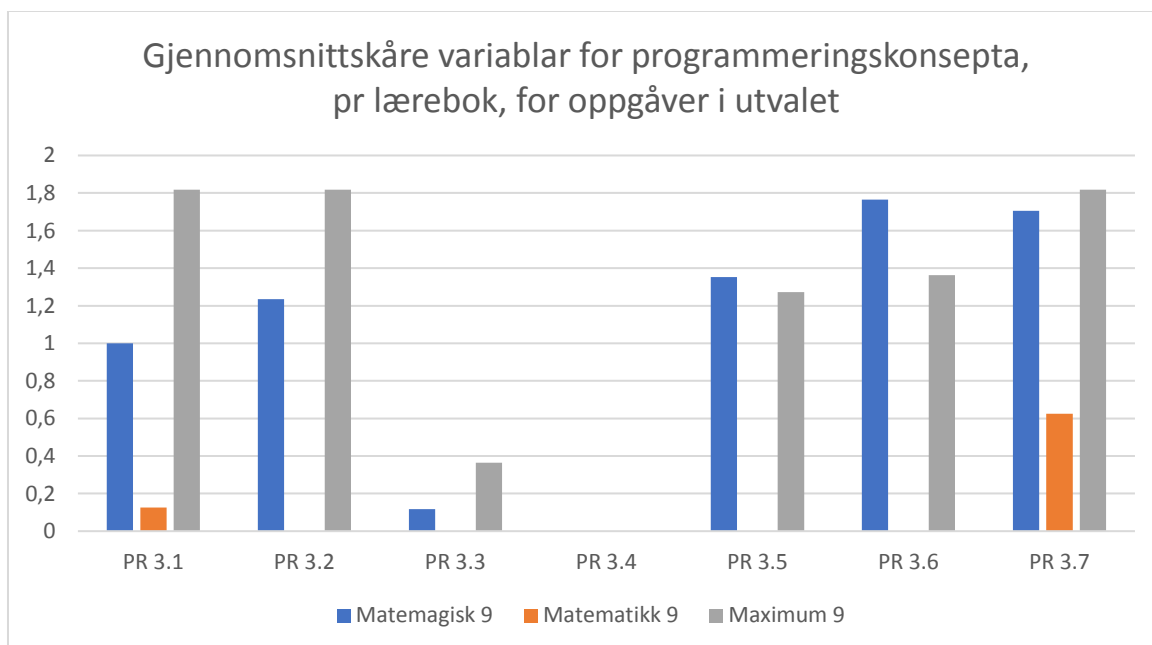
#### 4.2.4 Resultat knytt til programmeringskonsept

Resultat knytt til programmeringskonsept vert ikkje samanlikna med resultata frå analysen av oppgåver utanfor utvalet, fordi alle oppgåvene utanfor utvalet skåra 0 på programmeringsvariablane. Difor har ikkje resultata for programmeringsvariablane ein samanlikningskontekst. Resultata kan likevel nyttast i samanheng med antal oppgåver i kvart læreverk og sjå kor stor andel av oppgåvene kvart programmeringskonsept opptek. Resultata seier også noko om kva programmeringskonsept som er vektlagt for lærebøkene i studien.

Figur 31 Andel oppgaver i kat. 0, 1 eller 2 for PR 3-variablar, for oppgaver i utvalet



Figur 32 Gjennomsnittskåre variablar for programmeringskonsept, pr lærebok, oppgaver i utvalet



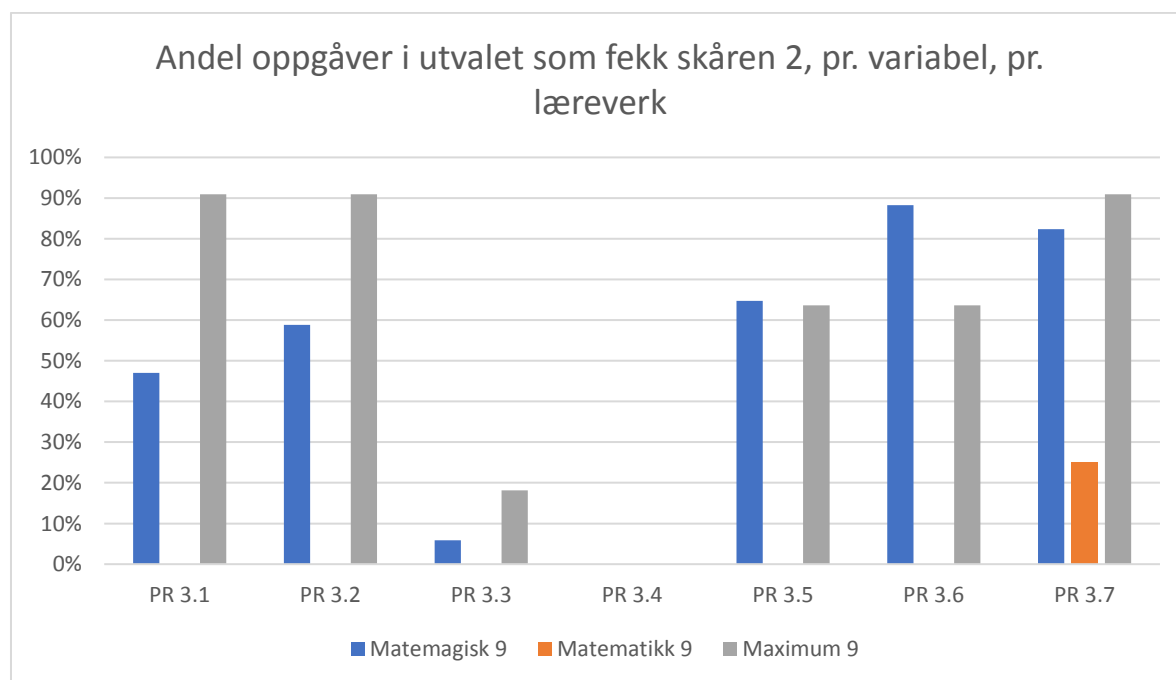
For oppgaver i utvalet er analysen av programmeringskonsept interessant for å belyse korleis lærebøkene inkluderer ferdigheitsbygging i oppgåvene. I Figur 31 og Figur 32 er det ein oversikt over korleis oppgåvene i utvalet skårar i dei forskjellige programmeringskonsept-variablane: *sekvensar/framgangsmåtar* (PR 3.1), *løkker* (PR 3.2), *handlingar* (PR 3.3), *parallellisme* (PR 3.4), *kontrollstrukturar* (PR 3.5), *operatorar* (PR 3.6) og *data* (PR 3.7). For å gje ei realistisk vurdering, er oppgaver tolka og koda slik liknande oppgaver eller

eksempel har vorte løyste i læreboka oppgåva står i, og tolka med tanke på kva type programmering elevane har lært fram til den aktuelle oppgåva. Kva kode som er hensiktsmessig er også nytta som eit ledd i å vurdere kva programmeringskonsept elevane kan eller må nytte. Det er skjeldan eksplisitt forklart i oppgåvene kva for programmeringskonsept som må nyttast i kvar oppgåve, men dersom oppgåva t.d. etterspør at programmet skal telje ei viss mengde tal automatisk, er det vurdert slik at oppgåva krev ei løkke, og oppgåva vert skåra i kategori 2 for løkker (PR 3.2).

Sidan programmering gir rom for mange framgangsmåtar å løyse eit problem på, er det kanskje rart at ikkje kategorien 1 er meir framherskande i figuren. I prosessen med å skåre oppgåvene i utvalet, måtte det gjerast ei vurdering av korleis kategorien 1 skal nyttast. I utgangspunktet kunne kategorien 1 ha vore nytta på nær alle oppgåvene sidan det skjeldan er eksplisitt forklart kva type programmeringskonsept eleven *ikkje* skal nytte. Dette vart løyst ved at det i testprogrammeringa av oppgåvene vart vurdert andre framgangsmåtar og programmeringskonsept som *også* kan brukast, men som ikkje er den *mest* hensiktsmessige måten.

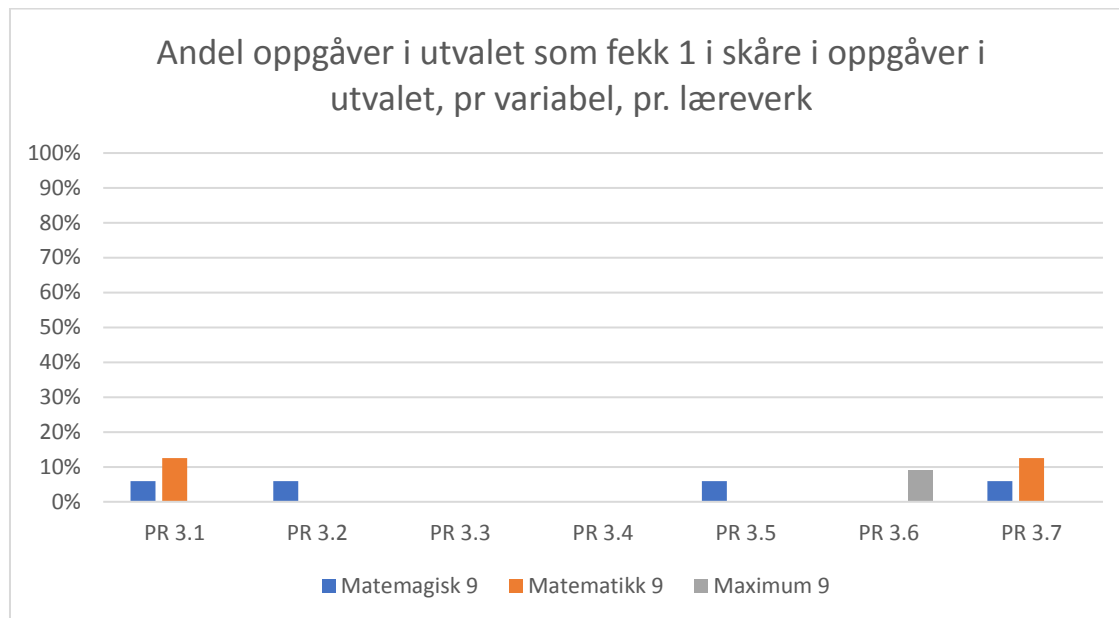
I analysearbeidet vart det stort sett kategorisert med skårane 0 og 2 for variablane knytte til programmeringskonsept. Figur 33 syner derfor variablar som er skåra med verdien 2, for kvart læreverk.

Figur 33 Antal oppgåver i utvalet som fekk skåren 2, pr variabel, pr. læreverk



Som vist i Figur 34, var det i tillegg 8 variablar som fekk 1 i skåre. Figur 34 Figuren har same lengde på andre-aksen som Figur 33 for å vise dei same proporsjonane på søylene.

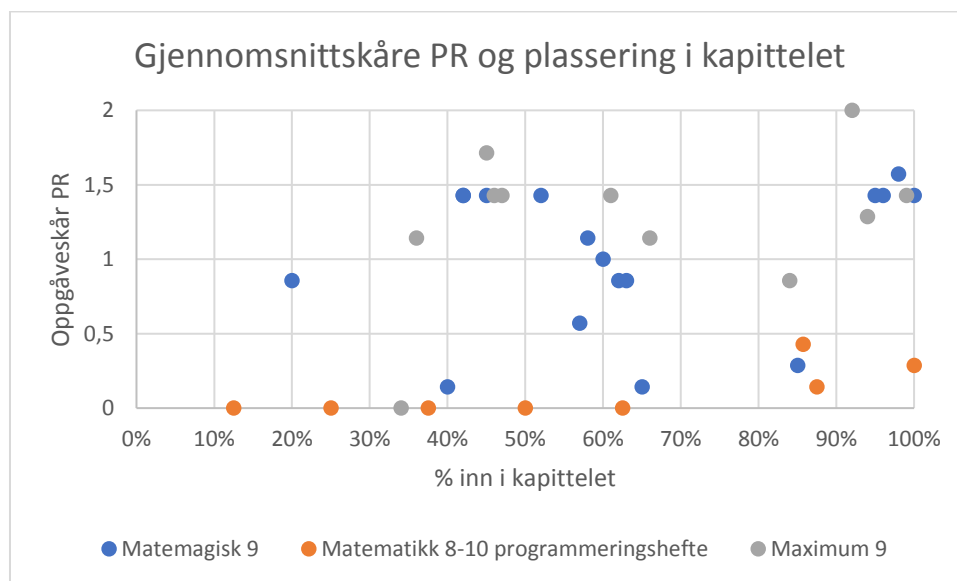
Figur 34 Antal variablar som fekk 1 i skåre i oppgåver i utvalet, pr variabel, pr læreverk



Mangelen på skår 1 eller 2 i variablane *handlingar* (PR 3.3) og *parallelisme* (PR 3.4) er påfallande og vert drøfta nærare i kap. 5.2 Men i korte trekk kan dette kome av at rammeverket for å analysere programmeringskonseptane knytte seg til bruken av Scratch.

I Figur 35 er gjennomsnittsskåren pr. oppgåve for kvar lærebok plassert i samanheng med oppgåveplasseringa i kapittelet.

Figur 35 Gjennomsnittsskåre PR for oppgåver i utvalet ,pr. lærebok og plassering i kapittelet



## 5 Drøfting

### 5.1 Drøfting knytt til resultat

Studien ser på oppgaver kategoriserte som programmeringsoppgåve og korleis desse skårar på variablar henta frå kjerneelementa i matematikk og frå programmeringskonsept knytt til programmeringsferdigheiter. Slik programmeringsoppgåver er definerte i studien, dekkjer dei ca. 12% av oppgåvene som vart vurderte. Alle oppgåvene som vart vurderte låg i eitt av fleire kapittel i lærebøkene. Slik blir prosentandelen oppgåver som vart analyserte i studien vesentleg mindre, sett opp mot det totale antal oppgåver i alle lærebøkene nytta i studien. Eit gjennomsyn av lærebøkene syner at andre kapittel i lærebøkene også har programmeringsoppgåver, baserte på korleis studien har definert programmeringsoppgåver.

Det er altså mange fleire oppgåver i lærebøkene enn det som har vorte analysert og studien kan ikkje sei noko om heilskapen til kvar lærebok. Studien seier derimot noko om oppgåver i kapitla om sannsynsrekning i dei vurderte matematikklæreverka (med tillegg). Desse resultatata, saman med resultatata frå utvalet av oppgåver, kan t.d. nyttast som innspel i vurdering av kva lærebok som passar for kvar einskild skule. Sjå kap. 5.2 for ei drøfting rundt rammeverket.

I nokre resultat knytte til oppgåveanalysane vert det nytta gjennomsnittsverdiar for å oppsummere anten skåringane for ein variabel eller skåringar for variablar som er knytte til eit kjerneelement. Variablane vert skåra innan kategoriane 0, 1 eller 2, med unntak av samlevariabelen knytt til programmering. Ein høg gjennomsnittskåre betyr ikkje at ei oppgåve er bra og ein låg gjennomsnittskåre betyr ikkje at ei oppgåve er dårleg. Rammeverket ser på kor mange variablar som er eksplisitt etterspurt eller har moglegheit for å kome til uttrykk gjennom ei oppgåve. For at ei oppgåve skal oppnå ein gjennomsnittskåre på 2 treng oppgåva å eksplisitt etterspørje alle variablane. Ei slik oppgåve vil nok oppfattast som kompleks og det gjer ikkje nødvendigvis oppgåva «god». Det finst ingen taksonomi i studien som beskriv kva verdiar ein gjennomsnittskåre må ha for å definere ei oppgåve som «god» eller «dårleg». I staden kan gjennomsnittskåren saman med andre resultat gje eit bilete av korleis studien har vurdert utvalet innanfor kvar lærebok. I tillegg vil resultat knytt til heile utvalet av oppgåver seie noko om korleis studien har vurdert oppgåver på tvers av lærebøkene og gi ein peikepinn på kva oppgåver som finst i dei mest nytta lærebøkene.

Drøftingane knytt til resultatata er delte i to: ein generell del og ein del som er knytt til programmeringsvariablane.

### 5.1.1 Drøftingar av hovudfunna

I resultatkapittelet vert resultatata diskuterte og delvis drøfta undervegs. I dette delkapittelet vil følgande hovudfunn drøftast:

- Sett under eitt gir oppgåver i utvalet meir trening på nokre kjerneelement enn på andre
- Dei forskjellige læreverka skil seg frå kvarandre når det kjem til kva kompetansar oppgåver i utvalet inkluderer
- Oppgåver i utvalet har tydeleg ein annan profil på kva kompetansar dei inkluderer enn oppgåver utanfor utvalet h
- Oppgåver i utvalet gir vesentleg meir trening på enkelte delkompetansar innanfor kjerneelementa
- Oppgåver i utvalet er veldig eksplisitte, og det er liten tvil om at oppgåvene skal løysast med programmering

#### 5.1.1.1 Ulikt vekting av kjerneelement i oppgåver i utvalet

Studien finn at oppgåver i utvalet på tvers av alle læreverka, ikkje gir like moglegheiter til å trene på kompetansane i kjerneelementa (Figur 15 og Figur 16). I figurane skil kjerneelementet *abstraksjon og generalisering* (AG) seg ut med ein markant lågare gjennomsnittskåre enn dei andre samlevvariablane knytt til kjerneelementa, medan *modellering og anvendingar* (MA) fekk høgare gjennomsnittskåre enn resten.

Figur 2 og 3 frå S. Eriksen og Vos (2022, s. 48) viser at gjennomsnittsverdien til kjerneelementet AG for faga 1P og 1P-Y også er låg, med hvv. verdiane 0,2 og 0,0. S. Eriksen og Vos (2022, s. 50) gir ingen klar grunn for kvifor det er slik, men frå figurane dei presenterer spekulerer dei på om kjerneelementet AG har lettare for å kome til uttrykk i eit meir teoretisk 1T og 1T-Y-fag, enn i eit meir praktisk retta 1P og 1P-Y. S. Eriksen og Vos (2022) kan ikkje forklare kvifor det er slik og påpeiker også at dersom AG hadde hatt lettare for å kome til uttrykk i dei meir teoretiske versjonane av matematikkfaget, burde gjennomsnittskåren for andre kjerneelement ha støtta opp om dette og slik er det ikkje.

Men er det slik at lærebøkene skal balansere bruken av kjerneelement? S. Eriksen og Vos (2022, s. 50) påpeikar at det ikkje har vorte spesifisert frå Utdanningsdirektoratet at dei ulike kjerneelementa skal vektast likt i matematikkfaget. I arbeidet med kjerneelementa i



matematikken vart kjerneelementet Utforsking og problemløysing gitt ei prioritert rolle i læreplanen (Nordbakke, 2018, s. 38) og dette signaliserer at det ikkje er slik at kjerneelementa skal vektast likt.

Det er ei moglegheit for at læreplanane legg opp til å styre vekting av kjerneelementa gjennom kompetansemåla. Kompetansemåla skal konkretisere kjerneelementa (Utdanningsdirektoratet, 2022b). Sidan kjerneelementa er like for heile grunnskuleløpet medan kompetansemåla er, for matematikken sin del, trinn-bestemte, vil ein slik måte å vekte kjerneelementa på vere eit vanskeleg kart å følgje.

Ei anna moglegheit er at Utdanningsdirektoratet vel å styre vektinga av kjerneelement ved bruk av eksamen. Eksamen er ein måte å innføre endringar i skulen på, og kan verte nytta av Utdanningsdirektoratet for å endre retning på kva som vert undervist (Fried & Amit, 2015; Morgan & Sfard, 2016, s. 92; Wall, 2006). Dersom Utdanningsdirektoratet vel ein slik måte å vise kva som skal ha fokus i undervisninga, gir dette ein fordel med tanke på programmering; fokus kan endrast etter kvart som det kjem elevar opp i 10. trinn som har meir og meir akkumulert programmeringskunnskap enn dei som går i 10. trinn, når læreplanen vert innført. Dette kan utførast utan å endre læreplanen. Baksida ved eit slikt opplegg er at lærarar og lærebokforfattarar ikkje veit kva dei skal vektlegge i undervisninga for å følgje Utdanningsdirektoratet sine ynskje, og spesielt vanskeleg er det for dei som lagar fysiske lærebøker. Matematikk 9 (Cappelen Damm, u.å.) har valt ei tilnærming som vil fungere godt dersom Utdanningsdirektoratet nyttar eksamen til å endre fokus, og har publisert programmeringsoppgåver som eit eige digitalt hefte gjennom ei teneste som skulane kan abonnere på.

#### *5.1.1.2 Forskjellig kompetanseprofil i oppgåvene i og utanfor utvalet*

Forskjellen på oppgåver i utvalet frå kvar lærebok heng saman med profilen på kompetansar som er representerte i oppgåvene i utvalet, og vert her drøfta under eitt. Sjølv om lærebøkene er forskjellig oppbygd, er likevel Matemagisk 9-læreboka og Maximum 9-læreboka likare i korleis dei presenterer både oppgåver i og utanfor utvalet samanlikna med Matematikk 9-læreverket.

Samanlikna med oppgåver utanfor utvalet, altså oppgåver som ikkje er kategoriserte som programmeringsoppgåver, er det tydeleg at fokuset er forskjellig i oppgåver i og utanfor utvalet. Medan kjerneelementa *utforsking og problemløysing* (UP) og *modellering og anvendingar* (MA) får høgast skåre av samlevariablane for kjerneelementa i oppgåver i

utvalet, får desse to kjerneelementa lågast skåre i oppgåvene utanfor utvalet (Figur 16). Det ligg ikkje nokon spesielle føringar i læreplanen for at kjerneelementet UP skal knytast til programmering, men det er tydeleg at oppgåvene i utvalet dekkjer dette kjerneelementet i større grad enn oppgåver utanfor utvalet. For kjerneelementet MA er det heilt tydeleg frå analysen at det er dataprogramma si moglegheit til å matematisere som kjem til uttrykk gjennom oppgåvene, og hevar gjennomsnittsverdien for dette kjerneelementet for oppgåver i utvalet.

Studien finn at når oppgåver i og utanfor utvalet vert sett under eitt, så jamnar forskjellane seg ut. Av dei 5 kjerneelementa er det relativt likt for 4 av dei (mellom 0,2 og 0,27 i gjennomsnitt), medan samlevariabelen *representasjon og kommunikasjon* (RK) får ein høgare gjennomsnittskåre på 0,36. Ei samanlikning av kvart enkelt læreverk bekreftar langt på veg at dette også er tilfelle for to av verka. Både Matemagisk 9 og Maximum 9-lærebøkene syner ein oppgåveprofil der samlevariablane for kjerneelementa UP og MA får høgast skåre for oppgåver i utvalet, medan samlevariablane for kjerneelementa RK, AG og *resonnering og argumentasjon* (RA) får høgast skåre for oppgåver utanfor utvalet (Figur 18). Kvar av dei to lærebøkene får då ein heilskapleg profil som for oppgåver i og utanfor utvalet tilseier at elevane får til dels lik trening innanfor alle dei 5 samlevariablane. Matematikk 9-læreverket har løyst dette på ein annan måte, og samlevariablane for kjerneelementa har lågare skåre enn dei to andre læreverka, men samtidig er oppgåver utanfor utvalet likare innbyrdes i læreverket.

På den eine sida kan ei løysing der læreverka passar på at den totale mengda oppgåver i eit kapittel eller i ei bok gir relativ lik trening i alle kompetansane, virke fornuftig. Læreverka i studien er knytt til eit spesifikt klassetrinn og dekker gjerne læreplanen sine mål for kva tema matematikkundervisinga for trinnet skal innehalde. Nyttar læraren læreverket gjennom heile skuleåret, vil elevane også få relativt lik trening i kompetansane. På den andre sida er læraren då knytt til læreboka for heile skuleåret. Kva skjer dersom læraren vel å nytte eit kapittel frå ei anna lærebok, eller eige opplegg? Læraren kan ikkje lenger vere trygg på at elevane har fått lik trening i kompetansane. Ei mogleg løysing er knytt til korleis Matematikk 9-læreverket har designa sine oppgåver, der trening på kompetansane er meir jamt fordelt i oppgåvene utanfor utvalet. Denne løysinga kan gje større tryggleik rundt at elevane får trene på kompetansane sjølv om læraren går vekk ifrå boka for ein periode.

### 5.1.1.3 Vesentleg meir trening på enkelt delkompetansar

Oversikta over alle oppgåvene i utvalet syner at det totalt sett vert trena vesentleg meir på enkelte delkompetansar enn andre i oppgåvene (sjå Figur 19 for samandrag og Figur 23 for individuelle læreverk). Som med kjerneelementa, kjem læreplanverket ikkje med ei oppskrift på kva delkompetansar innan kvart kjerneelement som skal vektast mest. I eksamen for matematikk 10. årstrinn for vår 2022, del 2, etterspør t.d. oppgåve 10 at eleven skal «... vise din kompetanse innen modellering og anvendelse» (Utdanningsdirektoratet, u.å.-c).

Eksamensoppgåva etterspør kompetansen frå heile kjerneelementet, ikkje berre deler av kjerneelementet.

Kvifor oppgåvene i utvalet har denne skeivprioriteringa av delkompetansar er ikkje lett å forklare innanfor rammene til studien. Ei mogleg forklaring kan vere at nokre delkompetansar passar betre saman med programmering enn andre. Programmering er sterkt knytt til utforsking, og mange av programmeringsoppgåvene har til formål å lage ein modell av t.d. terningkast for å finne mønster og samanhengar. Samtidig vert det nytta uformelt, matematisk språk i utviklinga av modellane. Desse faktorane kan gjere at delkompetansane (eller variablane) *avdekke mønster og samanhengar* (UP 1), *matematisere – lage ein modell* (MA 2) og *bruke matematisk språk* (RA 1) får høgast gjennomsnittskåre av variablane knytte til kjerneelementa.

Dersom det er slik at nokre delkompetansar passar betre saman med programmering enn andre, burde oppgåvene utanfor utvalet, ideelt sett, kompensere og vektlegge dei delkompetansane som ikkje vert dekte av oppgåvene i utvalet. Men det er ikkje tydeleg at det er slik. For læreverka Matematisk 9 og Maximum 9 er det ikkje ein tydeleg trend der oppgåvene utanfor utvalet dekkjer delkompetansar som oppgåver i utvalet ikkje gjer (Figur 24 og Figur 26). Matematikk 9-læreverket legg meir vekt på å dekke delkompetansar i oppgåver utanfor læreverket, medan dekking av delkompetansar i oppgåver i utvalet er nærast fråverande.

### 5.1.1.4 Eksplisitte oppgåver i utvalet

Studien finn at alle oppgåver i utvalet inneheld behovet for programmering eksplisitt nemnt i oppgåveteksten. Sjølv om eksplisitte krav om programmering var eit av to utvalskriterium, var det ikkje eit krav at oppgåvene skulle treffe begge kriteria. 31% av oppgåvene stod i eit programmeringskapittel og hadde vorte inkludert i utvalet sjølv utan eksplisitt krav om programmering.

På den eine sida kan eksplisitt krav om programmering gje ei utfordring når elevar skal vise kompetanse på å velje om det er behov for digitale hjelpemiddel eller ikkje. Figur 19 viser at variabelen som måler om elevane får moglegheit til å *vurdere bruk av digitale hjelpemiddel eller ikkje* (UP 3), får 0,0 i gjennomsnittskåre, og elevane får ikkje moglegheit til å vurdere dette for oppgåver i utvalet. Dersom alle oppgåver eksplisitt forklarar val av digitalt hjelpemiddel eller ikkje, og kva hjelpemiddel elevane kan nytte, kan elevar gå glipp av moglegheita til å bygge kompetanse på dette området sjølve.

På den andre sida er eit eksplisitt krav om programmering med på å legge til rette for at elevar bygger ferdigheiter innan programmering. Utan eit slikt eksplisitt krav til programmering kan elevar, som manglar motivasjon til eller ferdigheiter om programmering velje andre, gjerne analoge, metodar å løyse matematikkoppgåva på. Å auke ferdigheitene til elevane kan hjelpe elevane til å sjå kva tid det er hensiktsmessig å nytte programmering, eller andre digitale måtar, og dermed gje auka moglegheit til å vise kompetanse i variabelen UP 3.

### 5.1.2 Drøftingar rundt PR-variabelen

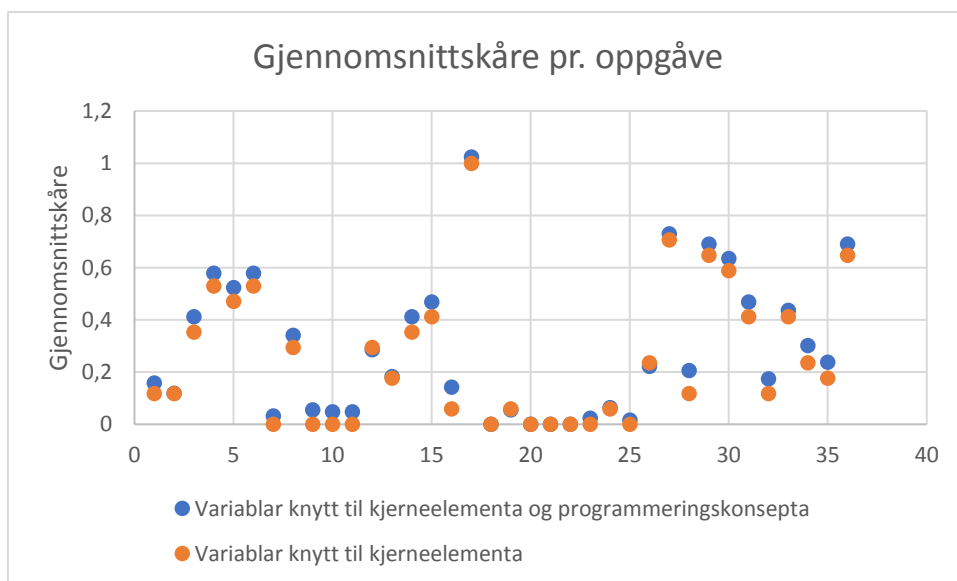
Samlevariabelen for programmering (PR) er den samlevariabelen som har den høgaste gjennomsnittskåren (Figur 15). På den eine sida kan det argumenterast for at ein samlevariabel for programmering som ligg langt over gjennomsnittet for dei forskjellige kjerneelementa (den samla gjennomsnittsverdien for variablane til kjerneelementa er 0,24), viser at programmeringsoppgåver i for stor grad underviser i programmering for programmeringa si skuld. Inntrykket etter analysinga av oppgåvene støttar også ei slik oppfatning, sjølv om dette ikkje gjeld alle oppgåver. På den andre sida kan det argumenterast med at ein samlevariabel for programmering i programmeringsoppgåver skal være så høg som mogleg, slik at elevane får trene på ferdigheita programmering. Denne ferdigheita kan ikkje nyttast til å bygge kompetanse innanfor kjerneelementa dersom elevane ikkje får nok programmeringstrening.

Måten studien lagar utvalet oppgåver på, gjer at det er oppgåver knytt til programmering som vert med i utvalet. Dette betyr *ikkje* at ingen andre oppgåver i kapittelet kan løysast med programmering. Men om ein elev nyttar programmering berre når oppgåvene ber om det, vil eit sterkt fokus på programmeringsopplæring i programmeringsoppgåvene kunne gi trening på programmering som *ferdigheit* og kan gje mindre moglegheit til å nytte programmeringsferdigheiter til å auke matematisk *kompetanse*.

For å avgjere kva som er eit hensiktsmessig nivå for ein variabel knytt til programmering, er det avgjerande å sjå læreverket for grunnskulen som læreboka er knytt til under eitt. Eleven må få tid og moglegheit til å trene på programmering som ferdighet for å kunne nytte ferdigheita til å bygge matematisk kompetanse. Ein studie på kompetansemåla i matematikk for kvart trinn, kan kanskje kaste lys over korleis læreplanen ser for seg at samanhengen mellom programmeringsferdighet og kompetansebygging skal løysast. Då vil gjennomsnittsvariabelen for programmering i dette rammeverket også kunne få ein kontekst å bli sett inn i.

Figur 36 syner korleis oppgåvene i utvalet skil seg frå kvarande når dei vert sorterte etter gjennomsnittskåre for variablar knytte til kjerneelementa, både saman med og utan samlevariabelen for *programmeringskonsept* (PR 3). Oppgåvene er nummererte og ei oppgåve i den blå gruppa korresponderer til oppgåva på same plass i førsteaksen for den oransje gruppa. Diagrammet tydeleg at det for dei fleste oppgåvene fører til høgare gjennomsnittskåre å inkludere PR-variabelen.

Figur 36 Gjennomsnittskåre pr. oppgåve i utvalet, nummerert og samanlikna med og utan PR 3-variabelen



Differansen mellom gjennomsnittskåre med og utan PR 3 syner at mesteparten av oppgåvene i utvalet har fokus på å trene programmeringskonsept. I utgangspunktet kan dette sjå ut som ein indikator på at oppgåvene trenar programmeringsferdigheiter i staden for å nytte programmering for å underbygge utvikling av kompetansar.

Men er det verkeleg slik at informasjonen frå figuren kan nyttast som ein indikator på kva fokus oppgåvene har? På den eine sida er det ein forskjell på verdiane for kvar oppgåve, sjølv

om differansen ikkje er så stor mellom grafane og heller ikkje alltid går i favør PR 3-variabelen. På den andre sida er det vanskeleg å drive programmering med berre eit programmeringskonsept uavhengig av kor mange variablar knytt til kjerneelementa som vert inkluderte. Dersom eleven skal lage eit program som skal simulere terningkast og legge saman resultatane av 10 kast etter kvarandre, involverer dette i utgangspunktet variabelen *matematisere – lage ein modell* (MA 2) frå kjerneelementa. Men frå programmeringskonseptane er variablane *data* (PR 3.7), *sekvensar* (PR 3.1) og *operatorar* (PR 3.6) involvert, og med moglegheit for at variabelen *løkker* (PR 3.2) kan nyttast. Desse 3-4 variablane er også involverte, sjølv om oppgåva er vesentleg meir kompleks og involverer mange fleire variablar frå kjerneelementa. Det er altså ingen direkte samanheng mellom gjennomsnittet for variablar knytt til kjerneelementa og gjennomsnittet for variabelen PR 3 sjølv om Figur 36 kan indikere at det er slik.

Vidare er det tydeleg frå analysen av oppgåver at variablane knytte til programmeringskonseptane stort sett vart skåra med verdien 2 (Figur 31). Slik oppgåvene i utvalet var formulerte, vert 5 av 7 variablar knytte til programmeringskonseptane nytta i 50% av tilfella eller meir (Figur 31). For variablar knytte til kjerneelementa, var det 1 variabel av 17 moglege som vart nytta 50% eller meir i oppgåvene. Ei mogleg forklaring på dette er, som nemnt over, at nokre grunnleggjande konsept må vere inkluderte når eit program skal programmerast i eit programmeringsspråk.

## 5.2 Drøftingar knytte til rammeverket

Eit mål med studien var å utvikle eit rammeverk som kan hjelpe lærarar og skuleeigarar med å kvalitetssikre, i første omgang, lærebokressursar, men også sjølvskapt programmeringsundervising i matematikk. Rammeverket i studien ser på korleis innhaldet i kjerneelementa vert spegla i programmeringsoppgåver, i tillegg til å vurdere om programmeringsoppgåvene gir programmeringsopplæring som elevane treng for å løyse oppgåver i matematikkundervisinga. Fag-læreplanane i LK20 er ikkje konkrete lister over kva elevane skal lære til kva tid, men er sett saman av fleire element som samla gir læraren rammer å arbeide innanfor. Dette gir læraren ansvar for at læreplanen vert implementert i undervisinga i tråd med Utdanningsdirektoratet sitt ynskje. I LK20 er det kjerneelementa som beskriv innhaldet i kvart fag, medan kompetansemåla konkretiserer korleis kompetansen som arbeidet med kjerneelementa har gitt, kan målast (Utdanningsdirektoratet, 2022b).

Forsking har gitt resultat som kan tyde på at innføringa av programmering i LK20 har gitt matematikklærarar ei oppgåve som kan opplevast krevjande (Hirsch, 2022, s. i). Lite programmeringskompetanse blant lærarane kan gje læreboka ein større plass i undervisninga. Tal som er henta frå tida med LK06 viser at lærarar stort sett nyttar læreboka i undervisninga (Mullis et al., 2012). Tala seier ingenting om korleis læraren nyttar læreboka, men dersom ein lærar opplever å mangle programmeringskompetanse, er det grunn til å tru at læreboka eit nyttig hjelpemiddel.

Rammeverket, slik det er nytta i studien, er sikta inn mot nettopp lærebøker, og funna er i stor grad knytt til statistikk generert frå ein større mengde data. Ei slik kvantitativ innhaldsanalyse gir moglegheit til å raskt få ein oversikt over oppgåveutvalet. Dette kan nyttast for å vurdere noverande eller komande læreverk opp mot behovet til skulen. Det gir òg rom for både endringar undervegs i undervisningsløpet og ein oversikt over kjerneelement og programmeringskonsept å arbeide vidare med utover det oppgåvene i utvalet dekker.

Ei slik tilnærming til analyse av oppgåver som studien nyttar, kan gjere det utfordrande å sjå kvar individuell oppgåve for seg sjølv. Rammeverket er ikkje velegna å nytte i prosessen med å lage/designe programmeringsoppgåver i matematikkfaget. Men utifrå erfaringar gjort i arbeidet med studien, er det to ting som kan trekkast fram i den samanhengen. Det eine er kompleksiteten/omfanget på oppgåva som vert laga, og det andre er korleis den passar inn i oppgåvene som allereie er laga/er planlagt.

Den oppgåva frå heile utvalet som har høgast snittskåre både når det gjeld programmering og kompetansemål, er å finne sist i sitt kapittel. Denne oppgåva skil seg ut frå dei andre oppgåvene i statistikken, og står litt for seg sjølv i diagramma. På same måten kan ein lærar nytte verktøyet til å vurdere sine eigne oppgåver, og sjå kor mange variablar ei oppgåve freistar å dekke samtidig, slik at variabelmengda kan justerast opp og ned, og plasseringa i kapittelet kan justerast, alt etter ynskje frå kvar lærar.

I tillegg kan rammeverket nyttast til å vurdere korleis ei oppgåve passar inn saman med dei andre oppgåvene som er gitt til elevane, eller som er planlagde for bruk. I eit undervisningsmiljø der dekking av kjerneelementa er eit kollektivt ansvar, kan kvar enkelt lærar leggje inn sine oppgåver og alle kan sjå kva kjerneelement og programmeringskonsept som er dekte, eller som treng meir fokus.

### 5.2.1 Utval av oppgåver

I kap. 3.5 tek oppgåva føre seg utval av oppgåver. Rammeverket har ein tydeleg definisjon av programmering, med fokus på koding. Utvalet av oppgåver i studien ber tydeleg preg av denne definisjonen. I prosessen med å velje ut oppgåver viste det seg derimot, at lærebøkene har varierende grad av definisjon av programmering, noko som kan gje utslag på utvalet frå framtidige læreverk. Ein måte ein kan endre utvalskriteriet på, og som kan inkludere andre typar oppgåver i utvalet, er å sjå på oppgåver der (lærebok-)forfattaren tydeleg har meint at oppgåva skal løysast med programmering utan å nemne det eksplisitt. Slike oppgåver vert ikkje tekne med i utvalet i studien, og det vart heller ikkje identifisert slike oppgåver i lærebok-kapitela som danna grunnlaget for utvalet av oppgåver. Eit eksempel på ei oppgåve som tydeleg er meint å løysast med programmering kan være «Kva vert summen av dei 100 første primtala?».

Utvalsfilteret tek heller ikkje med oppgåver der det vert nytta andre ord som er knytte til programmering, t.d. «simuler» eller «lag ein modell». Når programmering skal være ein måte å løyse og å utforske matematiske problem på, der elevane sjølve skal velje og vurdere bruk av digitale hjelpemiddel eller ikkje, kan programmering som løysingsmetode være meir implisitt enn eksplisitt etterspurt. I preanalysen vart det ikkje funne nokon oppgåver som inkluderte programmering implisitt.

### 5.2.2 Oppgåveanalyse

Hovuddelen av rammeverket er oppgåveanalysen. Som når utvalet vert laga, vert informasjonen også i denne fasen av metoden analysert og oversett frå ikkje-metriske data til tal som kan analyserast statistisk (Grønmo, 2016, s. 137–138). Kodeskjemaet for analysen består av operasjonaliserte kjerneelement henta frå S. Eriksen og Vos (2022) og programmeringsvariablar henta frå eit rammeverk av Brennan og Resnick (2012). Frå S. Eriksen og Vos (2022) vart det henta ut variablar for 5 av dei 6 kjerneelementa, og frå Brennan og Resnick (2012) var det henta ut programmeringsvariablar knytt til 7 forskjellige programmeringskonsept. I tillegg vart det lagt til to variablar som registrerer om ei programmeringsoppgåve krev å lese/tolke kode eller om eleven må skrive kode sjølv. Dette gir samla eit kodeskjema med 26 variablar som vert skåra med kategori 0, 1 eller 2. I tillegg var det ein samlevariabel for programmeringskonsept som var automatisk skåra.

Å kombinere to ulike rammeverk kan gje fleire fordelar. Ein fordel er at statistikken frå fleire variablar i større grad kan nyttast til å finne styrkar og svakheiter i oppgåveutvalet. Fleire variablar er med på å gje eit meir nyansert bilde. Med dei to rammeverka som er kombinerte i



studien er det også mogleg å undersøke både variablar om kjerneelement og om programmering, og generere statistikk for kvar av dei. Desse kan samanliknast for å gje eit betre bilde av kva kjerneelement, og dermed kompetansar, og kva programmeringskonsept eleven arbeider med når eleven løyser programmeringsoppgåver.

Men å nytte eksisterande rammeverk kan også gje utfordringar. Når S. Eriksen og Vos (2022) presenterer funna dei gjorde med sitt rammeverk, presenterte dei samtidig berre eksempel, eller kodeinstruks, for eit av dei 6 kjerneelementa. Dei variablane som var henta frå dei andre 4 kjerneelementa som vart nytta i studien, måtte tolkast og fyllast med innhald. Å arbeide med kjerneelementa er å arbeide med eit sett med kompetansar som er samanvevde (Kilpatrick et al., 2001). Når rammeverket tek sikte på å dele opp desse kompetansane og knyte dei til allereie eksisterande operasjonaliseringar, er det ingen garanti for at innhaldet variablane i rammeverket frå denne studien er det same som i rammeverket frå S. Eriksen og Vos (2022). I denne studien er operasjonaliseringa av kjerneelementa knytt i stor grad opp til kompetansemodellane frå KOM-prosjektet (Niss & Højgaard, 2011) og PISA (Turner et al., 2013)

I arbeidet med analyse av oppgåve, vart det klart at det var fleire variablar som hadde overlappende element og dermed vanskelege å skilje. Kva som t.d. er den heilt klare forskjellen på å *avdekke mønster og samanhengar* (UP 1) og å *oppdage matematiske samanhengar og formelle resonnement* (AG 3) er ikkje alltid tydeleg. Eit anna eksempel er å sjå kvar skiljet går mellom å *bruke matematiske representasjonar* (RK 2) og *byte mellom representasjonar* (RK 4). Studien legg m.a. Duval (2006) sin definisjon til grunn for tolkinga av skiljet mellom RK 2 og RK 4, men det er ikkje sikkert dette er likt med tolkinga til S. Eriksen og Vos (2022) eller med kjerneelementa i læreplanen (Utdanningsdirektoratet, 2019b).

Innanfor dei rammene som var tilgjengelege for denne oppgåva, var det berre ein som koda og analyserte oppgåver. Spørsmål som handla om forskjellen på, eller innhald i variablar, blir ofte hengande litt i lufta. I ein situasjon med fleire kodarar kan ein nytte forskjellige verktøy for å måle kodesamsvar. Cohen's Kappa er eit populært instrument for å måle kodesamsvar for data på nominal- eller ordinalnivå (Nagaraja & Choudhary, 2017, s. 289). Utan slike verktøy til rådigheit, vart det i studien i staden nytta fleire rundar med analyser av dei same oppgåvene, og svara vart samanlikna for kvar runde. Det finst ingen fasit på kor mange slike

analyse-rundar ein bør gjere. I studien vart det to rundar, men med fleire rundar kan kodaren lage statistikk som syner at avviks-raten mellom svara flatar ut.

### 5.2.3 Konstruksjon av rammeverket

Målet med rammeverket som er utvikla i studien er å tilby lærarar og andre eit rammeverk for å analysere to sider med programmeringsoppgåver: korleis oppgåvene dekkjer kjerneelementa og korleis oppgåvene dekkjer behovet for opplæring i programmering. Denne delinga kan opplevast kunstig. Når programmering er ein del av matematikken, kvifor skal ikkje programmeringsspråket då være ein del av eit matematisk språk (RK 1) eller eit formelt symbolspråk (AG 1)? Eit programmeringsspråk har faste kommando-ord, symbol, reglar og syntax (Mannila, 2017, s. 71), og er i høgste grad formelt.

Ein grunn til å skilje mellom kjerneelement og programmering er at programmering ikkje er rekna som ein kompetanse i kjerneelementa. Programmering ligg inne i læreplanen som ei ferdigheit (Utdanningsdirektoratet, u.å.-e), og er ein del av problemløysingsprosessen som vert kalla «algoritmisk tankegang» (Sevik, 2016, s. 13). I norsk skule er programmering meint å støtte utvikling av kompetansar hjå eleven (Sevik, 2016, s. 7), til tross for at omgrepet finst i matematikkens kompetansemål (Utdanningsdirektoratet, 2019b). Å nytte seg av datamaskiner er nemnt i KOM-prosjektet (Niss & Højgaard, 2011) og er knytt til kompetansen «Aid and tools», men denne kompetansen er ikkje teken direkte inn i kjerneelementa.

At programmering er skildra som ei ferdigheit i den norske skulen betyr at elevane må trene opp denne ferdigheita slik at den kan nyttast i problemløysinga. For å belyse programmeringsopplæringa i skulen slik den er gjennom lærebøkene, måtte rammeverket i studien utvidast med variablar knytt til måling av programmering. Variablane vart henta frå (ein del av) eit rammeverk som ser på korleis programmering støttar utvikling av algoritmisk tankegang (Brennan & Resnick, 2012). Figur 31 viser korleis oppgåvene skårar innanfor dei forskjellige programmeringsvariablane. I etterkant av oppgåveanalysen i studien viser det seg at to variablar knytte til programmering hadde ingen eller få oppgåver som ga skåre 1 eller 2, og desse variablane var hvv. PR 3.4 (parallellisme/multitasking) og PR 3.3 (handlingar).

Rammeverket til Brennan og Resnick (2012) tek utgangspunkt i programmerarar som nyttar programmeringsspråket Scratch til å lage m.a. interaktive program, slik som t.d. spel eller interaktive simuleringar. Det er få oppgåver som nyttar seg av interaktivitet i oppgåveutvalet for studien, og det er ingen oppgåver som krev programmeringsspråket Scratch. Det ligg i

essensen til interaktive program er at dei krev ei form for interaksjon frå brukaren, og det er her rammeverket vurderer om programmet nyttar handlingar (PR 3.3) i koden. Ei handling kan t.d. være å trykke pil opp i eit spel, eller skrive inn kor mange gongar ein terning skal trillast i ei simulering av terningkast (Brennan & Resnick, 2012). I eit spel vert det ofte nytta kode som vert køyrt samtidig. Dette vert kalla parallellisme/multitasking (Brennan & Resnick, 2012) og vert vurdert i variabel PR 3.4 . I Scratch er det tilrettelagt for å lage interaktive program med multitasking (Skeie, 2016). Det er eit blokkbasert programmeringsspråk der elevane kan, med relativt få taste-trykk, generere kode som t.d. både spelar av musikk og skriv ut noko på skjermen når ei handling inntreffer, og som gjer begge deler samtidig, ikkje etter kvarandre. Koden som vert utført kan gjerne stå forskjellige plassar i kodeområdet til Scratch, og treng ikkje henge saman.

Korkje blokkprogrammering eller multitasking er omgrep som er knytte utelukkande til Scratch, det finst mange andre programmeringsspråk som yter dei same moglegheitene. Lærebokoppgåvene i utvalet for studien nyttar programmeringsspråket Python (eller eit språk med liknande kommandoar og syntax), og dette programmeringsspråket er ikkje i like stor grad utvikla for grafiske interaktive program som t.d. Scratch er. I kompetansemåla for 9. trinn er tema simulering og programmering nemnt i same kompetansemål. Dette er kanskje grunnen til at lærebøkene ikkje fokuserer på dei programmeringskonseptane som dei gjer. Kvifor alle oppgåvene i utvalet nyttar programmeringsspråket Python, eller eit tilsvarande og likt språk, er usikkert. Utdanningsdirektoratet har ikkje lagt tydelege føringar i læreplanen når det kjem til val av programmeringsspråk å nytte i undervisinga, men i t.d. eksamen for matematikk 1P har det vore nytta kode som er tilsvarande slik koden hadde sett ut i Python.

#### 5.2.4 Kva er rammeverket ikkje ser på?

Som eit ledd i nyttevurderinga av rammeverket er det vesentleg å sjå på kva rammeverket ikkje ser på. Forskjellig undervisningspersonell har forskjellige behov, og det er vanskeleg å lage eit rammeverk som passar alle. Berre ved å ha ei klar oversikt over kva eit rammeverk inkluderer og ekskluderer er det mogleg å vurdere korleis resultatet frå rammeverket si analyse skal nyttast.

Det mest openlyse er kanskje at rammeverket ikkje ser på kompetansemål. Vurderinga i prosessen med å lage dette rammeverket er at kompetansemåla konkretiserer kjerneelementa (Utdanningsdirektoratet, 2022b) og at arbeid med kjerneelementa skal gje eleven kompetanse til å kunne løyse dei konkrete kompetansemåla. Sett på spissen vart det vurdert slik at

kompetansemåla er resultat av arbeidet med kjerneelementa, og at kjerneelementa representerer kompetansar som til saman gjer eleven i stand til å møte kompetansemåla på ein god måte.

Noko som i stor grad kan påverke læreboka sin effekt på programmeringsundervisinga for kvar enkelt elev, men som rammeverket ikkje tek omsyn til, er nivådifferensiering. I preanalysen og gjennomgang av lærebøkene var det tydeleg at nokre lærebøkene la opp til nivådifferensiering av oppgåver, sjølv om ikkje alle lærebøkene differensierte programmeringsoppgåver. Det kan derimot løysast av den som nyttar rammeverket ved at oppgåver som er felles for alle nivådifferensieringar og oppgåver knytte til ei nivådifferensiering, vert med i utvalet oppgåver, og at denne prosessen vert gjenteken for kvar nivådifferensiering.

Kvaliteten på oppgåvene, sett frå ein pedagogisk ståstad, er ikkje ein del av analysen til rammeverket. Det finst rammeverk som freistar gje svar på dette, og ei integrering av eit slikt rammeverk vert foreslått som ein veg vidare.

Rammeverket vurderer ikkje størrelsen på eller arbeidsmengda til ei oppgåve. Oppgåvene kan være komplekse eller enkle, ha ingen eller mange deloppgåver utan at det påverkar resultatet av analysen. Rammeverket vurderer heller ikkje om det er ein naturleg progresjon i arbeidet med kjerneelementa. Heller ikkje programmeringsundervisinga vert vurdert med tanke på progresjon for kvart enkelt programmeringskonsept. Figur 35 syner ei oversikt over gjennomsnittsverdien i programmering pr. oppgåve, og plasseringa i kapittelet. Rammeverket viser om lag kvar i hovudkapittelet programmeringsoppgåve er, men det vert ikkje innhenta data som kan seie noko om det er programmeringsoppgåver i alle delkapittel eller delemne.

### 5.2.5 Oppsummering og praktisk nytte av rammeverket

I denne drøftingsdelen har det vorte sett fokus på rammeverket som er utvikla for studien, både med tanke på utvikling av rammeverktøyet og praktiske implikasjonar. Sjølv om rammeverket kan vere eit godt hjelpemiddel for lærarar og skuleeigarar i arbeidet med å finne gode læreverk eller lage gode matematikkoppgåver sjølve, er det likevel ein jobb som må gjerast. Den som nyttar seg av rammeverket bør ha ei klar oppfatning av kva dei forskjellige variablane rommar. Kodeinstruksen er ei god hjelp, men det vil kome oppgåver som utfordrar kodeinstruksen. Er det fleire som skal kode oppgåvene i kodeskjemaet, bør det nyttast verktøy for kodesamsvar, eller at dei som kodar samanliknar resultata undervegs.

Resultata som kjem fram i studien er knytt til to faktorar: oppgåvene som vart vurderte, og vurderinga til den som koda. Resultatet som visast i Figur 12 «prosentvis programmeringsoppgåver pr. lærebok» gir kanskje mest verdi når dei vert sett inn i ein kontekst. Dersom dei som nyttar seg av rammeverket ynskjer læreverk som har mange programmeringsoppgåver, kan dette resultatet gi innspel i prosessen med å velje læreverk. Ein lærar vel kanskje å nytte seg av resultat som dei i Figur 30 «gjennomsnittskåre for variablar knytt til hvv. kjerneelementa og programmering, pr. oppgåve» for å vurdere om oppgåvene som vert produserte for elevar, er for komplekse eller ikkje omfattande nok. Rammeverket opnar for mange forskjellige typar resultat, og det er hensikta til den som nyttar rammeverket, som avgjer kva type resultat som er formålstenleg.

## 6 Oppsummering og vegen vidare

Studien ser på programmeringsoppgåver frå 3 matematikklæreverk og analyserer korleis desse oppgåvene knyter seg til kjerneelementa og innføring av programmering i matematikk i LK20. Problemstillinga er open og ikkje eigna for konklusjon. I staden vert forhold mellom programmeringsoppgåver og kjerneelementa skildra for å syne korleis læreverka i studien koplar desse to. For å lette arbeidet med både å svare på problemstillinga og å lese oppgåva, vart det utvikla 3 forskingsspørsmål, og desse er med på å legge grunnlaget for oppsummeringa. Dei 3 forskingsspørsmåla er:

- Kva kjenneteiknar den totale mengda programmeringsoppgåver i studien
- Kva er forskjellen mellom lærebøkene som er nytta i studien
- Kva er forskjellen på programmeringsoppgåver i matematikk og andre matematikkoppgåver

Forskingsspørsmåla må sjåast i lys av problemstillinga og dei avgrensingane som er gjorde for studien. Programmeringsoppgåvene er henta frå dei 3 læreverka og blir handsama ut frå slik dei er definerte i dette arbeidet.

### 6.1 Kommenterarar om kjerneelementa

Funn frå studien syner at oppgåver i utvalet ikkje legg like mykje vekt på alle kjerneelementa (Figur 15). Heller ikkje innanfor kvart kjerneelement vil oppgåver i utvalet prioritere delkompetansane likt (Figur 19). Eit fullgodt svar på korleis alle programmeringsoppgåvene vil dekke kjerneelementa, vil berre kome til syne når alle læreverka er analyserte frå start til slutt. Men funna i studien syner at dersom ein lærar vel å nytte seg av programmeringsoppgåver vert ikkje alle kjerneelementa dekte i like stor grad.

Lærebøkene ser ut til å prøve å jamne ut fokuset på kjerneelementa og delkompetansane ved at oppgåver utanfor utvalet, har ein annan profil enn oppgåver i utvalet, når det gjeld kva kjerneelement dei inkluderer. Like fullt gir ikkje oppgåvene, totalt sett i og utanfor utvalet, like mykje trening på alle kjerneelement eller delkompetansar (Figur 24, Figur 25 og Figur 26).

Lærebøkene skil seg til dels kraftig frå kvarande. Læreverka Matematisk 9 og Maximum 9 har ein relativt lik profil på oppgåvene i utvalet, medan Matematikk 9-læreverket har valt ei anna tilnærming (Figur 17). Dette gir seg utslag i at Matematikk 9-læreverket får svært låg

gjennomsnittsskåre for alle variablar knytte til kjerneelementa i studien, og mykje lågare enn dei to andre læreverka. Oppgåver i og utanfor utvalet, sett under eitt, syner at Matematikk 9-læreverket kompenserer noko ved å dekke fleire av delkompetansane i oppgåver utanfor utvalet enn dei to andre læreverka gjer.

Den store skilnaden på oppgåver i og utanfor utvalet, er profilen på kva kjerneelementvariablar dei inkluderer (Figur 18), og naturleg nok fokuset på programmering. Matemagisk 9-læreboka og Maximum 9-læreboka står for mesteparten av oppgåvene i studien, og i desse lærebøkene er det tydeleg forskjell på kva kjerneelement som vert fokusert på i og utanfor utvalet. Matematikk 9-læreverket har løyst dette på ein måte der differansen mellom oppgåver i og utanfor utvalet er stor. Men fordi læreverket har låg skåre for alle kjerneelementvariablar for oppgåver i utvalet, er det stort sett oppgåver utanfor utvalet som gjer at kjerneelementvariablar får ein skåre over 0.

Funna i studien indikerer at det ikkje er likegyldig kva læreverk lærarar eller skuleeigarar vel å nytte i undervisinga. Anten det vert nytta eitt og same læreverk til all undervising eller ein lærar vil hente undervisningsopplegg frå andre kjelder, og eventuelt kombinere dette med sitt eige undervisningsopplegg, er det tydeleg at når det kjem til kjerneelement og kompetansar, er det viktig å være bevisst i valet av læreverk. Dersom ein lærar ynskjer å nytte programmeringsoppgåver frå læreverka brukte i studien til å gje eit breitt og variert kompetansegrunnlag, er det sannsynleg at læraren må supplere programmeringsoppgåver sjølv.

## 6.2 Kommenterar om programmering

I tillegg til å sjå på kjerneelementa, analyserer studien korleis programmeringsoppgåvene inkluderer programmering i oppgåvene. Det er vanskeleg å slå fast om oppgåver i utvalet trenar programmeringsferdigheiter eller om oppgåvene nyttar programmering til å trene kompetansar, eller om det er litt begge deler. I Matematikk 9-læreverket har oppgåvene i utvalet så låg gjennomsnittsskåre for variablar knytte til kjerneelementa, at det er sannsynleg at desse oppgåvene trenar programmeringsferdigheiter. At Matematikk 9-læreverket sine oppgåver i utvalet er henta frå eit hefte som heiter «Programmering 8-10» (Hjardar, u.å.) forsterkar dette inntrykket.

Både resultata knytte til variablar for kjerneelementa og variablar knytte til programmering kan t.d. inngå i ein prosess der undervisningspersonell ser på lærebøkene i si heilheit, anten

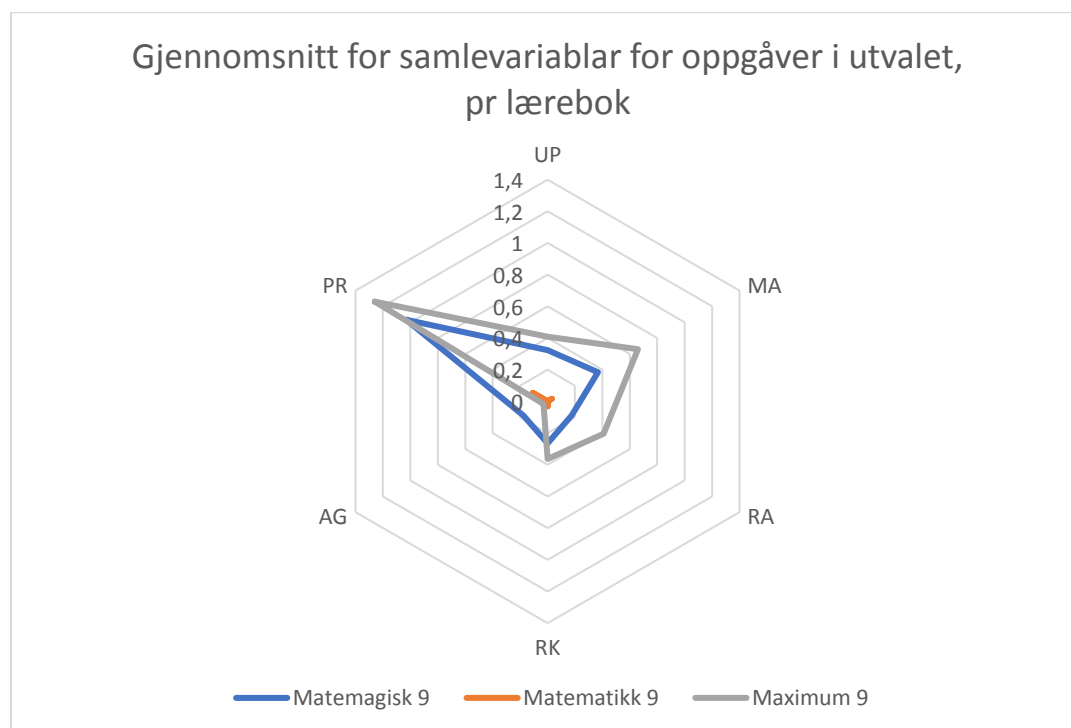
som eit ledd i å velje lærebøker til ein skule eller som ledd i å sjå korleis ein skal gå fram for å sikre at undervisinga held ein høg standard. Når 97% av lærarar i den norske skulen nyttar læreboka i undervisinga (Mullis et al., 2012, s. 392) kan det være hensiktsmessig å ha eit rammeverk som vurderer programmeringsoppgåvene i læreverka. Mullis et al. (2012) skriv ikkje noko om korleis lærarar nyttar lærebøkene, så resultatata må tolkast i tråd med ein lærar sine behov.

### 6.3 Kommentar om rammeverket

I tillegg til å analysere lærebøker kan rammeverket utvikla i oppgåva også nyttast til å analysere eigne undervisingsopplegg.

For å illustrere nytta av eit slikt rammeverk skildrast her to scenario. Første scenarioet er ein matematikklærer med lang erfaring, men som ikkje er trygg på bruken av programmering i matematikkundervisinga. Læraren støttar seg på læreverket skulen nyttar. Skulen skal skifte læreverk og læraren vil trygge seg sjølv på at læreverket dei skal velje lærer elevane det dei skal innan programmering, men også korleis at programmeringsoppgåvene knyter seg opp til kjerneelementa i faget. Læraren vel ut dei læreverka som er nytta i studien og ser på kapittel for sannsyn. Med dei tala studien har presentert, kan læraren generere ei samanlikning av lærebøkene lik den som er vist i Figur 37.

Figur 37 Gjennomsnitt for samlevariablar for oppgåver i utvalet, pr lærebok





Her kjem det tydeleg fram av figuren at i samanlikninga av lærebøkene kjem Matematikk 9 dårlegast ut på alle samlevvariablar, medan Maximum 9 kjem best ut på 5 av 6 samlevvariablar. Dersom læraren i scenarioet har tenkt å velje ut berre programmeringsoppgåver frå lærebøkene og nytte eigenproduserte matematikkoppgåver til den resterande undervisinga, kan læraren i dette tilfelle velje Maximum 9 som si prioriterte lærebok. Ein annan lærar i motsett situasjon vil kanskje føretrekke Matematikk 9-læreboka fordi læraren er trygg på programmeringsundervisning og ikkje ynskjer at læreboka skal inkludere rein programmeringsundervisning.

I eit anna scenario vil ein lærar nytte rammeverket til å kvalitetssikre si eiga matematikkundervisning. Etter LK20 ynskjer læraren å inkludere programmering i allereie eksisterande undervisningsopplegg, og vil nytte eit rammeverk for å sjå at alle programmeringskonseptane er representerte i samlinga av undervisningsopplegg. I tillegg vil læraren sjå kva kjerneelement som alt er representerte i undervisningsopplegga, og på den måte identifisere kva kjerneelement som må inn i dei resterande undervisningsopplegga. Etter avslutta analyse kan læraren produsere figurar lik Figur 23 og Figur 31 for å finne informasjonen læraren treng.

Rammeverket kan også nyttast for å analysere berre kjerneelement eller berre programmering. Men slik kodeinstruksjonen er utforma, skil rammeverket mellom dei matematiske kompetansane og programmeringsferdigheiter. Resultatet av å velje enten kjerneelement eller programmering kan være at den totale kompleksiteten i oppgåvene ikkje vert fanga like godt opp i analysen.

Å arbeide med eit rammeverk som dette har den fordel at det vert arbeidd med detaljert operasjonalisering av kjerneelementa. I prosessen med å utvikle kodeinstruks og å analysere oppgåvene i utvalet, kom det fram mange spørsmål om tolking av kjerneelementa. I arbeidet med å tolke oppgåver og vurdere kva kjerneelement oppgåvene inkluderte, ligg det også mykje refleksjonsarbeid som ga ny innsikt, både rundt kjerneelement og korleis desse er inkluderte i oppgåvene. I eit kollegium kan arbeid med desse avklaringane styrke trygghet i eigen forståing av kjerneelementa og gje grunnlag for felles forståing av innhaldet i kjerneelementa.

## 6.4 Veggen vidare

I studien har rammeverket vorte utforma og testa opp mot eit kapittel om sannsyn i utvalde lærebøker for 9. trinn, laga for LK20. Vidare bør rammeverket testast på fleire matematiske fagfelt og over fleire trinn. Dette vil bidra til kvalitetssikring av rammeverket og potensielt auke den ytre gyldigheita til studien. Replikastudier vil også gje innspel på kor godt rammeverket er designa og auke den indre gyldigheita til studien.

Det er fleire sider ved matematikkoppgåver rammeverket ikkje ser på, og det er rom for å utvide rammeverket frå studien med andre rammeverk. Ei mogleg utviding er å sjå på om programmeringsoppgåver også er kognitivt krevjande eller ikkje. Berge et al. (2022) har gjort ein studie der dei ser på matematikkoppgåver som inneheld programmering og vurderer desse oppgåvene for å sjå om dei er kognitivt krevjande innan både matematikk og programmering.

Eit område det kunne vore interessant å sjå nærare på, er korleis produsentane av læreverka vel å presentere oppgåver no når fleire av dei tilbyr digitale løysingar. Tradisjonelle læreverk med lærebøker er dyre og vert skjeldan skifta ut, medan læreverka i den digitale verda kan reviderast og publiserast fortløpande. Eit aktuelt spørsmål blir om forlaga vil ta i bruk moglegheita for både interaktive matematikkoppgåver og for adaptive oppgave-stiar der elevane får presentert oppgåver baserte på tidlegare svar og automatiske kartleggingar.

Eventuelle endringar av programmeringsoppgåver på sikt, kunne også vore interessant å sjå nærare på. No vert læreverka utvikla for alle trinna meir eller mindre samtidig, og skal passe også for dei som går i siste del av skuleløypa og som har låg eller inga programmeringsferdigheit. Korleis vil då programmeringsoppgåvene sjå ut når dagens småskuleelevar kjem til ungdomsskulen med eit heilt anna ferdigheitsnivå innan programmering samanlikna med slik det er no?

## Kjeldeliste

- Aschehoug. (2021, 1. juni). *Matemagisk 8-10*. aunivers.no.  
<https://aunivers.no/marked/ungdomsskole/laeremidler-8.-10/matematikk>
- Berge, R. L., Sæterås, B. & Brandsæter, A. (2022). *Integrated Programming and Mathematics in Schools—A Solid Foundation for a Future Engineering Education?* Volda University Collage.
- Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M. & Rumble, M. (2012). Defining twenty-first century skills. I P. Griffin, B. McGaw & E. Care (Red.), *Assessment and Teaching of 21st Century Skills* (s. 17–66). Springer Netherlands.  
[https://doi.org/10.1007/978-94-007-2324-5\\_2](https://doi.org/10.1007/978-94-007-2324-5_2)
- Boesen, J., Helenius, O., Bergqvist, E., Bergqvist, T., Lithner, J., Palm, T. & Palmberg, B. (2014). Developing mathematical competence: From the intended to the enacted curriculum. *The Journal of Mathematical Behavior*, 33, 72–87.  
<https://doi.org/10.1016/j.jmathb.2013.10.001>
- Botten, G., Daland, E. & Dalvang, T. (2008). Tilpasset matematikkopplæring i en inkluderende skole. *Tangenten*, (2), 23–27.
- Bratberg, Ø. (2021). *Tekstanalyse for samfunnsvitere* (3. utg.). Cappelen Damm akademisk.
- Bratholm, B. (2001). *Godkjenningsordningen for lærebøker 1889-2001, en historisk gjennomgang*. Høgskolen i Vestfold. <https://www-bib.hive.no/tekster/hveskrift/notat/2001-05/not5-2001-02.html>
- Brennan, K. & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada.  
<http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Burner, T., Alvestad, K. C., Brazier, E., Gustavsen, T. S., Kacerja, S., Ruud, L. C., Salvesen, G. S. & Schipor, D. (2022). EvaFag 2025: Evaluering av Fagfornyelsen : Realiseringer av læreplanen i spesifikke fag. Arbeidspakke 1, Delrapport 1. I 78 [Report]. Universitetet i Sørøst-Norge. <https://openarchive.usn.no/usn-xmlui/handle/11250/3001411>
- Cappelen Damm. (u.å.). *Matematikk 8-10 fra Cappelen Damm*. Henta 28. mars 2023 frå <https://www.cappelendammundervisning.no/verk/matematikk-8-10-fra-cappelen-damm-153429>
- Carlsson, S., Öberg, B., Hake, K.-B. & Hagen, M. B. (2007). *Tetra 9* (1. utg.). Fagbokforlaget.
- Cohen, L., Morrison, K. & Manion, L. (2018). *Research methods in education* (Eighth edition.). Routledge.
- Dalland, O. (2012). *Metode og oppgaveskriving* (5. utg.). Gyldendal Norsk Forlag.
- De nasjonale forskningsetiske komiteene. (2019, 10. februar). *Generelle forskningsetiske retningslinjer*. <https://www.forskningsetikk.no/>.  
<https://www.forskningsetikk.no/retningslinjer/generelle/>
- Dolonen, J. A., Kluge, A., Litherland, K. & Mørch, A. I. (2019). *Litteraturgjennomgang av programmering i skolen*. <https://www.duo.uio.no/handle/10852/76290>
- Duval, R. (2006). A Cognitive Analysis of Problems of Comprehension in a Learning of Mathematics. *Educational Studies in Mathematics*, 61(1/2), 103–131.
- Enge, O. & Valenta, A. (2022). Helhetlig matematikkundervisning. I T. S. Gustavsen, R. A. Rinvold, K. Hinna & T. Sundtjønn (Red.), *QED 1-7: Matematikk for grunnskolelærerutdanningen: Bind 1: Bd. Bind 1* (2. utgave., s. 619–657). Cappelen

- Damm akademisk.  
[https://www.nb.no/search?q=oaiid:"oai:nb.bibsys.no:999920207896202202"](https://www.nb.no/search?q=oaiid:)
- Engelsen, B. U. (2020). Læreplanens generelle del – et historisk perspektiv. *Norsk pedagogisk tidsskrift*, 104(2), 206–217. <https://doi.org/10.18261/issn.1504-2987-2020-02-10>
- Eriksen, A. E. & Bolme, J. T. (2021). *Fremmer nye læreverker i matematikk kjerneelementene i Fagfornyelsen?* [Masteroppgåve, Universitetet i Tromsø]. Universitetet i Tromsø.
- Eriksen, S. & Vos, P. (2022). Kjerneelementer og eksempeloppgaver. *Tangenten*, 33(3), 45–51.
- Fried, M. N. & Amit, M. (2015). Reform as an Issue for Mathematics Education Research: Thinking About Change, Communication, and Cooperation. I L. D. English & D. Kirshner (Red.), *Handbook of International Research in Mathematics Education* (3. utg.). Routledge.
- Gjone, G. (2003). Læreplaner og læreplanutvikling i matematikk. I B. Grevholm (Red.), *Matematikk for skolen* (s. 261–287). Fagbokforlaget.
- Grønmo, S. (2012). Kvalitative og kvantitative metoder: Begreper og distinksjoner. *Sosiologisk tidsskrift*, 20(1), 85–91. <https://doi.org/10.18261/ISSN1504-2928-2012-01-06>
- Grønmo, S. (2016). *Samfunnsvitenskapelige metoder* (2. utg.). Fagbokforl. [https://www.nb.no/search?q=oaiid:"oai:nb.bibsys.no:999919818099702202"&mediatype=bøker](https://www.nb.no/search?q=oaiid:)
- Gyldendal. (u.å.). *Maximum 9, 2. Utgave, Grunnbok*. Henta 28. mars 2023 frå <https://www.gyldendal.no/grs/maximum/9/maximum-9-2-utgave-grunnbok/p-10025937-no/>
- Heintz, F., Mannila, L., Nygård, K., Parnes, P. & Regnell, B. (2015). Computing at School in Sweden – Experiences from Introducing Computer Science within Existing Subjects. I A. Brodnik & J. Vahrenhold (Red.), *Informatics in Schools. Curricula, Competences, and Competitions* (s. 118–130). Springer International Publishing. [https://doi.org/10.1007/978-3-319-25396-1\\_11](https://doi.org/10.1007/978-3-319-25396-1_11)
- Hinna, K. R. C., Rinvold, R. A. & Gustavsen, T. S. (2011). *QED 5-10: Matematikk for grunnskolelærerutdanningen : B. 1: Bd. B. 1*. Høyskoleforl. [https://urn.nb.no/URN:NBN:no-nb\\_digibok\\_2018060148095](https://urn.nb.no/URN:NBN:no-nb_digibok_2018060148095)
- Hirsch, H. C. (2022). *Seks naturfaglæreres refleksjoner om innføringen av programmering i grunnskolen* [Masteroppgåve, Nord Universitet]. <https://nordopen.nord.no/nord-xmlui/bitstream/handle/11250/3022966/HenrikChristopherHirsch.pdf?sequence=1>
- Hjardar, E. (u.å.). *Programmering 8-10*. Cappelen Damm.
- Hjardar, E. & Pedersen, J.-E. (2021a). *Matematikk 9 Grunnbok* (Nynorsk[utgåve], utgåve 1.). Cappelen Damm.
- Hjardar, E. & Pedersen, J.-E. (2021b). *Matematikk 9 Oppgåvebok* (Nynorsk[utgåve], utgåve 1.). Cappelen Damm.
- Imsen, G. (2020). *Lærerenes verden: Innføring i generell didaktikk* (6. utg.). Universitetsforlaget.
- Jacob, S. & Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, 1(1), 1. <https://doi.org/10.26716/jcsi.2018.01.1.1>
- Kaufmann, O. T., Stenseth, B. & Forsström, S. (2022). Programmering i matematikkundervisningen. I T. S. Gustavsen, R. A. Rinvold, K. Hinna & T. Sundtjønn (Red.), *QED 1-7: Matematikk for grunnskolelærerutdanningen: Bind 1: Bd. Bind 1* (2. utgave., s. 545–617). Cappelen Damm akademisk. [https://www.nb.no/search?q=oaiid:"oai:nb.bibsys.no:999920207896202202"](https://www.nb.no/search?q=oaiid:)

- Kilpatrick, J. (2014). Competency Frameworks in Mathematics Education. I S. Lerman (Red.), *Encyclopedia of Mathematics Education* (s. 85–87). Springer Netherlands. [https://doi.org/10.1007/978-94-007-4978-8\\_27](https://doi.org/10.1007/978-94-007-4978-8_27)
- Kilpatrick, J. (Red.), Swafford, J. (Red.), Findell, B. (Red.), National Research Council, Division of Behavioral and Social Sciences and Education, Center for Education, & Mathematics Learning Study Committee. (2001). *Adding it up: Helping children learn mathematics*. National Academy Press. <https://ebookcentral.proquest.com/lib/hivolda-ebooks/detail.action?docID=3375421&pq-origsite=primo>
- Kleven, T. A. & Hjordemaal, F. (2018). *Innføring i pedagogisk forskningsmetode: En hjelp til kritisk tolking og vurdering* (3. utg.). Fagbokforlaget.
- Kongsnes, A. L. & Wallace, A. K. (2020a). *Matemagisk 8-10: Elevhandbok* (Nynorsk, 1. utgåve.). Aschehoug undervisning.
- Kongsnes, A. L. & Wallace, A. K. (2020b). *Matemagisk 9* (1. utgåve, nynorsk[utgåve].). Aschehoug undervisning.
- Kunnskapsdepartementet. (2018a, 26. juni). *Fornyelse innholdet i skolen*. Regjeringen.no. <https://www.regjeringen.no/no/dokumentarkiv/regjeringen-solberg/aktuelt-regjeringen-solberg/kd/pressemeldinger/2018/fornyelse-innholdet-i-skolen/id2606028/>
- Kunnskapsdepartementet. (2018b). *Kjerneelementer i fag*. <https://www.regjeringen.no/contentassets/3d659278ae55449f9d8373fff5de4f65/kjerne-elementer-i-fag-for-utforming-av-lareplaner-for-fag-i-lk20-og-lk20s-fastsatt-av-kd.pdf>
- Kvale, S. & Brinkmann, S. (2015). *Det kvalitative forskningsintervju* (3. utg.). Gyldendal akademisk.
- Leinwand, S., Brahier, D. & Huinker, D. (2014). *Principles to Actions: Ensuring Mathematical Success for All*. National Council of Teachers of Mathematics. <http://ebookcentral.proquest.com/lib/hivolda-ebooks/detail.action?docID=6478900>
- Lillejord, S., Nordahl, T. & Manger, T. (2010). *Livet i skolen: Grunnbok i pedagogikk og elevkunnskap. 2: Lærerprofesjonalitet*. Fagbokforlaget. [https://urn.nb.no/URN:NBN:no-nb\\_digibok\\_2013071008145](https://urn.nb.no/URN:NBN:no-nb_digibok_2013071008145)
- Mannila, L. (2017). *Att undervisa i programmering i skolan: Varför, vad och hur?* Studentlitteratur.
- Meld. St. 28 (2015–2016). *Fag – Fordypning – Forståelse: En fornyelse av Kunnskapsløftet*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/meld.-st.-28-20152016/id2483955/>
- Morgan, C. & Sfard, A. (2016). Investigating changes in high-stakes mathematics examinations: A discursive approach. *Research in Mathematics Education*, 18(2), 92–119. <https://doi.org/10.1080/14794802.2016.1176596>
- Mullis, I. V. S., Martin, M. O., Foy, P. & Arora, A. (2012). *Timss 2011 international results in mathematics*. TIMSS & PIRLS International Study Center.
- Møller, G., Bergsgard, N. A., Salvesen, G. S. & Burner, T. (2023). *EvaFag 2025: Evaluering av Fagfornyelsen Mellom frihet og struktur – evaluering av fagfornyelsen i grunnskolen. Arbeidspakke 2, Delrapport 1. I 121* [Report]. Universitetet i Sørøst-Norge. <https://openarchive.usn.no/usn-xmlui/handle/11250/3058284>
- Nagaraja, H. N. & Choudhary, P. K. (2017). *Measuring Agreement: Models, Methods, and Applications*. John Wiley & Sons, Incorporated. <http://ebookcentral.proquest.com/lib/hivolda-ebooks/detail.action?docID=5109071>
- Niss, M. (1996). Goals of Mathematics Teaching. I A. J. Bishop, K. Clements, C. Keitel, J. Kilpatrick & C. Laborde (Red.), *International Handbook of Mathematics Education* (Bd. 4, s. 11–47). Springer Netherlands. [https://doi.org/10.1007/978-94-009-1465-0\\_2](https://doi.org/10.1007/978-94-009-1465-0_2)

- Niss, M. & Blum, W. (2020). *The Learning and Teaching of Mathematical Modelling* (1. utg.). Routledge. <https://doi.org/10.4324/9781315189314>
- Niss, M. & Højgaard, T. (Red.). (2011). *Competencies and Mathematical Learning*. Roskilde University, Danmark.
- Niss, M. & Jensen, T. H. (2002). *Kompetencer og matematiklæring: Ideer og inspiration til udvikling af matematikundervisning i Danmark: Bd. nr 18-2002*. Undervisningsministeriet.
- Nordbakke, M. (2018). Utvikling av kjerneelementer. *Tangenten*, (4), 35–40.
- Nosrati, M. & Wæge, K. (2015). *Sentrale kjennetegn på god læring og undervisning i matematikk*. <https://www.matematikkcenteret.no/sites/default/files/attachments/resource/Sentrale%20kjennetegn%20p%C3%A5%20god%20l%C3%A6ring%20og%20undervisning%20i%20matematikk.pdf>
- NOU 2014: 7. (2014). *Elevenes læring i fremtidens skole—Et kunnskapsgrunnlag*. Kunnskapsdepartementet. <https://www.regjeringen.no/no/dokumenter/NOU-2014-7/id766593/?ch=8>
- OECD. (2018). *PISA 2022 mathematics framework (draft)*. <https://pisa2022-maths.oecd.org/ca/index.html>
- Opplæringslova. (1998). *Lov om grunnskolen og den vidaregåande opplæringa* (LOV-1998-07-17-61). Lovdata. <https://lovdata.no/dokument/NL/lov/1998-07-17-61>
- Postholm, M. B. & Jacobsen, D. I. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm akademisk.
- Remillard, J. T. (2005). Examining Key Concepts in Research on Teachers' Use of Mathematics Curricula. *Review of Educational Research*, 75(2), 211–246.
- Schoenfeld, A. H. (2007). What Is Mathematical Proficiency and How Can It Be Assessed? I A. H. Schoenfeld (Red.), *Assessing Mathematical Proficiency* (Bd. 53, s. 59–73). Cambridge University press.
- Scratch. (u.å.). *Scratch*. Henta 8. mai 2023 frå <https://scratch.mit.edu/>
- Sevik, K. (2016). *Programmering i skolen*. Senter for IKT i utdanningen. [https://www.udir.no/globalassets/filer/programmering\\_i\\_skolen.pdf](https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf)
- Sjøvoll, J. (2018). Masteroppgaven—Forskningsplanlegging. I M. Krogtoft & J. Sjøvoll (Red.), *Masteroppgaven i lærerutdanninga: Temavalg, forskningsplan, metoder* (2. utg., s. 21–25). Cappelen Damm akademisk.
- Skeie, J. H. (2016). *Scratch fra scratch: Enkel programmering for nybegynnere*. Kodegenet. [https://www.nb.no/search?q=oaiid:"oai:nb.bibsys.no:999919857395302202"&mediatype=bøker](https://www.nb.no/search?q=oaiid:)
- Skemp, R. R. (1976). Relational understanding and instrumental understanding. *Mathematics Teaching*, 77(1), 20–26.
- Stenhouse, L. (1975). *An introduction to curriculum research and development*. Heinemann.
- St.meld. nr. 30 (2003–2004). *Kultur for læring*. Utdannings- og forskningsdepartementet. <https://www.regjeringen.no/no/dokumenter/stmeld-nr-030-2003-2004-/id404433/>
- Strand, M. (2022). *Fagfornyelsens innføring av programmering i grunnskolen* [Masteroppgåve, Nord Universitet]. <https://nordopen.nord.no/nord-xmlui/bitstream/handle/11250/3023410/MortenStrand.pdf?sequence=1>
- Svingen, O. E. L. (2018). *Representasjoner i matematikk*. [https://www.matematikkcenteret.no/sites/default/files/attachments/Elever%20som%20presterer%20lavt/P4\\_M1Representasjoner-i-matematikk\\_fagtekst.pdf](https://www.matematikkcenteret.no/sites/default/files/attachments/Elever%20som%20presterer%20lavt/P4_M1Representasjoner-i-matematikk_fagtekst.pdf)
- Thrane, C. (2018). *Kvantitativ metode: En praktisk tilnærming*. Cappelen Damm akademisk.

- Tjora, A. H. (2018). *Viten skapt: Kvalitativ analyse og teoriutvikling*. Cappelen Damm akademisk.
- Tofteberg, G. N., Alseth, B., Bråthe, L. T., Stedøy, I. & Tangen, J. (2021). *Maximum 9, 2. utg.: Matematikk for ungdomstrinnet* (Nynorsk[utgåve], 2. utgåva.). Gyldendal.
- Turner, R., Dossey, J., Blum, W. & Niss, M. (2013). Using Mathematical Competencies to Predict Item Difficulty in PISA: A MEG Study. I M. Prenzel, M. Kobarg, K. Schöps & S. Rönnebeck (Red.), *Research on PISA: Research Outcomes of the PISA Research Conference 2009* (s. 23–37). Springer Netherlands. [https://doi.org/10.1007/978-94-007-4458-5\\_2](https://doi.org/10.1007/978-94-007-4458-5_2)
- Utdanningsdirektoratet. (u.å.-a). *2.2 Kompetanse i fagene*. Fastsett som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/overordnet-del/prinsipper-for-laring-utvikling-og-danning/kompetanse-i-fagene/>
- Utdanningsdirektoratet. (u.å.-b). *Fagfornyelsen*. Henta 9. januar 2023 frå <https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/>
- Utdanningsdirektoratet. (u.å.-c). *Finn eksamensoppgaver*. Henta 1. mars 2023 frå <https://sokeresultat.udir.no/eksamensoppgaver.html?query=matematikk&ExCatalogTypeName=Eksamensoppgaver&ExKL=Kunnskapsl%C3%B8ftet%202020&ExPeriodName=2022-1>
- Utdanningsdirektoratet. (u.å.-d). *Forsøkslæreplan i valgfag programmering (PRG1-01)*. Fastsett ved forskrift. Læreplanverket for Kunnskapsløftet 2006. Henta 9. januar 2023 frå <https://www.udir.no/laring-og-trivsel/lareplanverket/finn-lareplan/lareplan/>
- Utdanningsdirektoratet. (u.å.-e). *Grunnleggende ferdigheter—Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. Henta 4. november 2022 frå <https://www.udir.no/lk20/mat01-05/om-faget/grunnleggende-ferdigheter?TilknyttedeKompetansemaal=true&anchorId=GF5#GF5>
- Utdanningsdirektoratet. (u.å.-f). *Læreplanverket*. Henta 9. januar 2023 frå <https://www.udir.no/laring-og-trivsel/lareplanverket/>
- Utdanningsdirektoratet. (u.å.-g). *Om overordnet del*. Henta 24. januar 2023 frå <https://www.udir.no/lk20/overordnet-del/om-overordnet-del/?lang=no>
- Utdanningsdirektoratet. (2017). *Rammeplan for barnehagen*. <https://www.udir.no/laring-og-trivsel/rammeplan-for-barnehagen/>
- Utdanningsdirektoratet. (2019a). *Kjerneelementer—Læreplan i matematikk fellesfag vg1 praktisk (matematikk P) (MAT08-01)*. <https://www.udir.no/lk20/mat08-01/om-faget/kjerneelementer?lang=nno>
- Utdanningsdirektoratet. (2019b). *Læreplan i matematikk 1.–10. Trinn (MAT01-05)*. Fastsett ved forskrift. Læreplanverket for Kunnskapsløftet 2020. <https://www.udir.no/lk20/mat01-05>
- Utdanningsdirektoratet. (2019c, 27. mars). *Algoritmisk tenkning*. Utdanningsdirektoratet. <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet. (2019d, 18. november). *Hva er kjerneelementer?* <https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hva-er-kjerneelementer/>
- Utdanningsdirektoratet. (2021, 2. november). *Grunnskole*. <https://www.udir.no/tall-og-forskning/publikasjoner/utdanningsspeilet/utdanningspeilet-2021/grunnskole/>
- Utdanningsdirektoratet. (2022a, 29. juni). *Fag- og timefordeling og tilbudsstruktur for Kunnskapsløftet Udir-1-2022*. <https://www.udir.no/regelverkstolkninger/opplaring/Innhold-i-opplaringen/udir-1-2022/>

- Utdanningsdirektoratet. (2022b, 23. august). *Hvordan ta i bruk læreplanene?*  
<https://www.udir.no/laring-og-trivsel/lareplanverket/stotte/hvordan-ta-i-bruk-lareplanen/>
- Van den Brink, J. & Kaufmann, O. T. (2022). *En komplett innføring i programmering i matematikk*. Høyskoleforlaget.
- Waite, J. (2018). *Pedagogy in teaching Computer Science in schools: A Literature Review*.  
Lasta ned 2.3.2023 frå <https://royalsociety.org/computing-education>.
- Wall, D. (2006, 16. januar). *The Impact of High-Stakes Examinations on Classroom Teaching: A case study using insights from testing and innovation theory*.  
<https://www.semanticscholar.org/paper/The-Impact-of-High-Stakes-Examinations-on-Classroom-Wall/cdb6295af29428d9191b3354a21d2ecafe9ec218>



## Vedlegg 1: Kodeskjema

Kjerneelement	#	Variabel
Utforsking og Problemløysing (UP)	1	Avdekke mønster og samanhengar
	2	Dele eit problem opp i delproblem
	3	Vurdere bruk av digitale hjelpemiddel eller ikkje
	4	Utvikle ei metode i ein ukjent situasjon
Modellering og anvendingar (MA)	1	Knyte til dagleglivet, arbeidslivet eller samfunnet
	2	Matematisere – lage ein modell
	3	Kritisk vurdere ein modell
	4	Bruke ein modell i ein annan kontekst.
Resonnering og argumentasjon (RA)	1	Utvikle ei tankerekkje
	2	Grunngje eller bevise
Resonnering og argumentasjon (RK)	1	Bruke matematisk språk
	2	Bruke matematiske representasjonar
	3	Grunngje valet av representasjonsform
	4	Byte mellom representasjonar
Abstraksjon og generalisering (AG)	1	Nytte formelt symbolspråk eller formelle resonnement
	2	Oppdage matematiske samanhengar og generalisere
	3	Utforske tal, utrekningar og figurar. Formaliser med algebra og formålstenlege representasjonar.
Programmering (PR)	1	Lese/tolke koding (både pseudo og programmeringsspråk)
	2	Utføre koding (både pseudo og programmeringsspråk)
	3	Programmeringskonsept, samlevvariabel
	3.1	Sekvensar/framgangsmåtar
	3.2	Løkker
	3.3	Handlingar
	3.4	Parallellisme (eller multitasking)
	3.5	Kontrollstrukturar
	3.6	Operatorar
3.7	Data	