

MASTEROPPGAVE

Algoritmisk Tenkning i Matematikklærebøker på 4. Trinn

Fanny Borgås Olsen

15.05.2023

MAT405 Masteroppgave i matematikk

Grunnskolelærerutdanning for trinn 1-7

Høgskolen i Østfold

Sammendrag:

Temaet for denne masteroppgaven er algoritmisk tenkning. Målet med forskningen er å finne ut hvordan algoritmisk tenkning er ivaretatt i to matematikklærebøker for 4. trinn som er revidert etter LK20. Problemstillingen er: *Hvordan presenteres algoritmisk tenkning i trykte lærebøker i matematikk på 4. trinn?*

Å tenke algoritmisk er en problemløsningsstrategi som egner seg for å løse komplekse oppgaver på en effektiv måte, både med og uten datamaskin. De siste årene har flere land satset på å implementere algoritmisk tenkning i læreplanen. Det samme gjorde Norge, og det gjorde meg nysgjerrig på hvordan oppgavene i bøkene legger til rette for at eleven får arbeidet med strategien. Mange læreres forståelse for algoritmisk tenkning er mangelfull, noe som gjør det ekstra viktig at lærebøkene dekker temaet.

For å svare på problemstillingen har jeg valgt å gjennomføre en lærebokanalyse. Analysen gjennomføres i to læreverk; Multi og Matemagisk. Den deles inn i en horisontal analyse og en vertikal analyse. Den horisontale analyse gir meg svar på hvor mange oppgaver som kan fremme algoritmisk tenkning i lærebøkene. I den vertikale analysen blir oppgavene i ett kapittel i hvert læreverk analysert dypere, som gir svar på hvordan algoritmisk tenkning blir ivaretatt i oppgavene.

Resultatene fra den horisontale analysen avdekket at kapitlene som i størst grad inneholder oppgaver som fremmer algoritmisk tenkning er plassert lenger bak i bøkene, og inneholder færrest oppgaver.

Hovedfunnene fra den vertikale analysen viser at kun et fåtall av oppgavene ber eleven om å feilsøke. Når det gjelder i hvor stor grad algoritmisk tenkning og matematisk innhold er integrert i oppgavene, varierer svarene i de to kapitlene. Kapittelet i Multi integrerer matematiske temaer og algoritmisk tenkning i samme oppgaver, mens Matemagisk har et større antall oppgaver som fremmer algoritmisk tenkning, men ikke matematisk kompetanse. Dette er interessante funn som bør forskes videre på.

Abstract:

The topic of this master's thesis is computational thinking (CT). The aim of the research is to find out how CT is presented in two mathematics textbooks for 4th grade that have been revised according to LK20. The research question is: *How is CT presented in printed textbooks in mathematics at 4th grade?*

CT is a problem-solving strategy that is suitable for solving complex tasks in an efficient way, both with and without a computer. In recent years, several countries have invested in implementing CT in the curriculum. Norway did the same, and it made me curious about how the tasks in the books facilitate the student's work on the strategy. Many teachers' understandings of CT are deficient, which makes it extra important that the textbooks cover the topic.

In order to answer the question, have I chosen to carry out a textbook analysis. The analysis is carried out in the two textbooks: Multi and Matemagisk. It is divided into a horizontal analysis and a vertical analysis of the textbooks. In the horizontal analysis, I will find how many tasks that can promote CT in the textbooks. In the vertical analysis, the tasks in one chapter of each textbook are analyzed more deeply, which provides answers to how CT is presented in the tasks.

The results from the horizontal analysis revealed that the chapters that contain the most tasks that promote CT, are placed further back in the books and contain the fewest tasks. The main findings from the vertical analysis show that only a small number of the tasks urge the student to debug an algorithm. When it comes to the extent to which CT and mathematical content are integrated in the tasks, the findings in the two chapters vary. The chapter in Multi integrates mathematical concepts and CT in most of the tasks, while Matemagisk has a larger number of tasks that promote CT, but not mathematical competence. These are interesting findings that should be further researched.

Innholdsfortegnelse

1.0 Innledning	6
1.1. Det 21. århundret	6
1.2 Egne erfaringer	7
1.3 Algoritmisk tenkning	7
1.4 Læreplan	8
1.5 Lærebøker	8
1.6 Problemstilling og metode	9
1.7 Oppgavens oppbygning	9
2.0 Bakgrunn	10
3.0 Teori	13
3.1 Begrepsavklaring	13
3.1.1 Algoritmisk tenkning (AT)	13
3.1.2 Matematisk kompetanse	15
3.1.3 Problemløsning	16
3.2 Litteraturgjennomgang	16
3.2.1 AT i utdanning	16
3.2.2 AT og matematikk	17
3.2.3 ATO (AT-oppgave)	18
3.3 Rammeverk	19
3.3.1 Rammeverk for horisontal analyse	21
3.3.2 Rammeverk for vertikal analyse	22
4.0 Metode	24
4.1 Forskningsmetode – kvalitativ innholdsanalyse av lærebøker	24
4.2 Utvalg av lærebøker	25
4.3 Analysestrategier	26
4.3.1 Horisontal analyse	27
4.3.2 Vertikal analyse	30
4.4 Reliabilitet, validitet og etiske betraktninger	31
5.0 Analyse – presentasjon av funn	33
5.1 Horisontal analyse	33
5.1.1 Matemagisk 4	33
5.1.2 Multi 4	36
5.2 Vertikal analyse	38
5.2.1 Funn i Matemagisk 4b Programmering og Multi 4B Mønster og algoritmer	38
5.2.2 Å følge en algoritme	42

5.2.3 Å lage en algoritme	44
5.2.4 Feilsøke	45
5.2.5 Å finne et mønster	47
5.2.6 Forklare	49
5.2.7 Matematisk innhold.....	50
6.0 Diskusjon.....	52
6.1 Diskusjon om funnene fra den horisontale analysen.....	52
6.2 Diskusjon om funnene fra den vertikale analysen.....	53
6.2.1 Oppgavetyper og AT-ferdigheter	53
6.2.2 Matematisk innhold og matematisk kompetanse	56
7.0 Konklusjon	59
8.0 Litteraturliste.....	61
Vedlegg.....	65

1.0 Innledning

I denne masteroppgaven skal jeg undersøke hvordan algoritmisk tenkning blir presentert i trykte matematikklærebøker på småskoletrinnet. Bøkene skal være forankret i den nye læreplanen, heretter kalt LK20. Algoritmisk tenkning (AT) er en nødvendig ferdighet alle bør beherske for å kunne leve i, og forstå, den digitale verdenen vi lever i (Csizmadia et al, 2015). Strategien omtales som en «soft start» på senere involvering med digitale verktøy, som kunstig intelligens (Nordby et al., 2022a). Den innebærer å dele opp et problem i delproblemer som kan løses på en logisk og strukturert måte (Kunnskapsdepartementet, 2019). Videre kan denne fremgangsmåten brukes for å løse andre, lignende problemer. Dette kan forenkle kommende møter med ukjente, komplekse problemer du umiddelbart ikke ser en løsning på. Å lære å lære handler om å kunne bruke ulike metoder for å tilegne seg kunnskap på en selvstendig måte (Kunnskapsdepartementet, 2017), og forskning viser at AT kan være en godt egnet problemløsningsmetode (Brackmann et al., 2017).

1.1. Det 21. århundret

Vi lever i et samfunn som vokser og utvikler seg raskt. Dagens og fremtidens arbeidsplasser krever andre ferdigheter av oss enn de gjorde tidligere. Skolens opplæringslov sier at skolen skal gi eleven mulighet til å utvikle ferdigheter og kompetanser som trengs for å delta i arbeid og fellesskap i samfunnet (Opplæringslova, 1998). For å møte de nye kravene, behøvde skolen en fornyelse. Ludvigsen-utvalget hadde i oppgave å vurdere og komme frem til hvilke kompetanseområder som var nødvendig i fremtidens skole (NOU, 2015:8). De anbefalte fem kompetanseområder som burde være grunnlag for skolens fornyelse: fagspesifikk kompetanse, kompetanse i å lære, kompetanse i å kommunisere, samhandle og delta og kompetanse i å utforske og skape (NOU, 2015:8). I denne masteroppgaven vil jeg se nærmere på kompetanse i å lære. Det innebærer å kunne ta i bruk ulike læringsstrategier og regulere egen tenkning (NOU, 2015:8). En strategi som har vært belyst de siste årene er nettopp AT.

Hovedbegrunnelsen for å implementere AT i utdanningen i de fleste land er for å utvikle ferdigheter som er nødvendig for å delta i den digitale verdenen (Bocconi et al., 2016, s. 8). Digital kompetanse er ettertraktet i vårt digitale samfunn, og er tydeliggjort i LK20 som en grunnleggende ferdighet. Det innebærer blant annet å bruke regneark og programmering til å «utforske og løse matematiske problemer» (Kunnskapsdepartementet, 2019). Digital programmering er det mest brukte verktøyet for å utvikle AT-ferdigheter i undervisningen,

men oppgaver som ikke krever datamaskin er like verdifulle da dette kan øke elevens interesse for å lære digital programmering senere i utdanningsløpet (Brackmann et al., 2017). En viktig side av AT er at det er en problemløsningsstrategi som kan brukes for å forstå og løse komplekse problemer (Kunnskapsdepartementet, 2019).

1.2 Egne erfaringer

Som student på lærerutdanningen 1-7 har jeg stor interesse for elevens læringsutvikling i de første årene på skolen. Interessen min for AT er todelt. På den ene siden kan det trekkes tydelige koblinger til *programmering*, som er en ferdighet som er høyt ettertraktet i samfunnets digitale fremtid, også belyst i LK20. På den andre siden kan AT kobles til det «å lære å lære», hvor eleven lærer å utvikle strategier til å løse problemer. De øver seg på å bli gode problemløsere ved å være bevisste på prosessen med å løse et problem på en effektiv måte. Disse to ferdighetene gir eleven verktøy i deltagelsen i dagens kunnskapssamfunn.

Som praksisstudent og vikar på barneskolen har jeg erfart flere elever med begrenset utholdenhet og tro på egen mestring i matematikkfaget. I noen tilfeller virker det som at gode læringsstrategier er fraværende. Eleven kan dra nytte av å øve seg på strategier for å løse et problem på en effektiv og kreativ måte. Jeg lærte om AT på høyskolen, og forstod fort at det er en problemløsningsmetode som kan være nyttig for mange elever. Høgskolelærernes engasjement for AT gjorde meg oppmerksom på begrepet, og jeg erfarte at det var lite diskusjoner blant lærerne på barneskolen om AT; dette selv om det er belyst i både kjerneelementene og kompetansemålene i LK20. Av den grunn ble jeg nysgjerrig på om eleven får trening på tankestrategien ved hjelp av lærebøkene.

1.3 Algoritmisk tenkning

Wing (2006, s.1) definerte tidlig at AT er å omformulere et komplekst og vanskelig problem til noe som er mulig å løse. Det beskriver en kognitiv tankeprosess som engasjerer oss til å gå løs på oppgaver der man ikke umiddelbart ser en opplagt løsning. AT fungerer som en problemløsningsmetode for å løse et ukjent problem. Problemløsning er beskrevet i LK20 som både det å utvikle metoder for å løse ukjente problemer, og å kunne vurdere løsningenes gyldighet (Kunnskapsdepartementet, 2019). AT innebærer også å vurdere om problemet løses mest effektivt med digitale verktøy eller uten (Kunnskapsdepartementet, 2019).

Csizmadia et al. (2015, s. 6) trekker frem seks begreper som er viktige i AT. Disse er: logikk, algoritmer, dekomposisjon, mønstre, abstraksjon og evaluering. Dette er et sett med ferdigheter, eller evner, som til sammen beskriver en algoritmisk tenker. For å forstå hva en algoritmisk tenker gjør velger jeg også å presentere disse teknikkene: reflektere, kode, designe, analysere og anvende (Csizmadia et al., 2015, s. 9). Å arbeide med disse teknikkene kan sette i gang kognitive prosesser som utvikler evnene som er nevnt over. Disse tilnærmingene skal jeg bruke i analysen av studiet, og vil bli beskrevet nærmere i teori-kapittelet.

1.4 Læreplan

I 2020 ble den nye læreplanen LK20 innført i landets skoler. Skolen trengte en fagfornyelse for at eleven skal kunne utvikle de kunnskapene og ferdighetene som er nødvendig i dagens samfunn. Matematikkfaget skal «forberede eleven på et samfunn og arbeidsliv i utvikling ved å gi dem kompetanse i utforsking og problemløsning.» (Kunnskapsdepartementet, 2019).

Fagfornyelsen resulterte i LK20, som nå involverer begrepet AT. I kjerneelementene blir AT tydeliggjort som et viktig verktøy i prosessen med å bli en god problemløser. AT innebærer å «bryte ned et problem i delproblemer som kan løses systematisk» (Kunnskapsdepartementet, 2019).

Kompetansemål i matematikk på 4. trinn i LK20 har to mål som knyttes til utvikling av den algoritmiske tenkeren: «utforske og beskrive strukturer og mønstre i lek og spill» og «lage algoritmer og uttrykke dem ved bruk av variabler, vilkår og løkker» (Kunnskapsdepartementet, 2019).

1.5 Lærebøker

Lærebøker er en viktig formidler av læreplanen (Fan et al., 2013, s. 635). Det er enighet om at læreboka med sin sentrale rolle i undervisningen i stor grad er med på å bestemme hva slags matematisk innhold eleven blir eksponert for (Haggarty & Pepin, 2002, s. 3). Dette understreker lærebokas fremstående rolle i undervisning, og dens betydning for elevens læringsmuligheter.

Fagfornyelsen i 2020 resulterte i nye og reviderte lærebøker med formål om å dekke de nye kompetansemålene. Bøkene er en tolkning av læreplanen, der innholdet avgjør hva eleven

arbeider med i store deler av matematikkundervisningene. Derfor ønsker jeg å studere hvordan nye lærebøker presenterer AT i trykte lærebøker på småskoletrinnet. Jeg har valgt å analysere grunnbøkene på 4. trinn i læreverkene Multi og Matemagisk.

1.6 Problemstilling og metode

Problemstillingen min er: *Hvordan presenteres algoritmisk tenkning i trykte lærebøker i matematikk på 4. trinn?*

Jeg ønsker å få svar på hvordan AT framtrer i trykte lærebøker. Jeg har gjort et utvalg av to læreverk: Matemagisk og Multi. I dagens skole er det viktig at dette temaet har en sentral rolle i matematikkundervisningen. Det begrunner jeg med å presentere AT som en viktig ferdighet i dagens digitale kunnskapssamfunn, samt å se sammenhengen med matematisk kompetanse. Blant annet innebærer matematisk kompetanse å bruke begreper og velge effektive strategier for å løse et problem (Kunnskapsdepartementet, 2019). Derfor ser jeg etter om lærebøkene inneholder oppgaver som kan fremme AT. Jeg velger å se spesifikt på de trykte grunnbøkene i læreverkene, ikke tilleggsbøker og digitale nettressurser. Det kunne vært spennende å se på hva slags støtte lærerne får igjennom lærerveiledningen, men på grunn av omfanget på oppgaven har jeg valgt dette bort.

For å svare på problemstillingen skal jeg gjennomføre en kvalitativ lærebokanalyse av læreverkene Multi og Matemagisk sine grunnbøker for 4. trinn. Disse lærebøkene er utviklet etter målene og innholdet i LK20. Jeg er spesielt kjent med Multi fra praksis og vikarjobbing. Matemagisk er et læreverk jeg har mindre erfaring med, men som jeg synes har spennende elementer.

I analysen gjennomfører jeg en horisontal og en vertikal analyse. Dette gjør det mulig å kartlegge ATs plass i bøkene, både i hvor stor grad det gjennomsyrrer bøkene, og på hvilken måte.

1.7 Oppgavens oppbygning

I det neste kapittelet vil jeg beskrive bakgrunnen for temaet nærmere. Videre presenteres definisjoner av sentrale begreper i teorikapittelet, der jeg også går nærmere inn på tidligere forskning om temaet. Deretter gjør jeg rede for metoden for datainnsamling og dataanalysering i metodekapittelet. Her tydeliggjøres også rammeverkene som anvendes i

analysen. Videre presenteres funnene av analysen i analysekapittelet. Funnene drøftes i forbindelse med teorien i diskusjonskapittelet. Oppgaven avsluttes med en konklusjon som oppsummerer de viktigste funnene gjort i masteroppgaven.

2.0 Bakgrunn

I denne masteroppgaven skal jeg analysere lærebøker som er designet for å dekke målene i LK20, og jeg ønsker derfor å ta hensyn til hvordan AT fremstilles i de nasjonale styringsdokumentene. I dette kapitlet skal jeg beskrive hvordan LK20, som er fastsatt av Kunnskapsdepartementet (2019), og artikkelen «Algoritmisk tenkning» (Utdanningsdirektoratet, 2019) belyser AT.

I LK20 er det kompetansemål tilknyttet AT på alle trinn i grunnskolen. På småskoletrinnet handler målene om «å lage og følge regler og trinnvise instruksjoner i lek og spill» (Kunnskapsdepartementet, 2019). På 4. trinn introduseres programmeringsbegrepene: løkker, vilkår og variabler, men det er ikke før på 5. trinn programmering nevnes. Selv om kompetansemålene er tilknyttet AT, nevnes ikke begrepet i noen av de.

Læreplanen anser AT som en viktig faktor i problemløsning, spesielt i prosessen med å dele problemer inn i delproblemer som kan løses systematisk (Kunnskapsdepartementet, 2019). De skriver at problemløsning «handler om at eleven utvikler en metode for å løse et problem de ikke kjenner fra før» (Kunnskapsdepartementet, 2019). AT utvider problemløsning med at du i tillegg skal kunne vurdere om løsningen løses best med eller uten digitale verktøy (Kunnskapsdepartementet, 2019).

Som skrevet i innledningen er AT presentert i kjerneelementene i LK20. Under kjerneelementet «Utforskning og problemløsning» knyttes AT til det å utvikle fremgangsmåter for å kunne løse problemer på en systematisk måte (Kunnskapsdepartementet, 2019). Det blir spesielt trukket fram å løse problemer ved å dele de opp til delproblemer og vurdere om de best kan løses analogt eller digitalt. Dette er for øvrig det eneste stedet begrepet AT nevnes i læreplanen.

Utenom lærerplanen har Utdanningsdirektoratet publisert en artikkel som beskriver begrepet nærmere. I denne artikkelen skriver de at AT er «å vurdere hvilke steg som skal til for å løse et problem, og å kunne bruke sin teknologiske kompetanse for å få en datamaskin til å løse (deler av) problemet» (Utdanningsdirektoratet, 2019). En algoritmisk tenker vurderer altså hvordan et problem kan løses ved hjelp av steg som kan følges analogt eller digitalt. I tråd

med kjerneelementet i LK20, er disse stegene en del av fremgangsmåten som brukes for å løse problemet på en systematisk måte. Både kjerneelementet og artikkelen trekker frem det å bruke teknologisk kompetanse for å vurdere om stegene kan følges av en datamaskin, eller om løsningsforslaget løses best av et menneske. De belyser grunnleggende digital kompetanse som et element i AT.

I artikkelen presenterer Utdanningsdirektoratet (2019) en plakat (se Figur 1) som viser viktige elementer ved den algoritmiske tenkeren. Plakaten gir oversikt over nøkkelbegreper og arbeidsmåter den algoritmiske tenkeren bruker. Nøkkelbegrepene blir beskrevet i korte trekk. Dekomposisjon handler om å dele problemet inn i delproblemer; det innebærer å bruke logikk for å analysere informasjonen i delproblemene (Utdanningsdirektoratet, 2019). Videre viser de til at ved hjelp av abstraksjon siles den viktige informasjonen fra den uviktige slik at det kan lages abstraksjoner og modeller av den virkelige verdenen (Utdanningsdirektoratet, 2019). Abstraksjoner og modeller er ikke nevnt i kjerneelementet, men inngår i andre kjerneelementer: «Modellering og anvendelser» og «Abstraksjon og generalisering». Disse kjerneelementene beskriver modeller og abstraksjoner, men kobler det ikke til AT. Kort fortalt forklarer kjerneelementene at en modell er en representasjon av virkeligheten i matematisk språk, og abstraksjon er formalisering av tanker og matematisk språk (Kunnskapsdepartementet, 2019).

Utdanningsdirektoratet (2019) trekker frem arbeidsmåtene som minst like viktige som nøkkelbegrepene. Arbeidsmåtene dreier seg om å være skapende og utforskende i arbeidet med å løse problemer, samt å oppdage og rette feil underveis (Utdanningsdirektoratet, 2019). For dette trengs det en kognitiv kondis som gjør at du holder ut, og er villig til å fortsette og prøve igjen (Utdanningsdirektoratet, 2019). Til slutt fremheves samarbeid, som innebærer å dele ideer og jobbe sammen med andre, som en essensiell del av den algoritmiske tenkerens arbeidsmåte (Utdanningsdirektoratet, 2019). Plakaten viser på denne måten forskjellige ferdigheter som kan arbeides med på ulike måter. Den strukturelle delen er like viktig som den utforskende og skapende delen. Kjerneelementene som tar for seg disse begrepene er «*Utforskning og problemløsning*» og så vidt i «representasjon og *kommunikasjon*», men heller ikke disse kobler de til AT.



Figur 1 (Utdanningsdirektoratet, 2019)

For å definere nøkkelbegrepene nærmere bruker jeg rapporten til Csizmadia et al. (2015).

Logikk lar eleven forstå ting ved å trekke konklusjoner basert på egen kunnskap og interne modeller. *Abstraksjon* handler om å gjøre et problem håndterbart ved å redusere unødvendige detaljer. *Evaluering* ses på som en prosess som sikrer at algoritmen er egnet til formålet. *Algoritmer* handler om tydelige løsninger med klare trinn. *Dekomponering* er å behandle et problem som et sett komponenter, der delene kan forstås og løses separat. Denne tankegangen gjør det mulig å kunne behandle komplekse problemer, som for eksempel å forstå nye situasjoner og å designe store systemer. *Mønstre* gjør det lettere å behandle nye problemer basert på tidligere problemløsninger. Det handler om å finne fellestrekk i gjenstander og å justere løsninger slik at de er gjeldende for flere lignende problemer.

Kort oppsummert kan AT ses på i to deler, der den ene delen har fokus på elevens strukturelle fremgangsmåte, og den andre delen på tilnærmingen eleven har til problemet.

Hovedelementene i den strukturelle delen er å forstå problemet ved å dele det opp i delproblemer, og håndtere delproblemene slik at problemet kan løses digitalt eller analogt. Hovedelementene i arbeidsmåtene er å ha en utforskende og skapende innstilling.

Med dette som grunnlag, er jeg interessert i å se hvordan norske lærebøker i matematikk presenterer AT. I neste kapittel bruker jeg internasjonal forskningsteori for å utvide kunnskapen om AT og lage et teoretisk rammeverk som brukes til å forstå begrepet og dens plass i utdanning.

3.0 Teori

I dette kapittelet viser jeg til relevant teori for oppgaven. Kapittelet er delt inn i tre deler: begrepsavklaring, tidligere forskning og teoretisk rammeverk. I begrepsavklaringen definerer jeg AT ved hjelp av internasjonal forskning. Ettersom AT ofte presenteres som en problemløsningsmetode kommer jeg også til å definere problemløsning. Deretter viser jeg til tidligere forskning om AT i skolesammenheng, både fra Norge og resten av verden. Delen om tidligere forskning inneholder forskning om AT sin plass i skolen, AT i matematikkundervisning og AT oppgaver (heretter kalt ATO). Til slutt presenterer jeg rammeverket som skal brukes i analysen. Dette rammeverket bidrar til en tydelig forståelse av begrepet, og skal hjelpe meg å finne AT i oppgaver. Teorien brukes til å diskutere funnene senere i oppgaven og bidra til å gi svar på problemstillingen som er «*Hvordan presenteres algoritmisk tenkning i trykte læreverk i matematikk på 4. trinn?*».

3.1 Begrepsavklaring

I dette delkapittelet defineres de sentrale begrepene for masteroppgaven. Begrepene er AT, programmering, matematisk kompetanse og problemløsning.

3.1.1 Algoritmisk tenkning (AT)

Det norske begrepet algoritmisk tenkning er det nærmeste vi kommer det engelske «computational thinking» (Bocconi et al., 2018, s. 16). I forskningen er det ikke enighet om hva AT innebærer (Nordby et al., 2022a, s. 28). Definisjonene spriker i to retninger. Den ene retningen forstår AT som noe mer enn programmering, og den andre kobler AT direkte til programmeringsferdigheter (Bocconi et al., 2018, s. 11). En viktig hendelse var da Wing (2006, s. 33) tidlig beskrev AT; hun fastslo at det er en rekke mentale verktøy som til sammen etablerer en ferdighet som er nødvendig for alle, ikke bare informatikere. Med disse mentale verktøyene kan vanskelige og komplekse oppgaver omformes til en oppgave vi vet hvordan kan løses (Wing, 2006, s. 33). Bocconi et al. (2016, s. 15) deler Wing (2006) sin definisjon i to:

1. Algoritmisk tenkning er en tankeprosess og er dermed uavhengig av teknologi

2. Algoritmisk tenkning er en problemløsningsmetode som forutsetter ferdigheter i å designe løsninger som kan gjennomføres av mennesker, en datamaskin eller en kombinasjon.

Forskere påpeker at programmering og koding er verktøy som kan benyttes for å fremme AT (Bocconi et al., 2016, s. 7), men at oppgaver som ikke krever digitale verktøy også kan fremme AT i stor grad (Brackmann et al., 2017, s. 71). Det betyr at selv om AT innebærer å arbeide med algoritmer, innebærer det mer enn å bruke digitale verktøy. Bocconi et al. (2016, s. 7) skriver videre at det er enighet om at AT dekker mye mer enn programmering og koding, spesielt problemformulering og dekomponering. Mens digital kompetanse handler om bruken av digitale verktøy, handler AT om å forstå de underliggende kjernekonseptene. (Bocconi et al., 2016, s. 19).

Piatti et al. (2022, s. 2) definerer AT slik: «Roughly speaking, we could define computational thinking as the cognitive activity required to solve problems through algorithms». Wing (2006), Bocconi et al. (2016), og Piatti et al. (2022) er altså enige om at det er en kognitiv aktivitet, et tankesett, som brukes til å løse problemer. Som definisjonen til Bocconi et al. (2016) trekker frem, skal løsningen kunne utføres av mennesker eller datamaskiner, eller begge. Dette ses i sammenheng med Piatti et al. (2022) sin definisjon som bruker ordet algoritmer. Jeg velger derfor å definere AT slik: en tankeprosess som innebærer å omforme et komplekst problem til et håndterbart problem, og løse de ved hjelp av algoritmer som utføres med og/eller uten hjelp av datamaskiner.

Til tross for den store internasjonale satsningen på AT, er det uenigheter om hvilke ferdigheter og tilnærminger strategien innebærer (Bocconi et al., 2016, s. 17). I en litteraturanalyse fant Bocconi et al. (2016) begreper som gikk igjen. De kom frem til at AT innebære disse hovedbegrepene: abstraksjon, algoritmebehandling, automatisering, dekomposisjon, generalisering og feilsøking. Hovedbegrepene samsvarer med mye av Utdanningsdirektoratet (2019) sin beskrivelse som er presentert i kapittel 2, men begrepet automatisering er nytt. Med denne ferdigheten er det mulig å effektivisere repetitive prosedyrer ved å skape en automatisk gjentakelse ved hjelp av en datamaskin (Bocconi et al., 2016).

Aktiviteter som fremmer AT deles ofte inn i to tilnærminger: analoge aktiviteter og digitale aktiviteter (Brackmann et al., 2017, s. 65). Førstnevnte krever ingen form for digitale verktøy,

mens sistnevnte innebærer bruk av datamaskin, ofte i form av programmeringsoppgaver. (Brackmann et al., 2017, s. 65). Disse tilnærmingene beskrives nærmere senere i oppgaven.

3.1.2 Matematisk kompetanse

Denne masteroppgaven tar for seg AT i matematikkfaget, som gjør det relevant å se begrepet i sammenheng med matematisk kompetanse.

Matematisk kompetanse beskrives av Kilpatrick et al. (2001) som summen av fem ulike komponenter, eller tråder: begrepsmessig forståelse, beregning, anvendelse, resonnering og engasjement (Oversettelse fra Matematikksenteret). Disse trådene utvikles ikke hver for seg, men er integrerte i hverandre. Jeg skal kort beskrive de fem trådene.

Den første tråden er *begrepsmessig forståelse*, og handler om at eleven har forståelse for matematiske begreper og ideer, sammenhengen mellom dem og i hvilke kontekster de kan brukes (Kilpatrick et al. 2001, s. 118). Det vil si at begrepsmessig forståelse innebærer mer enn å pugge begreper, man skal også kunne bruke de i ulike kontekster. *Beregning* er evnen til å utføre prosedyrer på en effektiv og fleksibel måte (Kilpatrick, 2001, s. 121). Eleven bruker sin begrepsmessige forståelse til å ta hensiktsmessige valg som gjør prosedyren effektiv og nøyaktig. Et eksempel kan være å bruke løkke (gjenta en kode) i en algoritme, istedenfor å skrive koden «gå ett skritt frem» flere ganger etter hverandre. Kilpatrick et al. (2001, s. 124) sammenligner *anvendelse* med problemløsning. Eleven skal kunne formulere, representere og løse matematiske problemer (Kilpatrick et al., 2001, s. 124). Det vil si at eleven forstår problemet basert på tidligere erfaringer, og planlegger en løsning deretter. Dette gjelder spesielt i oppgaver der eleven ikke tydelig ser hva problemet er, og hvordan det kan løses (Kilpatrick et al., 2001, s. 124).

Eleven kan lære seg *resonnering* fra tidlig barneskolealder, og det dreier seg om å tenke logisk og begrunne sine matematiske ideer (Kilpatrick et al., 2001, s. 129-130). Eleven bruker logikk for å rettferdiggjøre og tydeliggjøre de matematiske valgene han tar. Den siste tråden i trådmodellen er *engasjement*. I denne ligger tiltroen til at matematikk er nyttig og at den er ikke tilfeldig (Kilpatrick et al., 2001, s. 131). Det er regler og strategier som kan læres og mestres. Elevens engasjement i matematikk påvirker de andre trådene, og omvendt. Høyt engasjement kan gi eleven mestringstro og utvikle for eksempel den begrepsmessige forståelsen, på samme måte som et mislykket forsøk på å løse et problem kan svekke engasjementet.

3.1.3 Problemløsning

Som tidligere nevnt har LK20 valgt å koble AT sammen med problemløsning. Derfor velger jeg å definere problemløsning kort i dette avsnittet.

Ifølge Szetela & Nicol (1992) er et problem en oppgave der eleven ikke umiddelbart ser en løsning. Oppgaven kan altså ikke ha en klar og tydelig løsningsmetode for å kunne bli kalt et problem. I motsetning til en oppgave eleven kjenner løsningen til, er problemløsning en mer omfattende prosess med å finne veien til målet. Det krever at eleven bruker forkunnskaper på en effektiv måte for å forstå og finne en løsning til problemet. For å få en oversikt over prosessen velger jeg å trekke frem Pólya (1971) sine fire steg. Disse er: å forstå problemet, å lage en plan, å følge planen og å se tilbake (Pólya, 1971).

I det første steget må eleven forstå problemet, hvilket vil si å se tydelig hva oppgaven faktisk ber om (Pólya, 1971, s. 6). Pólya (1971) trekker frem viktigheten av at vanskelighetsnivået tilpasses eleven som skal løse problemet, slik at det ikke blir for vanskelig å forstå. I neste steg må eleven lage en plan som kan løse problemet. Dette steget krever ifølge Pólya (1971) mye av eleven; han må bruke forkunnskaper i tillegg til å holde på konsentrasjonen. I denne delen av prosessen må eleven finne ut om deler av problemet kan minne om noe han har gjort fra før (Pólya, 1971, s. 34). Da kan eleven finne en løsning på en del av problemet, og kommer videre i problemløsningen. Eleven må deretter følge planen. Her er det viktig at stegene i planen er korrekte, noe eleven kan trenge å bli minnet på av læreren (Pólya, 1971, s. 13). Når problemet er løst, er det viktig at eleven ikke lukker boka og sier seg ferdig, men at han ser tilbake (Pólya, 1971, s. 14). Eleven effektiviserer løsningen ved å gjøre den så enkel som mulig, og kan i prosessen oppdage bedre løsninger (Pólya, 1971, s. 36).

3.2 Litteraturgjennomgang

Her vil jeg vise til tidligere forskning om AT i undervisningssammenheng.

3.2.1 AT i utdanning

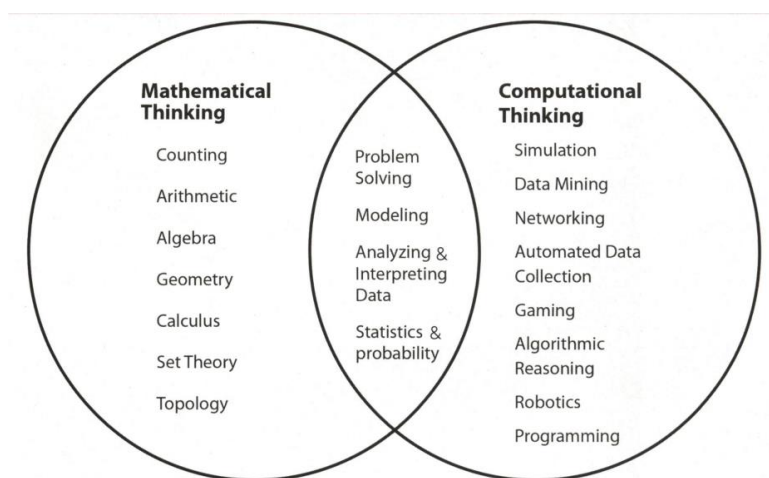
Ulike land har ulike meninger om hvor AT passer inn i opplæringen. Bocconi et al. (2016, s. 9) sine funn tilsier at de fleste land integrerer AT i ungdomsskolen, mens integrering i barneskolen er en økende trend. De fant også at AT i grunnskolen ofte integreres i eksisterende fag, mens det på høyere trinn er vanlig presentere det i et eget emne (Bocconi et al., 2016, s. 9). Norge har valgt at AT skal integreres i eksisterende fag som matematikk, musikk og naturfag, der matematikk har hovedansvar. I Norge er målet med involveringen av

AT i undervisning å fremme logisk tenkning, problemløsningsferdigheter og forståelse for samfunnet og teknologiens rolle i det (Bocconi et al., 2018, s. 23).

3.2.2 AT og matematikk

Nordby et al. (2022a, s. 39) forsket på læreres forståelse av AT, og konkluderte med at lærere sliter med å fastslå hva AT egentlig er, og hvilken sammenheng det har til matematikk. Derfor er det viktig å inkludere forskning som har tatt for seg koblingen mellom AT og matematikk.

Gandanidis (2017 s. 78) oppdaget i sin forskning en tendens å forstå AT som et isolert emne, og ikke integrert i eksisterende emner. Israel og Lash (2020) fant i sin forskning at lærere ofte, i 46,8% av tilfellene, lager undervisningsopplegg der matematikk og AT er isolert fra hverandre, altså ikke integrert. Dette bygger på faktumet at mange lærere ikke ser sammenhengen mellom AT og matematikk. Videre i forskningen fant de at 23,4% av undervisningsoppleggene integrerte matematikk og AT fullt ut (Israel & Lash, 2020), som beviser at full integrering heller ikke er uvanlig.



Figur 2 (Sneider et al., 2014)

AT har mye til felles både med matematisk tenkning og matematisk kompetanse. Matematisk tenkning er i likhet med AT et sett med ferdigheter. Disse ferdighetene utvikles fra barnets første leveår og videre igjennom utdanningen (Piatti et al., 2022). Sneider et al. (2014) sammenligner matematisk tenkning og AT (se figur 2). Matematisk tenkning skiller seg fra AT ved å inneholde matematiske begreper som telling og algebra. Begrepene de har til felles er problemløsning, modellering, analyse og tolkning av data og statistikk og sannsynlighet. Grover og Pea (2013, s. 40) oppdaget i sin litteraturgjennomgang at det at det var stilt

spørsmål om AT skilte seg nok fra andre typer tenkningsferdigheter barn utvikler i skolen. Likevel mener flere forskere at selv om matematisk tenkning har mange fellestrekk med AT, utvider AT tenkningen på en unik måte (Grover & Pea, 2013, s. 40).

Matematisk kompetanse deles, som nevnt tidligere, i fem tråder: begrepsmessig forståelse, beregning, anvendelse, resonnering og engasjement. Flere av disse trådene har flere AT-komponenter i seg. Begrepsmessig forståelse handler om å se sammenhengen i matematiske begreper og hvordan de brukes. Når det gjelder AT sin tilknytning til matematiske emner har tidligere forskning vist at AT-oppgaver ofte knyttes til arbeid med mønster, inkludert tallrekker (Bråting & Kilhamn, 2022). Beregning sammenlignes med å lage og utføre algoritmer. Anvendelse er å bruke matematikken i ulike situasjoner. Det handler om å ta valg som passer til settingen. Resonnering kan være å tenke og vurdere algoritmer og hvordan de påvirker utfall.

3.2.3 ATO (AT-oppgave)

For å kunne analysere innholdet i lærebøkene trenger jeg å gå nærmere inn på hva slags oppgavetyper som kan bidra med å lære eleven å bruke AT som strategi. Bråting og Kilhamn (2022) delte oppgavetyperne inn i seks typer: følge en prosedyre, finne ut, finne feil, skape, forklare og forestille seg. Funnene i studiet deres tilsa at de fleste oppgavene i de analyserte lærebøkene er kategorisert som «følge en prosedyre», etterfulgt av «skape» som innebærer at eleven lager stegvise instruksjoner (Bråting & Kilhamn, 2022, s. 601). De fant færrest antall oppgaver som passet i kategoriene «forestille seg» og «finne feil» (Bråting & Kilhamn, 2022, s. 601).

Nordby et al. (2022b) gjennomførte en litteraturanalyse som gir et overblikk over forskning på AT i matematikk på barneskolen. De fant at ATO ofte baserer seg på problemløsning. De deler ATO inn i to kategorier som går igjen i forskningen:

1. Ferdighetsorienterte aktiviteter
2. Prosessorienterte aktiviteter

De ferdighetsorienterte aktivitetene fokuserer på innlæring av ferdigheter som dekomponering og abstraksjon, mens de prosessorienterte aktivitetene tar for seg de mer overordnede temaet i AT, som utforskning, kommunikasjon og kreativitet (Nordby et al., 2022b). Den førstnevnte kategorien kan kjennes igjen i aktiviteter der for eksempel hensikten er å lage algoritmer. Den

andre kategorien kan kjennes igjen i aktiviteter der eleven utforsker ulike løsninger, og ofte samarbeide med andre. Nordby et al. (2022b) fant i flere av artiklene at prosessorienterte aktiviteter medbringer verdifulle faktorer inn i arbeidet med AT. For eksempel bidro det positivt til elevens engasjement i matematikk, som er en av trådene i trådmodellen Kilpatrick et al., (2001) utviklet for matematisk kompetanse. Nordby et al. (2022b) oppdaget at de ferdighetsorienterte aktivitetene dominerte, og at en mulig forklaring på det er at lærere er ukjente med AT og strever med å se sammenhengen mellom AT og matematikk (Nordby et al. 2022b).

Digitale aktiviteter er den mest brukte tilnærmingen for å lære bort AT (Brackmann et al., 2017, s. 65). Å løse oppgaver med digitale verktøy gir god trening på de sentrale nøkkelbegrepene i AT, spesielt algoritmedesign. Analoge aktiviteter er aktiviteter som ikke krever tilgang på digitale verktøy (Brackmann et al., 2017, s. 65). I denne masteroppgaven avgrensers jeg aktiviteter til å kun gjelde de som fremmer AT. Eksempler på slike oppgaver er logiske spill, fysiske bevegelser og andre aktiviteter som bygger på AT-strategien (Brackmann et al., 2017, s. 1). Analoge aktiviteter ble tidlig kritisert for å ta ifra eleven muligheten til å ha avgjørende dataopplevelser (Grover & Pea, 2013). Nyere forskning har derimot funnet mange fordeler med analoge aktiviteter. Relkin et al. (2021) og Brackmann et al. (2017) konkluderte med at analoge aktiviteter promoterer AT. Fordelen med disse aktivitetene er at det egner seg for yngre elever uten programmeringsferdigheter (Román-González, 2017), og gjør det mulig for eleven å utvikle seg som algoritmiske tenkere før de får opplæring i programmering. Aktivitetene er uavhengig av elektronikk og er derfor tilgjengelig for flere klasser enn det de digitale aktivitetene er (Brackmann et al., 2017, s. 65). Nordby et al. (2022b) anbefaler å ta analoge aktiviteter i betraktning fordi det kan kobles til hverdagsaktiviteter.

3.3 Rammeverk

Dette delkapittelet viser det teoretiske rammeverket gjeldende for denne masteroppgaven. Rammeverket tar for seg AT-ferdigheter, AT-praksiser og ATO. Det vises til ett rammeverk som passer den horisontale analysen, og ett som passer til den vertikale analysen. Felles for begge analysene er AT-ferdighetene Anderson (2016) viste til i sin artikkel. Det vil si at det er forstått i denne oppgaven at ATO kan fremme akkurat de ferdighetene. I den horisontale analysen er disse ferdighetene med på å danne en forståelse for hvordan en ATO kan se ut,

sammen med AT-praksiser. Denne forståelsen avgjør hva jeg ser på som en ATO og ikke, og blir brukt i analysen av alle bøkene. Dette presenteres tydeligere i delkapittel «3.2.1 Rammeverk for horisontal analyse». I den vertikale analysen avgjør jeg hvilke av AT-ferdighetene oppgaven kan fremme, i tillegg til hva slags ATO det er. De ulike ATO kategoriene er presentert i en tabell i delkapittel «3.2.2 Rammeverk for vertikal analyse». Jeg skal først utdype mer om AT-ferdighetene fra artikkelen til Anderson (2016), før jeg deler kapittel 3.2 i det todelte rammeverket.

I denne masteroppgaven er det AT-ferdighetene Anderson (2016) viser til som blir brukt: dekomposisjon, mønstergjenkjenning, abstraksjon, algoritmedesign og evaluering. Alle ferdighetene samsvarer med Utdanningsdirektoratets beskrivelse, bortsett fra logikk som Anderson ikke har med i sin artikkel. Med utdypning fra andre artikler beskrives AT-ferdighetene slik:

Dekomposisjon brukes for å forenkle problemet ved å dele det inn i delproblemer, som videre innebærer å kunne løse disse systematisk (Anderson, 2016, s. 228). Denne ferdigheten gjør det lettere for eleven å både forstå hva oppgaven ber om, og å finne løsninger på problemet.

Mønstergjenkjenning innebærer at eleven ser etter et mønster, enklere sagt repetisjon, i delproblemene (Anderson, 2016, s. 228). Så fort mønsteret er identifisert, kan eleven lage løsninger som kan brukes der mønsteret i problemet gjentar seg. På denne måten automatiseres noe av løsningen (Anderson, 2016, s. 228).

Abstraksjon handler om å sette overflødig informasjon i bakgrunnen, og viktig informasjon i forgrunnen (Csizmadia et al., 2015, s. 7). Gjennom abstraksjon lages en abstrakt representasjon av det oppdagede mønsteret som kan brukes som et «blueprint» på hvordan resten av mønsteret kan løses (Anderson, 2016, s. 228). Det kan lages modeller eller simulasjoner (Shute et al., 2014, s. 153). Videre innebærer abstraksjon å kunne hente og vurdere informasjon som er relevant for prosjektet (Shute et al., 2014, s. 153).

Algoritmedesign er å lage en algoritme som kan løse problemet (Anderson, 2016, s. 228). Algoritmen har alltid et start- og slutt punkt, og imellom disse må eleven bestemme stegene som må følges for å løse problemet (Anderson, 2016, s. 228-229). Shute et al. (2014, s. 153) trekker frem begreper som effektivitet og automatisering, hvilket innebærer å effektivisere løsningen ved å lage færrest mulig instruksjoner, og å automatisere løsningen så den kan brukes til å løse lignende oppgaver. I programmeringssammenheng kalles dette å kode (Anderson, 2016, s. 229).

Evaluering er det siste steget, og det er å vurdere om algoritmen er omfattende og effektiv (Anderson, 2016, 2. 229). Dersom algoritmen ikke inneholder all nødvendig informasjon, og/eller inneholder irrelevant informasjon må dette endres på. Dette kan ses i sammenheng med arbeidsmåten «feilsøking», der hensikten er å oppdage og rette opp i feil når løsningen ikke fungerer slik den skal (Shute et al., 2014, s. 153).

3.3.1 Rammeverk for horisontal analyse

Fire praksiser er belyst for å forklare hvordan elever arbeider med ATO: å være inkrementell og iterativ, testing og feilsøking, gjenbruk og remiksing og abstrahering og modularisering (Brennan & Resnick, 2012, s. 7). Den første praksisen er viktig når eleven skal designe et prosjekt. Den kjennetegnes ved at eleven tar små steg og repeterer disse med justeringer underveis (Brennan & Resnick, 2012, s. 7). På denne måten blir prosessen adaptiv og løsningsforslaget tilpasses og forbedres etter de oppdagelsene som oppstår underveis. Å ta små steg kan ses i sammenheng med ferdigheten *dekomposisjon*, der eleven bryter ned problemet i delproblemer. Jeg vil også trekke frem AT-ferdigheten *evaluering*, fordi eleven evaluerer løsningen for å forbedre den underveis.

Testing og feilsøking er en viktig praksis hos den algoritmiske tenkeren. Det innebærer at eleven tester algoritmen og avdekker og løser eventuelle feil (Brennan & Resnick, 2012, s. 7). Her brukes AT-ferdigheten *evaluering* for å vurdere om algoritmen er effektivt og inneholder det som er nødvendig.

Gjenbruk og remiksing handler om at eleven bygger videre på noen annens verk (Brennan & Resnick, 2012, s. 8). Dette gjør at eleven ikke alltid må finne opp kruttet selv, men kan bruke andres ideer som grunnlag i egne designprosesser. For at eleven skal kunne bruke andres verk må eleven se mønstret og reglene i det eksisterende verket for å vurdere hvordan det kan bygges videre på. Her brukes ferdigheten *mønstergjenkjenning*. Ferdigheter i *algoritmedesign* brukes videre for å utforme en algoritme som passer til oppgavens mål.

Den siste praksisen, modularisering og abstrahering, beskrives som å lage noe stort med en kolleksjon av små deler (Brennan & Resnick, 2012, s. 8). Det kan kobles til AT-ferdigheten *dekomposisjon*, der eleven bryter problemet ned i delproblemer, før det designes løsninger for delproblemene som brukes til å løse det store problemet. Ved hjelp av *abstrahering* fjernes overflødig informasjon, og hvert trinn blir mer konkret og lettere å utføre. *Algoritmedesign*

brukes videre for å sette sammen de trinnvise instruksjonene til en algoritme som kan følges av et menneske eller en datamaskin.

For å vurdere om innholdet i bøkene tar for seg AT bruker jeg AT-praksisene Brennan og Resnick (2012) presenterte i sin forskningsartikkel, og AT-ferdighetene Anderson (2016) presenterte. I kombinasjon med definisjonen av AT «å omforme et komplekst problem til et håndterbart problem, og løse de ved hjelp av algoritmer som utføres med og/eller uten hjelp av datamaskiner», utviklet jeg følgende kriterier for å vurdere om en oppgave kan fremme AT:

1. Forstå problemet ved å dele det inn i delproblemer
2. Lage, følge eller evaluere trinnvise prosedyrer på en strukturert og logisk måte, enten analogt eller digitalt, som også kan brukes til å løse andre lignende problemer.

Mer om hvordan jeg analyserer etter denne forståelsen av ATO beskrives i kapittel 4.3 *Analysestrategier*.

3.3.2 Rammeverk for vertikal analyse

Bråting og Kilhamn (2022) analyserte svenske lærebøkers programmeringsinnhold, og så på programmering som et aspekt av AT. I analysen oppdaget de at rammeverkene de valgte ikke passet til å analysere oppgaver i en lærebok, fordi de dreide seg om tenkning, og ikke oppgavetyper. Derfor lagde de et eget rammeverk med kategorier ut ifra det oppgavene i boken ba eleven om å gjøre. Kategoriene er «follow a procedure, figure out, find mistakes, form and create, explain and envisage» (Bråting & Kilhamn, 2022, s. 599).

Slik beskriver Bråting og Kilhamn (2022) kategoriene:

- a. «Follow a procedure» er oppgaver som oppfordrer eleven til å følge en algoritme eller et mønster
- b. «Figure out» er oppgaver som ber eleven om å finne regelen eller mønsteret som fører til en gitt handling
- c. «Find mistakes» er oppgaver der eleven skal feilsøke en kode/algoritme/prosedyre
- d. «Form and create» er oppgaver der eleven skal utvikle instruksjoner eller lage mønstre
- e. «Explain» er oppgaver som legger til rette for at eleven forklarer en prosedyre eller et mønster

- f. «Envisage» er oppgaver som ber eleven om å forestille seg utfallet av algoritmen dersom vilkår og verdier endres

Jeg bruker disse oppgavetyperne for å forstå hvordan en ATO kan se ut, og oversetter de slik: å følge en algoritme, å lage en algoritme, å finne et mønster, å forklare en algoritme og å forestille seg et utfall av en algoritme. Disse kategoriene representerer ulike oppgavetyper som krever at eleven bruker AT-strategien. Kategoriene er plassert i analyserammeverket for den vertikale analysen (se Tabell 1). Hver oppgave analyseres og blir plassert i den kategorien den tilhører. Den siste kategorien er «matte», som jeg har inkludert for å kartlegge om AT-oppgaven har matematisk innhold. Det innebærer at den inneholder matematiske begreper og matematiske ideer som gjør at eleven kan fremme sin matematiske kompetanse. Jeg er interessert i å se hvordan de ulike ATO kan fremme AT-ferdigheter; derfor innebærer analysen også å vurdere hvilke AT-ferdigheter oppgaven utfordrer eleven til å bruke. Ferdighetene er: dekomposisjon, mønstergjenkjenning, abstrahering, algoritmedesign og evaluering (Anderson, 2016).

Tabell 1

Oppgave- nummer	Følge	Lage	Finne	Forklare	Forestille seg	Matte
--------------------	-------	------	-------	----------	-------------------	-------

4.0 Metode

I dette kapittelet skal jeg presentere metoden som brukes for å hente og analysere data som skal brukes for å svare på problemstillingen: hvordan presenterer trykte lærebøker i matematikk algoritmisk tenkning?

Starten av metodekapittelet vil fortelle om forskningsmetoden som er valgt, før utvalget av lærebøker begrunnes. Videre presenteres analysestrategiene etterfulgt av analyserammeverket som er en kombinasjon av Udir sin modell for AT og kategoriene i artikkelen til Nordby et al. (2022b). Kapittelet avsluttes med redegjørelse for hvordan jeg har ivaretatt oppgavens reliabilitet og validitet.

4.1 Forskningsmetode – kvalitativ innholdsanalyse av lærebøker

I denne masteroppgaven har jeg valgt å ha en kvalitativ tilnærming til datamaterialet. En slik tilnærming gjør det mulig å analysere datamaterialet i dybden, i motsetning til en kvantitativ tilnærming som krever en større mengde datamateriale som kan generaliseres. Likevel vil noen kvantitative elementer dukke opp i forskningen, spesielt i den horisontale analysen der et større antall oppgaver analyseres. Det vil si at selv om jeg bruker tall og prosenter for å presentere funnene mine, er ikke datamaterialet stort nok til å kunne kalle det en kvantitativ forskning. For å svare på problemstillingen min må jeg gå i dybden i lærebøkene, og forskningsmetoden er derfor kvalitativ.

I en kvalitativ forskningsmetode beskrives virkeligheten enten gjennom sitater fra andre eller observasjoner forskeren gjør selv (Postholm & Jacobsen, 2018, s. 89). Formålet med min forskning er å undersøke hvordan trykte lærebøker i matematikk presenterer AT. Derfor har jeg valgt å gjennomføre en innholdsanalyse av trykte lærebøker. En kvalitativ innholdsanalyse gjør det mulig å beskrive datamaterialet gjennom et perspektiv som er spesifisert av problemstillingen (Schreier, 2012, s.3-4), som i dette tilfellet er innholdets presentasjon av AT.

Schreier (2012, s. 1) definerer kvalitativ innholdsanalyse som en systematisk måte å beskrive kvalitativt datamateriale gjennom å klassifisere det som tilfeller av kategorier i et koderammeverk. Det kan ses i sammenheng med at Cohen et al. (2007, s. 476) skriver at innholdsanalyse inneholder koding, kategorisering, sammenligning og konkludering. Det vil si at innholdet i lærebøkene vil bli kategorisert ut ifra fenomenet som skal undersøkes, i dette

tilfellet AT. De aktuelle kategoriene som brukes i analysen er presentert i kapittel 3.2 Rammeverk, og gjøres videre rede for i kapittel 4.4 Analysestrategier.

4.2 Utvalg av lærebøker

Jeg har valgt å analysere to matematiske læreverk på småskoletrinnet. Å velge to læreverk fremfor ett gir et mer oversiktlig bilde av hvordan AT presenteres i lærebøker som brukes i Norge. Hver bok blir analysert både horisontalt og vertikalt. Den horisontale analysen kartlegger bøkens innhold som en helhet, og den vertikale går i dybden i fenomenet som undersøkes (Charalambos et al., 2010, s. 122). Hovedhensikten med studien er ikke å sammenligne de, men å skape en bredere forståelse av hvordan AT blir presentert i innholdet. Eventuelt store forskjeller eller likheter vil allikevel diskuteres senere i masteroppgaven. Det er enighet om at lærebøkene er hovedformidler av læreplanen, og bøkene spiller en dominerende rolle i undervisningen (Fan et al., 2013, s. 635). Forskning viser også at de emnene som ikke dekkes i lærebøkene, heller ikke dekkes i klasserommet (Stein et al., 2007). Med lærebøkens viktige rolle i undervisning er det interessant å analysere hvordan innholdet i bøkene ivaretar matematiske emner.

Lærebøkene jeg har valgt er Multi og Matemagisk. Multi det læreverket jeg har fått mest erfaring med i utdanningsløpet, spesielt i forbindelse med praksis og vikarjobbing. Det er utgitt av Gyldendal og utviklet etter LK20. I den reviderte versjonen av Multi er det lagt mer vekt på spill, aktiviteter og utforskningsoppgaver (Gyldendal, 2021). Spill er ofte nevnt som positivt for utvikling av AT (Brackmann et al., 2017). Derfor har jeg valgt å analysere Multi.

Matemagisk er utgitt av Aschehoug. Jeg har litt erfaring med læreverket fra praksis, men også fra et tidligere arbeidskrav i studiet. Læreverk fanget interessen min med sitt fokus på utforskende og varierte oppgaver som virker relevant for LK20. På nettsiden opplyser de om at Matemagisk inneholder «snakke-matte» oppgaver som designet for at elevene skal forklare, resonnere og argumentere for hvordan de tenker (Aschehoug, u.å). I tillegg inneholder lærebøkene problemløsning og utforskende aktiviteter (Aschehoug, u.å). Det gjør det spennende å se om oppgavene legger opp til at eleven bruker AT som problemløsningsstrategi.

Læreverkens digitale ressurser er levende og kan utvikles kontinuerlig. De trykte læreverkene er derimot låst med fast innhold, som kun kan oppdateres ved å kjøpe inn nye bøker. Dette gjør at interessen min for å analysere innholdet i de trykte lærebøkene øker. Den

trykte delen av verkene inneholder grunnbøker fordelt på en A-bok og en B-bok, og diverse tilleggsbøker som oppgavebøker og parallellbøker. Tilleggsbøkene er ment for tilrettelegging og ekstra øving utover grunnboka. For å sikre at analysen min kun tar for seg det innholdet alle eleven har tilgang på, velger jeg å analysere grunnbøkene, og ikke tilleggsbøker. Det vil si at mulighetene eleven har til å utvikle sin AT gjennom tilleggsbøkene ikke inkluderes i denne masteroppgaven.

De siste kompetansemålene i LK20 på hvert trinn dreier seg om AT, enten analogt eller digitalt. Det er ikke før 5. trinn at ordet programmering dukker opp i kompetansemålene. Derfor ønsker jeg å se hvordan AT presenteres for eleven før de begynner på mellomtrinnet og offisielt introduseres for programmering. I denne oppgaven er det kun de trykte grunnbøkene for 4. trinn som analyseres. I en større oppgave ville det vært interessant å se på overgangen fra 4. trinn til 5. trinn, og hvordan AT presenteres i de ulike bøkene når eleven skal begynne med programmering.

4.3 Analysestrategier

Forskning innebærer mer enn datainnsamling. Datamaterialet, som er oppgavene i lærebøkene, skal analyseres med rammeverket jeg presenterte i teorikapittelet. Her er det deler av helheten som hentes ut og brukes for å konstruere ny kunnskap om fenomenet (Johannessen et al., 2016, s. 233). Det vil i denne masteroppgaven si at jeg henter ut ATO fra bøkene for å få kunnskap om hvordan bøkene presenterer AT. Analysen består av to deler, inspirert av Charalambous et al. (2010). Jeg velger å starte med en horisontal analyse, der jeg ser på boka som en helhet for å få en oversikt over AT sin plass i boka. Altså hvor mange ATO som finnes i de trykte grunnbøkene, og hvor de er plassert. Denne analysen har som hensikt å gi en oversikt over lærebøkens innhold, slik som kapitler og antall oppgaver (Charalambous et al., 2010, s. 122). I denne analysen kartlegges det hvor mange av oppgavene i boka som kan kategoriseres som AT-oppgaver, og i hvilke kapitler disse er plassert. Deretter gjennomfører jeg en vertikal analyse som er en dypere undersøkelse av det matematiske innholdet (Charalambous et al., 2010, s. 122). Her analyseres AT-oppgavene i ett kapittel i hvert læreverk med et rammeverk som går mer i dybden enn det horisontale rammeverket. Analysen har som formål å strukturere innholdet i boka etter kategoriene i analyserammeverket.

I dette delkapittelet presenterer jeg analysekriteriene og -kategoriene jeg bruker i den horisontale og den vertikale analysen, med eksempler.

4.3.1 Horisontal analyse

Målet med den horisontale analysen er å gi meg svar på i hvor stor grad ATO gjennomsyrrer lærebøkene. Jeg vil se hvor stor andel av oppgavene som er ATO, og i hvor stor grad de finnes i de ulike kapitlene.

Jeg startet tidlig med å kategorisere oppgavene som «ATO» og «Ikke ATO». I analyseprosessen fant jeg fort ut at skillet mellom disse er utydelig. Det ble derfor vanskelig for meg å avgjøre om en oppgave var ATO eller ikke. Denne observasjonen medførte at jeg endret kategoriene i rammeverket fra de to kategoriene «ATO» og «Ikke ATO», til tre kategorier med som tar hensyn til de utydelige skillene: «ATO i stor grad», «ATO i noen grad» og «ATO i liten grad». ATO i stor grad er oppgaver som dekker kriteriet jeg satt for en oppgave som kan fremme AT. ATO i noen grad er oppgaver som legger til rette for utvikling av en AT-ferdighet, men som likevel mangler kjennetegn og ikke oppfyller kriteriet. Oppgaver som kategoriseres ATO i liten grad er rutineoppgaver der eleven øver på matematiske regler, og som ikke oppfordrer eleven til å bruke AT-ferdighetene.

Her vil jeg gjenta definisjonen jeg på AT: å omforme et komplekst problem til et håndterbart problem, og løse de ved hjelp av algoritmer som utføres med og/eller uten hjelp av datamaskiner. Ut ifra denne definisjonen utformet jeg disse kriteriene:

1. Forstå problemet ved å dele det inn i delproblemer
2. Lage, følge eller evaluere trinnvise prosedyrer på en strukturert og logisk måte, enten analogt eller digitalt, som også kan brukes til å løse andre lignende problemer.

Typiske oppgaver som kan fremme AT er: oppgaver som legger opp til å kjenne igjen mønster, oppgaver der eleven må følge trinnvise steg og regler, oppgaver der eleven må lage en algoritme og oppgaver der eleven må evaluere et løsningsforslag



- 17 4. trinn har aktivitetsdag ved sjøen. De kan bruke to typer kanoer. De store tar seks personer og de litt mindre tar fire personer.

Finn minst to ulike måter å fordele 44 elever og 8 voksne på slik at alle får plass og alle båtene som brukes, er fulle.



Bilde 1 (Multi 4B: Regning)

Tabell 2

Kapittel	ATO i stor grad	ATO i noen grad	ATO i liten grad
8: Regning	U17		

Bildet ovenfor er et eksempel på en oppgave jeg kategoriserer i «ATO i stor grad» (se tabell). Det er tydelig at oppgaven ber eleven om å lage prosedyrer på en logisk og strukturert måte, som også kan brukes til å løse lignende problemer, som er kriteriet jeg analyserer etter. Oppgaven har et tydelig mål; at 44 elever og 8 voksne skal få plass i kanoer, og har noen regler å forholde seg til. Reglene, eller vilkårene, er at båtene kun har plass til enten seks eller fire personer. Det vil si at alle kombinasjoner som ikke oppfyller vilkårene utgår. Oppgaven presiserer at eleven skal finne minst to måter å fordele menneskene på, som betyr at eleven må lage to løsningsforslag, eller algoritmer, for å svare på oppgaven.

Eksemplet på bildet nedenfor er hentet fra Multi 4A. Det viser på en tydelig måte forskjellen på en oppgave som kan fremme AT i stor grad, og en som kan fremme AT i liten grad.

Oppgave 34 ber eleven om å velge ut varer som til sammen oppfyller kravet om hvor mye de skal koste. Eleven har valgmuligheter og kan utforske ulike løsningsforslag. En slik oppgave legger godt til rette for at eleven kan utvikle AT, og kategoriseres derfor som «ATO i stor grad». Oppgave 35 ber derimot eleven om å regne ut hvor mye varene koster til sammen, som gjør at eleven kun må følge ett trinn (å regne ut). Derfor plasseres denne oppgaven i kategorien «ATO i liten grad». Det vil være stor forskjell på hvordan en elev løser oppgaven «finn to varer som koster 200kr til sammen» og «hvor mye koster disse varene til sammen?». For å finne varene som oppfyller vilkåret som gjelder i den førstnevnte oppgaven, brukes logisk tenkning, testing, iterative prosedyrer og evaluering. Sistnevnte krever kun at eleven regner ut et addisjonsstykke for å finne svaret, og blir derfor ikke kategorisert som en ATO.



- 34 Velg blant varene over og finn
- to varer som koster 90 kr til sammen.
 - to varer som koster 200 kr til sammen.
 - to varer hvor den ene koster 50 kr mer enn den andre.
 - tre varer som koster omtrent 500 kr til sammen.

- 35 Hvor mye koster varene til sammen? Regn i hodet.

a 	b 	c
d 	e 	f

Bilde 2 (Multi 4A: Tall og regning)

Tabell 3

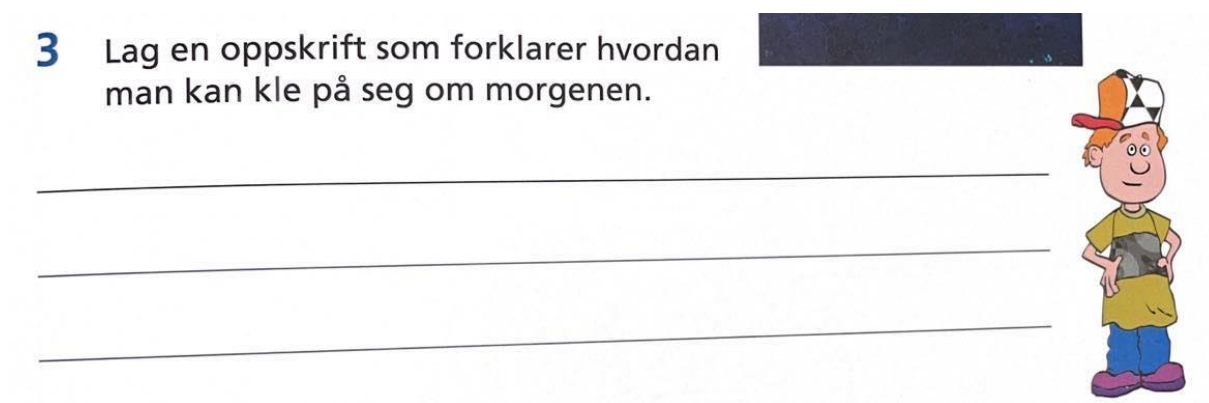
Kapittel	ATO i stor grad	ATO i noen grad	ATO i liten grad
1: Tall og regning	34		35

Slike type oppgaver kan dekke AT-begreper som dekomposisjon, mønstergjenkjenning, feilsøking og algoritmedesign. Andre eksempler er sortering, kode eller dekode hemmelig budskap, følge instruksjoner med regler (vilkår) og spill. En oppgave som ber deg om å sortere en mengde, ber samtidig eleven om å bruke en kodesekvens som gjentar seg, ved at hvert objekt må vurderes ut ifra vilkåret som er satt. Eleven bruker i disse tilfellene AT-strategien for å finne en løsning som er hensiktsmessig for oppgaven. Mange spill utfordrer den logiske tankegangen til spilleren, der eleven må tenke lurt for å vinne. Eleven tar i disse situasjonene hensiktsmessige valg for å gjøre spillets utfall best mulig. Derfor kan spill utvikle AT-ferdigheter i stor grad.

4.3.2 Vertikal analyse

Den vertikale analysen gjennomføres deskriptivt. Det vil si at datamaterialet struktureres og kategoriseres (Postholm & Jacobsen, 2016, s. 104). Postholm og Jacobsen (2016) foreslår å kode avsnittet på høyre siden av teksten, og skrive kategorien på venstre side. I både den horisontale og den vertikale analysen velger jeg å plassere oppgavene i en tabell som inneholder kategoriene for hvert rammeverk. Tabellene presenteres i de to neste underkapitlene 4.4.1 og 4.4.2.

Den vertikale analysen gjør det mulig å gjennomføre en dypere analyse av det matematiske innholdet (Charalambos, 2010, s. 122). Analyserammeverket som brukes i denne delen er tredelt, og er utviklet med inspirasjon fra Bråting og Kilhamn (2022), Anderson (2016) og Nordby et al. (2022), som alle er presentert i teorikapittelet. Kategoriene Bråting og Kilhamn (2022) brukte i sin lærebokanalyse kategoriserer hva slags ATO det er. For å vurdere hvilke AT-ferdigheter oppgaven kan fremme, brukes ferdighetene Anderson (2016) beskrev i sin artikkel: dekomposisjon, abstraksjon, mønstergjenkjenning, algoritmedesign og evaluering. Nordby et al. (2022) delte ATO inn i to tilnærminger, ferdighetsorientert og prosessorientert. Jeg vil derfor vurdere om oppgaven oppfordrer til at eleven kommuniserer, arbeider utforskende og kreativt.



Bilde 3 (Matemagisk 4B: Programmering)

Tabell 4

Følge	Lage	Feilsøke	Mønstergjenkjenning	Forklare	Forestille seg	Matte
	X					

Oppgaven i bilde 2 kategoriseres som «å lage en algoritme» (se tabell 4) fordi den ber eleven om skal lage en oppskrift. At eleven feilsøker algoritmen sin kan skje, men er ikke kategorisert som det da det ikke er formålet med oppgaven. Oppskriften, eller algoritmen, skal vise hvordan man kler på seg om morgenen, og eleven må dekomponere for å formulere delproblemer som er mer håndterbare. Han må da tenke på hva «å kle på seg» faktisk innebærer. Delproblemene brukes til å lage stegene algoritmen skal inneholde, som kan utvikle elevens ferdigheter i algoritmedesign. Oppgaven ber ikke om at eleven skal evaluere, abstrahere eller finne mønster i algoritmen. Evaluering kan oppstå i samtaler i klassen dersom eleven presenterer sin algoritme for andre, men oppgaven ber ikke om det. Tabell 5 viser hvordan jeg kan se om en oppgave kan fremme de ulike AT-ferdighetene.

Tabell 5

Dekomposisjon	Mønstergjenkjenning	Abstraksjon	Algoritmedesign	Evaluering
Oppgaven legger til rette for at eleven skal bryte et problem ned i delproblemer.	Oppgaven ber eleven om å finne et mønster eller regel i en algoritme.	Oppgaven ber eleven om å fjerne unødvendige steg, fjerne ukorrekte steg eller lage modeller/tabeller for å behandle informasjon	Oppgaven trener eleven på hva en algoritme inneholder. Hva som er start- og slutt punkt, og hvilke steg som passer imellom disse, avhengig av problemet som skal løses	Oppgaven ber eleven om å feilsøke eller effektivisere en algoritme. Det kan også være oppgaver der eleven må vurdere om en algoritme passer til formålet.

4.4 Reliabilitet, validitet og etiske betraktninger

I arbeidet med teorikapittelet, innså jeg hvor omfattende AT er. Forskernes ulike perspektiv på begrepet gjør at det kan virke flytende. Norge har valgt å se begrepet som en problemløsningsmetode som ikke er direkte knyttet til programmering, mens mye av forskningen ser begrepet i lys av programmering og koding. Etter mye litteratursøk fant jeg definisjoner jeg mener er passende for masteroppgaven.

I analysen oppdaget jeg hvor vanskelig det var å skille AT fra matematisk tenkning, da disse har mange likheter. For å kunne svare på problemstillingen var jeg nødt til å skille disse

tankesettene fra hverandre. På denne måten sikret jeg å kategorisere AT-oppgavene på en rett måte. Det var viktig å kun kategorisere en oppgave som en ATO dersom den direkte ga eleven mulighet til å tenke algoritmisk, ikke kun matematisk.

For å ivareta validiteten i denne oppgaven har jeg utformet et rammeverk ved hjelp av tidligere forskning med hensyn til den norske forståelsen av AT. Jeg utviklet kategorier og kriterier i et koderammeverk til den horisontale analysen, og ett til det vertikale. Med dette sikrer jeg en tydelig definisjon av begrepet og hvordan det kan oppdages i ulike oppgavetyper. Johannessen et al. (2016, s. 232) skriver at validitet «dreier seg om i hvilken grad forskerens framgangsmåter og funn på en riktig måte reflekterer formålet med studien og representerer virkeligheten». En kvalitativ innholdsanalyse er fleksibel, og krever at koderammeverket skreddersys til datamaterialet (Schreier, 2012, s. 7). For å sikre at koderammeverket passet til datamaterialet måtte jeg teste det gjentatte ganger med oppgavene i lærebøkene. I løpet av prosessen har kategoriene i koderammeverkene både blitt endret og justert flere ganger, helt til jeg fant kategoriene som passet til datamaterialet.

En utfordring jeg møtte på i utviklingen av rammeverket er at det ikke er tydelig enighet i forskningen om hva AT innebærer. Det har vært viktig for meg at rammeverket ikke distanserer seg for mye fra hvordan begrepet fremstilles i styringsdokumenter i Norge. Derfor har jeg nøye studert ulike forskningsartikler som resulterte i en rekke artikler som presenteres i teorikapittelet. Artikkelen danner en forståelsesramme for begrepet og for hvordan ulike AT-oppgaver kan se ut. Derfor kan det være mulig å for andre å få resultater som avviker fra denne studien dersom det brukes andre forståelser av AT.

Med tanke på den nylige integreringen av AT i læreplanen, var det vanskelig å ha en hypotese om hvor mange AT-oppgaver jeg kunne finne i analysen. At det er nytt i læreplanen, og fortsatt litt sprikende forskning, kan skape en forventning om at det er få slike oppgaver. På en annen side kan det tenkes at den store internasjonale satsningen har ført til at AT-strategien har fått stor plass i lærebøkene. Derfor har det vært viktig for meg å være nøytral og åpen for alle svar i analyseprosessen.

5.0 Analyse – presentasjon av funn

I dette kapitlet presenteres resultatene av analysen. For å svare på problemstillingen *Hvordan presenteres algoritmisk tenkning i trykte læreverker i matematikk på 4. trinn?*, har jeg som beskrevet i metodekapitlet gjennomført en horisontal og en vertikal analyse. Den horisontale analysen vil gi meg svar på i hvor stor grad lærebøkene integrerer ATO. I den vertikale analysen analyseres oppgavene ett kapittel i hvert læreverker. Disse funnene skal hjelpe meg med å svare på *hvordan* AT blir presentert. Resultatene fra den horisontale analysen legger grunnlag for valget av hvilket kapittel som skal analyseres vertikalt.

Analysekapitlet er delt inn i to underkapitler der resultatene fra den horisontale analysen presenteres først, før resultatene fra den vertikale analysen presenteres.

5.1 Horisontal analyse

Her presenteres funnene fra den horisontale analysen. Først presenteres funnene fra Matemagisk, deretter funnene fra Multi.

5.1.1 Matemagisk 4

Matemagisk 4 grunnbok er utgitt i 2022 av Aschehoug. Boka er en del av læreverket Matemagisk 1-10 som er nyskrevet etter fagfornyelsen, og følger LK20. Grunnboka er delt inn i to deler, en som er ment for høstsemesteret, og en for vårsemesteret. De kalles Matemagisk grunnbok 4A og Matematisk grunnbok 4B. I tillegg til disse tilbyr også læreverket ulike oppgavebøker, parallellbøker og en lærerveiledning. Disse analyseres ikke i denne masteroppgaven.

Matemagisk 4 grunnbok inneholder 313 oppgaver, jevnt fordelt i A og B bøkene. Innholdet er delt inn i 6 kapitler, der kapittel 1-3 tilhører bok A, og kapittel 4-6 tilhører bok B. Kapitlene heter Geometri, Multiplikasjon, Volum, Divisjon, De fire regneartene og Programmering. I den horisontale analysen har jeg plassert oppgavene i alle kapitlene i disse kategoriene «ATO i stor grad», «ATO i noen grad» og «ATO i liten grad». På denne måten ser jeg hvilke oppgaver som har potensiale til å fremme elevers algoritmiske tenkning, og i hvilke kapitler de forekommer.

Tabell 6

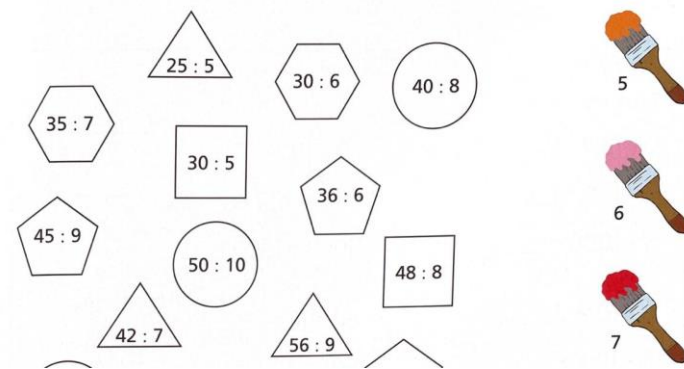
Matemagisk 4 grunnbok: 95 av 313 oppgaver (30,4%)	
4A: 47 av 156 oppgaver (30,1%)	4B: 48 av 157 oppgaver (30,6%)
1 Geometri: 15 av 53 oppgaver (28,3%)	4 Divisjon: 14 av 78 oppgaver (18%)
2 Multiplikasjon: 22 av 55 oppgaver (40%)	5 De fire regneartene: 14 av 59 oppgaver (23,7%)
3 Volum: 9 av 48 oppgaver (18,8%)	6 Programmering: 20 av 20 oppgaver (100%)

Av Matemagisk 4 sine totale 313 oppgaver, som er en blanding av grunnoppgaver, aktiviteter og «snakke matte», er 95 av de oppgaver som er kategorisert som «ATO i stor grad». Det vil si at 30,4% av oppgavene i boka kan bidra med å fremme elevers AT. AT-oppgavene, som er en blanding av grunnoppgaver, snakke-matte og aktiviteter, er fordelt jevnt mellom bok A og bok B, hvor de utgjør omtrent 30% av oppgavene i begge bøkene. Av de resterende oppgavene er 35 (11,2%) av de «ATO i noen grad» og 183 (58,5%) «ATO i liten grad».

Kapittelet 6 Programmering inneholder kun ATO. Det er likevel ikke det som inneholder flest ATO, men det er mer konsentrert da det ikke inneholder andre oppgaver som kategoriseres som ATO i liten grad. Det er verdt å nevne at kapittelet er skrevet av en annen forfatter enn de som har skrevet resten av boka, nemlig Ilse-Marie Restani. Dette kapittelet er plassert sist i boka og har færrest antall oppgaver. Eleven blir guidet gjennom å «følge en oppskrift», «å lage en oppskrift», «å bruke løkker», «å bruke variabler» og «å styre hva som skal skje», som er underoverskriftene i kapittelet.

Kapittel 3 Volum og 4 Divisjon er de med minst andel ATO. I kapittel 3 Volum øver eleven mye på å måle og regne volum. Innimellom dukker det opp oppgaver som kan fremme AT ved at eleven blir bedt om å lage prizmer som skal inneholde 48 kuber. Slike oppgaver kan fremme AT fordi eleven får trening på å løse et problem på ulike måter. Eleven må arbeide strukturert og trinnvis ved å tegne eller skrive størrelsen på prismene. For å lage disse må reglene for hva som kjennetegner et prisme følges. Til slutt skal det telles hvor mange prizmer han fikk til å lage. Kapittel 4 Divisjon inneholder mange oppgaver som innebærer at eleven regner ut eller lager divisjoner. Noen av oppgavene utfordrer eleven til å følge flere steg, som gjør at den i større grad kan fremme AT. En slik oppgave kan i tillegg til å be eleven om å regne ut et stykke, be han om å fargelegge regnestykket i fargen som er bestemt av hvilket svar regnestykket har. Se et utdrag av denne oppgaven i bildet på neste side (bilde 3).

20 Regn ut og fargelegg formene i riktig farge.



Bilde 4 (Matemagisk 4B: Divisjon)

Kapittel 2 Multiplikasjon inneholder nest høyest prosentdel ATO, og det høyeste antallet ATO av alle kapitlene i elevbøkene. Oppgavene som kan fremme AT er av typene «finn hemmelig kode», «finn ulike kombinasjoner», oppdage mønster og bruke og vurdere ulike løsningsstrategier. Flere av oppgavene på slutten skal løses med «HAUK», der eleven først lager arbeidstegning, skriver utregning og til slutt vurderer om løsningen kan stemme ved å følge et flytskjema. Se bildet under for et eksempel på en slik oppgave.

HAUK 40 Løs problemet med HAUk.

Hva er spørsmålet?
Det er 15 elever i 4C. I friminuttet hopper de med hoppetau. Elevene får fem poeng for hvert minutt de klarer å hoppe. Seks av elevene klarte å hoppe tre minutter hver. De andre elevene hoppet i to minutter hver. Hvor mange poeng fikk elevene i 4C til sammen?

Arbeidstegning

Utregning

Kan det stemme?

```

    graph TD
      A[Høres svaret ditt riktig ut?] -- ja --> B[Har andre elever fått samme svar?]
      A -- nei --> C[Se på arbeidstegningen og utregningen en gang til og prøv igjen.]
      B -- ja --> D[Nå har du jobbet godt med å finne en løsning på oppgaven.]
      B -- nei --> E[Diskuter løsningene deres.]
      E --> D
  
```

Bilde 5 (Matemagisk 4A: Multiplikasjon)

Det som kjennetegner AT i disse oppgavene er at eleven må følge instruksjoner med tydelige regler. Et eksempel er en oppgave som ber eleven om å tegne alle spisse vinkler røde, alle stumpe vinkler blå og alle rette vinkler grønne. Et annet eksempel er oppgaver som ber eleven om å fullføre en påbegynt figur, der eleven må finne mønsteret eller reglen for å få til oppgaven. Oppgaver som spør slik «Hvor stor er flaten av ...» har blitt analysert som «i liten grad ATO». Disse oppgavene ber i hovedsak eleven om å bruke en formel de allerede kan, og trenger ikke å bruke AT-ferdigheter for å løse oppgaven.

5.1.2 Multi 4

Multi 4 er utgitt av Gyldendal i 2021, og følger LK20. Multi 4 er et sett trykte og digitale læremidler. De trykte bøkene er delt i tre – elevbok, øvebok og lærerens bok. I dette tilfellet skal kun elevbøkene, som er delt i bok A og bok B, analyseres.

Multi 4 elevbøker er oppdelt i 8 kapitler. Kapitlene heter 1 Tall og regning, 2 Multiplikasjon, 3 Divisjon, 4 Geometri, 5 Brøk og desimaltall, 6 Mønster og algoritmer, 7 Måling og 8 Regning. Kapittel 1-4 er i bok A, og kapittel 5-8 er i bok B. I denne boka er det kapittel 6 Mønster og algoritmer som er direkte koblet til kompetansemålene som omhandler AT i LK20. Oppgavene i kapitlene er plassert i kategoriene «ATO i stor grad», «ATO i noen grad» og «ATO i liten grad».

Tabell 7

Multi 4 elevbok: 133 av 589 (41,3%)	
Multi 4A 110 av 300 (36,7%)	Multi 4B: 133 av 289 (46%)
1 Tall og regning: 37 av 104 (35,6%)	5 Brøk og desimaltall: 46 av 83 (55,4%)
2 Multiplikasjon: 12 av 86 (14%)	6 Mønster og algoritmer: 35 av 36 (97,2%)
3 Divisjon: 18 av 63 (28,6%)	7 Måling: 23 av 75 (30,7%)
4 Geometri: 43 av 47 (91,5%)	8 Regning: 29 av 95 (30,5%)

Tabellen ovenfor (tabell 7) viser en oversikt over hvor mange av oppgavene i bøkene som er kategorisert som «ATO i stor grad». Disse oppgavene består av grunnoppgaver, utforskningsoppgaver, aktiviteter og spill. Alle spillene i bok A og bok B er i kategorien «ATO i stor grad». Jeg vil også påpeke at 258 oppgaver er kategorisert som «ATO i liten grad», 43,8%, og 88 som «ATO i noen grad», 30,5%.

4 Geometri og 6 Mønster og algoritmer er kapitlene som i størst grad inneholder oppgaver som i stor grad kan fremme elevers AT. I kapittelet om geometri arbeider eleven med kjennetegn ved ulike geometriske former, og må sortere figurer etter disse og tegne former som tar hensyn til kjennetegnene. Eleven må altså ha hensyn til regler og mønstre for å arbeide med mange av oppgavene i kapittelet. Det forklarer hvorfor mange av oppgavene er kategorisert som «ATO i stor grad». Kapittel 6 Mønster og algoritmer utfordrer eleven på å arbeide mye med mønstergjenkjenning og med algoritmer. I en stor del av kapittelet skal eleven skal arbeide med figurtall, som i stor grad handler om å bygge videre på et gitt mønster. Eleven får også trening på å lage egne algoritmer med løkker og vilkår.

Kapittel 2 Multiplikasjon utpeker seg som det som inneholder færrest ATO. Mange oppgaver ber eleven om å skrive gangestykker til informasjon de får. Et gangestykke kan beskrives som en algoritme, men for at eleven skal få trening på å tenke algoritmisk må det innebære flere steg enn å kun arbeide med regnestykker. Et eksempel på oppgaver som i større grad kan fremme AT er vist i bildet på neste side. Her må eleven lage gangestykker samtidig som de følger reglen til ulikhetstegnene og det de symboliserer. Det vil si at de må vurdere hvilket tall som gir riktig verdi, slik at ulikheten stemmer.

- 23** Hva er den største faktoren som kan stå i ruta?
- | | | |
|---------------------------------|---------------------------------|---------------------------------|
| a $6 \cdot \square < 56$ | b $\square \cdot 5 < 44$ | c $63 > \square \cdot 7$ |
| d $49 > 8 \cdot \square$ | e $9 \cdot \square < 70$ | f $87 > 8 \cdot \square$ |
- 24** Hva er den minste faktoren som kan stå i ruta?
- | | | |
|---------------------------------|---------------------------------|---------------------------------|
| a $4 \cdot \square > 39$ | b $\square \cdot 3 > 29$ | c $75 < \square \cdot 9$ |
| d $56 < 8 \cdot \square$ | e $8 \cdot \square > 30$ | f $49 < 4 \cdot \square$ |

Bilde 6 (Multi 4A: Multiplikasjon)

Kapittel 1 Tall og regning, 3 Divisjon, 5 Brøk og desimaltall og 7 Måling ligger nærmere gjennomsnittet for hvor stor andel ATO det er i elevbøkene. Det som kjennetegner mange av oppgavene som i liten grad kan fremme AT er at eleven kun arbeider med regnestykker. Noen av oppgavene oppfordrer eleven til å utforske ulike fremgangsmåter, som gjør at disse plasseres i kategorien for oppgaver som i stor grad kan fremme AT. Det gjør at kapittel 5 Brøk og desimaltall skiller seg litt ut med en større prosent ATO, er at eleven i større grad må sammenligne og sortere størrelser. Eleven må også følge algoritmer i oppgaver og spill som for eksempel å hoppe fremover på en tallinje. I disse oppgavene må eleven følge

instruksjonene gitt i oppgaven, der det enten brukes brøkdeler eller desimaltall for å definere lengden på hoppene. I spillet er det terningen som bestemmer hvor stor brøkdel eleven kan hoppe fremover, som krever at eleven leser av informasjonen i en modell.

Den horisontale analysen pekte ut kapitlene «Programmering» fra Matematisk 4B, og «Mønster og algoritmer» fra Multi 4B som de som inneholder mest ATO, og er derfor valgt ut som de kapitlene jeg skal gå nærmere inn på i den vertikale analysen.

5.2 Vertikal analyse

Jeg har gjennomført en vertikal analyse av ett kapittel i hver bok. Kapitlene er «Programmering» i Matemagisk 4B, og «Mønster og algoritmer» i Multi 4B. Jeg har brukt rammeverket for den vertikale analysen for å undersøke hvordan oppgavene presenterer AT. Oppgavene kategoriseres ut ifra kategoriene: å følge en algoritme, å lage en algoritme, feilsøking, mønstergjenkjenning, forklare og forestille seg. Det er det oppgaven ber eleven om å gjøre som avgjør hva oppgaven blir analysert som. Dersom en oppgave for eksempel ber eleven om å rette opp i en ukorrekt algoritme, blir oppgaven kategorisert som feilsøking. Samtidig som oppgavene blir kategorisert, noterer jeg også hva slags AT-ferdigheter oppgaven kan fremme. Disse ferdighetene er: dekomposisjon, mønstergjenkjenning, abstraksjon, algoritmedesign og evaluering. Noen oppgaver kan passe i flere enn én kategori. En oppgave som ber om mønstergjenkjenning kan også be eleven om å lage en algoritme, og blir kategorisert som en kombinasjon av «å lage en algoritme» og «mønstergjenkjenning».

Funnene fra den vertikale analysen presenteres på en strukturert måte, delt inn i hver enkelt kategori. I hver kategori viser jeg til én utvalgt oppgave fra hvert læreverk, og hvilke AT-ferdigheter oppgaven kan fremme. Hver oppgave vises med et bilde som er scannet av oppgaven fra den trykte boken, med tilhørende analysetabell fra koderammeverket. Til slutt gir jeg en oversikt over hvor mange av oppgavene i kapitlene som inngår i de ulike kategoriene, og hvor mange oppgaver som kan fremme de ulike AT-ferdighetene.

5.2.1 Funn i Matemagisk 4b Programmering og Multi 4B Mønster og algoritmer

I dette delkapittelet fremstiller jeg funnene fra den vertikale analysen. De neste delkapitlene tar for seg hver enkelt kategori med eksempler fra hvert læreverk. Denne måten legger til rette for en helhetlig forståelse for kategoriene, og deres plass i bøkene. Først skriver jeg om generelle funn i læreverkene, før jeg legger frem hvor mange oppgaver i hvert kapittel som er

plassert i de ulike kategoriene. Til slutt legger jeg ved en tabell som viser en oversikt over hvor stor del av oppgavene som er plassert i de ulike kategoriene, i tillegg til hvor mange av de som inneholder matematikk. Det er viktig å understreke at en oppgave kan være plassert i flere kategorier.

Begge kapitlene inneholder kun oppgaver som i stor grad kan fremme elevens AT, utenom én oppgave i «Mønster og algoritmer» i Multi 4B som er kategorisert som «ATO i noen grad». Alle oppgavene er analoge, som vil si at de ikke trenger digitale verktøy for å løses. Kapitlene inneholder flere spill og ulike aktiviteter. Det er ulikt hvor integrert matematikken er i oppgavene. I Multi 4B Mønster og algoritmer har 91,7% av oppgavene matematisk innhold, mens i Matemagisk 4B Programmering har 40% av oppgavene matematisk innhold.

I kapittelet «Programmering» i Matemagisk 4B er det kompetansemålet «lage algoritmer og uttrykke dem ved bruk av variabler, vilkår og løkker» som dekkes. Eleven trenes først på å følge oppskrifter, før de utfordres til å lage oppskrifter i riktig rekkefølge. Underveis i prosessen dukker det opp oppgaver der eleven må feilsøke en algoritme. Videre får eleven mulighet til å lære hvordan en algoritme (oppskrift) effektiviseres ved hjelp av løkker, variabler og vilkår.

7 av 20 oppgaver ble kategorisert som «å følge algoritme», og 13 oppgaver ble kategorisert som «å lage algoritme». Her vil jeg understreke at oppgaver kan bli plassert i flere kategorier, som vil si at noen oppgaver er kategorisert som både «å følge» og «å lage» en algoritme. Et eksempel på en slik oppgave er en to-delt oppgave som først ber deg om å følge en oppskrift på å brette et papirfly, før du selv lager en egen oppskrift. 6 av 20 oppgaver ble kategorisert som «feilsøking». Dette er oppgaver som for eksempel ber eleven om å finne feilen i en oppskrift, eller «sant eller usant» oppgaver. 3 av 20 oppgaver ble kategorisert som «mønsterkjennelse/finne regel eller mønster». Disse oppgavene er de som oppfordret eleven til å bruke løkker som en måte å effektivisere en algoritme på. Eleven må finne et gjentakende mønster for å skrive inn løkke. 5 av 20 oppgaver ble kategorisert som «forklare». Dette er oppgavene som er kalt «snakke matte» der eleven ikke trenger penn og papir, kun å prate med hverandre. 0 av 20 oppgaver ble kategorisert som «forestille seg».

13 av totalt 20 oppgaver er uten matematisk innhold. Det vil si at det er 7 oppgaver med matematisk innhold. 3 av disse oppgavene defineres kun som oppgaver med matematisk innhold på grunn av at de nevner «snu 90 grader». 2 av oppgavene innebærer rutenett, og kun 2 handler om geometriske former.

Kapittelet som ble analysert i Multi 4b er «Mønster og algoritmer». Kapittelet inneholder totalt 36 oppgaver. Den første delen av kapittelet handler om mønster, og den siste delen handler om algoritmer, hvorav 23 av oppgavene er i delen om mønster, og 13 i delen som handler om algoritmer. Underoverskriftene er som følger: geometrisk mønster, figurtall og tallmønster og algoritmer. Igjennom kapittelet kan eleven lære å finne og følge et mønster, både i rutenett og i tallmønster, og algoritmedesign med løkker og vilkår. Kapittelet tar for seg kompetansemålene «utforske og beskrive strukturer og mønstre i lek og spill» og «lage algoritmer og uttrykke dem ved bruk av variabler, vilkår og løkker».

12 av 35 oppgaver er kategorisert som «å følge en algoritme», og 9 er kategorisert som «å lage en algoritme». De fleste av disse er plassert i delkapittelet som heter Algoritmer. Det er kun 1 oppgave som er kategorisert som «feilsøking». Denne oppgaven ba eleven om å plassere operasjoner for å lage spaghetti i riktig rekkefølge. 19 av oppgavene er kategorisert som «mønstergjenkjenning/finne regel eller mønster». Her er derimot alle oppgavene plassert i delen som handler om mønster. Oppgavene som skal lære eleven om løkker ber aldri eleven om å lage en løkke. Det er forklart og vist eksempel på hva det er, så de kan likevel fremme AT-ferdigheten *mønstergjenkjenning*, selv om de ikke er kategorisert som det i denne analysen. 2 av 35 oppgaver er kategorisert som «forklare». Disse oppgavene stiller eleven spørsmålet «Hvordan er disse forskjøvet».

32 av 35 oppgaver er med matematisk innhold. De fleste går innunder temaet «mønster», med tallmønster, forskyvning og geometri, der flere av oppgavene krever at eleven også bruker regneartene. Oppgavene i delen «Algoritmer» er kun 3 av 13 oppgaver som er uten matematisk innhold. De andre oppgavene inneholder rutenett, vinkler og/eller regning.

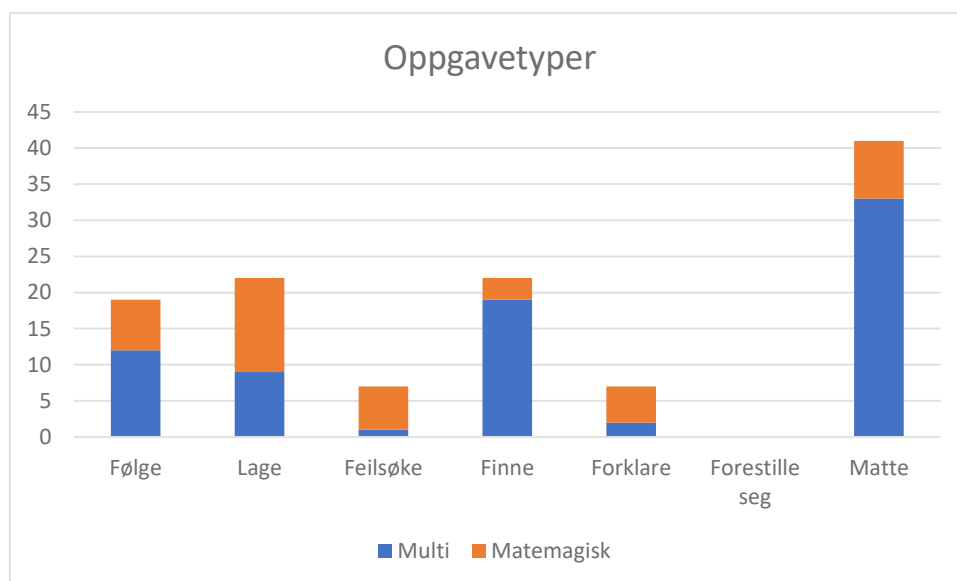


Diagram 1

I den vertikale analysen har jeg også notert hva slags AT-ferdigheter som kan fremmes i hver oppgave. Resultatene viser at alle ferdighetene dekkes i begge bøkene (se tabell 5).

Algoritmedesign, dekomposisjon og mønstergjenkjenning er de ferdighetene oppgavene dekker i størst grad. Alle de 55 AT-oppgavene i kapitlene gir eleven erfaringer med algoritmedesign, der 22 av de innebærer at eleven lager algoritmer selv. I 30 av oppgaven bruker eleven dekomposisjon for å forstå hva oppgaven ber om. Mønstergjenkjenning er den ferdigheten kapitlene dekker på mest ulik måte. Kapittelet i Multi har 19 slike oppgaver, mens Matemagisk kun har 6. Kapitlene har like mange oppgaver som dekker abstraksjon, nemlig 7 i hvert kapittel. Ferdigheten det er lagt minst vekt på er evaluering. Jeg fant totalt 9 oppgaver, hvorav 2 er i kapittelet i Multi, og 7 i Matemagisk.

Tabell 8

Ferdigheter	Multi	Matemagisk
Dekomposisjon	20	10
Mønstergjenkjenning	19	6
Abstraksjon	7	7
Algoritmedesign	35	20
Evaluerings	2	7

5.2.2 Å følge en algoritme



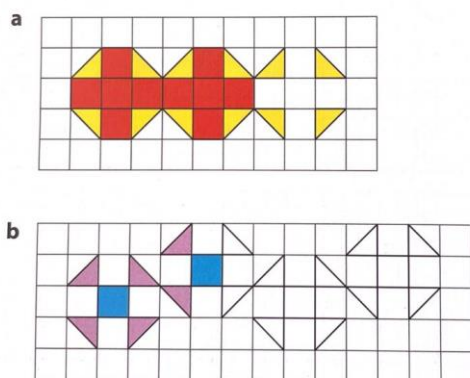
Bilde 7 (Matemagisk 4B: Programmering)

Tabell 9

Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Aktivitet	★						★90 grader
	Algoritmedesign						

Den første oppgaven i kapittelet er det Matemagisk omtaler som en aktivitet. Aktiviteten innebærer at eleven skal følge en algoritme som er presentert i oppgaven. I dette tilfellet er det kun det oppgaven ber om, og er derfor kategorisert som «å følge en algoritme». Ved å følge stegene, øver eleven på å oppdage mønster og algoritmedesign. Å følge gitte instruksjoner og finne ut hva de fører til, kan gi eleven forståelse for hvordan instruksjoner fungerer. Oppgaven kan fremme forståelse for abstraksjon, ved at algoritmen inneholder steg som kunne vært effektivisert. Ettersom at oppgaven ikke ber eleven om å evaluere algoritmen, velger jeg å definere dette som en oppgave som ikke fremmer abstraksjon og feilsøking. For å følge algoritmen må eleven forholde seg til hva «90 grader» betyr, som gjør at jeg har valgt å plassere dette som en oppgave med matematisk innhold.

- 3 Tegn av figuren. Forskyv figuren flere ganger på samme måte, slik at du får et mønster.



Bilde 8 (Multi 4B: Mønster og algoritmer)

Tabell 10

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
3	★ Algoritmedesign			★ Mønstergjenkjenning			Mønster med forskyvning

Dette er en oppgave som inngår i to kategorier. Den kan kategoriseres som «å følge en algoritme», fordi eleven må fortsette mønsteret for å svare på oppgaven. Det gjøres ved å forskyve figuren, og eleven må finne mønsteret som allerede er gitt i oppgaven. Derfor kategoriseres den også som «å finne et mønster». Dette er en oppgave som viser at små steg kan utgjøre noe stort. Mønsteret dannes av en figur som gjentas flere ganger. Jeg vil si at eleven får kunnskap om dekomponering, men oppgaven gir ikke eleven mulighet til å dekomponere selv. Derfor vil jeg ikke si at oppgaven fremmer elevens evne til å dekomponere. Derimot fremmer den elevens ferdigheter i mønstergjenkjenning. For å løse oppgaven trengs det ikke abstraksjon, da mønsteret allerede er tydelig presentert. Eleven får derimot mulighet til å utvikle sine ferdigheter i algoritmedesign ved å følge algoritmen videre på en strukturert og logisk måte.

5.2.3 Å lage en algoritme

A Lag mønster.

Arbeid sammen i par. Lag et mønster av figurer med en mønsterdel som gjentas. Dere kan bruke farger, bokstaver, brikker, klosser, geometriske figurer og liknende. Bytt mønster med et annet par og be dem finne ut hva som blir figur nummer 17, eller 25, eller et annet tall mellom 10 og 30.



Bilde 9 (Multi 4B: Mønster og algoritmer)

Tabell 11

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Aktivitet side 42		★ Dekomposisjon Algoritmedesign		★ Mønstergjenkjenning			Mønster

Her skal elevene jobbe sammen i par. Oppgaven er todelt. Den første delen er å lage et mønster av figurer, som gjør at den passer i kategorien «å lage en algoritme». Det første elevene må gjøre er å dekomponere problemet for å forstå hva oppgaven krever av dem. Videre arbeider elevene med forståelse for mønster og gjentakelse i form av tydelig struktur. De bruker ferdigheten algoritmedesign for å planlegge delene som til sammen skaper mønsteret. Den andre delen innebærer at elevene bytter mønster med et annet par, før de skal finne hva som blir figur nummer 17, eller andre nummer i deres mønsterrekke. For å svare på oppgaven må paret oppdage mønsteret det andre paret har laget, og fortsette mønsterrekka. Derfor kategoriseres oppgaven som «å finne et mønster».

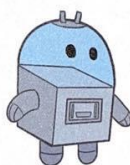
Snakke matte

Roboten skal fargelegge bokstavene H, E og I i rutenettet.

Dere bestemmer:

- ☞ hvor roboten skal starte på rutenettet
- ☞ hvordan roboten skal beveges seg for å komme til bokstavene H, E og I
- ☞ hvilke farger bokstavene H, E og I skal ha

Lag programmet til roboten.



S	F	G	Z	R
Å	P	N	I	V
A	U	Ø	T	M
O	H	L	A	D
B	Æ	Y	J	E

Bilde 10 (Matemagisk 4B: Programmering)

Tabell 12

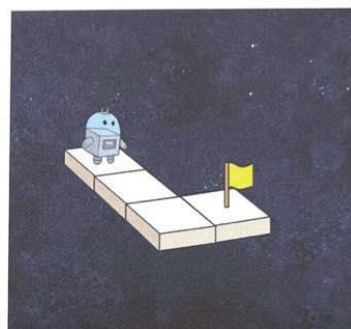
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Snakke matte		★ Dekomposisjon Algoritmedesign			★		Rutenett

Den siste setningen i oppgaven ber elevene om å lage programmet til roboten, som gjør at den kategoriseres som «å lage en algoritme». Dette er en «snakke matte» oppgave. Formålet med disse oppgavene er at elevene skal forklare det de tenker og gjør, og den er derfor kategorisert som «forklare en algoritme» i tillegg. Elevene har frihet til å ta egne valg innenfor rammene. Det vil si at de selv bestemmer start- og slutt punkt, og alle stegene imellom, så lenge utfallet av algoritmen svarer på oppgaven. Elevene må dekomponere problemet for å skape en forståelse over hvilke steg han kan ta. Underveis må det tas hensyn til at algoritmen gjør at roboten fargelegger bokstavene H, E og I i rutenettet. Å sette sammen stegene på denne måten fremmer elevenes ferdigheter i algoritmedesign. Det kan tenkes at elevene evaluerer de ulike algoritmene og velger den som er mest hensiktsmessig og effektiv, men kan ikke forventes da oppgaven ikke ber direkte om dette. Oppgaven hadde derimot lagt til rette for bruk av evaluering dersom den ga en tilleggsopplysning som for eksempel «roboten skal gå færrest mulig skritt».

5.2.4 Feilsøke

2 Roboten skal gå til flagget. Stryk det som ikke passer inn i oppskriften.

Gå ett steg fram.
Gå ett steg fram.
Snu til venstre 90 grader.
Gå ett steg fram.
Gå ett steg fram.



Bilde 11 (Matemagisk 4B: Programmering)

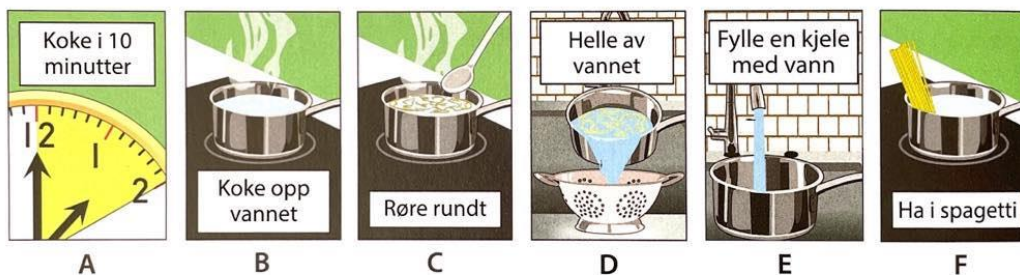
Tabell 13

Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
2			★ Evaluering Abstraksjon				★90 grader

Oppgave 2 (bilde 11) ber eleven om å stryke instruksjonene som ikke passer inn i oppskriften. Den er kategorisert som feilsøking, der eleven må *evaluere* om algoritmen er effektiv for formålet.

Ferdigheten *abstraksjon* brukes for å fjerne stegene som er ukorrekte i algoritmen. Eleven må da vurdere hvilke steg algoritmen må inneholde for at roboten når flagget, og hvilke som må fjernes.

U 21 Her er seks operasjoner for å lage spaghetti. I hvilken rekkefølge må operasjonene gjøres?



Bilde 12 (Multi 4B: Mønster og algoritmer)

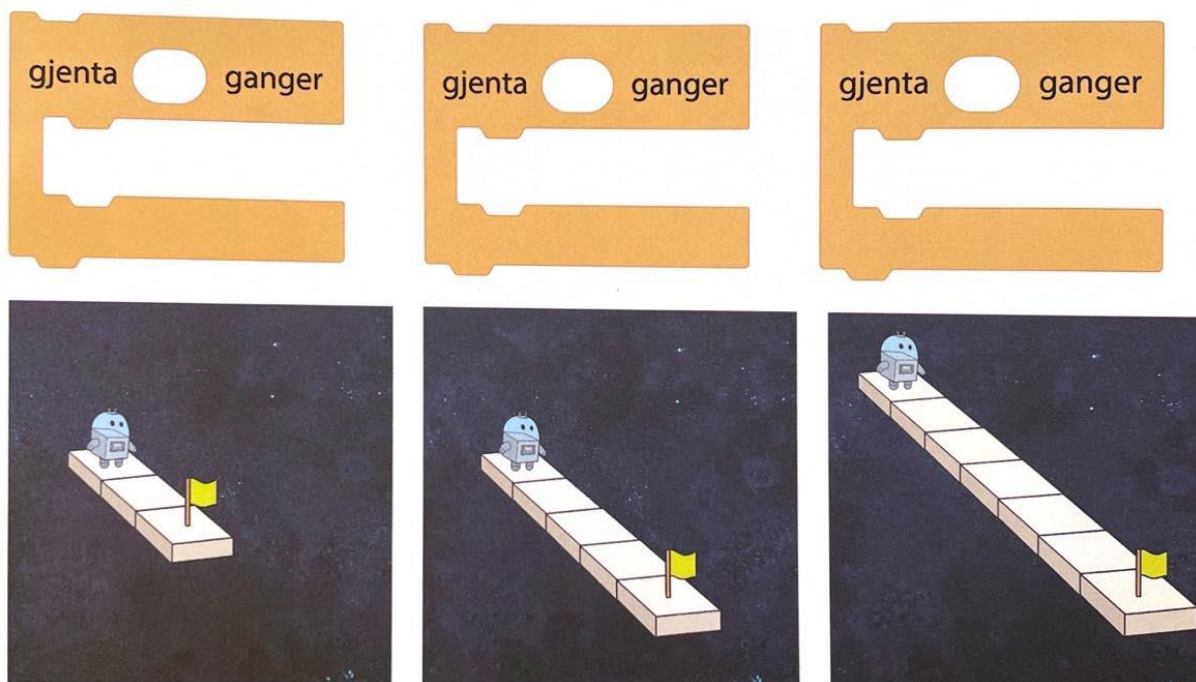
Tabell 14

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
21 Utforskning			★ Evaluering Algoritmedesign				

Oppgave U21 (Bilde 12) er den eneste oppgaven i kapittelet i Multi 4B som krever at eleven feilsøker en algoritme. Her må eleven bruke logikk for å plassere operasjonene i riktig rekkefølge. Oppgaven viser tydelig hvilke steg som skal til for å nå målet, som vil si at eleven slipper å dekomponere problemet. Eleven får trening på algoritmedesign ved at han må vurdere hvordan de ulike stegene skal plasseres imellom start- og sluttpunktet, slik at algoritmen blir rett. Å arbeide med en slik oppgave fremmer ferdigheten *evaluering*. Før eleven kan si seg ferdig med oppgaven, må han evaluere om algoritmen er effektiv og inneholder nødvendig informasjon for å nå målet.

5.2.5 Å finne et mønster

- 5** Lag en oppskrift på hvordan roboten kommer seg til flagget ved å bruke løkke.



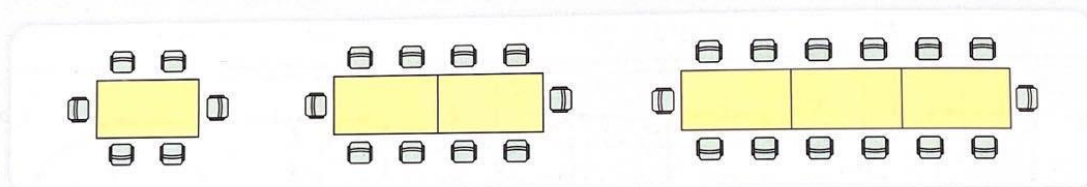
Bilde 13 (Matemagisk 4B: Programmering)

Tabell 15

Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
5		★ Dekomposisjon		★ Mønstergjenkjenning			

I oppgave 5 skal eleven lage en algoritme ved å bruke løkker. Til å starte med *dekomponerer* eleven problemet for å forstå hva oppgaven ber om. For å finne ut hvilket tall løkken skal være, må eleven identifisere gjentakelsene av instruksene. Derfor passer oppgaven i kategorien «Finne et mønster» der eleven bruker *mønstergjenkjenning* for å oppdage mønsteret. Målet er definert i oppgaven, og eleven bruker løkkene til å lage algoritmen som gjør at roboten når målet. Derfor har jeg også kategorisert denne oppgaven som «Lage algoritme».

- 18 Bordene kan også settes sammen på denne måten.



- Hvordan blir plasseringen nå med fire bord? Hvor mange personer blir det plass til?
- Lag en tabell som den over, som viser antall personer det er plass til rundt 1–7 bord.
- Hvordan vokser disse figurtallene?
- Hvor mange bord må nå settes sammen for at det skal være plass til 35 personer?

Bilde 14 (Multi 4B: Mønster og algoritmer)

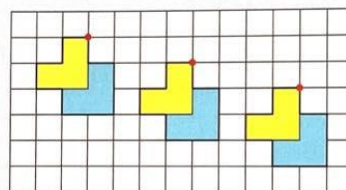
Tabell 16

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
18 Utforskning				★			Figurtall
				Mønstergjenkjenning			Regning
				Abstraksjon			
				Dekomposisjon			
				Algoritmedesign			

Oppgave U18 (bilde 14) handler om hvordan et figurtall vokser. Det er elevens oppgave å finne mønsteret figurtallene følger. Så fort mønsteret er oppdaget, kan eleven lage en algoritme som kan brukes til å finne ut av hvor mange personer det er plass til rundt x antall bord. I del B av oppgaven veiledes eleven til å lage en tabell som strukturer informasjonen på en oversiktlig måte. For å plassere riktig informasjon i tabellen, må eleven bruke ferdigheten *abstraksjon*; all irrelevant informasjon må fjernes. Algoritmen skal benyttes i del D, der eleven skal finne ut av hvor mange bord som settes sammen for å få plass til 35 personer. Figurtall er et matematisk emne, som gjør at oppgaven kategoriseres som en «ATO med matematisk innhold».

5.2.6 Forklare

U 4 Hvordan er figuren forskjøvet?



Bilde 15 (Multi 4B: Mønster og algoritmer)

Tabell 17

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
4 Utforskning				★ Mønstergjenkjenning	★ "Hvordan er figuren forskjøvet?"		Forskyvning i rutenett

Dette er en utforskende oppgaver der eleven skal finne mønsteret figuren er forskjøvet etter. Derfor passer den i «Finne et mønster» og eleven bruker *mønstergjenkjenning*. I tillegg skal eleven forklare hvordan mønsteret forskyver seg i rutenettet, som gjør at den også er kategorisert som «Forklare» og «Matte».

Snakke matte

Kan dere lese en oppskrift?

Raketten må programmeres for å komme til de ulike planetene.

1 Bruk oppskriftene og se hvilken planet raketten kommer fram til. Skriv bokstavene på planetene.

A

Gå en rute fram.
Gå en rute fram.
Snu til høyre
90 grader.
Gå en rute fram.
Gå en rute fram.

B

Gå en rute fram.
Gå en rute fram.
Snu til venstre
90 grader.
Gå en rute fram.

C

Snu til venstre
90 grader.
Gå en rute fram.
Gå en rute fram.

D

Gå en rute fram.
Snu til høyre
90 grader.
Gå en rute fram.
Gå en rute fram.
Snu til høyre
90 grader.
Gå en rute fram.

E

Snu til venstre
90 grader.
Gå en rute fram.
Snu til høyre
90 grader.
Gå en rute fram.
Gå en rute fram.

2 Gjem en ny planet i rutenettet. Lag en oppskrift til raketten.

3 Løs hverandres oppskrifter. Finner dere de gjemte planetene?

Bilde 16 (Matemagisk 4B: Programmering)

Tabell 18

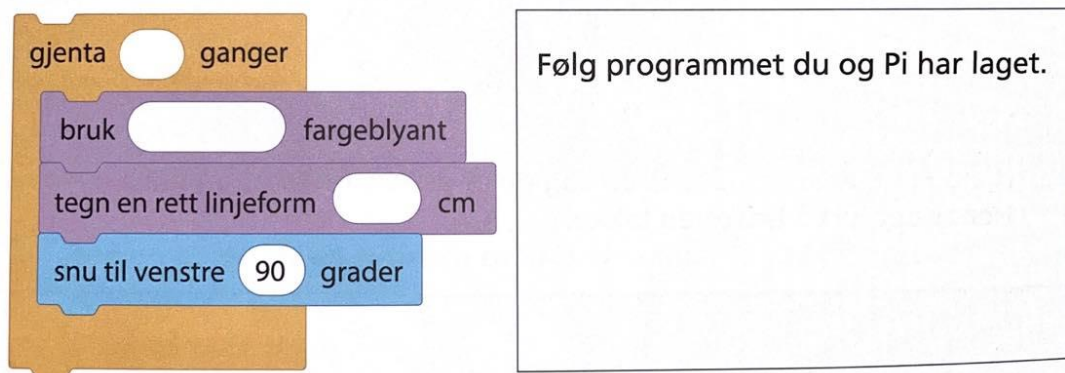
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Snakke matte	★	★ Dekomposisjon Algoritmedesign Evaluering			★		★ 90 grader Rutenett

Bilde 16 viser en «Snakke matte» oppgave som er kategorisert som «Følge en algoritme», «Lage algoritme» og «Forklare». Elevene skal arbeide sammen, og diskutere hvilke algoritmer som hører til hvilken planet. De bruker *evaluering* for å både finne ut av hvilke algoritmer som hører til hvilken planet, og å evaluere om deres egen algoritme (del 3 i oppgaven) gir riktig utfall. Når elevene skal plassere flere planeter i rutenettet (del 2 i oppgaven) må de dekomponere for å finne ut hva som skal til for å lage algoritmen. Deretter lager de løsninger til de ulike delene som settes sammen til en algoritme. Derfor fremmer oppgaven også *algoritmedesign*.

5.2.7 Matematisk innhold

8 Pi vil tegne et kvadrat. Han vet at alle vinklene er rette, og at alle sidene er like lange.

Skriv det som mangler i programmet.



Bilde 17 (Matemagisk 4B: Programmering)

Tabell 19

Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
8		★ Algoritmedesign		★ Mønstergjenkjenning			"Tegne et kvadrat"

Dette er en analog programmeringsoppgave som ber eleven om å lage et program som tegner et kvadrat. Kvadrat inngår i det matematiske temaet geometri, og er derfor plassert i denne kategorien. Eleven må ta hensyn til hvilke kjennetegn et kvadrat har; til dette brukes *mønstergjenkjenning*. Oppgaven gir samtidig eleven valgmuligheter når det gjelder størrelse og farge, som åpner for

kreativitet. Eleven bygger videre på et grunnlag som allerede er lagt, og tilpasser den for å lage en hensiktsmessig algoritme som svarer på oppgaven. Oppgaven passer derfor i kategoriene «Lage en algoritme» og «Finne et mønster». Eleven trener på *algoritmedesign* ved å vurdere hvilke variabler som passer inn i de tomme boksene. Det trengs derimot ikke dekomposisjon eller abstrahering for å løse denne oppgaven. Om algoritmen fungerer på en riktig måte, skal det tegnes et kvadrat; da er oppgaven besvart. Hvis ikke, må eleven feilsøke algoritmen for å finne feilen, og rette opp i den. Feilsøking kan derfor oppstå, men ikke nødvendigvis.

- 29** Musa går inn i rutenettet ved START.
Den går framover ei rute etter disse reglene:

Del tallet i ruta på 3 og finn ut hvor mye det blir i rest. Hvis resten er

- 1, sving 90° til venstre.
- 2, sving 90° til høyre.
- 0, fortsett rett fram.

Ved hvilken bokstav går musa ut av rutenettet?



START

	O	N	M	L	K	
P	23	6	33	8	28	J
Q	11	25	17	10	20	I
R	7	12	21	16	9	H
S	5	27	19	18	14	G
	15	22	13	26	3	F
	A	B	C	D	E	

Bilde 18 (Multi 4B: Mønster og algoritmer)

Tabell 20

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
29	★ Algoritmedesign						Divisjon Rutenett

Oppgave 29 i Multi 4B kan kun løses dersom eleven dividerer tallet musa er på i rutenettet med 3. Det gjør at oppgaven vurderes som en oppgaver med matematisk innhold. Deretter bruker eleven vilkårene for å finne ut av det neste steget i algoritmen som flytter musa. Eleven utvikler ferdigheter i algoritmedesign ved at han må følge en algoritme og får erfaring med hvordan vilkår er med på å bestemme hvordan en algoritme utformes.

6.0 Diskusjon

Dette kapittelet inneholder diskusjon av funnene koblet til teorien presentert i kapittel 3. Jeg deler diskusjonen inn i to deler, der den første delen inneholder diskusjon om funnene fra den horisontale analysen, og den andre delen om den vertikale. Det er viktig å påpeke at den horisontale analysen tar for seg alle oppgavene i bøkene, og den vertikale kun ett kapittel i hver bok. Diskusjonen om den vertikale analysen dreier seg altså ikke om bøkene i sin helhet, men kun et utdrag.

6.1 Diskusjon om funnene fra den horisontale analysen

Funnene fra den horisontale analysen viser at lærebøkene integrerer AT relativt jevnt igjennom bøkene. De legger til rette for at eleven kan arbeide med AT i minst 30% av oppgavene de gjør i boka i løpet av skoleåret. Rundt halvparten av alle de analyserte oppgavene er kategorisert som «ATO i liten grad», og de resterende oppgavene kan fremme AT, men ikke nødvendigvis. Alle kapitlene inneholdt minst 9 ATO, som gjør at eleven har mulighet til å utvikle sin AT fra start til slutt i boka. Disse funnene gir et positivt, og litt overraskende, svar på problemstillingen. Med tanke på at begrepet nylig ble introdusert i læreplanen, og at teorien påpeker at det fortsatt er misforståelser rundt begrepet, var jeg forberedt på å finne færre ATO i lærebøkene.

Begge lærebøkene inneholder ett kapittel der tilnærmet alle oppgavene i stor grad kan fremme AT. Det som er interessant er at de begge er plassert i 4B bøkene, og i Matemagisk - helt i slutten av boka. Det vil si at eleven får mest øving i AT mot slutten av skoleåret. En slik plassering kan være positivt for videre arbeid med programmering i 5. klasse, men kan også være negativt på grunn av manglende øving på strategien. Selv om alle kapitlene i bøkene inneholder ATO, inneholder disse kapitlene oppgaver som kan gi en grundigere opplæring i å bruke strategien, da alle oppgavene handler om AT. Det kan være interessant å vurdere effekten av å plassere disse kapitlene lenger frem i bøkene.

Alle AT-oppgavene var analoge aktiviteter. Eleven får derfor trent på å løse problemer på en måte som ikke krever digitale verktøy. Disse oppgavene krever ikke at eleven har kunnskaper i programmeringsspråk for å kunne løse dem. Dette kan ifølge Román-González (2017) være positivt for yngre elever som får mulighet til å utvikle AT-strategien før de får opplæring i programmering. En annen positiv side med analoge oppgaver er at de kan benyttes i flere

klasserom som ikke har tilgang på internett. Slik kan flere elever få muligheten til å lære seg å tenke algoritmisk.

Ingen av oppgavene nevnte digitale verktøy. Det vil si at eleven på 4. trinn, igjennom boka, ikke får øvd seg på å vurdere om problemene løses best ved hjelp av digitale verktøy eller ikke. Dette er imidlertid noe de kan få mer kunnskap om på høyere skoletrinn. En fordel med dette kan være at eleven får trening på å løse problemene analogt, slik at blir ikke er avhengig av å ha digitale verktøy tilgjengelig. Den kunnskapen de får om algoritmedesign kan være positivt når de senere skal få en datamaskin til å følge algoritmen deres. Det er likevel viktig at de ikke venter for lenge med å bruke digitale verktøy, da dette kan ta fra eleven muligheten til å få avgjørende dataopplevelser (Grover & Pea, 2013).

I neste delkapittel skal jeg diskutere nærmere hvordan et utvalg av ATO er utformet, hvordan de kan bidra til elevens utvikling av AT-ferdigheter og hvordan de kobles til matematikkfaget.

6.2 Diskusjon om funnene fra den vertikale analysen

Den vertikale analysen tok for seg 56 av oppgavene i lærebøkene. Disse ble kategorisert ut ifra hvilke oppgavetyper de er, hvilke AT-ferdigheter de kan fremme og om de har matematisk innhold. Funnene fra analysen diskuteres i dette delkapittelet, samt hvordan AT-strategien kan kobles til matematisk kompetanse.

6.2.1 Oppgavetyper og AT-ferdigheter

Begge lærebøkene legger til rette for at eleven kan arbeide med alle AT-ferdighetene. Mest trening får eleven på å dekomponere og utvikle ferdigheter i algoritmedesign, mens de får minst trening på å evaluere algoritmer. Denne studien tar ikke for seg om det er tilstrekkelig for å utvikle AT-ferdighetene eller ikke, men det er naturlig å tenke at det ville vært en fordel at disse kapitlene inneholdt like mange oppgaver som de andre kapitlene i lærebøken. Å legge til flere oppgaver som oppfordrer eleven til å evaluere algoritmer kan gjøre eleven mer bevisst over prosessen med å utvikle gode løsningsforslag og algoritmer, istedenfor å kun fokusere på å løse problemet.

Alle oppgavene som er analysert, gir eleven mulighet til å øke sin forståelse for algoritmedesign. I begynnelsen av analysen antok jeg at kun oppgavene som ber eleven om å lage en algoritme kunne fremme ferdigheter i algoritmedesign, men jeg oppdaget raskt at

oppgavene i de andre kategoriene også fremmet nettopp dette. Uansett om eleven følger, lager eller feilsøker en algoritme, kan eleven få erfaring med algoritmedesign. Det samme gjelder de oppgavene som krever at eleven finner et mønster. Algoritmedesign handler om å ha kunnskaper om algoritmenes trinnvise steg, hvordan de fungerer og hva de kan inneholde. Jeg mener at eleven utvikler sin forståelse for algoritmedesign ved å få erfaringer med å arbeide med algoritmer på ulike måter. Det er også viktig at eleven får muligheten til å designe egne algoritmer, noe de får muligheten til i begge kapitlene.

Resultatene fra den vertikale analysen viser hvordan oppgavene legger opp til at eleven brukes AT som strategi på ulike måter. De ulike kategoriene brukt i analysen representerer ulike oppgavetyper Bråting og Kilhamn (2022) fant i deres forskning. Av disse oppgavetyperne er det å følge en algoritme, og å lage en algoritme, som har fått mest plass i de to kapitlene. Oppgaver i disse kategoriene utgjorde omtrent halvparten av oppgavene. Det samsvarer med funnene Bråting og Kilhamn (2022) gjorde i sin lærebokanalyse. Å følge en algoritme gir eleven muligheten til å forstå algoritmedesign ved å følge trinnvise steg og se hva de skaper. De ser hvor stor betydning kodenes rekkefølge har for resultatet av algoritmen. Dette kan forberede eleven på AT-praksisen «abstrahering og modularisering», som eleven gjør i oppgavene der eleven oppfordres til å «lage en algoritme». Det er i kapittelet i Matemagisk eleven får mest trening på å lage algoritmer. Oppgavene varierer i hva slags prosjekter de ber eleven om å lage algoritme til. Algoritmene kan være til alt fra å brette papirfly til å programmere en rute for en robot.

Underveis i kapittelet i Matemagisk lærer eleven å effektivisere algoritmen ved å bruke løkker, variabler og vilkår. På denne måten ser de at en algoritme kan forkortes og fortsatt inneholde den samme informasjonen. Disse programmeringsbegrepene ses i sammenheng med mønstergjenkjenning. Løkker er en måte å effektivisere en gjentakende handling ved å skrive hvor mange ganger koden skal gjentas, der en variabel gjør det mulig å endre antallet ganger, ut ifra det oppgaven ber om. Eleven må oppdage gjentakelser i mønsteret for å definere hvor mange ganger en handling skal gjentas. På denne måten kan samme kode benyttes uansett om en robot skal forflytte seg fem eller tjue ganger fremover. Vilkår bestemmer hva som skjer dersom en bestemt handling oppstår, med andre ord en regel. Eleven har erfaring med å følge regler fra småtrinnet, der kompetansemålene har handler om å «følge regler og trinnvise instruksjoner». Slike type oppgaver kan hjelpe eleven med å lage strukturerte og effektive algoritmer.

Av alle de analyserte oppgavene er et stort antall kategorisert som «å finne et mønster». Det er en stor forskjell mellom kapitlene, der kapitlet i Multi inneholder mange flere slike oppgaver enn kapitlet i Matemagisk. En forklaring på det er at Multi kombinerte kapitlet med «mønster», som Matemagisk ikke gjorde. En stor del av oppgavene i mønster-delen i kapitlet i Multi innebærer å finne og arbeide med mønster i for eksempel forskyvning og tallrekker. Oppgavene ber eleven ofte om å finne mønsteret i tallrekken og regne ut figurtall som kommer senere i rekken. Eleven trener derfor mye på å finne underliggende mønster og regler før delen som heter «algoritmer» i kapitlet kommer. Å oppdage mønster, altså mønstergjenkjenning, er en av AT-ferdighetene. Denne type oppgaver gir god trening på elevens utvikling av nettopp denne ferdigheten som videre kan brukes i AT-praksisen «gjenbruk og remixing». I arbeidet med å gjenbruke eksisterende prosjekter er det essensielt å ha gode ferdigheter i mønstergjenkjenning for å klare å tydelig se det underliggende mønsteret i prosjektet, «blueprintet» som Anderson (2016) kalte det. Uten dette kan ikke eleven utvikle prosjektet videre til sitt eget.

At jeg fant et få antall oppgaver som oppfordret eleven til å feilsøke overrasket meg. Å kunne feilsøke og evaluere en algoritme er en stor del av AT og er en ferdighet jeg mener kan bidra til å øke elevens kognitive utholdenhet i problemløsningsprosessen. Om eleven er god på å feilsøke og evaluere vil det være lettere å gå tilbake og oppdage feil i algoritmen, for så å løse de. Det er viktig å påpeke at kapitlet i Matemagisk ga eleven mulighet til å feilsøke en algoritme 6 ganger, mens kapitlet i Multi inneholdt kun én slik oppgave. Et annet interessant funn er at ingen av oppgavene passet i kategorien «forestille seg». Hvilke årsaker som ligger til grunn for at denne kategorien mangler i kapitlene er ikke kjent for meg, men likevel noe jeg ble oppmerksom på.

Som tidlig nevnt er AT sett på som en problemløsningsferdighet. Problemløsning foregår i fire steg: forstå problemet, lage en plan, følge planen og se tilbake. En måte å forstå problemet på er å bruke AT-ferdigheten *dekomposisjon*. Det vil si at problemet må brytes ned i mindre deler, delproblemer, som gjør problemet mer oversiktlig og lettere å forstå. Begge kapitlene inneholder oppgaver som legger til rette for elevens utvikling av dekomposisjon. Det gjøres i oppgaver som presenterer et problem i oppgaveteksten, som vil den ikke har en innlysende løsning. En oppgave som gir eleven tydelige instruksjoner legger derfor ikke til rette for at eleven bruker dekomposisjon. En oppgave som derimot ber eleven om å flytte roboten til bestemte plasser, og eleven kan velge veien selv, krever at eleven dekomponerer. Han må ta forstå hva som er målet i oppgaven, hvor han skal begynne og hva som foregår imellom dem.

Når problemet er forstått, bruker eleven mønstergjenkjenning for å se om delproblemene ligner på noe han har gjort tidligere. Om det gjør det, blir det lett å legge en plan for delproblemet. Om delproblemet er helt ukjent for eleven, må han lete andre steder etter en løsning. Det er lagt spesielt til rette for utvikling av mønstergjenkjenning i kapittelet i Multi. Et viktig poeng er at alle de oppgavene er knyttet til mønster i form av mønsterskyvning og tallrekker. De gir likevel grundig trening i å forstå et underliggende mønster som kan brukes til å fortsette mønsteret. Når planen er lagt, må planen følges. Løsningene på delproblemene løses systematisk i riktig rekkefølge, som fører til at eleven kan løse hele problemet. Eleven arbeider dermed med trinnvise steg i en logisk rekkefølge.

I det siste steget skal eleven gå tilbake og se over løsningen sin. Her brukes AT-ferdigheten *evaluering* for å vurdere om løsningen kan forbedres på noen måte. Løsningen kan kanskje gjøres kortere eller mer presis. Til dette kan eleven bruke *abstraksjon* for å fjerne overflødig informasjon. Denne prosessen kan gjøre det lettere for eleven å huske løsningen slik at den kan brukes i lignende problemer senere. I kapittelet i Multi får eleven lite trening på å evaluere; kun i to oppgaver. Av de totalt 56 analyserte oppgavene, legger 9 opp til evaluering. 14 av oppgavene kan fremme abstraksjon. Selv om eleven ved flere anledninger får brukt ferdighetene i oppgavene, er det ingen som ber eleven forbedre sin egen algoritme. 7 av oppgavene ber eleven om å feilsøke en gitt algoritme, der kun 1 av disse oppgavene er i kapittelet i Multi. Derfor vil jeg si at det er lagt lite til rette for at eleven bruker dette steget. Med veiledning og tydelig beskjed fra lærer, kan det likevel gjøres mulig at eleven gjennomfører steget. Ulempen med det er at mange lærere mangler forståelse for hvordan AT kan brukes i undervisning.

6.2.2 Matematisk innhold og matematisk kompetanse

Når det gjelder ATO med matematisk innhold fant jeg store forskjeller i de to kapitlene. Disse oppgavene utgjorde en mye større del i Multi enn i Matemagisk. Dette kan begrunnes med at mye av kapittelet i Multi dreier seg om mønster med geometriske begreper. Her blir eleven instruert til å oppdage mønster og lage et løsningsforslag som kunne brukes for å kartlegge hvordan mønstrene utviklet seg. Med løsningsforslaget kunne de enten fortsette et mønster, eller regne ut for eksempel hva figurtall nr 20 er i en tallrekke. Her kreves det at eleven tenker logisk og er strukturert i arbeidet. Mange av disse oppgavene krevde også at eleven brukte regneartene for å løse problemet. I delen som handlet om algoritmer fortsatte Multi å integrere matematikken i flere av oppgavene. Slike oppgaver krevde for eksempel at eleven brukte

divisjon for å kunne bruke vilkårene i oppgaven, eller undersøke hvordan en figur forflytter seg i et koordinatsystem. Matemagisk har derimot et større antall ATO som ikke inneholder matematisk innhold, enn ATO med matematisk innhold. Et eksempel på en oppgave som ikke hadde matematisk innhold er der eleven skulle lage en algoritme for å bevege en robot fremover på en strekning (se bilde Matemagisk: Programmering 5). En slik oppgave kunne integrert matematikken ved å endre omstendighetene rundt roboten til et koordinatsystem, slik at eleven kunne jobbet med nøyaktig plassering av roboten. Dette understreker forskningens påstand om at AT og matematikk kan forstås som isolerte deler, og ikke alltid integreres i hverandre i undervisningen. Selv om eleven kan få god trening på AT-strategien ved å gjøre oppgavene i kapittelet i Matemagisk, kan det også skape forvirring dersom eleven ikke ser hvilken sammenheng den har med matematikken.

Analysen kartla at «snu 90 grader» og rutenett er de matematiske begrepene som ble mest brukt i ATO. Flere av disse oppgavene kunne med fordel integrert matematikken i større grad. Kapittelet i Matemagisk har brukt rutenett i to av oppgavene, hvilket gir et tynt grunnlag for å konkludere at AT og matematikk er integrert i slike oppgaver. Det er overraskende at de har valgt å ikke bruke koordinatsystem. Spesielt fordi nettopp dette er et av kompetansemålene etter 3. trinn, med forventning om at eleven har arbeidet med instruksjoner knyttet til koordinatsystem tidligere. I delen om algoritmer i Multi er tre av oppgavene knyttet til koordinatsystem, og eleven må henvise til koordinatene for å løse oppgavene. Disse oppgavene er i større grad knyttet til matematikk enn de i Matemagisk. De tre siste oppgavene i kapittelet i Multi krever at eleven bruker regnearter og ulikheter. Den ene oppgaven ber eleven om å finne svaret i ett regnestykke, og om svaret er et partall, skal eleven velge veien til høyre, og om svaret er oddetall skal han velge veien til venstre. Det er en fin måte å jobbe med vilkår på, i full integrering med matematikk.

Selv om resultatene viste ulik integrering av AT og matematikk, kan utvikling av AT-strategien likevel ha positiv innvirkning på matematisk kompetanse. En av de fem trådene i Kilpatrick et al. (2001) sin trådmodell om matematisk kompetanse er *begrepsmessig forståelse*, som handler om å forstå matematiske begreper, og bruke de i ulike kontekster. Dersom målet er å se sammenhengen mellom AT og matematikk, bør denne tråden tas hensyn til i utforming av oppgavene. Dette er gjort i større grad i kapittelet i Multi enn i Matemagisk. Den neste tråden i trådmodellen er *beregning*, som innebærer å arbeide med prosedyrer på en strategisk og effektiv måte. I denne tråden passer AT-strategien utmerket inn. Begge kapitlene har lagt vekt på oppgaver som trener eleven på å både lage og følge algoritmer. Det har også

vært fokus på å effektivisere algoritmene ved å bruke løkker og vilkår. Det kunne vært en fordel om oppgavene i større grad oppfordret eleven til å evaluere løsninger og algoritmer, da dette kan skape en større forståelse om hva en effektiv algoritme er.

Tråden *anvendelse* kan også kobles til AT da den har mange likheter med problemløsning. Det handler om å formulere et problem basert på tidligere erfaringer, spesielt i ukjente problemer der eleven ikke umiddelbart ser for seg løsningen. Resultatene fra analysen viser at totalt 30 av oppgavene kan fremme elevens ferdigheter i dekomposisjon. Denne ferdigheten henger sammen med tråden *anvendelse* fordi det handler om å dele et komplekst problem inn i delproblemer for å kunne løse disse på en enklere måte. Når problemet deles opp, er det lettere å kjenne igjen problemer du har løst tidligere, og det er enklere å lage en plan for hvordan hele problemet kan løses. AT handler om å oppdage mønster som brukes for å løse problemer, og etter problemet er dekomponert, brukes mønstergjenkjenning til å kunne kjenne igjen problemene. Kombinasjonen av dekomponering og mønstergjenkjenning gjør det altså lettere å behandle ukjente og komplekse problemer. Når eleven blir god på dette, vil det være gode sjanser for å lykkes i anvendelse på andre områder, også i andre fag enn matematikk.

Tråden *resonnering* handler om at eleven kan begrunne og rettferdiggjøre de matematiske valgene han tar. Kapittelet i Matemagisk ber eleven om å forklare en algoritme i 5 av oppgavene; som er mer enn kapittelet i Multi som kun inneholder 2 slike oppgaver. Likevel er det få av oppgavene som ber eleven om å begrunne en algoritme som er laget av eleven selv. De fleste oppgavene ber dermed eleven om å forklare en gitt algoritme. Derfor mener jeg at eleven får begrensede muligheter til å resonnerer rundt egne valg i begge kapitlene. Ingen av de analyserte oppgavene i den vertikale analysen var av typen «forestille seg». Å forestille seg et utfall dersom en kode i algoritmen endres, kan sette i gang en resoneringsprosess hos eleven. Eleven gjøres da oppmerksom på hvordan ulike valg og endringer påvirker resultatet. Dette kan være et argument for å inkludere flere slike oppgaver i kapitlene. Den siste tråden er *engasjement*. Her er det viktig at eleven ser meningen og verdien av matematikken. Elevens engasjement påvirkes av de andre trådene. Dersom eleven har forståelse for matematiske begreper og anvendelse, kan også engasjementet øke. En svakhet i Matemagisk er at ATO som ikke har matematisk innhold, kan føre til at eleven ikke ser verdien i å lære AT i matematikken. Dette kan svekke elevens engasjement til temaet. Likevel kan kapittelets oversiktlige gjennomgang av både å følge, lage og effektivisere algoritmer øke elevens engasjement; dersom han opplever mestring.

7.0 Konklusjon

Problemstillingen i masteroppgaven er *Hvordan presenteres algoritmisk tenkning i trykte læreverker i matematikk på 4. trinn?* Begge lærebøkene gir eleven muligheten til å trene opp AT-ferdigheter jevnt igjennom skoleåret, med ekstra tyngde mot slutten. Kapitlene med flest AT-oppgaver inneholder færre oppgaver enn de andre. Det kunne vært en fordel at kapitlene var like store, men det kan rettferdiggjøres ved at AT-oppgaver er relativt jevnt fordelt i de andre kapitlene i tillegg.

De fleste oppgavene som er analysert vertikalt ber eleven om å følge og å lage algoritmer. En god del ber eleven om å finne et mønster, der de fleste av disse oppgavene er i Multi, og ikke Matemagisk. De analyserte AT oppgavene oppfordrer i liten grad eleven om å feilsøke algoritmer. Dette kan være en svakhet i kapitlene, da det å feilsøke er sett på som en viktig del av AT-strategien. Ingen av oppgavene ba eleven om å forestille seg et utfall. For å øke elevens mulighet til å resonnere rundt ulike løsningsforslag, kunne det vært interessant å implementere flere slike oppgaver.

Eleven øver som nevnt på AT-strategien på ulike måter, gjennom bruk av de forskjellige bøkene. Begge bøkene dekker likevel alle AT-ferdighetene. Alle oppgavene i den vertikale analysen gir eleven verdifulle erfaringer med algoritmedesign på ulike måter. Mange av oppgavene krever at eleven dekomponerer problemet for å forstå hva oppgaven ber om. Bøkene gir ulik mulighet til å øve på mønstergjenkjenning, med større muligheter i Multi. Muligheten for abstraksjon finner jeg derimot i likt antall oppgaver i begge kapitlene. Ferdigheten det er lagt minst vekt på i kapitlene, er evaluering. Mange av oppgavene kan likevel tilpasses slik at eleven må evaluere. Lærerveiledningen er ikke analysert, og det er mulig den inneholder retningslinjer for hvordan inkludere evaluering i undervisningen. Dersom lærerveiledningen ikke dekker dette, kreves det økt kompetanse om AT hos læreren.

Den vertikale analysen kartla store forskjeller når det gjelder AT-oppgaver med matematisk innhold. AT-oppgavene i kapittelet i Multi inneholdt matematiske begreper i langt større grad enn de i kapittelet i Matemagisk. AT-oppgavene i Matemagisk kunne med små justeringer hatt et mer gjennomgående matematisk innhold, som igjen kan gjøre det lettere for elever og lærere å se sammenhengen mellom AT og matematikk. Ved bruk av flere matematiske begreper og ideer, kan AT-oppgavene bidra til å øke elevens matematiske kompetanse. I tillegg kan det få eleven til å innse at det er nyttig å anvende matematikken i ulike situasjoner.

Et resultat kan være at elevens engasjement øker, spesielt om eleven opplever mestring i arbeidet sitt.

AT er en problemløsningsmetode, og funnene fra den vertikale analysen viser at AT-oppgavene kan fremme AT-ferdighetene som trengs. Funnene viser også at få av oppgavene krever at eleven ser tilbake på løsningen sin. Det vil si at ingen disse oppgavene krever at eleven bruker alle stegene i problemløsningsmetoden. Det er likevel mulig at dette gjøres i andre oppgaver i de andre kapitlene i bøkene. Om dette er tilfellet, kunne det vært en fordel å vurdere å flytte kapittelet lenger frem i bøkene, slik at eleven får brukt AT-strategien som problemløsningsmetode i videre arbeid med oppgavene i bøkene.

Det er tydelig at forskningen bærer preg av ulike vinklinger og forståelser for AT. Det kan være nyttig å forske mer på hva slags oppgaver som fremmer AT, slik at det kan skapes en tydeligere forståelse for alle som jobber i utdanningssektoren. Av forskningsartiklene jeg leste, fant jeg få som tok for seg AT-oppgaver i lærebøker. Det kan svekke validiteten i denne oppgaven, ved at det gir et smalt sammenligningsgrunnlag. Flere lignende studier kan utfylle disse funnene, og eventuelt bidra med å vurdere denne studiens gyldighet. Jeg vil også foreslå å gjøre en dypere analyse i en lærebok for å se hvordan AT-strategien kan læres og brukes igjennom hele boka, fra start til slutt. For å vurdere om læreboka faktisk fremmer AT, og i så fall hvilke AT-ferdigheter, kan forskningen rettes mot å vurdere elevens tankesett før, underveis og etter bruk av boka, istedenfor å vurdere bøkene.

Funnene fra den horisontale analysen viser at minst 30% av oppgavene i lærebøkene legger til rette for AT. Funnene fra den vertikale analysen viser at flere av oppgavene ikke kobles opp mot matematikk. Dette stiller større krav til lærerens evne til å koble AT mot matematikkfaget. I begge bøkene gis rom for at eleven kan bruke strategien på ulike måter. Kategoriene «feilsøking» og «forestille seg» er underrepresentert i forhold til de andre ATO-kategoriene. Variasjonen av ATO-nivået i lærebøkene kan være en utfordring for elevers mulighet for læring. Funnene i lærebøkene tilsier at det kan være interessant å se på lærerens kompetanse innen AT. Ved å heve lærerens kompetanse på dette fagfeltet, kan det muliggjøre en mer lik praksis i undervisningen. Hvordan dette kan gjøres, kan det være verdt å se nærmere på.

8.0 Litteraturliste

- Alseth, B., Arnås, A.-C., Røsseland, M. & Nordberg, G (2021). *Multi 4A: matematikk for barnetrinnet: Elevbok* (3. utgave). Gyldendal.
- Alseth, B., Arnås, A.-C., Røsseland, M. & Nordberg, G (2021). *Multi 4B: matematikk for barnetrinnet: Elevbok* (3. utgave). Gyldendal.
- Anderson, N. D. (2016). A Call for Computational Thinking in Undergraduate Psychology. *Psychology Learning & Teaching*, 15(3), 226–234.
<https://doi.org/10.1177/1475725716659252>
- Aschehoug. (u.å.). *Matemagisk 1–7 (2020)—Læremidler*. Hentet 15. mai 2023, fra
https://skole.aschehoug.no/laremiddel/matemagisk-1-7-2020/grunnbok-grunnbok_1-4-bokmal
- Bocconi, S., & Chiocciariello, A. (2018). THE NORDIC APPROACH TO INTRODUCING COMPUTATIONAL THINKING AND PROGRAMMING IN COMPULSORY EDUCATION. *CNR Edizioni*. <https://doi.org/10.17471/54007>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing Computational Thinking in Compulsory Education—Implications for policy and practice. *JRC Publications Repository*. <https://doi.org/10.2791/792158>
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72. <https://doi.org/10.1145/3137065.3137069>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American*

educational research association, Vancouver, Canada (Vol. 1, p. 25).

<http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>

- Bråting, K., & Kilhamn, C., (2022). The Integration of Programming in Swedish School Mathematics: Investigating Elementary Mathematics Textbooks, *Scandinavian Journal of Educational Research*, 66:4, 594-609. <https://doi.org/10.1080/00313831.2021.1897879>
- Charalambous, C. Y., Delaney, S., Hsu, H.-Y., & Mesa, V. (2010). A Comparative Analysis of the Addition and Subtraction of Fractions in Textbooks from Three Countries. *Mathematical Thinking and Learning*, 12(2), 117–151. <https://doi.org/10.1080/10986060903460070>
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking—A guide for teachers. *Computing at School*. <https://eprints.soton.ac.uk/424545/>
- Fan, L., Zhu, Y., & Miao, Z. (2013). Textbook research in mathematics education: Development status and directions. *ZDM*, 45(5), 633–646. <https://doi.org/10.1007/s11858-013-0539-x>
- Fritzen, I.-L., Nilsen, E. K., Nilsen, M. & Nyborg, S. (2022). *Matemagisk 4A: matematikk for barnetrinnet: Grunnbok* (2. utgave). Aschehoug.
- Fritzen, I.-L., Nilsen, E. K., Nilsen, M., Nyborg, S., & Restani, I.-M. (2022). *Matemagisk 4B: matematikk for barnetrinnet: Grunnbok* (2. utgave). Aschehoug.
- Gadanidis, G., Hughes, J. M., Minniti, L., & White, B. J. G. (2017). Computational Thinking, Grade 1 Students and the Binomial Theorem. *Digital Experiences in Mathematics Education*, 3(2), 77–96. <https://doi.org/10.1007/s40751-016-0019-3>
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>

Gyldendal. (2021, 2. desember). *Oppgavetyper i nye Multi*.

<https://www.gyldendal.no/artikler/oppgavetyper-i-nye-multi/>

Haggarty, L., & Pepin, B. (2002). An Investigation of Mathematics Textbooks and Their Use in English, French and German Classrooms: Who Gets an Opportunity to Learn What? *British Educational Research Journal*, 28(4), 567–590.

Johannessen, A., Christoffersen, L., & Tufte, P. A. (2016). *Introduksjon til samfunnsvitenskapelig metode* (5. utg.). Abstrakt

Kilpatrick, Swafford, J., Findell, B., & National Research Council. (2001). Adding it up : helping children learn mathematics. National Academy Press.

Kunnskapsdepartementet (2017). *Overordnet del – verdier og prinsipper for grunnopplæringen*.

Fastsatt som forskrift ved kongelig resolusjon. Læreplanverket for Kunnskapsløftet 2020.

Kunnskapsdepartementet (2019). *Læreplan i matematikk (MAT01-05)*. Fastsatt som forskrift.

Læreplanverket for Kunnskapsløftet 2020.

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022a). Primary Mathematics Teachers'

Understanding of Computational Thinking. *KI - Künstliche Intelligenz*, 36(1), 35–46.

<https://doi.org/10.1007/s13218-021-00750-6>

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022b). Computational Thinking in the Primary

Mathematics Classroom: A Systematic Review. *Digital Experiences in Mathematics*

Education, 8(1), 27–49. <https://doi.org/10.1007/s40751-022-00102-5>

NOU 2014: 7 Elevens læring i fremtidens skole. Et kunnskapsgrunnlag. Kunnskapsdepartementet

NOU 2015: 8 Fremtidens skole. Fornyelse av fag og kompetanser. Kunnskapsdepartementet

Piatti, A., Adorni, G., El-Hamamsy, L., Negrini, L., Assaf, D., Gambardella, L., & Mondada, F.

(2022). The CT-cube: A framework for the design and the assessment of computational

thinking activities. *Computers in Human Behavior Reports*, 5, 100166.

<https://doi.org/10.1016/j.chbr.2021.100166>

Pólya, G. (1971). *How to solve it: a new aspect of mathematical method* (2nd ed). Princeton University Press.

Postholm, M. B., Jacobsen, D. I., & Søbstad, R. (2018). *Forskningsmetode for masterstudenter i lærerutdanningen*. Cappelen Damm akademisk.

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 169, 104222.

<https://doi.org/10.1016/j.compedu.2021.104222>

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking.

Educational Research Review, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Exploring the Science Framework and the NGSS: Computational Thinking in Elementary School Classrooms. *Science & Children*, 52(3), 10–15. https://doi.org/10.2505/4/sc14_052_03_10

Szetela, W., & Nicol, C. (1992). Evaluating problem solving in mathematics. *Educational Leadership*, 49(8), 42.

<https://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=9207130844&site=ehost-live>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.

<https://doi.org/10.1145/1118178.1118215>

Vedlegg

Vedlegg 1: Horisontal analyse av Matematisk 4A:

■

Kapittel	ATO i stor grad	ATO i liten grad	ATO i noen grad	Totalt
1: Geometri	A6, 1, 3, 5, 6, 7, 9, 16, SM17, SM19, SM25, 22, 23, A31, A33, 25 16 30,19%	SM7, 2, 4, SM10, SM13, 10, SM15, SM18, 11, 12, 13, 14, 15, SM22, SM23, SM24, SM27, 17, SM27, 18, SM29, 19, 20, SM31, 24, SM32, A34, 26, 27, 28, 29, 30, 31 33	8, SM20, SM21, SM39, 4	53
2: Multiplikasjon	7, 9, 11, 13, 15, 17, 19, 21, SM60, 29, 31, 33, 35, A76, 38, 39, 40, 41, 42, 43, 44, 45, 22 40%	A48, 2, SM50, 3, 4, 5, 8, 10, 12, 16, 20, 23, 24, 25, 26, 27, 28, 17	SM49, 1, 6, 14, 18, 22, SM63, SM64, 30, 32, 34, SM72, 36, 37, SM77, SM79 16	55
3: Volum	13, 20, SM116, 25, 26, 27, 28, 29, A121, 9 18,75	SM94, SM94, SM95, 1, 2, A99, A100, 3, 4, SM102, 5, 6, 7, 8, 9, 10, SM106, 11, 109, A110, 12, 14, 15, 16, 17, SM114, 18, 19, 21, 22, 23, 24, SM120, 30, 31, 32, 33, 37	A96, A108, 2	48
Totalt	47 30,13%	87	22	156

Vedlegg 2: Horizontal analyse av Matematisk 4B:



Kapittel	I stor grad ATO	I noen grad ATO	I liten grad ATO	Totalt
4: Divisjon	A6, A16, 19, 20, SM31, 28, 29, A34, 39, S45, SM50, 55, SM59, SM60 14 17,97%	SM8, 1, 12, SM24, SM29, SM46, 52, 7	SM7, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, SM17, 13, 14, 15, 16, 17, 18, 21, 22, SM26, 23, 24, 25, 26, 27, 30, 31, SM36, 32, 33, 34, 35, 36, 37, 38, SM38, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 53, SM56, 54, 56, 57, 58, SM58, 59 57	78
5: De fire regnearterne	A78, SM79, 20, 26, 29, 32, 33, A99, 41, 42, 43, 44, 47, 48 14 23,73%	SM74, SM81, A93, 30, 31, SM103, 6	SM76, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, SM87, 21, 22, 23, 24, 25, 27, 28, 34, 35, 36, 37, 38, 39, 40, 45, SM104, 46, SM105, 39	59
6: Programmering	Alle 20 100%			20
	48 30,57%	13	96	157

Vedlegg 3: Horizontal analyse av Multi 4A

Kapittel	ATO i stor grad	ATO i noen grad	AT i liten grad	Totalt
1: Tall og regning				
	10, 14, 16, 21, 30, 34, 42, 44, 57, 61, 68, 88, 89, 91, 92, 93, 94A7, U4, U11, Spill 11 A13, Spill : rund av til hundre, U32, U36, Spill : Hvem får det største svaret?, U45, U51, 60, Spill : skyt sifrene bort, Spill : addisjonsspill, Spill : subtraksjonsspill, U83, Spill : krig med negative tall, U86, U90, Spill 39,	22, 28, 29, 46, 47, 48, 49, 50, 52, 53, 54, 55, 59, 72, 73,74, 75, 76, 77, 78, 79, 80, 81, 82,U27, U41, U56,	1, 2, 3, 5, 6, 7, 8, 13, 15, 18, 19, 20, 24, 25, 26, 33, 35, 37, 38, 40, 43, 58, 63, 65, 67, 69, 70, 84, 85, 87U9, U12, U17, U23, U31, U39, U62, U64, U66, U71	104
	37	27	40	
	35,58%			
Kapittel 2: Multiplikasjon				
	20, 21, 23, 24, 71, Spill 47, U22, spill 49, spill 51, U33, spill 55, U80,	26, 30, 31, 34, 35, 36, 42, 51, 58, 63, A46, A48,	2, 3, 4, 5, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 27, 28, 32, 37, 38, 39, 41, 43, 44, 45, 46, 49, 50, 52, 53, 54, 55, 56, 59, 60, 61, 64, 65, 66, 67, 68, 69, 70, 72, 73, 74, 75, 76, 77, 78, 79,U1, U6, U10, U15, U29, U40, U47, U48, U57, U62,	86
	12	12	62	
	13,95			
Kapittel 3: Divisjon				
	21, 23, 27, 28, 39, 40, 41, 44, 45, 48, Spill 71, Spill 77, Spill 79, U29, Spill 81, Spill 87, Spill 89, U15,	15, 20, 30, 31, 32, 37, 42, 43, 51, 52, A69, U19, A82, A86,	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16, 17, 22, 25, 26, 33, 35, 46, 47, 50, 53, 54,U2, U13, U14, U18, U24, U34, U36, U38,	63
	18	14	31	
	28,57%			
Kapittel 4: Geometri				
Prosessorientert	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 19, 20, 22, 23, 24, 25, 26, 28, 29, 32, 34, 35, 36, 38 A95, U2, A99, A101, U13, A102, A104, U18, A106, A108, U27, U30, A113, U33, A117	U37,	14, 21, 31,	47
	43	1	3	
	91,49%			
Totalt	110	54	136	300
	36,67%			

Vedlegg 4: Horisontal analyse av Multi 4B

Kapittel 5: Brøk og desimaltall	ATO i stor grad	ATO i noen grad	ATO i liten grad	Totalt
	4, 5, 14, 15, 17, 25, 29, 33, 37, 40, 43, 44, 45, 47, 48, 56, 57, 58, 59, 68, 69, 70, 71, U1, A6, 3, 7, A7, U13, A10, U16, A12, spill13 , spill15 , U26, U30, A17, A18, spill21 , U42, U46, spill25 , U55, spill31 , U67, spill33 ,	2, 12, 27, 28, U6, 5	9, 10, 11, 19, 20, 21, 23, 24, 32, 34, 35, 36, 39, 41, 50, 51, 52, 53, 54, 60, 61, 62, 63, 64, 65, 66, U8, U18, U22, U31, U38, U49, 32	83
	46 55,42%			
Kapittel 6: Mønster og algoritmer	ATO i stor grad	ATO i noen grad	ATO i liten grad	Totalt
	35 STK 97,22%	1		36
Kapittel 7: Måling	ATO i stor grad	ATO i noen grad	ATO i liten grad	Totalt
Proessorientert	9, 13, 20, 28, 51, 55, 62, U2, A58, spill59 , U14, spill62 , U17, A64, spill67 , U26, U27, A69, U40, U41, A75, spill77 , U50,	A57, 12, 19, 30, 31, 61, 63, 64, A70, U32, A76, 11	1, 3, 4, 6, 7, 8, 11, 15, 16, 18, 22, 23, 25, 29, 33, 34, 35, 36, 37, 38, 39, 43, 44, 46, 47, 48, 49, 52, 53, 56, 57, 58, 60, U5, U10, U21, U24, U42, U45, U54, U59, 41	75
	23 30,67%			
Kapittel 8: Regning	ATO i stor grad	ATO i noen grad	ATO i liten grad	Totalt
Proessorientert	12, 13, 38, 44, 45, 55, 66, 67, 69, 73, 82, Spill89 , U5, spill91 , U9, U17!! , 21, spill95 , spill97 , U37, spill101 , spill103 , U46, spill105 , U57, U62, U68, U76, U83,	19, 20, 25, 30, 51, 54, 59, 71, 72, 74, 79, 80, 87 U41, 43, U56, A106	2, 3, 4, 6, 7, 8, 10, 11, 14, 15, 18, 22, U23, 24, 26, 27, 28, 31, 32, 33, 34, 35, 36, 42, 47, 48, 49, 50, 52, 53, 58, 60, 61, 63, 64, 65, 70, 75, 77, 78, 81, 84, 85, 86, U1, U16, U29, U39, U40	95
	29 30,53	17	49	
Totalt	133 46,02%	34	122	289

Vedlegg 5: Vertikal analyse av Matemagisk 4B: Programmering

Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Aktivitet	★ Mønstergjenkjenning Algoritmedesign						★ 90 grader
Aktivitet	★	★ Dekomposisjon Algoritmedesign					
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Snakke matte	★	★ Dekomposisjon Algoritmedesign Evaluering			★		★ 90 grader Rutenett
1			★ Evaluering Algoritmedesign				
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
2			★ Algoritmedesign Evaluering Abstraksjon				★ 90 grader
3		★ Dekomposisjon Algoritmedesign					
4		★ Mønstergjenkjenning Dekomposisjon Algoritmedesign					
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Snakke matte	★				★ Mønstergjenkjenning Algoritmedesign		
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
5		★ Mønstergjenkjenning Dekomposisjon Algoritmedesign		★			
6			★ Algoritmedesign Abstraksjon Evaluering				"Snu 90 grader"
7		★ Mønstergjenkjenning Dekomposisjon Algoritmedesign		★			Legger opp til å snu 90 grader
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
8		★ Mønstergjenkjenning Algoritmedesign		★			"Tegne et kvadrat"
Oppgaver	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
Snakke matte		★ Abstraksjon Dekomposisjon Algoritmedesign			★		Rutenett
9		★ Abstraksjon Algoritmedesign					
Snakke matte	★	★ Abstraksjon Dekomposisjon Algoritmedesign			★ "Snakke matte"		
10			★ Evaluering Algoritmedesign Abstraksjon				
11			★ Evaluering Abstraksjon Algoritmedesign				
Snakke matte	★	★ Dekomposisjon Algoritmedesign			★		
12		★	★ Evaluering Algoritmedesign				
13		★ Dekomposisjon Algoritmedesign					Geometri

Vedlegg 6: Vertikal analyse av Multi 4B: Mønster og algoritmer

Oppgave	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
1	★			★ Liten grad			Symmetri
2				★ Mønstergjennkjenning			Symmetri av figur i rutenett
Aktivitet side 38		★ Dekomposisjon		Algoritmedesign			Mønster med forskyvning
		Algoritmedesign					
Oppgave 3	★			★ Mønstergjennkjenning			Mønster med forskyvning
				Algoritmedesign			
Oppgave 4	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
4 Utforskning				★ Algoritmedesign	★ "Hvordan er figuren forskjøvet?"		Forskyvning i rutenett
				Mønstergjennkjenning			
5				★ Algoritmedesign	★ "Hvordan er disse figurene forskjøvet?"		Forskyvning i rutenett
				Mønstergjennkjenning			
6				★ Algoritmedesign			Geometriske former, forskyvning i koordinatsystem
Aktivitet side 40	★	★ Algoritmedesign		Mønstergjennkjenning			Mønster og geometriske former
		Dekomposisjon		Dekomposisjon			
7				★ Mønstergjennkjenning			Mønster med rotasjon
				Dekomposisjon			
8	★						Mønster med rotasjon
	Algoritmedesign			★ Algoritmedesign			
9				Mønstergjennkjenning			Forskyvning, spelling og rotasjon
10 Utforskning				★			Figurtall
				Mønstergjennkjenning			
				Dekomposisjon			
				Algoritmedesign			
Oppgave Aktivitet side 42	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
		★ Mønstergjennkjenning		★			Mønster
		Dekomposisjon					
		Algoritmedesign		★			Figurtall
11				Mønstergjennkjenning			
				Dekomposisjon			
				Algoritmedesign			Figurtall
12				Mønstergjennkjenning			
				Dekomposisjon			
				Algoritmedesign			Figurtall
13 Utforskning 8				★			Regning
				Mønstergjennkjenning			
				Dekomposisjon			
				Algoritmedesign			
14 Utforskning 8				★			Figurtall
				Mønstergjennkjenning			Regning
				Dekomposisjon			
				Algoritmedesign			
				Abstraksjon			
15				★			Figurtall
				Mønstergjennkjenning			Regning
				Dekomposisjon			
				Abstraksjon			
				Algoritmedesign			
16				★			Figurtall
				Mønstergjennkjenning			Regning
				Abstraksjon			
				Dekomposisjon			
				Algoritmedesign			
17 Utforskning 8				★			Figurtall
				Mønstergjennkjenning			Regning
				Abstraksjon			
				Dekomposisjon			
				Algoritmedesign			
Oppgave 18	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
18 Utforskning 8				★			Figurtall
				Mønstergjennkjenning			Regning
				Abstraksjon			
				Dekomposisjon			
				Algoritmedesign			Figurtall
19				★			Regning
				Mønstergjennkjenning			
				Abstraksjon			
				Dekomposisjon			
				Algoritmedesign			Figurtall
20				★			
				Mønstergjennkjenning			
				Abstraksjon			
				Dekomposisjon			
				Algoritmedesign			
Oppgave 21	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
21 Utforskning 8			★				
			Evaluerings				
			Algoritmedesign				
22		★					
		Dekomposisjon					
		Algoritmedesign					Rutenett
23		★					
		Dekomposisjon					
		Algoritmedesign					Rutenett
24	★						
	Algoritmedesign						Rutenett
Spill - kongen befaler	★	★					
	Evaluerings	Dekomposisjon					Rutenett
	Algoritmedesign	Algoritmedesign					
25		★					90 grader
25 Utforskning 8		Dekomposisjon					Rutenett
		Algoritmedesign					
26	★						Rutenett
	Algoritmedesign						
27		★					Rutenett
		Dekomposisjon					
		Algoritmedesign					
28	★						Rutenett
	Algoritmedesign						
Spill - Blått lys	★	★					
	Algoritmedesign						
Oppgave 29	Følge algoritme	Lage algoritme	Feilsøke	Finne et mønster	Forklare	Forestille seg	Matte
	★						Divisjon
	Algoritmedesign						Rutenett
30	★						
	Algoritmedesign						Regning med vilkårene
Spill - Karibia rundt	★						Penger
	Algoritmedesign						Ulikhetstegn