Haaga-Helia
ammattikorkeakoulu Oy

Ohjelmoinnin opetus peruskoulussa

Laura Hiltunen

Opinnäytetyö
Tietojenkäsittely
2021

Tiivistelmä

Haaga-Helia ammattikorkeakoulu Oy

Tekijä
Hiltunen Laura

Koulutusohjelma
Tietojenkäsittelyn koulutusohjelma

Raportin/Opinnäytetyön nimi	**Sivu- ja liitesivumäärä**
Ohjelmoinnin opetus peruskoulussa	36 + 3

Tämän systemaattisen kirjallisuuskatsauksen tarkoituksena oli selvittää, millaiset tekijät edesauttavat ohjelmoinnin opettamista ja oppimista. Tavoitteena oli selvittää, mitä asioita tulee huomioida ohjelmoinnin opetuksessa suomalaisen peruskoulun kontekstissa.

Tässä tutkimuksessa tarkasteltiin vuosien 2017–2019 välillä julkaistua kansainvälistä tutkimuskirjallisuutta, joka käsittelee ohjelmoinnin opettamista ja opiskelua. Tutkimusaineisto on kerätty kansainvälisistä tieteellisistä artikkeleista koostuvasta EBSCOhost Academic Search Premier -artikkelitietokannasta. Yhdeksästä artikkelista koostuva aineisto on kerätty kesällä 2020. Artikkeleista koostuva aineisto on valittu tutkimuskysymykseen pohjaavien sisään- ja poissulkukriteereiden perusteella. Tutkimus aineiston keruun sisältäen on toteutettu Finkin mallin mukaisesti. Aineisto on analysoitu sisällönanalyysin keinoin.

Tutkielman tulosten mukaan ohjelmoinnin opettamiseen ja oppimiseen vaikuttavat tekijät jakautuvat kolmen yläluokan alle. Yläluokkia ovat (1) ohjelmointityökaluun liittyvät asiat, (2) opettajaan, opetukseen tai koulunkäyntiin liittyvät asiat sekä (3) oppilaan tekemiseen, ajatteluun ja taitoihin liittyvä asiat. Yläluokkien alle muodostui lisäksi kahdeksan alaluokkaa, joista neljä liittyi opettajan taitoihin ja neljä oppilaan taitoihin tai ominaisuuksiin. Ohjelmointityökaluun liittyviä alaluokkia ei ollut lainkaan.

Ohjelmoinnin oppimiseen vaikuttaa opetuksessa käytettävän ohjelmointityökalun selkeä ja miellyttävä visuaalinen ilme ja käyttöliittymän helppokäyttöisyys, sekä mahdollisuus saada reaaliaikaista palautetta. Ohjelmoinnin oppimista edesauttaa ohjelmointia opettavan opettajan hyvä sisältöosaaminen sekä opettajan käyttämät tarkoituksenmukaiset ja itselleen sopivat pedagogiset ja kommunikoinnin keinot. Ohjelmoinnin oppimiseen vaikuttavat myös oppilaan kommunikointi- ja yhteistyötaidot sekä algoritmisen ajattelun taidot. Lisäksi jo opitut ohjelmoinnin taidot edesauttavat uuden oppimista.

Asiasanat
ohjelmointi, algoritminen ajattelu, opettaminen, oppiminen, peruskoulu, systemaattinen kirjallisuuskatsaus

Sisällys

1 Johdanto

Suomalainen perusopetus on Opetushallituksen säätelemää ja opetuksen sisällöistä määrätään Perusopetuksen opetussuunnitelman perusteissa, joka uusitaan säännöllisesti vastaamaan koulutuksen tarpeita. Näin pyritään takaamaan laadukas ja tasa-arvoinen opetus kaikille oppilaille ja luomaan edellytykset kasvulle, kehitykselle ja oppimiselle. (POPS 2014.) Voidaan ajatella, että jokainen opetussuunnitelma on oman aikansa peili. Opetukseen halutaan sisällyttää kulloinkin niitä asioita ja aiheita, jotka nähdään yhteiskunnan ja tulevaisuuden kannalta arvokkaina.

Ohjelmointi saapui alakoulun opetukseen opetussuunnitelman perusteiden mukaisesti vuonna 2016 ja porrastetusti yläkouluun vuodesta 2017 lähtien. Tätä opinnäytetyötä kirjoittaessani, muutoksesta on kulunut viisi vuotta. Ohjelmoinnin opettaminen on nostanut viime vuosina esiin kritiikkiä sekä hämmennystä opetuksen ammattilaisten keskuudessa. Peruskouluikäisten ohjelmoinnin opetukselle on ollut myös monia puolestapuhujia, ja tästä johtuen se on alun perin tuotu osaksi perusopetusta.

Ohjelmointi on pitkään ollut vain rajatun ihmisryhmän erityisosaamisalaa, kun tietokoneita ja ohjelmistoja ovat käyttäneet vain asiantuntija-asemassa olevat ihmiset. Tällä vuosituhannella kehitys on kuitenkin kääntynyt täysin päinvastaiseksi. Internetin myötä, lähes jokaisella länsimaalaisella ihmisellä on käytössään yksi tai useampi verkkoon yhdistetty tietokone. Myös yhä useammassa kodin sähkölaitteessa on nykyisin tietokone, puhumattakaan yrityselämän tai teollisuuden laitteista. Tietokoneita ja niillä olevia ohjelmistoja on kaikkialla, ja niiden määrän voidaan olettaa kasvavan entisestään.

Perusopetuksen digitalisaatio sekä siihen kuuluva ohjelmoinnin opetus tuntuu tästä näkökulmasta katsottuna perustellulta. Vaikka kaikista peruskoululaisista ei tule ohjelmoijia, niin kuin ei heistä tule matemaatikkoja tai biologejakaan, on ohjelmoinnin ja ohjelmistojen ymmärtäminen kuitenkin yleissivistyksen kannalta tärkeää. Ohjelmoinnin opetuksen tarkoituksena on tuoda ymmärrystä siitä, miten meitä ympäröivät laitteet toimivat, ja lisätä tietoisuutta ihmisestä ohjelmistojen toiminnan takana (POPS 2014, 157). Niin kuin kaikkea ihmisen toimintaa, myös ohjelmointia sekä ohjelmistojen rakentamista tulee pohtia myös eettisestä näkökulmasta.

Opetussuunnitelman perusteissa mainitaan yhdeksi ohjelmoinnin opiskelun tavoitteeksi algoritmisen ajattelun kehittäminen. Algoritmisella ajattelulla tarkoitetaan laajempia ongelmanratkaisuun liittyviä ajatteluntaitoja, joita ovat esimerkiksi ongelman purkaminen osiin

sekä ratkaisun vaiheittainen tuottaminen. (Mykkänen & Liukas 2014.) Algoritmisen ajattelun taitoja on harjoiteltu perusopetuksessa jo pitkään, joten mistään uudesta asiasta ei ole kyse. Yksi esimerkki alakoulussa opiskeltavasta algoritmista on allekkain jakaminen, joka muodostuu ennalta määritellyistä, ratkaisuun johtavista vaiheista. Vastaavanlaisia vaiheittaisia malleja voidaan käyttää myös ohjelmointiongelmien ratkaisemiseen.

Vaikka algoritmisen ajattelun taitojen harjoittelu ei ole koulumaailmassa uutta, tuottaa ohjelmoinnin opettaminen monelle päänvaivaa ja haasteita. Koska ohjelmointi ja siihen liittyvät sisällöt ovat osana matematiikan ja jossain määrin myös käsityön oppiaineiden sisältöjä, on ohjelmoinnin opetusvastuu alakoulussa luokanopettajilla ja yläkoulussa matematiikan aineenopettajilla. Vaikka täydennyskouluttautumiseen on ollut monilla mahdollisuus, saattaa opettajien ohjelmoinnin sisältöosaaminen silti olla vajavaista. Tästä näkökulmasta katsottuna, on hyvin ymmärrettävää, että varsin nopeasti alkanut ohjelmoinnin opettaminen peruskouluissa on saattanut monet opettajat epämukavuusalueelleen.

Olen itse koulutukseltani luokanopettaja, ja sen lisäksi minulla on pätevyys toimia myös maatiedon aineenopettajana sekä erityisopettajana. Aloitin opettajan urani samaan aikaan kun ohjelmoinnin opetuksen oli tarkoitus alkaa perusopetuksessa, ja olen luokanopettajana toimiessani opettanut ohjelmointia alakoulun ensimmäisillä luokilla. Tätä opinnäytetyötä kirjoittaessani, työskentelen yläkoulussa erityisopettajana ja olen työssäni saanut opettamaa ohjelmointia valinnaisella ohjelmointikurssilla yhdeksäsluokkalaisille. Opetustyön ohella olen opiskellut ohjelmointia sekä tietotekniikkaa, ja näin ollen kokenut oman ohjelmointiosaamiseni riittäväksi peruskoulun kontekstiin. Kuitenkin olen nähnyt myös asian kääntöpuolen, kun osa kollegoista on tuskaillut ohjelmointia koskevien sisältötavoitteiden parissa.

Valitsin opinnäytetyöni aiheeksi ohjelmoinnin opettamisen peruskoulukontekstissa, sillä se on nivoutunut omaan arkielämääni. Samalla koen tämän aiheen myös hyvin tärkeäksi, sillä ohjelmointi on taito, jolle tarvitaan kasvavassa määrin lisää osaajia. Itse olen opettanut pienille oppilaille ohjelmointia leikin ja tekemisen kautta. Yläkouluikäisten kanssa olemme harjoitelleet ohjelmointia graafisen käyttöliittymän avulla sekä tutustuneet Python-ohjelmointikielen perusteisiin. Vaikka suomalaisella opetuksella on vankka tieteellinen pohja, tulee ajoittain vastaan myös opetustekniikoita ja -metodeja, jotka noudattavat trendejä ja ovat kaupallistettuja. Tämän opinnäytetyön tavoitteena onkin tutkia, millaista tieteellistä tutkimusta ohjelmoinnin opetuksesta on tehty ja millä tavoin ohjelmoinnin opetus ja oppiminen on tuloksekasta. Kiinnostavaa on, ovatko nykyiset opetuksessa käytetyt ohjelmoinnin opetuksen menetelmät tieteellisesti relevantteja vai seurausta trendeistä.

Opinnäytetyössä käytettyjä käsitteitä:

algoritmi	Algoritmi on kuvaus jonkin tehtävän suorittamiseen vaadituista toimenpiteistä (Mykkänen & Liukas 2014).
algoritminen ajattelu	Algoritmiseen ajatteluun (eng. Computational thinking) kuuluvat luovaan ongelmanratkaisuun liittyvät ajattelun taidot: ttaito purkaa ongelmat osiin, taito tunnistaa kaavoja, taito luoda algoritmeja sekä taito yleistää ja automatisoida ratkaisuja (Mykkänen & Liukas 2014).
digitalisaatio	Digitalisaatiolla tarkoitetaan digitaalisuuden ja tietotekniikan yleistymistä ja hyödyntämistä arki- ja yrityselämän toiminnoissa.
konstruktivismi	Konstruktivismi on psykologiaan pohjautuva oppimiskäsitys, jonka mukaan oppija oppii rakentamalla, eli konstruoimalla, uutta tietoa aiemmin opitun päälle (Bada & Olusegun 2015, 66).
lähikehityksen vyöhyke	Lähikehityksen vyöhyke on kognitiivinen etäisyys lapsen nykyisen osaamisen ja aikuisen avustuksella saavutettavan osaamisen välillä (Gibbons 2002, 13–14).
scaffolding	Scaffolding tarkoittaa opettajan tai kasvattajan antamaa oppimisen tukea, jonka määrää säädellään lapsen taitojen mukaan. Tukea annetaan juuri sen verran, kuin lapsi tarvitsee harjoiteltavan taidon suorittamiseksi. (Hammond & Gibbons 2005, 8–9.)

2 Konstruktivistinen oppimiskäsitys

Ihminen aloittaa oppimisen heti syntymänsä jälkeen. Aivot muovautuvat ja kehittyvät paljon jo sitä ennen. Oppimiseen vaikuttaa biologia sekä ajan myötä tapahtuva kasvaminen ja kehittyminen. Kasvaminen ei kuitenkaan tapahdu tyhjiössä, vaan kehittyäkseen ja oppiakseen, ihminen tarvitsee muita ihmisiä ja sitä kautta saatavaa sosiaalista vuorovaikutusta. Näihin asioihin liittyy olennaisesti koulutusjärjestelmä, opettaminen ja opiskelu. Esittelen seuraavaksi kasvatuksen ja oppimisen tutkimukseen pohjautuvia teorioita sekä tällä hetkellä vallalla olevaa oppimiskäsitystä.

2.1 Konstruktivismi ja kieli konstruktion välineenä

Tämänhetkinen oppimiskäsitys pohjautuu konstruktivismiin. Psykologiaan pohjautuvan konstruktivismin perusajatuksen mukaan oppija oppii rakentamalla, eli konstruoimalla, uutta tietoa aiemmin opitun päälle. Aikaisemmat opit muovautuvat uuden tiedon myötä, ja oppija tarkastele nykyistä ymmärrystään uuden valossa. Oppiminen on siis aktiivinen tapahtuma, jonka keskeinen toimija on oppija itse. (Bada & Olusegun 2015, 66.)

Konstruktivistisen oppimiskäsityksen mukaan, oppimistapahtumaan vaikuttaa se konteksti, jossa oppiminen tapahtuu. Oppiminen pohjautuu siis oppijan kokemuksiin. Lisäksi oppimiseen vaikuttavat oppijan aiemmat uskomukset sekä asenteet. (Bada & Olusegun 2015, 66.) Myös kieli ja ajattelu ovat kytköksissä oppimiseen, sillä ihmiset ovat symbolisia olentoja. Tällä tarkoitetaan sitä, että ihmisillä on tapana esittää ja jäsentää tietoa erilaisten merkkien ja merkkijärjestelmien avulla. (Amsel, Amsel, Byrnes & Jean Piaget Society Staff 2002, 3.) Oppimisen ja kielen yhteys on nähtävillä myös koulujärjestelmässämme, jossa suuri osa opetettavasta ja opittavasta informaatiosta esitetään kielen kautta. Tavoitteellinen opiskelu aloitetaan usein myös siinä iässä, kun kieltä opitaan jäsentämään kirjoittamisen avulla. (Painter 2005.) Suomessa lukemaan opettelu alkaa esiopetuksessa, joka käydään noin kuuden vuoden ikäisenä, ennen ensimmäisen luokan alkamista.

Oppimista tapahtuu siis oppijan kokemusten sekä kielen kautta. Näiden lisäksi vuorovaikutus on tärkeä osa oppimista. Ihmislapsi oppii ensimmäiset taitonsa vuorovaikutuksessa lähimpien aikuisten vahvistamana. (Wells & Bridges 1981.) Oppimista tapahtuu vuorovaikutuksen lisäksi laajemmassa sosiokulttuurisessa kontekstissa, johon vaikuttavat oppijan sosiaaliset, kulttuuriset ja historialliset kokemukset. Oppiminen voidaan siis nähdä sosiaalisena tapahtumana. (Gibbons 2002, 13.)

Oppimisen sosiaalisuuteen liittyy viime vuosisadalla vaikuttaneen venäläisen psykologin Lev Vygotskyn teoria *lähikehityksen vyöhykkeestä*. Lähikehityksen vyöhykkeellä viitataan kognitiiviseen etäisyyteen lapsen nykyisen osaamisen ja aikuisen avustuksella saavutettavan osaamisen välillä. Oppimista tapahtuu, kun aikuinen antaa lapselle juuri sopivan verran tukea ja apua, jotta lapsi kykenee suorittamaan opettelemaansa tehtävän. (Gibbons 2002, 13–14.)

Lähikehityksen vyöhykkeeseen liitetään usein termi *scaffolding*, joka tarkoittaa kirjaimellisesti käännettynä rakennustelinettä. Sillä tarkoitetaan opettajan tai kasvattajan antamaa oppimisen tukea. Aluksi tukea annetaan tehtävän tai toiminnon suorittamiseksi paljon ja myöhemmin tuen ja avun määrää voidaan vähentää, kunnes oppilas suoriutuu tehtävästä itsenäisesti. Fyysisten aktiviteettien oppimisen lisäksi scaffolding toimii myös ajattelun taitojen opettamisessa. Aikuisen avustuksella oppilas kykenee suoriutumaan tehtävistä, joihin hän ei yksin kykenisi. Pikkuhiljaa hän omaksuu oikeanlaiset toimintamallit ja aikuinen voi vähentää tarjottua tukea. Lopulta oppilas suoriutuu tehtävästä täysin ilman ulkopuolista apua. (Hammond & Gibbons 2005, 8–9.)

Oppiminen tapahtuu sosiaalisessa kontekstissa ja siihen vaikuttavat eritoten kokemukset ja kieli. Konstruktivismin mukaista oppimiskäsitystä sovelletaan monin paikoin suoraan nykyopetuksessa ja -koulutuksessa. Konstruktivismin voidaan ajatella toimivan opetuksen pohjana, mutta sen lisäksi opetuksessa ja koulutuksessa näkyvät monet muut oppimisen teoriat ja trendit. (Bada & Olusegun 2015, 66.)

2.2 Lapsen kasvaessa ajattelu kehittyy

Kasvaessa, vartalon kehittymisen lisäksi kasvaa ja kehittyy myös ajattelu. Kehityspsykologia on tieteenala, joka tutkii ihmisten ikään liittyvää kehitystä ja muutoksia. Yksi 1900-luvun tunnetuimpiin tutkijoihin kuuluva kehityspsykologi Jean Piaget tutki ihmisen ajattelun kehitystä erityisesti varhaislapsuudesta aikuisuuteen.

Yksi Piaget'n tunnetuimpia teorioita on ajattelun kehittymistä kuvaava vaiheteoria. Piaget'n teoria tukee konstruktivistista oppimiskäsitystä, ja moni koulujärjestelmä ja opetusideologia pohjautuu vaiheteorian varaan. (Huitt & Hummel 2003.) Piaget'n teoria koostuu neljästä ajattelun kehityksen vaiheesta, joita ovat sensomotorisen, esioperationaalisen, konkreettisten operaatioiden ja formaalin ajattelun vaiheet.

Sensomotorinen vaihe: Syntymästä taaperoikään asti, lapset ovat kehityksseltään sensomotorisen ajattelun vaiheessa. Tämän kauden aikana lapsi alkaa hahmottaa itseään toimijana ja muodostamaan toimintaan pohjautuvia ajatusmalleja.

Esioperationaalinen vaihe: Alle kouluikäisinä, esioperationaalisessa vaiheessa, lapsi oppii käyttämään kieltä ja kuvaamaan todellisuutta symbolien ja kielen kautta. Tässä vaiheessa ajattelu on vielä hyvin kirjaimellista ja ajatusten ja tekojen erottaminen toisistaan on vaikeaa.

Konkreettisten operaatioiden vaihe: Kouluikäisenä, konkreettisten operaatioiden vaiheessa, ajattelu kehittyy niin, että lapsi osaa luokitella ja lajitella asioita ominaisuuksiensa perusteella sekä ajatella loogisesti. Samaan aikaan ajantaju ja ajallinen ymmärrys sekä välimatkojen hahmottaminen kehittyvät. Lapsi alkaa myös ymmärtää yksinkertaisia abstrakteja käsitteitä suhteessa konkreettisiin asioihin sekä asioiden ominaisuuksien säilyvyyttä. Lapselle kehittyy myös kyky harkita tekoja tarkoituksien mukaan.

Formaalien operaatioiden vaihe: Nuoruusiässä ihmisen ajattelu kypsyy, jolloin abstrakti ajattelu mahdollistuu ja päättelykyvyn loogisuus kehittyy entisestään. Piaget'n teorian mukaan formaali ajattelu kehittyy murrosiän alussa 11–12 -vuotiaana, mutta myöhemmät tutkimukset ovat osoittaneet, että formaalin ajattelun kehitykseen voi mennä paljon kauemmin. Vain noin kaksi kolmesta korkeakouluikäisestä on saavuttanut formaalin ajattelun tason. Toisin sanoen, kaikki nuoret tai nuoret aikuiset eivät välttämättä ole kehittyneet formaalisen ajattelun tasolle.

(Huitt & Hummel 2003.)

Klassinen esimerkki esioperationaalisen ajattelun ja konkreettiseen ajatteluun välisestä erosta on testi, joka tehdään vedellä ja erimuotoisilla astioilla. Testin aluksi kannuun laitetaan tietty määrä vettä. Sitten lapselta kysytään mitä vedelle tapahtuu, kun se kaadetaan ensin korkeaan ja kapeaan lasiin. Konkreettisten operaatioiden vaiheessa oleva lapsi vastaa, että veden määrä pysyy samana. Esioperationaalisella ajattelun tasolla oleva lapsi vastaa kuitenkin eri tavalla. Hänen ymmärryksensä mukaan veden määrä lisääntyy, sillä veden pinta on korkeammalla kuin suuressa kannussa. Samaan tapaan, kun korkeassa

lasissa oleva vesi kaadetaan matalaan, laakeaan astiaan, sama lapsi toteaa veden määrän vähentyneen, sillä veden pinta on matalalla. (Piaget 1964) Esioperationaalisessa ajattelun vaiheessa lapsi ei siis vielä ymmärrä asioiden ominaisuuksien säilyvyyttä.

Vaikka konkreettisten operaatioiden vaiheessa lapsi pystyy abstraktiin ajatteluun suhteessa konkreettisiin asioihin, ei hän vielä kykene korkeampaan abstraktiin ajatteluun. Aineettomien asioiden sekä johdonmukaisen abstraktin ajattelun mahdollistava formaali ajattelu kehittyy paitsi iän myötä, mutta myös ympäristöllä on suuri vaikutus sen kehitykseen. (Huitt & Hummel 2003.) On siis mahdollista, että lapsi tai nuori ei saavuta formaalin ajattelun tasoa peruskoulun aikana. Tämä tulee huomioida myös oppimisen ja opettamisen kontekstissa.

3 Ohjelmointi peruskoulussa opetettavana sisältönä

Ohjelmointi ja ohjelmoinnin opettaminen lisättiin vuonna 2016 voimaan tulleeseen, ja tällä hetkelläkin voimassaolevaan, perusopetuksen opetussuunnitelman perusteisiin. Ohjelmointi ei ollut ainoa muutos edelliseen opetussuunnitelmaan verrattuna, mutta se oli yksi eniten keskustelua herättäneistä sisällöistä. Ohjelmoinnin opiskelua tuodaan perusopetukseen osana laajempaa koulutuksen ja opetuksen digitalisaatiota. Vaikka opetussuunnitelmassa tuodaan esille tavoitteet ohjelmoinnille, ei ohjelmoinnin opetusta kuvailla kovinkaan yksiselitteisesti. Esittelen seuraavaksi, mitä valtakunnallinen opetussuunnitelma sanoo ohjelmoinnin opettamisesta sekä perehdyn siihen, miten näitä asioita tuodaan konkreettisesti osaksi opetusta.

3.1 Ohjelmointi perusopetuksen opetussuunnitelman perusteissa

Valtakunnallisen perusopetuksen opetussuunnitelman perusteissa opetuksen sisällöt, osaaminen ja tavoitteet on jaettu alakouluun vuosiluokille 1–2, 3–6 sekä yläkouluun vuosiluokille 7–9. Lisäksi opetussuunnitelmassa kuvataan kuudennen ja yhdeksännen luokan hyvää, eli arvosanan kahdeksan arvoista, osaamista eri oppiaineissa. Opetussuunnitelmassa opetuksen tavoitteet ja sisällöt on jaettu laaja-alaisiin osaamisen alueisiin. Näiden lisäksi opetussuunnitelmasta löytyy oppiainekohtaisia sisällöllisiä tavoitteita. (POPS 2014.)

Ohjelmointi on sisällytetty tämänhetkiseen valtakunnalliseen opetussuunnitelmaan sekä laaja-alaisen osaamisen tavoitteina että osana joidenkin oppiaineiden sisällöllisiä tavoitteita. Alakoulun puolella ohjelmoinnin sisällölliset tavoitteet on liitetty osaksi matematiikan

opiskelua, mutta yläkoulun puolella ohjelmointia koskevia sisältöjä on kirjattu matematiikan lisäksi käsityön opetukseen. Yläkoulun käsityön opetuksessa tulisi esimerkiksi käyttää sulautettuja järjestelmiä valmistettavissa tuotteissa. (POPS 2014.)

Vuosiluokilla 1–2 matematiikan oppiaineessa aloitetaan ohjelmoinnin alkeisiin tutustuminen. Tutustuminen aloitetaan harjoittelemalla vaiheittaisten toimintaohjeiden laatimista. (POPS 2014, 129.) Vuosiluokilla 3–6 opetuksen tavoitteena on innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmina graafisessa ohjelmointiympäristössä (POPS 2014, 235). Yläkoulussa yhdeksi matematiikan opetuksen tavoitteeksi listataan, että opetuksen tulisi ohjata oppilasta kehittämään algoritmista ajatteluaan sekä taitojaan soveltaa matematiikkaa ja ohjelmointia ongelmien ratkaisemiseen (POPS 2014, 375).

Edellä mainittujen tavoitteiden lisäksi sekä vuosiluokkien 3–6 että 7–9. matematiikan opetuksen tulisi osittain koostua ajattelun taitojen opettamisesta. Alakoulun puolella näihin ajattelun taitojen kehittämiseen lukeutuvat esimerkiksi yhtäläisyyksien, erojen ja säännönmukaisuuksien löytäminen sekä vertailun ja luokittelun harjoittelua ja havainnointia. Lisäksi ajattelun taitojen harjoittelussa mainitaan ohjelmien suunnittelu ja toteutus graafisessa ohjelmointiympäristössä. Yläkoulussa ajattelun taitojen sisältöihin kuuluvat loogisen ajattelun harjoittelu, johon kuuluu sääntöjen ja riippuvuuksien etsiminen. Lisäksi tavoitellaan päättelykyvyn vahvistamista sekä harjoitellaan todistamista. Ohjelmointiin liittyen yläkoulussa tulisi syventää algoritmista ajattelua, ohjelmoida ja harjoitella hyviä ohjelmointikäytäntöjä. (POPS 2014, 235, 375.)

Laaja-alaisen osaamisen osa-alue L5 on tieto- ja viestintäteknologinen osaaminen, jonka yksi osa on ohjelmointi. Alkuopetuksessa, eli vuosiluokilla 1–2, tähän laaja-alaiseen osaamisalueeseen on kirjattu tavoite, jonka mukaan oppilaiden tulisi saada ikätasolleen sopivia ohjelmointikokemuksia. Tämän lisäksi mainitaan näppäintaitojen harjoittelu sekä pelillisyyden hyödyntäminen opetuksessa. (POPS 2014, 101.) Vuosiluokilla 3–6 tavoitellaan oppilaiden ymmärrystä siitä, että koneiden ja teknologian toiminta on seurausta ihmisten toiminnasta, kuten ohjelmoinnista (POPS 2014, 157). Yläkoulussa eli vuosiluokilla 7–9 ohjelmointia tulisi harjoitella osana eri oppiaineiden opintoja (POPS 2014, 284).

Peruskoulun opetussuunnitelman perusteiden mukaan kuudesluokkalaisen tulisi osata ohjelmoida toimivia ohjelmia graafisessa ohjelmointiympäristössä. Yhdeksäsluokkalaisen tulisi osata soveltaa algoritmisen ajattelun periaatteita ja ohjelmoida yksinkertaisia ohjelmia. (POPS 2014, 239, 379.)

3.2 Ohjelmoinnin osaksi peruskoulua

Suomalainen perusopetus pohjautuu Opetushallituksen laatimaan perusopetuksen opetussuunnitelman perusteisiin. Tämä valtakunnallinen opetussuunnitelma on Opetushallituksen antama määräys, jonka mukaan koulujen ja kuntien paikalliset opetussuunnitelmat laaditaan. (Opetushallitus 2020.) Tämä tarkoittaa sitä, että kaikki oppivelvollisuutta Suomessa suorittavat opiskelijat opiskelevat näiden valtakunnallisten ohjeiden mukaisesti.

Perusopetuksen opetussuunnitelman perusteissa 2016 ohjelmointi on osana matematiikan sisältöjä. Ohjelmointia opettavat alakoulussa siis luokanopettajat ja yläkoulussa matematiikan ja käsityön aineenopettajat. Ohjelmoinnin opettamiselle ei ole muodollisia pätevyysvaatimuksia, mutta yleisessä keskustelussa on tullut esiin opettajien omat huolet osaamisestaan ohjelmoinnin opettamisessa (Mykkänen & Liukas 2014).

Tähän huoleen on vastattu tarjoamalla opettajille mahdollisuutta kouluttautua. Digiaiheista täydennyskoulutusta on tarjottu enenevissä määrin, ja noin kolmasosa peruskoulujen opettajista on kehittänyt osaamistaan. Silti suurin osa opettajista kokee, että osaaminen ei ole vielä riittävää. (Tanhua-Piiroinen, Kaarakainen, Kaarakainen, Viteli, Syvänen, & Kivinen 2019, 39.) Osaamisensa lisäksi opettajat ovat olleet huolissaan siitä, että ohjelmointi lohkaisee aikaa matematiikan opiskelusta (Mykkänen & Liukas 2014, 71).

Kokonaisvaltainen huoli on ollut myös koulujen laite- ja verkkoresursseista. Ohjelmoinnin ja tietotekniikan opiskeluun käytettyjen laitteiden määrä vaihtelee laajasti eri koulujen välillä. Monin paikoin laitteet ovat myös vanhentuneita tai lähes toimimattomia ja tämä lannistaa intoa uuden opettelulta. (Mykkänen & Liukas 2014, 71.)

Ohjelmointia on pyritty tuomaan kouluihin osana opetuksen digitalisaatiota. Keskeistä perusopetuksen digitalisaatiolle ovat kuntien ja koulujen digistrategiat ja rehtoreiden rooli oppilaitosten digitalisaatioprosessin johtamisessa. Koulujen digitaalinen ympäristö laitteineen, muodostaa pohjan teknologioiden käyttämiselle opetustilanteissa. (Tanhua-Piiroinen ym. 2019, 12.) Opettajien huoli laiteresursseista sekä olemassa olevien laitteiden huonosta kunnosta on siis perusteltu.

Yhteenvetona voidaan sanoa, että koulujen digitalisaatio ja sen osana ohjelmoinnin tuominen osaksi peruskoulujen opetusta, vaatii selkeää toimintaympäristön muutosta, johon kuuluu uudenlaisen toimintakulttuurin luominen sekä laitteiston ja muiden resurssien saatavuudesta huolehtiminen. Näiden lisäksi tulisi kehittää henkilöstön osaamista ja saada

heidät sitoutumaan muutokseen. Opetushenkilöstö on avainasemassa ohjelmoinnin opetuksen kehittämisessä.

3.3 Algoritminen ajattelu

Valtakunnallinen opetussuunnitelma nostaa ohjelmoinnin opetuksen tavoitteeksi algoritmisten ajattelutaitojen kehittämisen. Tarkoituksena ei ole opiskella ohjelmointikieliin pohjautuvaa ohjelmointia, vaan kehittää ajattelun taitoja ohjelmoinnin kautta. Varsinaista ohjelmointia tärkeämpää on siis harjoitella algoritmisen ajattelun perustaitoja. (Mykkänen & Liukas 2014, 77.)

Algoritmisesta ajattelusta on ensimmäisiä todisteita jo ennen ajanlaskun alkua. Noin 1650 vuotta ennen ajanlaskua egyptiläiset kuvasivat papyrukselle tapoja selvittää numeraalisia pulmia. Algoritmin käsite liitetään usein matematiikkaan ja näin asia onkin. Kuitenkin algoritmeja ja algoritmista ajattelua vaativia toimintoja löytyy kaikkialta ympäriltämme. (Ferragina & Luccio 2018, 1–2.) Käsitteenä algoritmi esiintyi ensimmäisen kerran vuonna 1202 Leonardo Fibonaccin matemaattisessa tutkielmassa. Nykyisin käsite liitetään matematiikan lisäksi yleisesti myös ohjelmointiin. Voidaan ajatella, että algoritmi on ajatus ennen koodin kirjoittamista ja koodi on tuon ajatuksen muotoileminen askelittaisiksi toimintaohjeiksi. (Ferragina & Luccio 2018, 11.) Toisin sanoen, algoritmi tarkoittaa kuvausta jonkin toimenpiteen suorittamiseksi (Mykkänen, Liukas 2014, 18).

Algoritmisen ajattelun taitoja ei tarvitse välttämättä harjoitella tietokoneella. Niitä voidaan harjoittaa myös erilaisten leikkien tai pelien avulla. (Mykkänen & Liukas 2014, 77.) Algoritminen ajattelu voidaan purkaa neljään erilaiseen taitoon, joita kaikkia voidaan harjoitella erillään varsinaisesta ohjelmoinnista. Yksi algoritmisen ajattelun taidoista on taito purkaa ongelma osiin (*decomposition*) (Mykkänen & Liukas 2014, 77). Tähän liittyy olennaisesti toinen algoritmisen ajattelun taito: taito luoda algoritmeja, eli tuottaa ongelmalle vaiheittainen ratkaisu, jota voidaan harjoitella esimerkiksi tekemällä kulkuohjeet paikasta toiseen. Esimerkiksi matka kotoa koululle koostuu erilaisista vaiheista, jotka alkavat kotioven sulkemisesta ja päättyvät koulun pihalle astumiseen. Algoritmin muodostamisessa on tärkeää, että jokainen vaihe on ohjeistettu tarkasti. Samoin on myös kulkuohjeissa, joissa askeleiden tarkat määrät ja kaikki käännökset suuntineen on oltava oikein, jotta matka onnistuu.

Kolmas algoritmiseen ajatteluun kuuluva taito on kaavojen tunnistaminen (*pattern recognition*). Toisin sanoen, tämä tarkoittaa toistuvien sääntöjen tunnistamista. Neljäs algoritmisen ajattelun taito liittyy kykyyn yleistää ratkaisuita ja automatisoida ne. (Mykkänen & Liukas 2014, 77–78). Tällaisia taitoja harjoitellaan paljon jo nyt matematiikan tunneilla sekä monissa reaaliaineissa. Esimerkiksi yläkoulussa tutustutaan erilaisiin ratkaisukaavoihin matematiikan ja fysiikan tunneilla. Niiden avulla voidaan ratkaista ongelmanasettelultaan samankaltaiset tehtävät.

Algoritmisen ajattelun taidot eivät siis kuulu vain osaksi ohjelmoinnin taitoa, vaan ne ovat osa laajempia ajattelun ja ongelmanratkaisun taitoja. Näitä taitoja kehitetään perusopetuksessa jo osana muiden sisältöjen opiskelua. Tätä kautta avautuu myös mahdollisuus sisällyttää algoritmisen ajattelun taitojen harjoittelua ja ohjelmointia laajasti osaksi monia oppiaineita ja laaja-alaisia oppimiskokonaisuuksia.

3.4 Luovan ongelmanratkaisun harjoittelua erilaisilla välineillä

Nykyaikainen ohjelmointi on luovaa ongelmanratkaisua. Ohjelmointia voisi kuvata myös eräänlaiseksi taiteeksi. Hyvä ohjelmoija on taitava ongelmanratkaisussa vaikka moni aloitteleva ohjelmoija kokee sen ensin haastavaksi. Yksi tätä selittävä tekijä on se, että ohjelmoinnin harjoittelu aloitetaan usein ohjelmointikielen ja syntaksien harjoittelemisella. Tällöin harjoittelun pääpaino on muistamiseen ja toistoon perustuvassa harjoittelussa. Ohjelmoinnin ydin on kuitenkin aivan muualla. Olennaista on osata pohtia loogisia syitä ja seurauksia sekä asioiden riippuvuuksia. (Mykkänen, Liukas 2014, 18, 32–33.) Ongelmanratkaisu vaatii luovuutta ja kykyä soveltaa muistiin painettua tietoa. Opitun soveltaminen ja hyödyntäminen uudenlaisessa tilanteessa voi alkuun olla haastavaa. (Spraul 2012.)

Ohjelmointi on yksinkertaisesti sanottuna ohjeiden antamista tietokoneelle. Ohjelmoinnissa yhdistyy ohjelmointikielen käyttäminen, ohjelman rakentamisen suunnittelu sekä ohjelmointiongelman luova ratkaiseminen. (Mykkänen & Liukas 2014, 32–33.) Alkeisohjelmoinnin tehtävät eivät edellytä varsinaista ohjelmointiosaamista. Ne edellyttävät kuitenkin ongelmanratkaisutaitoja, loogista päättelykykyä sekä kykyä lukea ja antaa toimintaohjeita. (Tanhua-Piiroinen ym. 2019, 29.) Ohjelmoinnin opettaminen onkin pohjimmiltaan näiden taitojen opettelua. Vaikka tosielämän ohjelmointiin kuuluukin osana ohjelmointikielien osaaminen, lopulta kuitenkin korostuu edellä mainittujen taitojen hallinta. (Mykkänen & Liukas 2014, 78.)

Ohjelmoinnin harjoitteluun tietokoneella on olemassa erilaisia graafisia ohjelmointiympäristöjä, joiden avulla algoritmisen ajattelun ja ohjelmoinnin taitoja harjoitellaan pelillisesti. Joissain graafisissa ohjelmointiympäristöissä ohjelmoidaan esimerkiksi tietokoneen ruudulla liikkuvaa hahmoa ja autetaan sitä selviytymään sokkeloista tai tehtävistä. Näiden lisäksi on olemassa fyysisiä robotteja, joiden toimintaa hallitaan graafisen käyttöliittymän kautta. Robotti voi auttaa konkretisoimaan ohjelmointia, ja se sopii erityisesti nuoremmille lapsille ja ohjelmoinnin aloittelijoille. Yksinkertaisilla ohjelmointiroboteilla voivat leikkiä myös alle kouluikäiset lapset. Kuitenkin ohjelmoidakseen robottia, on lapsen ymmärrettävä tilan, etäisyyksien ja suuntien käsitteet sekä hahmotettava lukumääriä. Ennen kuin lapselle esitellään ohjelmoitava robotti, on ohjelmointia hyvä harjoitella ensin fyysisten harjoitusten avulla. (Morris, David, Uppal, Gurmit & Wells, David 2017, 29.) Ohjelmoinnin harjoittelu voidaan aloittaa esimerkiksi erilaisilla leikeillä ja peleillä.

Vanhempien ja jo ohjelmointiin tutustuneiden oppilaiden kanssa voidaan siirtyä myös ohjelmoimaan varsinaisia ohjelmointikieliä hyödyntäen. Ohjelmointikielet ovat ohjelmoijien työkaluja pulmien ratkaisemisessa. Uudenlaisen ongelman ratkaisemiseen saatetaan jopa kehittää uusi kieli. (Mykkänen, Liukas 2014, 35.) Eri kielet soveltuvat erilaisiin tehtäviin. Ohjelmointikieliä on satoja, mutta niiden perusajatus on se, että niiden avulla ohjelma saa tietoja ihmiseltä tai ihmisen valitsemasta lähteestä. Tämän jälkeen ohjelma käsittelee tiedot ja saa aikaan halutun lopputuloksen. (Mykkänen, Liukas 2014, 17.)

Kuten muissakin kielissä, ohjelmointikielissäkin on omat sanastonsa ja kielioppisääntönsä, joita voidaan kutsua myös syntakseiksi. Syntaksit ovat tiettyjä toimintoja tarkoittavia ilmaisuja ja jokaisella ohjelmointikielellä ne ovat omanlaisiaan. Ohjelmoijan kirjoittama koodi muodostaa ylhäältä alas luettavan listan komennoista, jotka ohjelma suorittaa rivi kerrallaan. (Mykkänen, Liukas 2014, 18.) Ohjelmoinnin opiskelun kannalta, on hyödyllistä käyttää ja opiskella sellaista kieltä, jonka avulla voi saada nopeasti näkyviä tuloksia (Mykkänen, Liukas 2014, 35). Lapset aloittavat ohjelmoinnin harjoittelun usein tästä syystä esimerkiksi ohjelmoimalla graafisissa ohjelmointiympäristöissä tai ohjelmoitavilla roboteilla.

Algoritmisen ajattelun ja ohjelmointitaitojen lisäksi on tärkeää pohtia oppilaiden kanssa sitä, minkälaista työtä ohjelmoijat tekevät ja minkälaisiin arjen tilanteisiin heidän tekemänsä työ vaikuttaa. Ohjelmoijan tehtävänä on esimerkiksi kirjoittaa ohjelmia, joiden tehtävänä on vastaanottaa tietoa, käsitellä sitä ja lopulta tulostaa ulos käsitelty tieto. Ohjelmoijien tekemiä ohjelmia käytämme päivittäin esimerkiksi tietokoneella ja mobiililaitteilla. (Mykkänen & Liukas 2014, 18.) Näin lisäämme oppilaiden ymmärrystä ympäröivästä maailmasta ja ohjelmoinnista osana sitä.

4 Tutkielman tavoite ja tutkimuskysymys

Tämän tutkimuksen tavoitteena on selvittää millaiset tekijät edesauttavat ohjelmoinnin oppimista peruskoulussa, ja tutkia ohjelmoinnin opetusta sekä oppimisen että opettamisen näkökulmasta. Olen kiinnostunut siitä, millaiset menetelmät tai työkalut ovat toimivia ohjelmoinnin opetuksessa. Koska ohjelmoinnin opetus on tullut osaksi perusopetuksen sisältöjä vasta hiljattain, on mielekästä tutkia, millaisia taitoja ohjelmoinnin opettaja tarvitsee tai millaista ohjelmoinnin opetuksen tulisi olla peruskoulun alaluokilla. Lisäksi olisi kiinnostavaa selvittää, millä tavoin ohjelmoinnin opetuksen tulee kehittyä peruskoulun alakoulusta yläkouluun siirryttäessä?

Tutustun opinnäytetyöni kautta aiheesta tehtyyn tutkimukseen systemaattisen kirjallisuuskatsauksen keinoin. Systemaattisessa kirjallisuuskatsauksessa kerätään joukko vertaisarvioituja ja tutkimukseen soveltuvien muiden kriteerien mukaan rajattuja artikkeleita. Tässä tutkimuksessa tarkastellaan kansainvälisiä tutkimusartikkeleita, jotka on julkaistu aikavälillä 2017–2020. Artikkelit toimivat tutkimuksen aineistona ja niiden pohjalta etsitään vastauksia tutkimuskysymykseen. Avaan tarkemmin tutkimusmenetelmää luvussa 5.1.

Systemaattinen kirjallisuuskatsaus rakennetaan perinteisesti yhden tutkimuskysymyksen varaan. Tutkimuskysymys ohjaa aineiston käsittelyä sekä rajaamista. Olen muotoillut tutkimuskysymyksen niin, että sen avulla on mahdollista selvittää monipuolisesti ohjelmoinnin opetukseen ja oppimiseen liittyviä tekijöitä.

Tutkimuskysymys:

> Millaiset tekijät edesauttavat ohjelmoinnin oppimista peruskoulussa?

5 Aineisto ja menetelmät

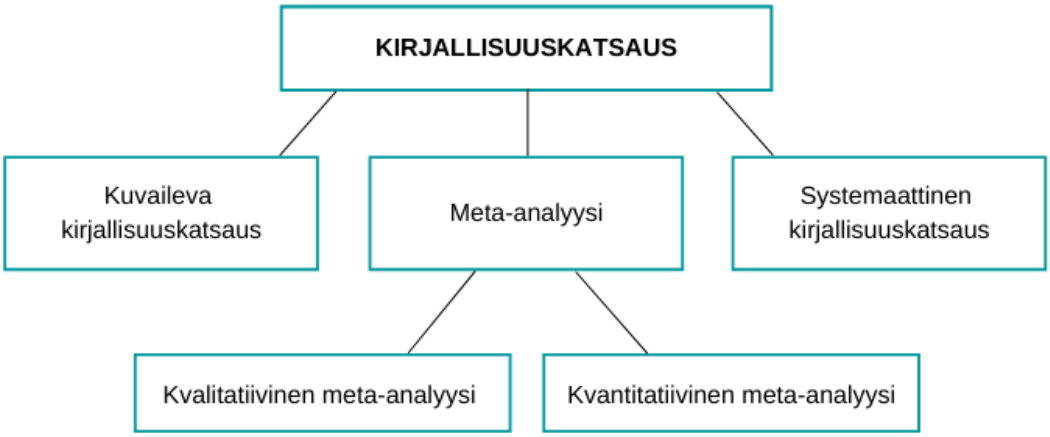
Tässä systemaattisessa kirjallisuuskatsauksessa noudatetaan laadullisen tutkimuksen perinnettä. Laadullisessa tutkimuksessa nojataan laadulliseen analyysiin, jonka kautta tutkimusaihetta ja siihen liittyvää, tutkimuksen kannalta tarkoituksenmukaista aineistoa tarkastellaan aluksi kokonaisuutena. Kokonaisuudesta pyritään tämän jälkeen nostamaan esiin tutkimuksen kannalta olennaiset löydökset. Tutkimusaineistoa pelkistetään näitä yksittäisiä, esiin nostettuja havaintoja yhdistämällä. Lopulta pelkistetystä aineistosta voidaan muodostaa aineistoa kuvaavia luokituksia, joiden pohjalta laadullisen tutkimuksen tulokset muodostuvat. (Alasuutari 2011.)

5.1 Systemaattinen kirjallisuuskatsaus tutkimusmetodina

Kirjallisuuskatsaus on metodi ja tutkimustekniikka, jolla tutkitaan olemassa olevaa tutkimusta ja sen avulla kootaan aiemman tutkimuksen tuloksia suuremmiksi kokonaisuuksiksi. Kirjallisuuskatsauksen avulla tavoitellaan usein olemassa olevan teorian kehittämistä sekä uuden teorian rakentamista. Sen avulla voidaan myös arvioida teoriaa ja rakentaa kokonaiskuvaa tutkittavista asiakokonaisuuksista. Kirjallisuuskatsaus voi toimia myös keinona tutkia teorioiden historiallista kehitystä. (Salminen 2011, 3–4.)

Hyvässä kirjallisuuskatsauksessa aineisto on hankittu kattavalla kirjallisuushaulla, ja aineiston keruu hakukriteereineen tulee olla selostettuna mahdollisimman tarkasti. Erityisesti artikkeleiden etsintä ja valikointikriteerit tulee esittää selkeästi. (Mäkelä, 1999; Isojärvi 2015.) Kirjallisuuskatsauksen periaatteisiin kuuluu huolellinen suunnittelu, jolla pystytään edesauttamaan luotettavuutta. Luotettavuutta edistetään myös etsimällä kaikki tutkimuksen kannalta relevantit tiedonlähteet sekä raportoimalla tarkasti tutkimuksen kulkua. Näin myös mahdollistetaan tutkimuksen toistettavuus. (Isojärvi 2015.)

Kirjallisuuskatsaukset voidaan jaotella kolmeen tyyppiin: kuvaileva kirjallisuuskatsaus, systemaattinen kirjallisuuskatsaus ja meta-analyysi (ks. kuva 1). Edellä mainituista viimeinen, eli meta-analyysi, on mahdollista toteuttaa sekä kvalitatiivisena että kvantitatiivisena. (Salminen 2011, 6.) Tämä tutkimus on toteutettu systemaattisen kirjallisuuskatsauksen metodein, mutta esittelen lyhyesti myös kaksi muuta katsaustyyppiä.



Kuva 1. Kirjallisuuskatsauksen tyypit (mukaillen Salminen 2016)

Kuvaileva kirjallisuuskatsaus on yleisimmin käytetty perustyyppi, jonka toteutus on suhteellisen vapaamuotoinen yleiskatsaus aineistoon ilman tiukkoja aineistonvalintaa tai metodologiaa koskevia sääntöjä. Kuvailevan kirjallisuuskatsauksen etuna on, että sen avulla tutkittavaa ilmiötä pystytään kuvaamaan laaja-alaisesti. (Salminen 2011, 6.)

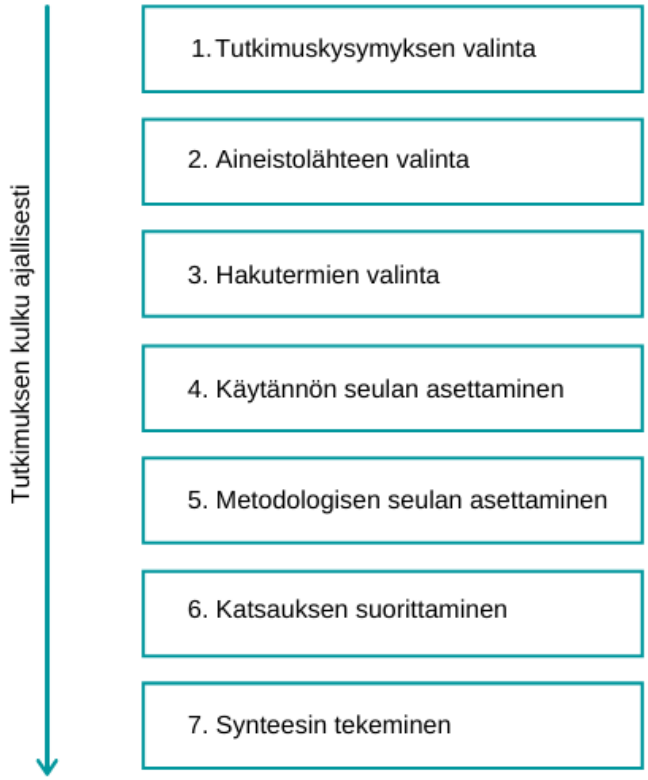
Meta-analyysi voidaan toteuttaa joko kvantitatiivisesti tai kvalitatiivisesti. Kvantitatiivisessa meta-analyysissa kvantitatiivisia tutkimuksia yhdistetään ja yleistetään tilastotieteen menetelmin. Kvalitatiivinen meta-analyysi taas pyrkii selittämään tutkittavaa ilmiötä (metasynteesi) tai laadullisen tutkimuksen tiivistämiseen kvantitatiivisilla menetelmillä (metayhteenveto). (Salminen 2011, 13.)

Tässä tutkimuksessa käytettävä systemaattinen kirjallisuuskatsaus eroaa muista katsaustyypeistä aineiston valinnan sekä analysoinnin perusteella. Kirjallisuuskatsauksesta tulee systemaattinen, kun katsauksen aineiston valinnassa kiinnitetään huomiota käytettyjen lähteiden keskinäiseen yhteyteen ja tekniikkaan, jolla tulokset on saatu. (Salminen 2011, 4.) Systemaattisen kirjallisuuskatsauksen aineistona ovat pääasiassa ensimmäisen asteen tutkimukset. Siinä kootaan systemaattisesti yhteen tieteellistä tutkimusaineistoa jonka pohjalta tietoa tiivistetään tieteellisen tutkimuksen keinoin. (Bettany-Saltikov 2012, 5, 7.) Systemaattinen kirjallisuuskatsaus siis tiivistää olennaisen sisällön tietyn aihepiirin tutkimuskirjallisuudesta.

Systemaattisessa kirjallisuuskatsauksessa tutkija pyrkii asettamaan tutkimuksensa oman tieteenalansa kontekstiin ja toteuttaa tämän käymällä läpi laajan määrän tiiviissä muodossa olevaa tutkimusmateriaalia (Salminen 2011, 9). Katsauksen systemaattisuudella

pyritään luomaan tutkimukselle tieteellistä luotettavuutta ja tästä syystä katsaus tulee toteuttaa kriteerien mukaisesti. Tällä tavoin varmistetaan myös se, että tutkimukseen valikoitunut aineisto on yhtenäinen. (Salminen 2011, 10–11.)

Tämän tutkimuksen toteutus mukailee Finkin (2019) mallin mukaisia vaiheita, jotka olen esittänyt kuvassa 2. Se toimii pohjana systemaattisen kirjallisuuskatsauksen tekemiselle, ja kuvaa tutkimuksen kulun ajallista kaarta. Mallin mukaan kirjallisuuskatsauksen tekeminen aloitetaan tutkimuskysymyksien valinnalla ja asettamisella (Fink 2019, 7). Systemaattinen kirjallisuuskatsaus rakentuu perinteisesti yhden tutkimuskysymyksen varaan (Bettany-Saltikov 2012, 5). Tutkimuskysymys muodostetaan tarkaksi ja yksiselitteiseksi tutkimusta ohjaavaksi kysymykseksi (Fink 2019, 7). Tämän tutkimuksen tutkimuskysymys sekä tutkimuksen tavoite on esitelty luvussa 4.



Kuva 2. Finkin malli systemaattisen kirjallisuuskatsauksen tekemiseen (mukaillen Fink 2019)

Kun tutkimuskysymys on asetettu, valitaan aiheeseen sopiva aineistolähde, joka sisältää tarkoituksenmukaista aineistoa (Fink 2019, 7). Systemaattisen kirjallisuuskatsauksen tulisi

aina pohjautua vertaisarvioituun tietoon (Bettany-Saltikov 2012, 6). Aineisto voi koostua esimerkiksi bibliografioista, tietokannoista tai verkkosivuilta kerätyistä artikkeleista (Fink 2019, 7).

Aineistolähteen sisältöä rajataan hakutermien sekä käytännön seulan ja metodologisen seulan avulla. Sopivien hakutermien avulla aineistolähteestä voidaan etsiä tutkimuksen kannalta olennainen sisältö eli artikkelit. Tämän jälkeen muodostetaan sisään- ja poissulkukriteerit, joiden mukaan aineistolähteen artikkeleita rajataan. Metodologisen seulan avulla suodatetaan pois metodologialtaan tai luotettavuudeltaan sopimattomat artikkelit. (Fink 2019, 7.)

Kun aineisto on rajattu, aloitetaan katsauksen suorittaminen. Aineistosta nostetaan esille säännönmukaisesti haettua informaatiota. Informaation poimimisessa ja esille nostamisessa on tärkeä noudattaa yhtenäistä ja johdonmukaista tapaa. Lopuksi katsauksen aikana nousseesta informaatiosta muodostetaan synteesi, jonka pohjalta tutkimuksen tekijä tekee johtopäätökset. (Fink 2019, 7.)

5.2 Aineisto ja valintakriteerit

Tämän tutkimuksen aineisto on kerätty EBSCOhost Academic Search Premier -hakua käyttäen heinäkuussa 2020. Kyseinen tietokanta sisältää laajasti ja monipuolisesti korkealaatuisia, eri tieteenaloille sijoittuvia tutkimusartikkeleita. Laajoissa kirjallisuuskatsauksissa käytetään usein monista tietokannoista valikoituja artikkeleita. Koska tämä tutkimus on opinnäytetyö eli hieman suppeampi katsaus, päädyin valikoimaan vain yhden, mutta monipuolisen tietokannan tutkimusaineiston keräämiseen. Tutkimuksen aineisto on kerätty rajaamalla artikkelitietokannasta ne artikkelit, jotka vaikuttivat vastaavan tutkimuskysymykseen.

Systemaattista kirjallisuuskatsausta tehdessä tulee kuvata tarkasti käytetyt menetelmät, jotta lukijan on myös mahdollista arvioida tuloksia (Oxman 1999). Avaan seuraavaksi aineistohaun kulkua sekä esitän ja perustelen käyttämäni sisäänotto- ja poissulkukriteerit. Aineistohaun sisäänottokriteerit on esitetty taulukossa 1 ja aineistoon lopulta valikoituneet artikkelit taulukossa 2.

EBSCOhost -artikkelitietokannassa artikkelihakua on mahdollista tehdä useammalla hakusanalla yhtäaikaisesti. Hakusanat on mahdollista liittää hakuun OR, NOT tai AND hakukomennolla. NOT -komennolla on mahdollista rajata pois hakusanan sisältävät tulokset. OR

-komennon avulla hakukone antaa tulokset joissa esiintyy vähintään toinen OR -komennolla määritellyistä hakusanoista. AND -komento rajaa tulokset niin, että molemmat hakusanat löytyvät listatuista artikkeleista. Lisäksi hakua on mahdollista rajata erilaisten ominaisuuksien kuten julkaisuvuoden ja kirjoituskielen mukaan tai rajata pois ne artikkelit, joita ei ole vertaisarvioitu.

Aloitin aineistohaun Finkin (2019) mallin mukaan hakutermien määrittelyllä. Aineistohaussa käytetyt hakutermit olivat *education*, *programming* ja *computation*. Määrittelin hakutermit tutkimuskysymykseen pohjaten. Hakutermi *education* (suom. koulutus) pohjasi siihen, että tässä tutkimuksessa käsitellään peruskouluikäisten oppimisympäristössä ohjelmoinnin oppimiseen vaikuttavia tekijöitä. Hakutermit *programming* (suom. ohjelmointi) ja *computing* (suom. tietojenkäsittely) taas ovat aiheeseen suoraan liittyviä asiasanoja.

Hakutermit oli liitetty AND sekä SU Subject terms -komentoja käyttäen mikä tarkoitti, että jokaisen hakutermin tuli esiintyä artikkelin asiasanoissa. Hakutuloksia tuli ensimmäisellä haulla 154 kappaletta. Tämän jälkeen toteutin käytännön seulonnan. Aluksi rajasin pois ne artikkelit, joita ei ollut mahdollista lukea kokonaan sähköisesti. Tämä rajaus kavensi tulosten määrän 56 kappaleeseen. Seuraavaksi rajasin pois vertaisarvioimattomat artikkelit, joka kavensi hakutuloksen 49 kappaleeseen, sekä ne artikkelit, joiden julkaisuvuosi ei sijoittunut vuosien 2015 ja 2020 väliin. Hakutuloksia oli käytännön seulan asettamisen jälkeen 42 kappaletta. Seulonnan kulku sisäänottokriteereineen on havainnollistettu taulukossa 1.

Taulukko 1. Sisäänottokriteerit ja hakutuloksina tulleiden artikkeleiden lukumäärä rajausten mukaan

Sisäänottokriteerit	**Hakutuloksia**
asiasanoissa esiintyy hakutermit education, programming ja computation	N = 154
koko teksti luettavissa sähköisesti	N = 56
vertaisarvioidut tekstit	N = 49
julkaisuvuosi sijoittuu välille 2015–2020	N = 42

Jatkoin hakutuloksien rajaamista Finkin (2019) mallin mukaisesti metodologisella seulonnalla. Kävin hakutuloksina tulleet 42 artikkelia manuaalisesti läpi ja valikoin lupaavien otsikoiden sekä abstraktien perusteella luettavaksi 17 artikkelia. Tässä vaiheessa rajasin pois

ne artikkelit, joissa ei käsitelty peruskouluikäisten ohjelmoinnin opetusta tai oppimista. Lisäksi suljin ulkopuolelle sellaiset artikkelit, jotka olivat muita kuin tutkimuksia tai eivät olleet relevantteja tutkimuskysymyksen kannalta. Tutkimuksen aineistoon valikoitui lopulta 9 artikkelia, jotka on listattu taulukossa 2.

Taulukko 2. Aineistoon valikoituneet artikkelit kirjoittajineen sekä artikkelin julkaisuvuosi

Nro	Otsikko	Kirjoittajat	Vuosi
1	A Cross-Analysis of Block-Based and Visual Programming Apps with Computer Science Student-Teachers	João, Piedade; Nuno, Dorotea; Fábio, Sampaio Ferrentini; Ana, Pedro	2019
2	Robotics to Develop Computational Thinking in Early Childhood Education	Muñoz-Repiso, Ana García-Valcárcel; Caballero-González, Yen-Air.	2019
3	Integrating the Constructionist Learning Theory with Computational Thinking Classroom Activities	Csizmadia, Andrew; Standl, Bernhard; Waite, Jane	2019
4	Computational Concepts Reflected on Scratch Programs	Kwon, Kyungbin; Lee, Kyungbin; Chung, Jaehwa	2019
5	Abstraction in Action: K-5 Teachers' Uses of Levels of Abstraction, Particularly the Design Level, in Teaching Programming	Waite, Jane Lisa; Curzon, Paul; Marsh, William	2018
6	Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space	Kynigos, Chronis; Grizioti, Marianthi	2018
7	Teaching Computing in a Multidisciplinary Way in Social Studies Classes in School – A Case Study	von Wangenheim, Christiane Gresse; Alves, Nathalia Cruz; Rodrigues, Pedro Eurico	2017
8	Students' Perception towards Program Visualization on Smartphone – Case of SunLab Initial Investigation	Kumalija, Elhard James; Fatih, Ymran; Sun, Yi	2019
9	Classroom Talk and Computational Thinking	Jenkins, Craig W.	2017

Kaikki tutkimukseen valikoituneet artikkelit käsittelevät ohjelmointia joko opettamisen tai oppimisen näkökulmasta. Osa tutkimuksista tutkii ohjelmointia opettavien opettajien kompetenssia ja loput käsittelevät ohjelmoinnin opetuksen menetelmiä tai oppilaiden käsityksiä. Vaikka artikkelihaussa artikkelit oli suodatettu julkaisuvuoden mukaan välille 2015–

2020, kaikki tutkimukseen valikoituneet artikkelit sijoittuvat vain kolmen vuoden aikavälille: vuodesta 2017 vuoden 2019 loppuun asti.
Finkin (2019) mallin mukaan aineiston keruun jälkeen on vuorossa katsauksen suorittaminen ja lopuksi synteesin tekeminen. Seuraavassa luvussa 5.3 toteutan katsauksen suorittamisen eli analysoin katsaukseen valikoituneet artikkelit. Tämän jälkeen on vuorossa synteesin tekeminen, joka on luettavissa tutkimuksen tulosten muodossa luvussa 6.

5.3 Aineiston analyysi

Tämän kirjallisuuskatsauksen aineisto on järjestetty sisällönanalyysin keinoin. Sisällönanalyysi on perusanalyysimenetelmä, jota voidaan käyttää kaikissa laadullisissa tutkimuksissa. Sisällönanalyysi on tekstianalyysia ja sopii näin erilaisten kirjallisten dokumenttien analysoimiseen. Aineistolähtöinen sisällönanalyysi soveltuu hyvin teoreettisen tutkimuksen piiriin kuuluvan systemaattisen kirjallisuuskatsauksen toteuttamiseen. Perusanalyysimenetelmänä se ei pohjaudu mihinkään tiettyyn teoreettiseen tai epistemologiseen näkökulmaan. (Tuomi ja Sarajärvi 2018.)

Olennainen osa kirjallisuuskatsausta on aineiston ja tulosten kriittinen tarkastelu (Salminen 2011, 5). Tämän tutkimuksen aineisto koostuu tieteellisistä tutkimusjulkaisuista. Sisällönanalyysi on toteutettu tutustumalla aineistoon tutkimuskysymyksen pohjalta, nostamalla esiin tutkimustehtävän kannalta relevantit asiat. Esiin nostetut asiat on pelkistetty ja luokiteltu ala- ja yläluokkiin kokonaiskuvan jäsentämistä varten. (Alasutari 2011; Tuomi & Sarajärvi 2018.)

Käytännössä tutkimus on toteutettu merkitsemällä tutkimustehtävään liittyvät virkkeet tutkimusartikkeleista. Analyysiyksikkönä toimivat siis yksittäiset virkkeet tai lyhyet ja peräkkäiset, samaa asiaa käsittelevät, muutaman virkkeen kokonaisuudet. Aineistosta nousi yhteensä 85 havaintoa, jotka taulukoin artikkeli kerrallaan. Esitän artikkeleista nousseiden havaintojen määrän taulukossa 3. Analyysiyksiköt ja niiden luokittelu on nähtävillä myös liitteessä 1.

Taulukoinnin jälkeen tiivistin kunkin havainnon lyhyellä kuvauksella, joita käytin luokittelun tukena. Aineisto jakautui aluksi selkeästi kolmeen teemaan: 1. Ohjelmointityökaluun liittyvät tekijät (N=12), 2. Opettajaan, opetukseen ja kouluun liittyvät tekijät (N=33), 3. Oppilaan tekemiseen, ajatteluun ja taitoihin liittyvät tekijät (N=40). Teemojen sisällä havainnoista oli mahdollista muodostaa vielä alaluokkia, joten luokittelin teemat vielä alaluokkiin, joita tuli lopulta yhteensä 8. Olen esittänyt nämä luokat alaluokkineen taulukossa 4.

Taulukko 3. Analyysiyksiköt tutkimusartikkeleittain

Artikkeli nro	**Analyysiyksiköt kpl**
1	9
2	3
3	12
4	16
5	16
6	3
7	9
8	4
9	13
Yhteensä	N=85

Taulukko 4. Aineistosta nousseiden havaintojen määrät sekä artikkelit, joista analyysiyksiköt on nostettu (ks. liite 1)

Luokka	**N**	**Artikkelit**
Ohjelmointityökaluun liittyvät asiat	12	1, 2, 3, 4, 5, 7, 8
Opettajaan liittyvät asiat	4	1,5
Pedagogiset asiat	14	1,3,4,5,7
Opettajan kommunikointiin liittyvät asiat	10	4, 5, 9
Oppilaan kommunikointiin liittyvät asiat	12	3, 5, 7, 9
Oppilaan tekemiseen tai ajatteluun liittyvät asiat	5	3, 5, 7
Ajattelutaitoihin liittyvät asiat	14	3, 4, 5, 6
Ohjelmointitaitoihin liittyvät asiat	9	4, 5, 7, 8
Muuhun opetukseen tai kouluun liittyvät tekijät	5	1, 2, 8

6 Tulokset

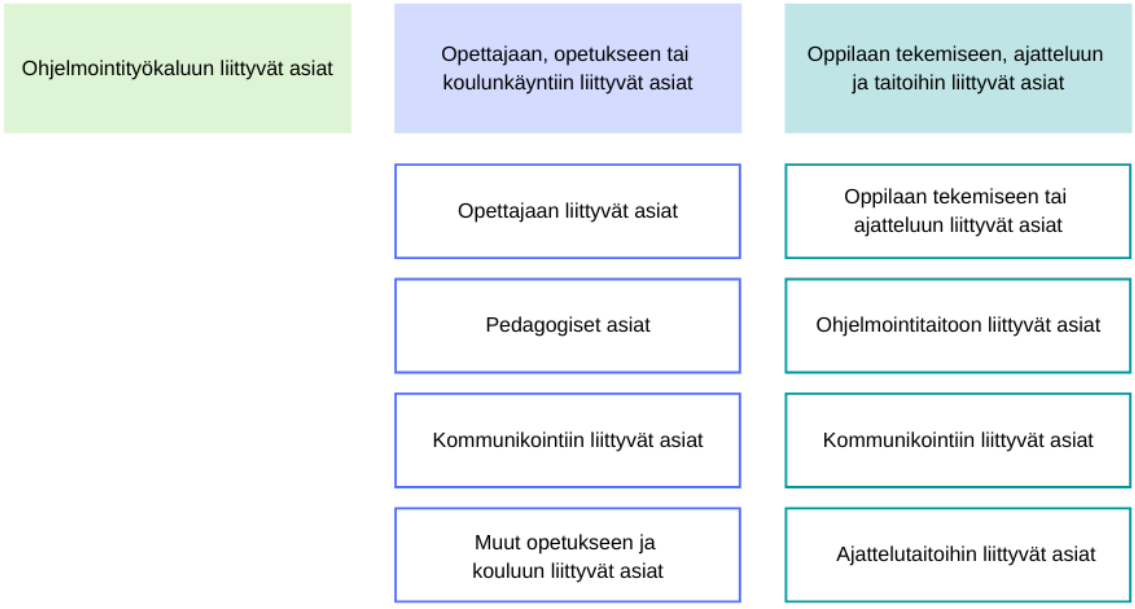
6.1 Aineistosta nousseet ylä- ja alaluokat

Havaintojen taulukointi ja luokittelu ovat aineiston käsittelyä, eivätkä itsessään ole vielä aineiston analysointia (Tuomi & Sarajärvi 2018). Luokittelun jälkeen suoritetaan itse analyysi. Analyysi koostuu tutkijan sisällönanalyysin lopputuloksena saaduista jaotteluista ja luokista tekemistä johtopäätöksistä. (Hakala 2017; Tuomi & Sarajärvi 2018.) Olen koonnut tähän lukuun aineistosta nousseet havainnot ja esittelen tulosten pohjalta tehdyt johtopäätökset luvussa 7.1.

Aineistosta nousi esiin kolme yläluokkaa, joista kahden alle muodostui kahdeksan alaluokkaa – neljä kumpaankin. Yläluokat alaluokkineen on kuvattu kuvassa 3.

Yläluokkia ovat:

1. Ohjelmointityökaluun liittyvät asiat
2. Opettajaan, opetukseen tai koulunkäyntiin liittyvät asiat
3. Oppilaan tekemiseen, ajatteluun ja taitoihin liittyvä asiat



Kuva 3. Havaintojen jakautuminen kolmeen yläluokkaan ja kahdeksaan alaluokkaan

Niin kuin ylä- ja alaluokkien määrästä voi päätellä (ks. kuva 3), määrällisesti suurin osa aineiston havainnoista lukeutui kahden jälkimmäisen yläluokan alle. Ohjelmointityökaluun

liittyviä havaintoja oli selkeästi vähiten ja tästä syystä siihen liittyviä alaluokkia ei ole. Tarkemmat havaintojen määrät löytyvät edellisen kappaleen taulukoista 3 ja 4.

6.2 Ohjelmointityökaluun liittyvät asiat

Ohjelmointityökaluun liittyviä havaintoja nousi aineistosta yhteensä 12 kappaletta. Kaikki aineistosta nousseet ohjelmointityökaluun liittyvät havainnot eivät koskeneet tietokoneella tai mobiililaitteella käytettäviä työkaluja tai välineitä. Sisällytin tähän luokkaan myös erilaiset fyysiset leikit tai pelit, joiden avulla ohjelmointia ja algoritmista ajattelua harjoiteltiin. Aineistosta nousi esiin, että ohjelmoinnin oppimista edesauttoi fyysisten aktiviteettien, kuten kartan avulla reitin kävelyohjeiden laatiminen (Waite, Curzon, Marsh, Sentence & Hadwen-Bennett 2018).

Ohjelmoinnin oppimista edesauttaa oikein valittu graafinen ohjelmointiympäristö, jonka ulkoasu on yksinkertainen ja vetoava. Graafisen ohjelmointiympäristön symbolinen kieli mahdollistaa myös sellaisten oppilaiden ohjelmoinnin harjoitelun, jotka eivät osaa vielä lukea. (João, Nuno, Fábio, & Ana 2019.) Graafiset ohjelmointiympäristöt, joiden avulla voi ohjata robotteja, edistävät algoritmisen ajattelun kehittymistä (Muñoz-Repiso & Caballero-González 2019).

Ohjelmoinnin oppimiselle ovat hyödyllisiä myös sellaiset ohjelmointiympäristöt, joissa käyttäjä voi lisätä ohjelmaansa kokonaisen ohjelmablokin yksittäisten käskyjen sijasta (Waite ym. 2018). Oppilaiden motivaatiota lisää se, että ohjelmoimalla voi tuottaa pelejä tai muita tuotoksia, joita he pystyvät lopuksi itse pelaamaan (Von Wangenheim, Alves, Rodrigues, & Hauck 2017). Hyvässä ohjelmointiympäristössä voidaan huomioida erilaiset oppimistyylit ja ajattelutavat (Csizmadia, Standl & Waite 2019). Lisäksi hyödyksi on se, että ympäristö arvioi oppilaan kirjoittamaa koodia ja ohjelmaa automaattisesti (Kwon, Lee & Chung 2018).

Vaikka graafinen ohjelmointiympäristö koettiin monissa tutkimusaineiston artikkeleissa toimivaksi, ei se sulkenut pois ohjelmointikielien onnistunutta käyttämistä. Aineiston mukaan ohjelmointikielillä tehtävä ohjelmointi sujui hyvin, mikäli siirtymä graafisesta ohjelmointiympäristöstä tehtiin vaiheittain ja ohjelmointikielen käyttäminen alustettiin perusteellisesti. (Kwon ym. 2018.)

6.3 Opettajaan, opetukseen tai koulunkäyntiin liittyvät asiat

Ohjelmoinnin oppimista edesauttaa, jos ohjelmoinnin harjoittelu pystytään aloittamaan mahdollisimman varhain (João ym. 2019; Muñoz-Repiso & Caballero-González 2019) ja sen harjoittelua yhdistettiin luonnontieteiden opetukseen (Muñoz-Repiso & Caballero-González 2019). Lisäksi algoritmisen ajattelun oppimista edistää kokonaisvaltainen luovan ajattelun stimuloiminen sekä vapaa-ajan ohjelmointimahdollisuudet (Kumalija, Faith & Sun 2019).

Yksi tärkeä ohjelmoinnin oppimiseen vaikuttava tekijä koulussa on opettaja ja hänen valmiutensa opettaa ohjelmointia. Oppimista edesauttaa, jos opettajalla on tietämystä ohjelmoinnista ja hän kykenee opetuksessaan käyttämään oikeita ohjelmointiin liittyviä termejä (Waite ym. 2018). Ohjelmoinnillisen tietämyksen lisäksi opettajan tulee olla perehtynyt opetuskäytössä olevaan ohjelmointiympäristöön tai -välineeseen. Pelkkä perehtyneisyys ei kuitenkaan täysin riitä, vaan opettajan täytyy voida opettaa sellaisilla pedagogisilla strategioilla, jotka ovat hänen mukavuusalueellaan. (João ym. 2019.)

Opettajan opetukseen liittyy olennaisesti hänen kykynsä ja tapansa kommunikoida ja antaa palautetta oppilaille. Ohjelmoinnin oppimiselle on hyödyksi dialoginen ote opetukseen. Tällä tarkoitetaan sitä, että opettaja ei vedä opetusta ennalta valmistellun kaavan mukaisesti, vaan antaa tietoa oppilaille silloin, kun aiheen nousevat keskustelussa esiin. Vielä pitemmälle menevä ajatus on se, että opetustilanteissa oppilas saa johtaa keskustelua ja viedä sitä tarvitsemaansa suuntaan. Tällaista tapaa voidaan kutsua oppilasjohtoiseksi dialogiksi. Tähän dialogiin liittyy se, että opettaja myös kommentoi ja antaa palautetta oppilaiden kysymyksistä ja ajatuksista. Täysin oppilasjohtoista ei opetuksen tarvitse olla, vaan opettajan rooli on kysymyksiin vastaajan lisäksi olla myös kysymysten esittäjä. Opettaja voi ohjata oppilaiden oppimista oikeaan suuntaan kysymysten kautta. Kysymykset tulee asetella niin, että oppilas voi niiden avulla löytää ja korjata tekemänsä virheet sekä keksiä seuraavat askeleet ohjelmansa rakentamiseen. Kysymyksien avulla opettaja ohjaa myös oppilaiden ajattelua ja ohjaa heidät pohtimaan, miksi ja miten opiskeltavia asioita tehdään. (Jenkins 2017.)

Kasvokkain tapahtuvan palautteen lisäksi opettajan on hyvä antaa palautetta myös oppilaiden ohjelmista ja käydä manuaalisesti läpi oppilaiden tuotoksia (Kwon ym. 2018). Hyödyllistä on, jos opettaja voi jättää oppilaille ohjelmointipohjaan tai -tiedostoon vihjeitä ja kommentteja, jotka auttavat oppilaita keksimään ja selvittämään, miten heidän tulisi lähestyä ohjelmointipulmaa. Myös ohjelmien kommentointi jälkikäteen auttaa oppimisessa, ja

näin oppilaat saavat palautetta siitä, mikä oli hyvää ja mitä voisi tai kannattaisi tehdä toisin. (Waite ym. 2018.)

Kommunikoinnin lisäksi opettaja voi tehdä ohjelmoinnin oppimista edesauttavia pedagogisia ratkaisuja, kuten pyrkiä kehittämään oppilaiden sanavarastoa ohjelmoinnin termeillä ja sanastoilla (Waite ym. 2018). Ohjelmoinnin opiskelu kannattaa suunnitella niin, että jokaisella oppilaalla on koko ajan tarpeeksi haastetta, mutta kuitenkin niin, että tehtävät ovat helposti suoritettavissa. Mikäli on mahdollista, kannattaa hyödyntää pelillistämistä ohjelmoinnin ja ongelmanratkaisun harjoitteluun. (João ym. 2019.)

Ohjelmoinnin opettamiselle hyödyllistä on tehdä sitä algoritmisen ajattelun taitojen pohjalta ja oppilaita tulee ohjata pohtimaan erilaisia metodeita löytää ratkaisuita. Tähän kuuluu myös se, että oppilaat oppivat arvioimaan erilaisten ongelmanratkaisumetodien hyödyllisyyttä ongelman ratkaisemisen näkökulmasta. Yhtä tärkeää on myös tukea oppilaita silloin, kun he kohtaavat pulmia tai esteitä ohjelmoinnin harjoittelussa. (Kwon ym. 2018.)

Ohjelmoinnin harjoittelu kannattaa integroida osaksi eri oppiaineita ja tarkastella arkielämästä tuttuja tilanteita algoritmisen ajattelun kautta. Aluksi jokin esitetty ongelma voidaan ratkaista esimerkiksi matematiikan keinoilla ja tämän jälkeen tehdä ratkaisu myös ohjelmoinnin avulla. Oppimisen kannalta on hyödyllistä, jos oppilaat kykenevät yhdistämään oppimaansa johonkin tuttuun ajatukseen tai ilmiöön. Tämän jälkeen mallinnettua ratkaisua voidaan koettaa soveltaa uudenlaiseen ympäristöön ja erilaiseen tilanteeseen. (Csizmadia ym. 2019.)

Muistamista ja oppimista helpottaa myös se, että otetaan opetuksessa käyttöön useita oppimiskanavia. Kuuntelemisen ja katsomisen lisäksi voidaan hyödyntää fyysisyyttä muuttamalla fyysistä toimintaa algoritmeiksi eli sanoittamalla niitä toimintaohjeiksi. Tällaista voidaan tehdä esimerkiksi erilaisten leikkien ja pelien kautta. Tämän kaltaisten aktiviteettien jälkeen on huolellisesti pyrittävä linkittämään fyysisesti tuotettuja algoritmeja ohjelmoinnin kontekstiin eli pohdittava ja selitettävä, miltä ne näyttäisivät varsinaisessa ohjelmassa. (Waite ym. 2018.)

6.4 Oppilaan tekemiseen, ajatteluun ja taitoihin liittyvät asiat

Jokainen oppilas on erilainen oppija, joka kehittyy omassa tahdissaan ja itselleen sopivalla tavalla. Jokaiselle löytyy varmasti hyviä tapoja ohjelmoinnin harjoitteluun. On kuitenkin joitakin persoonallisuuden piirteitä tai toimintamalleja, jotka ovat eduksi ohjelmoinnin

oppimisessa. Ne oppilaat, jotka kokevat tekemänsä aktiviteetit mielekkäiksi ja merkityksellisiksi, kykenevät pitämään yllä opiskelumotivaatiota. (Csizmadia ym. 2019.) Tätä edesauttaa, jos oppilailla on mahdollisuus jossain määrin vaikuttaa siihen, mitä ja millä tavalla he harjoittelevat tai minkälaista ohjelmaa he tahtovat rakentaa (Von Wangenheim ym. 2017). Hyvät itsesäätelykyvyt edesauttavat oman toiminnan ja ohjelman suunnittelemisessa. Tähän liittyy olennaisesti kyky oman toiminnan arvioimiseen. Näitä taitoja voidaan myös harjoitella ohjelmoinnin opiskelun kautta. (Waite ym. 2018.) Sinnikkyys nousi aineistossa esiin sellaisena persoonallisuuspiirteenä, jolla on myönteinen vaikutus ohjelmoinnin oppimiseen (Von Wangenheim ym. 2017).

Kuten opettajien, myös oppilaiden kommunikaation ja yhdessä tekemisen taidot korostuivat aineistossa ohjelmoinnin oppimista edistävänä tekijänä. Monenlaiset päättelyn taidot edesauttavat ohjelmoinnin oppimista ja opiskelua. Sellaiset oppilaat, jotka osaavat sanallistaa ohjelmointiongelmia ja muodostaa hypoteeseja ja ennustuksia ratkaisusta ja sen mahdollisesta toimivuudesta, oppivat ohjelmointia paremmin muihin verrattuna. Myös ne oppilaat, jotka pystyvät sanallistamaan ajatuksiaan sekä sanallisesti selittämään omaa ongelmanratkaisuaan, pärjäsivät vertaisiaan paremmin. Yhteistyötaidoissa korostui vertaisoppiminen sekä vertaistuen antaminen (Jenkins 2017). Ohjelmoinnin oppimista edisti, jos oppilaat ohjattiin ohjelmoimaan pareittain ja keskustelemaan sanallisesti siitä, mitä he olivat tehneet ja ohjelmoineet (Waite ym. 2018; Von Wangenheim ym. 2017; Csizmadia ym. 2019). Myös omien tuotosten muille esittämisellä ja esittelyllä oli positiivinen vaikutus (Von Wangenheim ym. 2017; Csizmadia ym. 2019).

Ne oppilaat, joilla algoritmisen ajattelun taidot ovat mahdollisimman monipuolisesti kehittyneet, kykenevät opiskelemaan ohjelmointia muita tehokkaammin (Csizmadia ym. 2019; Kwon ym. 2018; Waite ym. 2018; Kynigos & Grizioti 2018). Varsinainen ohjelmointiosaaminen ja ohjelmien erityispiirteiden ymmärtäminen hyödyttää myös luonnollisesti oppimista. Ne oppilaat, jotka pystyvät ymmärtämään erilaisia ohjelmoinnin konsepteja, kuten esimerkiksi laskutoimitusten, muuttujien, listojen ja ehtolauseiden toimintaa pystyvät edistymään ohjelmoinnin oppimisessa. (Kwon ym. 2018; Kumalija ym 2019.) Ohjelmoimisen ohessa oppilaiden tulee myös hallita ohjelman suunnittelutaitoja. Yksityiskohtien ja ominaisuuksien rajaamisen harjoittelu on oleellista. Suunnitteluun kuuluu olennaisesti suunnitelman toteuttamisen toteutettavuuden arvioiminen. (Waite ym. 2018.)

7 Pohdinta

7.1 Tulosten tarkastelu

Kokosin tässä tutkimuksessa systemaattisen kirjallisuuskatsauksen keinoilla ohjelmoinnin opetusta käsitteleviä tutkimusartikkeleita. Artikkeleiden aineisto oli koottu laajasti esi- ja perusopetuksesta. Tavoitteenani oli selvittää millaiset asiat edesauttavat ohjelmoinnin oppimista. Tekemäni katsaus osoitti, että ohjelmoinnin oppimista voidaan edistää monin eri keinoin. Ohjelmoinnin oppimiseen vaikuttaa niin opettajaan kuin oppijaankin liittyviä asioita sekä opetuksessa käytettyyn ohjelmointiympäristöön liittyviä asioita.

Esitän seuraavia johtopäätöksiä systemaattiseen kirjallisuuskatsaukseen perustuen:

1) Ohjelmoinnin oppimiseen vaikuttaa opetuksessa käytettävän ohjelmointityökalun selkeä ja miellyttävä visuaalinen ilme ja käyttöliittymän helppokäyttöisyys, sekä mahdollisuus saada reaaliaikaista palautetta.
2) Ohjelmoinnin oppimiseen vaikuttaa ohjelmointia opettavan opettajan sisältöosaaminen ohjelmoinnista sekä opettajan käyttämät pedagogiset ja kommunikoinnin keinot.
3) Ohjelmoinnin oppimiseen vaikuttaa oppilaan kommunikointi- ja yhteistyötaidot sekä algoritmisen ajattelun taidot. Lisäksi jo opitut ohjelmoinnin taidot edesauttavat uuden oppimista.

Katsauksen mukaan ohjelmoinnin oppimiseen vaikuttaa opetukseen käytetty ohjelmointityökalu tai -väline. Aloittelevaa ohjelmoijaa hyödyttävät fyysiset harjoitukset, pelit ja leikit. Myös graafiset ohjelmointiympäristöt, joihin on yhdistetty pelillistämistä tai robotteja, edesauttavat ohjelmoinnin oppimista. Tällä tavoin rakennetun pohjan päälle voidaan rakentaa opetusta varsinaisilla ohjelmointikielillä. Tulos on hyvin samankaltainen kuin Mykkäsen ja Liukkaan (2014) ohjeistus ohjelmoinnin opetuksesta.

Lapsen kehitystä ajatellen, tällainen vaiheittainen eteneminen ei oikeastaan poikkea muidenkaan aiheiden oppimisesta. Kouluikäisten oppilaiden ajattelu on konkreettisten operaatioiden vaiheessa eli he tarvitsevat konkreettisia esimerkkejä ja monia eri aistikanavia hyödyntävää opetusta. (Huitt & Hummel 2003.) Koska tällä tasolla ajatteleva lapsi suhtautuu asioihin hyvin kirjaimellisesti ja harjoittelee vasta kielen ja symbolien kautta ympäröi-

vän maailman hahmottamista, on opetusta mielestäni luonnollistakin antaa mahdollisimman konkreettisella ja visuaalisesti havainnollistavalla tavalla. Ikänsä ja ajattelunsa puolesta kehittyneemmät oppilaat alkavat ymmärtämään abstrakteja käsitteitä ja ovat valmiita kehittymään ohjelmoinnissa graafisen tai tekstipohjaisen ohjelmointiympäristön parissa. Tämä tulos ei siis ole nähdäkseni kovinkaan yllättävä, mutta kuitenkin huojentava, sillä näillä keinoilla ohjelmoinnin opetusta ohjeistetaan annettavaksi tällä hetkellä Suomen peruskouluissa (Mykkänen, Liukas 2014; POPS 2016).

Katsauksen tulokset viittaavat siihen, että opettajan opetukseen ja pedagogiikkaan liittyvät asiat ovat kiinteästi yhteydessä oppilaiden ohjelmoinnin oppimiseen. Tämäkin tulos tuntuu luontevalta ja loogiselta. Tulos on nähdäkseni hyvinkin paljon konstruktivistisen oppimiskäsityksen mukainen. Suuri osa niistä asioista, jotka edesauttavat ohjelmoinnin oppimista, ovat sellaisia, joissa oppija on keskiössä ja konstruoi itse sosiaalisen vuorovaikutuksen tukemana. Opettajan tehtävänä on ohjata oikeaan suuntaan, asettaa ohjaavia kysymyksiä ja ohjata oppilaita ratkaisemaan ohjelmointiongelmia. Lisäksi tuloksissa korostuu opettajan oma osaaminen ohjelmointiin liittyen, sekä opettajan taidot hyödyntää oppilaskeskeisiä opetusmetodeita. Myös nämä huomiot voisi mielestäni liittää minkä tahansa aiheen tai oppiaineen menestyksekkääseen opettamiseen.

Katsauksen tulosten perusteella myös oppilaan ominaisuudet ja valmiudet vaikuttavat ohjelmoinnin oppimiseen. Tämäkin on uskoakseni kaikkeen oppimiseen ja opetukseen vaikuttava luonnollinen asia. Yksi luonteenpiirre näyttää kuitenkin tulosten pohjalta olevan ohjelmoinnin oppimiseen positiivisesti vaikuttava. Sinnikkyys auttaa erityisesti ohjelmoinnin oppimisessa.

Katsaus vahvistaa myös opetussuunnitelman perusteissa (POPS 2016) korostetun algoritmisen ajattelutaidon merkityksen, sillä tulosten mukaan ohjelmointitaitojen kehittymisen kannalta ratkaisevaa on algoritmisen ajattelutaidon kehittyminen. Näiden tulosten lisäksi kirjallisuuskatsaukseni tukee tämänhetkistä näkökulmaa siitä, miten ohjelmoinnin opetusta tulisi lähteä toteuttamaan peruskoulussa. Pähkinänkuoressa tämän voisi esittää niin, että ohjelmoinnin opetuksen, niin kuin kaiken muunkin opetuksen, on edettävä lapsen ajattelun ja kehityksen vaiheiden sallimissa rajoissa, konstruktivistisen oppimiskäsityksen mukaisella tavalla.

7.2 Tutkielman luotettavuus

Suuri osa tutkimuksen toteuttamisesta on valintojen tekemistä ja niiden tarkastelua kriittisesti. Esittelen seuraavaksi tässä tutkimuksessa tekemiäni valintoja ja kerron niiden taustalla käymääni pohdintaa. Tarkastelen pääasiassa aineiston keräämiseen vaikuttaneita tekijöitä.

Systemaattinen kirjallisuuskatsaus pohjautuu tutkimusta ohjaavaan tutkimuskysymykseen. Laadullisessa tutkimuksessa on tärkeä tarkasti määritellä tutkimusongelma sekä rajata tutkimuskysymys. (Fink 2019.) Tutkimuskysymykseni on tässä katsauksessa rajattu, mutta laaja. Perustelen tutkimuskysymyksen laajuutta sillä, että halusin löytää laajasti erilaisia kuvauksia ohjelmoinnin opettamisesta. Tavoitteenani oli löytää monipuolisesti ohjelmoinnin opettamiseen käytettyjä tekniikoita ja välineitä, jotta saisin muodostettua mahdollisimman kattavan näkökulman siitä, mitkä tekijät vaikuttavat suotuisasti ohjelmoinnin opiskeluun ja oppimiseen.

Systemaattiseen kirjallisuuskatsaukseen kuuluu tarkka aineistonhakumenetelmä, jossa nojataan ennalta päätettyihin sisään- ja poissulkukriteereihin. Tämän tutkimuksen aineisto on kerätty EBSCOhost -tietokannasta, joka koostuu tieteellisistä artikkeleista. Tarkka aineistonhakuprosessi on kuvattu luvussa 4.2. Jälkikäteen ajatellen, olen sitä mieltä, että käyttämäni hakutermit olisivat voineet olla myös erilaiset. Termi *computational thinking*, eli algoritminen ajattelu, olisi voinut tuottaa hakutuloksena parempia artikkeleita. Toisaalta on mielenkiintoista, että tuo termi nousi esille aineistossa, vaikka se ei esiintynyt hakutermeissä.

Kokonaisuudessaan aineistonhakuprosessi oli suhteellisen lyhyt, eikä sisään- ja poissulkukriteereitä ollut kovin montaa. Olen tyytyväinen siihen valintaan, että kävin manuaalisesti läpi eli luin tiivistelmät 42 artikkelista. Mikäli rajausten jälkeen läpi käytäviä artikkeleita olisi ollut vähemmän, olisi voinut tutkimuksen aineisto jäädä kovin pieneksi. Mielestäni 9 artikkelia on sopivan kokoinen aineisto tämän laajuiseen kirjallisuuskatsaukseen.

Systemaattinen kirjallisuuskatsaus voi toisintaa eräitä aineiston valintaan liittyviä harhoja. Tällaisia harhoja ovat julkaisuharha, kieliharha ja toistojulkaisemisen harha. Julkaisuharhalla tarkoitetaan sitä vääristymää, joka liittyy julkaistaviin tutkimuksiin. Tilastollisesti merkittäviä lopputuloksia sisältävät tutkimukset julkaistaan todennäköisemmin kuin ne, joiden lopputulokset eivät ole tilastollisesti merkittäviä. (Oxman 1999.) Tämä harha ei kuitenkaan uskoakseni ole kovin voimakas tässä tutkimuksessa, sillä aineistoni koostuu laadullisista tutkimuksista.

Kieliharha sen sijaan on tämän tutkimuksen kannalta olennaisempi. Kieliharhalla tarkoitetaan sitä, että kirjallisuuskatsauksen aineisto muodostuu vain yhdellä kielellä kirjoitetuista artikkeleista, jolloin muunkieliset tutkimukset jäävät seulan ulkopuolelle. Vaikka monet tutkimukset julkaistaan nykyään englannin kielellä, on silti tavallista, että tilastollisesti epämerkitseviä tuloksia sisältävät tutkimukset julkaistaan vain tutkimuksen kotimaassa kotimaisella kielellä. (Oxman 1999.) Olen valinnut tähän tutkimukseen vain englanniksi kirjoitettuja artikkeleita, jotka ovat ilmestyneet vertaisarvioiduissa ja arvostetuissa julkaisuissa. Toisaalta se on hyvä asia, mutta samalla se saattaa vahvistaa harhaa aineistossa.

Toistojulkaisemisen harha esiintyy silloin, kun jokin tutkimus julkaistaan uudestaan johtuen päivitetyistä tiedoista tai silloin, kun niihin on lisätty seurantaselvityksiä. Aina toistojulkaistuja artikkeleita ei merkitä erikseen, jolloin niitä on vaikea erottaa. (Oxman 1999.) Tämä kirjallisuuskatsaus on niin suppea, ettei toistojulkaisemisen harha näy aineistossa. Lisäksi aineisto on manuaalisesti seulottu, jolloin duplikaattitutkimukset olisivat luultavasti nousseet esille.

Tekemäni kirjallisuuskatsauksen aineisto edustaa vain hyvin marginaalista osaa kokonaisuudesta. Tämä varmasti vaikuttaa suuresti tulosten luotettavuuteen. Vaikka tämän katsauksen aineisto on voinut vääristyä monenlaisista tekijöistä johtuen, on se silti nähdäkseni kokoonsa nähden kattava. Aineiston sisältämät tutkimukset on tehty Portugalissa, Espanjassa, Iso-Britanniassa, Yhdysvalloissa, Kreikassa, Brasiliassa ja Tansaniassa. Jakauma on maiden välillä epätasainen, sillä tutkimuksista kolme on tehty Iso-Britanniassa, kun muita maita edustaa vain yksi tutkimus. Kokonaisuutta katsoen, aineistoa on kuitenkin neljältä mantereelta. Aasiassa tehtyjä tutkimuksia ei kuitenkaan ole aineistossa yhtään. Kokonaisuutta tarkastellen, tämän katsauksen tarkoituksena on ollut tarkastella laajasti ohjelmoinnin opettamiseen ja oppimiseen vaikuttavia tekijöitä, mikä on mielestäni onnistunut hyvin.

7.3 Oman oppimisen pohdinta

Opinnäytetyö on usein yksi laajimmista töistä, jonka opiskelija tekee opintojensa aikana. Itselleni tämä on jo neljäs opinnäytetyö, jonka olen kirjoittanut, ja olen muodostanut itselleni sopivan tavan työstää ja tehdä tutkimuksia. Olen kerran aikaisemmin toteuttanut systemaattisen kirjallisuuskatsauksen, ja halusin toteuttaa tämänkin opinnäytetyön ennestään tuntemallani tavalla. Yksi syy tähän valintaa oli se, että olen tätä kirjoittaessani työskennellyt täyspäiväisesti erityisopettajana ja lisäksi suorittanut kahta korkeakoulututkintoa yhtäaikaisesti.

Olen tyytyväinen tekemääni tutkimukseen ja uskon, että aikapaineesta huolimatta olen onnistunut tuottamaan tarkoituksenmukaisen opinnäytetyön ja laadukkaan kirjallisuuskatsauksen. Valitsin katsauksen aiheen oman mielenkiintoni sekä erityisosaamiseni mukaan. Olen aiemmalta koulutukseltani luokanopettaja ja viimeisimpien opiskeluiden myötä ammatiltani erityisopettaja. Olen vuosia ollut kiinnostunut myös ohjelmoinnista, joten ohjelmoinnin opetus peruskoulun kontekstissa oli aiheena mielekäs. Olen myös tämän opinnäytetyön tekemisen aikana päässyt itse opettamaan ohjelmoinnin valinnaista kurssia yhdeksäsluokkalaisille. Tästäkin näkökulmasta katsottuna aiheeni on ollut minulle ajankohtainen.

Ammatillisen kehittymisen näkökulmasta tämän opinnäytetyön kirjoittaminen on ollut osin hyödyllistä, mutta osittain myös olemassa olevaan tietämykseeni pohjaavaa. En osaa vielä arvioida, minne ja millaisiin työtehtäviin päädyn tulevaisuudessa. Itse toivon, että pääsisin työskentelemään jollain tavalla koulutuksen ja ohjelmistokehityksen rajapintaan. Tästä näkökulmasta katsottuna tietämys ja ymmärrys ohjelmoinnin opetuksesta ja oppimisesta on hyödyllistä. Olisi myös kiinnostavaa päästä mukaan kehittämään opetuksen digitalisaatiota sekä vaikuttamaan digiopetusmateriaalien ja opetussovellusten kehitykseen. Mikäli päädyn puhtaasti ohjelmistokehityksen työtehtäviin, tämä opinnäytetyön aikana kartuttamani tietämys tuskin tuo lisäarvoa työskentelyyni. Koen kuitenkin tärkeäksi sen, että olen tätä tutkimusta tehdessäni perehtynyt itseäni kiinnostavaan aiheeseen ja päässyt lukemaan aiheeseen liittyvää tutkimuskirjallisuutta.

Arvostan tutkimusperustaista tietoa ja pidän ajankohtaiseen tutkimustietoon tutustumista tärkeänä. Tämän opinnäytetyön tekemisen aikana pääsin tutustumaan parin viime vuoden aikana tuotettuun tietoon. Suuri osa tutkimuksista on saatavilla vain maksullisten tietokantojen ja kokoelmien kautta. Opintojeni päättyessä päättyy myös oikeuteni tietokantojen aineistojen lukemiseen. Opinnäytetyön tekeminen on mielestäni ollut aina se viimeinen mahdollisuus tutkia ajan kanssa itseä kiinnostavia artikkeleita ja saada ajankohtaisinta tietoa.

Vaikka tekemäni kirjallisuuskatsaus ei varsinaisesti tuottanut aiemmasta tutkimuksesta poikkeavaa tietoa tai mullistanut käsitystäni ohjelmoinnin opetuksesta, olen tyytyväinen tuotokseeni. Se, että ei tullut uutta tietoa, on sekin itsesään tärkeä tulos. Laadulliset kirjallisuuskatsaukset kai harvoin tuottavat uutta tietoa, mutta ne voivat avata uusia näkökulmia. Itselleni tämän opinnäytetyön kirjoittaminen avasi ymmärrystä erilaisten ohjelmointityökalujen sekä opetusmetodien taustalla olevalle tieteelliselle tiedolle.

7.4 Jatkotutkimusaiheet

Peruskoulun ohjelmoinnin opetuksesta voisi nostaa lukuisia jatkotutkimusaiheita. Itse olen erityisen kiinnostunut kolmesta teemasta, jotka sivuavat tätä tutkimusta: ohjelmoinnin opetuksen toteutuminen peruskoulussa, opettajien käsitykset ohjelmoinnista ja sen opettamisesta sekä oppilaiden käsitykset ohjelmoinnista.

Ohjelmoinnin opetusta olisi mielenkiintoista tutkia siitä näkökulmasta, miten se on käytännössä tuotu osaksi peruskoulua. Itse uskon, että monin paikoin olisi vielä parannettavaa. Kiinnostavaa on erityisesti se, miten ja millä keinoilla ohjelmoinnin opettamista tuetaan tai miten siihen on varustauduttu. Onko henkilökunta saanut täydennyskoulutusta tai onko koulun laite- tai muihin resursseihin tehty muutoksia? Lisäksi olisi mielenkiintoista selvittää, onko ohjelmoinnin opetusta laiminlyöty ja jos on, mikä siihen on syynä. Ovatko nämä syyt ulkoisia vai esimerkiksi opettajien taitotasoon liittyviä? Esimerkiksi laitekanta saattaa monissa kouluissa olla vanhentunutta tai toimimatonta. Miten ohjelmointia on opetettu niissä kouluissa, joissa laitteita ei ole ollut käytettävissä? Myös rehtoreiden rooli digitalisaatioprosessin sekä ohjelmoinnin opetuksen johtamisessa on kiinnostava.

Moni opettaja on varmasti joutunut opettelemaan ja sisäistämään paljon uutta ohjelmoinnin opetuksen myötä. Olisi kiinnostavaa tietää, kuinka moni opettaja opettaa ohjelmointia tällä hetkellä, ja millaisia keinoja he opetuksessaan käyttävät. Ohjelmoinnin opetus herätti paljon keskustelua ennen kuin se lisättiin opetussuunnitelman perusteisiin. Yksi jatkotutkimusaihe voisi liittyä tämänhetkisiin käsityksiin ja vallalla olevaan diskurssiin. Ovatko ohjelmoinnin opettamista koskevat mielipiteet muuttuneet viiden vuoden jälkeen?

Erityisen mielenkiintoista olisi myös selvittää toimivia tapoja opettaa ohjelmointia kestävästi. Tällä tarkoitan sitä, että ohjelmoinnin opettelu aloitetaan jo varhaisessa vaiheessa, ja taitoja kartutetaan koko koulu-uran ajan. Tähän liittyen voisi selvittää myös oppilaiden näkökulmaa ja käsityksiä ohjelmoinnista. Mitä he ajattelevat ohjelmoinnista ja minkälainen kokonaiskuva heille on muodostunut siitä? Olisi esimerkiksi kiinnostavaa tietää, osaavatko oppilaat yhdistää leikin tai fyysisen harjoituksen ohjelmoinnin tai algoritmisen ajattelun taitojen harjoitteluun. Myös koululaisten hallitsemaa käsitteistöä ja ymmärrystä olisi kiinnostavaa selvittää.

Lähteet

Alasoini, T. 2015. Digitalisaatio muuttaa työtä – millaista työelämää uudistavaa innovaatiopolitiikkaa tarvitaan. *Työpoliittinen aikakauskirja*, *2*(2015), 26–37. Työ- ja elinkeinoministeriö.

Alasuutari, P. 2011. Laadullinen tutkimus 2.0. Helsinki: Vastapaino.

Amsel, E. B., Amsel, E., Byrnes, J. P. & Jean Piaget Society Staff. 2002. Language, Literacy, and Cognitive Development: The Development and Consequences of Symbolic Communication. Mahwah: Lawrence Erlbaum Associates.

Bada, S. O. and Olusegun, S. 2015. Constructivism learning theory: A paradigm for teaching and learning. Journal of Research & Method in Education, 5(6), s. 66–70.

Bettany-Saltikov, J. 2012. How To Do A Systematic Literature Review In Nursing: A StepBy-Step Guide. Maidenhead: McGraw-Hill Education.

Ferragina, P. K. & Luccio, F. K. 2018. Computational Thinking: First Algorithms, Then Code. Cham: Springer International Publishing.

Fink, A. 2019. Conducting research literature reviews: From the internet to paper. Sage publications.

Gibbons, P. 2002. Scaffolding language, scaffolding learning. Portsmouth, NH: Heinemann.

Hammond, J., & Gibbons, P. 2005. What is scaffolding. Teachers' voices, 8, s. 8–16.

Huitt, W., & Hummel, J. 2003. Piaget's theory of cognitive development. Educational psychology interactive, 3(2), s. 1–5. Luettavissa: https://intranet.newriver.edu/images/stories/library/stennett_psychology_articles/Piagets%20Theory%20of%20Cognitive%20Development.pdf. Luettu 25.3.2021.

Isojärvi, J. 2015. Kirjallisuushaku. THL – Terveyden ja hyvinvoinnin laitos.

Morris, David, Uppal, Gurmit, and Wells, David. 2017. Teaching Computational Thinking and Coding in Primary Schools. London: SAGE Publications.

Mykkänen, J., Liukas, L. & Tammi, M. 2014. Koodi2016: Ensiapua ohjelmoinnin opettamiseen peruskoulussa. 1. p. Linda Liukas ja Juhani Mykkänen. Helsinki

Mäkelä, M. 1999. Systemaattiset katsaukset tieteellisen työn perustana. Teoksessa Götzsche, P., Semberg, V., Teikari, M. & Varonen, H. Tieteestä käytäntöön: Systemaattiset kirjallisuuskatsaukset terveydenhuollossa. Terveydenhuollon menetelmien arviointiyksikkö, Stakes. Helsinki.

Opetushallitus 2020. Perusopetuksen opetussuunnitelman perusteet. Luettavissa: https://www.oph.fi/fi/koulutus-ja-tutkinnot/perusopetuksen-opetussuunnitelman-perusteet Luettu 13.10.2020.

Oxman, A. 1999. Systemaattisen katsauksen metodologisia kysymyksiä. Teoksessa Götzsche, P., Semberg, V., Teikari, M. & Varonen, H. Tieteestä käytäntöön: Systemaattiset kirjallisuuskatsaukset terveydenhuollossa. Terveydenhuollon menetelmien arviointiyksikkö, Stakes. Helsinki.

Painter, C. 2005. Learning through language in early childhood. A&C Black.

Piaget, J. 1964. The early growth of logic in the child. Routledge and Kegan Paul. Lontoo.

POPS 2014. Perusopetuksen opetussuunnitelman perusteet 2014. Opetushallitus. Helsinki. Luettavissa: https://www.oph.fi/sites/default/files/documents/perusopetuksen_opetussuunnitelman_perusteet_2014.pdf. Luettu 13.10.2020.

Salminen, A. 2011. Mikä kirjallisuuskatsaus?: Johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin. Vaasan yliopiston julkaisuja: Vaasa. Luettavissa: https://www.univaasa.fi/materiaali/pdf/isbn_978-952-476-349-3.pdf. Luettu 13.10.2020.

Spraul, V. A. 2012. Think like a programmer: An introduction to creative problem solving. San Francisco: No Starch Press.

Tanhua-Piiroinen, E., Kaarakainen, S. S., Kaarakainen, M. T., Viteli, J., Syvänen, A., & Kivinen, A. 2019. Digiajan peruskoulu. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 6/2019. Luettavissa: http://urn.fi/URN:ISBN:978-952-287-634-8. Luettu 25.3.2021.

Tuomi, J., Sarajärvi A. 2018. Laadullinen tutkimus ja sisällönanalyysi (Uudistettu laitos). Helsinki: Kustannusosakeyhtiö Tammi.

Wells, G. and Bridges, A., 1981. Learning through interaction: volume 1: the study of language development (Vol. 1). Cambridge University Press.

Artikkelit:

Csizmadia, A., Standl, B., & Waite, J. 2019. Integrating the Constructionist Learning Theory with Computational Thinking Classroom Activities. Informatics in Education, 18(1), 41–67.

Jenkins, C. W. 2017. Classroom Talk and Computational Thinking. International Journal of Computer Science Education in Schools, 1(4).

João, P., Nuno, D., Fábio, S. F., & Ana, P. 2019. A cross-analysis of block-based and visual programming apps with computer science student-teachers. Education Sciences, 9(3), 181.

Kumalija, E. J., Fatih, Y. and Sun, Y. 2019. Students' Perception towards Program Visualization on Smartphone - Case of SunLab Initial Investigation. International Association for Development of the Information Society.

Kwon, K., Lee, K. and Chung, J. 2018. Computational Concepts Reflected on Scratch Programs. International Journal of Computer Science Education in Schools, 2(3).

Kynigos, C. and Grizioti, M. 2018. Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space. Informatics in Education, 17(2), s. 321–340.

Muñoz-Repiso, A. G.-V. and Caballero-González, Y.-A. 2019. Robotics to Develop Computational Thinking in Early Childhood Education. Comunicar: Media Education Research Journal, 27(59), s. 63–72.

Waite, J. L., Curzon, P., Marsh, W., Sentance, S., & Hadwen-Bennett, A. 2018. Abstraction in action: K-5 teachers' uses of levels of abstraction, particularly the design level, in

teaching programming. International Journal Of Computer Science Education In Schools, 2(1), s. 14–40.

Von Wangenheim, C. G., Alves, N. C., Rodrigues, P. E., & Hauck, J. C. 2017. Teaching Computing in a Multidisciplinary Way in Social Studies Classes in School – A Case Study. International Journal of Computer Science Education in Schools, 1(2), s. 3–16.

Liitteet

Liite 1 Analyysiyksiköistä tehdyt suomennokset ja niiden luokittelu

Havainto	Artikkeli (nro)
Yläluokka 1: Ohjelmointityökaluun liittyvät asiat	
Ohjelmointityökaluun liittyvät asiat	
oikein valittu visuaalinen ohjelmointiympäristö	1
ohjelmointiympäristön yksinkertainen ja vetoava ulkoasu	1
symbolinen kieli (lukutaidottomille)	1
robotteihin pohjautuvat ohjelmat edesauttavat ajattelutaitojen kehitystä (debugging, sequenses, action-instruction correspondense)	2
erilaiset oppimis- ja ajattelutyypit huomioiva työväline	3
siirtymä visuaalisesta ohjelmointiohjelmasta tekstipohjaiseen ohjelmaan täytyy alustaa hyvin	4
fyysisen aktiviteetin kautta ohjelmoinnin opettelu (esim. kartalla liikkuminen ohjeiden mukaan)	5
ohjelmointiblokeista koostuva ohjelmoinnin harjoitteluohjelmisto	5
Mobiililaitteiden käyttäminen ohjelmoimisessa --> helppo käyttää ja hyödyntää myös vapaa-ajalla	8
ohjelma arvioi oppilaan kirjoittamaa koodia	4
ohjelmoinnin opetusohjelman tekemä arviointi	4
se, että voi itse käyttää/pelata omia ohjelmoimalla tehtyjä tuotoksia (esim. pelejä) motivoi	7
Yläluokka 2: Opettajaan, opetukseen ja koulunkäyntiin liittyvät asiat	
Opettajaan liittyvät asiat	
opettajien tietämys ohjelmointityökalusta / ympäristöstä	1
opettajan itselleen sopivat (mukavuusalueella olevat) pedagogiset strategiat	1
opettajien oikeiden termien käyttäminen ohjelmointia harjoiteltaessa	5
opettajan ymmärrys ohjelmoinnista	5
Muuhun opetukseen ja kouluun liittyvät tekijät	
ohjelmoinnin harjoittelun aloittaminen varhain	1
ohjelmoinnin harjoittelun aloittaminen varhain	2
Vapaa-ajalla mahdollisuus ohjelmoida	8
Luovan ajattelun stimuloiminen	8
ohjelmoinnin yhdistäminen muihin luonnontieteen oppianeisiin	2
Pedagogiset asiat	
oikeiden termien ja sanavaraston kehittäminen ohjelmoinnista	5
ongelmanratkaisun / ohjelmoinnin pelillistäminen (pelit ja puzzlet)	1
oppilasta haastava taso	1
helposti suoritettavat tehtävät	1
saman (arkielämän) ongelman ratkaiseminen muilla keinoin (esim. matematiikalla) ja sen jälkeen ohjelmoinnin avulla	3
use, modify, create -lähestymistapa (mallinnetaan ratkaisu, jota oppilaat soveltavat uuteen ympäristöön)	3

ohjelmointiongelmien tai ohjelman rakenteen käsittely arkielämän pulmien tai toimintojen kautta --> linkittäminen johonkin tuttuun	3
oppilaita tulee ohjata arvioimaan käyttämiensä metodien tehokkuutta	4
oppilaille tulee ohjata käyttämään erilaisia metodeita	4
ohjelmointia tulee opettaa algoritmisen (computational thinking) ajattelumallin avulla	4
opettajan antama laaja tuki oppilaan / ryhmän kohdatessa pulmia ohjelmoidessaan	7
autonomian lisääminen omien ratkaisuiden tuottamisessa	3
algoritmista ajattelua kehittää fyysisen toiminnan muuttaminen algoritmeiksi, eli sanoittaminen toimintaohjeiksi	5
fyysisten ohjelmointiharjoitusten huolellinen linkittäminen koodaamiseen (miten kyseinen asia näkyy ohjelmassa)	5
Kommunikointi	
manuaalinen (opettajan tekemä) arviointi	4
opettajien lisäämät kommentit koodauspohjiin, joita oppilaat käyttävät --> vinkkejä mitä koodia tulee käyttää	5
koodin kommentointi koodiin ennen ja koodauksen aikana	5
Oppilasjohtoinen keskustelu ohjelmoinnin opetuksessa	9
Oppilaiden opettajalta kysyminen ja ideoiden kommentointi	9
Oppilasjohtoinen dialogi	9
Opettajan ohjaavat kysymykset, kun oppilas ohjelmoi ja keksii ratkaisuita (--> auttaa selvittämään seuraavat askeleet ohjelmointiprosessissa)	9
Opettaja ohjaa kysymyksillä oppilasta löytämään ohjelman virheet	9
Opettajan tekemät kysymykset, jotka ohjaavat oppilaan järkeilyyn --> miten, miksi	9
Dialoginen opettaminen --> ei opettajan monologia, joka kulkee kohti opettajan tavoitteita	9

Yläluokka 3: Oppilaan tekemiseen, ajatteluun ja taitoihin liittyvät asiat

Oppilaan tekemiseen/ajatteluun liittyvät asiat	
itsesäätelykeinojen harjoittelu suunnittelemisen harjoittelulla	5
itsearviointi	5
valinnanvapaus siinä, mitä ollaan tekemässä	7
sinnikkyys	7
aktiviteetin merkityksellisyys oppijalle	3
Ohjelmointitaidot	
flow control	4
oppilaiden tulee ymmärtää ohjelman erityispiirteet	4
ohjelmoinnin eri konseptien ymmärtäminen	4
Ohjelmoinnin konseptien (laskutoimitukset, muuttujat, listat, if-else, loopit, funktiot) ymmärtäminen	8
algoritmisen ongelmanratkaisun perusteiden harjoittelu	7
ohjelman designin yksityiskohtien rajaamisen harjoittelu	5
sen ymmärtäminen, että ohjelmointi on sarja peräkkäisiä käskyjä	7
ohjelman suunnittelun harjoitteleminen	5
oman suunnitelman tehtävyyden (do-ability) arvioiminen	5
Ajattelutaidot	
taito purkaa ongelmat osiksi	4

osiin hajottaminen	6
pattern recognition	6
loogisen ajattelun taidot	4
tiedon esittämisen taidot	4
abstraktointi	6
abstraktointi	4
ohjelmointiongelman ymmärtäminen syvemmällä tasolla	3
ohjelmointiongelman ratkaiseminen usealla vaihtoehtoisella tavalla	3
ohjelmointiongelmien purkaminen osiin	3
tarvittavien tietojen löytäminen ohjelmointiongelmasta ratkaisua varten ja tarpeettomien tietojen pois sulkeminen	3
ohjelmointiongelmat tulee osata pilkkoa	4
ohjelmointia tulee osata suunnitella	4
ymmärtäminen, mikä on ohjelmoinnin avulla mahdollista tai saavutettavissa	5

Kommunikointi

omien tuotosten / tulkintojen esitteleminen / läpi käyminen	3
ohjelmointiongelmasta keskusteleminen opiskelutovereiden kanssa	3
yhdessä /pareittain ohjelmoiminen	5
pareittain ohjelmoiminen	7
omien tuotosten esittely muulle luokalle/ryhmälle	7
vertaisten toisilleen selittäminen, miten olivat tehneet tai toteuttaneet jonkin ratkaisun	7
Vertaistuki ja vertaisilta oppiminen	9
Oppilaan ohjelmointipulmien sanallistaminen	9
Oppilaan tekemät hypoteesit ohjelmointipulmien ratkaisuista	9
Oppilaan tekemä ohjelman toiminnan ja ongelmien ennustaminen ennen kokeilemista	9
Oppilaan verbaalinen järkeily ongelmien ratkaisemiseksi	9
Oppilaan omien ajatusten sanallistaminen silloin opettaja kysyy ohjelmointia ohjaavia kysymyksiä	9