

**"Se on mun mielest taas yks tapa rikastuttaa sitä
opiskelua ja oppimista, itekki oppii sit uusia juttui."
- Alakoulun opettajien käsityksiä ohjelmoinnin
opettamisesta**

Jenni Makkonen & Anniina Pyykönen

**Pro-gradu tutkielma
Kevät 2018
Opettajankoulutuslaitos
Jyväskylän yliopisto**

Tiivistelmä

Makkonen, Jenni & Pyykönen, Anniina. 2018. "Se on mun mielest taas yks tapa rikastuttaa sitä opiskelua ja oppimista, itekki oppii sit uusia juttui." - Alakoulun opettajien käsityksiä ohjelmoinnin opettamisesta. Kasvatustieteen Pro-gradu tutkielma. Jyväskylän yliopisto. Opettajan koulutuslaitos. 79 sivua.

Tässä tutkimuksessa selvitettiin, kuinka luokanopettajat ovat ottaneet ohjelmoinnin opetuksen kouluissa vastaan. Tutkimuksen tavoitteena oli saada selville opettajien käsityksiä ohjelmoinnin opettamisesta ja opettajien motivaatiota. Tutkimuksessa selvitettiin myös, millaista tukea ja resursseja opettajat ovat saaneet ohjelmoinnin opettamiseen, sekä kuinka ohjelmointi on otettu heidän työyhteisössään vastaan.

Tutkimus tehtiin laadullisen sisällönanalyysin keinoin. Tutkimuksessa haastateltiin seitsemää alakoulussa työskentelevää opettajaa puolistrukturoidulla haastattelulla.

Aineistosta nousi esiin viisi teemaa. Tutkimuksessa saatiin selville, että ohjelmoinnin opetuksen monipuolisuus on pitkälti opettajan omasta kiinnostuksesta ja asenteesta riippuvainen. Opettajille on melko hyvin tarjolla hyödyllisiä koulutuksia ohjelmoinnista, mutta työajalla on mahdollista käydä vain rajattu määrä täydennyskoulutuksia. Opettajien asenne ohjelmointia kohtaan on melko positiivinen, mutta sen asemaan suhtauduttiin kriittisesti. Ohjelmoinnilla on paikkansa koulumaailmassa ja sitä perusteltiin muun muassa nyky-yhteiskunnan tarpeilla. Ohjelmoinnin opetuksen keskeisimmiksi haasteiksi nostettiin aika ja raha.

Avainsanat: Ohjelmointi, algoritminen ajattelu, opetussuunnitelma, tieto- ja viestintäteknologia-aidot, alakoulu

Sisällysluettelo

1	Johdanto.....	1
2	Ohjelmointi.....	5
2.1	Ohjelmointiprosessin eteneminen.....	6
2.2	Ohjelmointikielet.....	8
2.3	Ohjelmoinnin sanastoa	9
2.4	Robotiikka ja sen käyttö peruskoulussa	10
2.5	Algoritminen ajattelu	11
3	Ohjelmointi peruskoulussa.....	15
3.1	Ohjelmointi peruskoulun opetussuunnitelmassa	15
3.2	Ohjelmointi valtakunnallisessa opetussuunnitelmassa	15
3.3	Ohjelmointi Englannin opetussuunnitelmassa verrattuna Suomen opetussuunnitelmaan	16
3.4	Miten ohjelmoinnin opetus alakoulussa on perusteltu?	18
4	Aiempaa tutkimusta ohjelmoinnin opetuksesta	22
4.1	Tutkimusta Suomessa	22
4.2	Esimerkkejä kansainvälisistä tutkimuksista ohjelmoinnin opetuksesta	25
4.2.1	Algoritminen ajattelu arjen ongelmissa	25
4.2.2	Ohjelmointi ja ongelmanratkaisu.....	26
4.2.3	Ohjelmointi ja motivaatio	26
4.2.4	Ohjelmoinnin opettaminen opettajien näkökulmasta.....	27
5	Millaista tukea opettajille on tarjolla?	31
5.1	Täydennyskoulutukset ja koulun tuki	31
5.2	Millaisia opetusmateriaaleja ja -ympäristöjä opettajat käyttävät ohjelmoinnin opetuksessa?.....	32
6	Tutkimustehtävä ja tutkimuskysymykset.....	34
7	Tutkimuksen toteuttaminen	36
7.1	Tutkimuksen lähestymistapa.....	36
7.2	Aineiston keruu ja analysointi.....	37
7.2.1	Haastatteluaineiston keruumenetelmänä	37
7.2.2	Laadullinen sisällönanalyysi aineiston analysointimenetelmänä.....	39
7.3	Tutkimukseen osallistujat.....	40
7.4	Eettiset ratkaisut	41
8	Tulokset	43
8.1	Ohjelmoinnin opetus alakoulussa	44
8.1.1	Opettajan oma opetus	44

8.1.2	Ohjelmoinnin opetuksen eheyttäminen	47
8.1.3	Oppilaiden rooli ohjelmoinnin opetuksen toteutuksessa	48
8.2	Koulutus ja tuki.....	49
8.2.1	Tarjolla olevat koulutukset.....	49
8.2.2	Tutor-toiminta.....	51
8.2.3	Kollegoiden ja ystävien tuki.....	52
8.2.4	Opettajien tukitoiveet	52
8.3	Kouluyhteisön asenne ohjelmointia kohtaan	53
8.3.1	Oppilaiden innostus ohjelmointia kohtaan.....	54
8.3.2	Oma kiinnostus ja osaaminen	56
8.3.3	Itsensä kehittäminen.....	57
8.3.4	Työyhteisön ilmapiiri	58
8.4	Ohjelmoinnin merkitys osana opetusta	59
8.4.1	Ohjelmoinnin sopivuus koulumaailmaan.....	60
8.4.2	Tulevaisuuden tarpeet	61
8.4.3	Ohjelmoinnin kautta opittavat taidot	62
8.5	Ohjelmoinnin opetuksen haasteet.....	64
8.5.1	Aika.....	64
8.5.2	Taloudelliset resurssit ja laitteiden ylläpito	66
8.5.3	Teknologian muuttuva luonne	67
8.5.4	Asennehaasteet.....	68
8.5.5	Oppimisen arviointi	69
8.6	Tulosten koonti	70
9	Pohdinta	72
9.1	Tutkimustulosten tarkastelu ja johtopäätökset.....	72
9.2	Tutkimuksen luotettavuus ja jatkotutkimuksen mahdollisuudet	77
9.2.1	Tutkimuksen rajoitukset ja luotettavuus	77
9.2.2	Tulosten hyödyntäminen ja jatkotutkimushaasteet	79
	Lähteet	80
	Liitteet	86

1 JOHDANTO

Uusi perusopetuksen opetussuunnitelman perusteet (POPS) julkaistiin vuonna 2014 ja se astui voimaan alakouluissa syksyllä 2016. Uudistuneen opetussuunnitelman myötä digitalisaation rooli koulussa kasvoi huomattavasti. Lisäksi ohjelmointi lisättiin osaksi opetusta. Tämä edellyttää, että jokaisen alakoulun opettajan tulisi kyetä opettamaan ohjelmointia. Uudistus tarkoitti monille opettajille hyppyä tuntemattomaan. Opintojemme aikana olemme keskustelleet alakoulussa työskentelevien opettajien kanssa uudesta opetussuunnitelmasta. Keskustelu on monesti kääntynyt ohjelmointiin ja tämä keskustelu on ollut melko negatiivisävytteistä. Tämän esioletuksen pohjalta halusimme lähteä selvittämään, millaisia käsityksiä alakoulun opettajilla on ohjelmoinnin opetuksesta.

Teknologiaa sisällytetään jatkuvasti enemmän osaksi arkea, mutta sen itseisarvo ei pitäisi olla keskiössä. Tärkeämpää on se, mitä ihmiset voivat sen avulla luoda ja miten he voivat olla milloin vain yhteydessä yhteisöönsä. (Opetushallitus 2009, 35.) Ohjelmoinnin lisäystä opetussuunnitelmaan on perusteltu sillä, että maailma, jossa elämme, on täynnä ohjelmoituja laitteita ja sovelluksia. Tällöin ohjelmoinnin voidaan ajatella olevan osana yleissivistystä, asiana, josta kaikkien meidän pitäisi tietää edes niiden peruseriaatteet. (Luostarinen & Peltomaa 2016, 101-102.) Tämä ei tarkoita sitä, että kaikkien tulisi osata kirjoittaa koodia, vaan sitä, että ymmärretään algoritmista ajattelua, jolla kaikki ympäröimämme laitteet periaatteeltaan toimivat. Tämä ajattelutaito helpottaa ohjelmoinnin oppimista, jota voidaan ajatella yhtenä taitona esimerkiksi lukemisen ja laskemisen rinnalla. Kalelioglun (2015, 200) mukaan teknologian kanssa toimiessa tärkeimpiä ominaisuuksia on olla joustava ja innovatiivinen, koska teknologia kehittyy vauhdilla. Koko ajan on mahdollista kehittää uutta, ja jo lapsille voisi antaa mahdollisuuden toimia tuottajana, eikä vain vastaanottajana. Algoritmisen ajattelun harjoittaminen tulisi aloittaa jo varhaisessa vaiheessa. (Kalelioglun 2015, 200.) Ajattelun taitojen kehittäminen on

suuressa osassa ohjelmoinnin opetusta. Tässä tulee ottaa huomioon ohjelmoinnin tuoma itseilmaisun mahdollisuus ja ympäröivään maailmaan osallistuminen uudella tasolla, esimerkiksi sosiaalisen median kautta. Algoritmisen ajattelun lisääminen kouluun on yksi tapa ohjata oppilaiden tieto- ja viestintäteknologian taitoja. (Burke & Kafai, 2014, 9.)

Kansainvälisesti verraten Suomi on ollut mukana muuttamassa opetussuunnitelmaansa teknologian kehityksen mukana muiden maiden kanssa. Esimerkiksi Englanti otti tietotekniikan omaksi oppiaineekseen 2013 (Education of Department 2013, 7). Virossa taas ohjelmointi otettiin mukaan opetukseen jo vuonna 2012. Tässä uudistuksessa oli tavoitteena, että kaikki 7-vuotiaat aloittavat ohjelmoinnin opiskelun. (Vänskä 2012). Esimerkiksi Yhdysvalloissa tietotekniikan opinnot ovat vapaavalintaisia, jolloin ne jäävät usein vähemmälle huomiolle (Burke & Kafai 2014, 115-116). Monilla mailla on myös eroja siinä, millaisella kokonaisuudella tietotekniikkaa ja tätä myöten ohjelmointia opetetaan (Eurodice 2012, 142).

Suomen perusopetuksen opetussuunnitelmassa ohjelmointia lähestytään leikkien, pelien ja visuaalisen ohjelmointikielen muodossa, sekä harjoitellen ohjeiden ja käskysarjojen antamista algoritmiseen ajatteluun nojautuen. Kun ajatellaan algoritmisesti, ajattelun kohde puretaan osiin, samalla tunnistetaan ja muodostetaan siitä kaavoja. Tavoitteena on muodostaa ja antaa mahdollisimman yksiselitteisiä käskyjä. (Luostarinen & Peltomaa 2016, 101-102.) Algoritmisen ajattelu sisältää myös ymmärryksen tietotekniikasta ja tietokoneiden ohjelmoinnista (Koodiaapinen 2015). Opetussuunnitelman mukaan tulisi myös opettaa, että tekniset laitteet tarvitsevat ihmistä valintojensa tekoon. (POPS 2014, 129, 235.) Perusopetuksen opetussuunnitelmassa ohjelmointia, sekä tieto- ja viestintäteknologiaa ei olla erikseen laitettu omaksi oppiaineekseen, vaan se on sisällytetty laaja-alaisiin osaamistavoitteisiin ja näin ollen sitä opetetaan eheyttynä muihin oppiaineisiin. (POPS 2014, 23.)

Ohjelmoinnin lisäys on vielä melko uusi muutos, joten ohjelmoinnin opetuksen tilanteen kartoittaminen on vielä hieman hankalaa peruskoulun puolella. Digiajan peruskoulu 2017 - Tilannearvio ja toimenpidesuosituksukset -

selvityksen (Kaarakainen, Kaarakainen, Tanhua-Piiroinen, Viteli, Syvänen, & Kivinen, 2017, 39) mukaan opettajien ohjelmointitaidot ovat heikot ja vain viidesosa opettajista on kokeillut ohjelmointia oppilaidensa kanssa. Ohjelmoinnin osalta opettajat toivoivat lisää täydennyskoulutusta (Kaarakainen, ym. 2017, 59). Lisäksi OAJ:n tekemässä selvityksessä saatiin selville, että opettajat ovat saaneet täydennyskoulutusta tv-taitoihin liittyen, mutta niiden sisällöt on mielletty suppeiksi, eikä niitä ole ollut riittävästi (Hietikko, Ilves & Salo 2016, 16-17).

Opettajien käsityksiä ohjelmoinnin opetuksesta on tutkittu Suomessa erittäin vähän, joten tutkimuksen tekeminen aiheesta on tarpeellista. Omien esioletusten ja aikaisempien tutkimuksien perusteella tämän tutkimuksen tarkoituksena on siis tutkia alakoulun opettajien käsityksiä, valmiuksia ja asenteita ohjelmoinnin opetuksesta. Pyrimme selvittämään, kuinka paljon ja millaista tukea ohjelmoinnin opetukseen on tarjottu, kuinka sitä eheytetään muihin oppiaineisiin sekä kuinka tärkeäksi opettajat mieltävät ohjelmoinnin opetuksen alakoulussa. Tutkimusongelmat ovat tutkimustavoitteiden pohjalta seuraavat:

1. Millaisia käsityksiä alakoulun opettajilla on ohjelmoinnin opettamisesta alakoulussa?
2. Millaista tukea alakoulun opettajille on tarjottu ohjelmoinnin opetukseen?
3. Miten alakoulun opettajat eheyttävät ohjelmointia muuhun opetukseen?
4. Millainen merkitys alakoulun opettajien mielestä ohjelmoinnilla on koulussa?

Tämä tutkimus on toteutettu laadullisena tutkimuksena ja tutkimuksessa on tehty yhteistyötä Innokas-verkoston (innokas.fi 2018) kanssa. Tutkimuksemme perustana on Suomen valtakunnallinen opetussuunnitelma.

Luvussa kaksi esittelemme mitä ohjelmointi on, sen peruseriaatteita sekä ohjelmoinnin käsitteistöä. Otamme käsitteistöön mukaan myös robotiikan, sillä se on yksi yleisistä tavoista harjoittaa ohjelmointia alakoulun puolella. Kolmannessa luvussa tarkastelemme ohjelmointia alakoulun näkökulmasta ja

perehdymme, mitä eri opetussuunnitelmissa on aiheesta kirjoitettu. Esittelemme Suomen valtakunnallisen opetussuunnitelman ja vertaamme sitä Englannin opetussuunnitelmaan saadaksemme laajan kuvan ohjelmoinnin asemasta koulumaailmassa. Lisäksi kerromme, miten ohjelmoinnin opetusta on perusteltu. Luvussa neljä esitämme aiempaa tutkimusta ohjelmoinnin opetuksesta Suomesta ja ulkomailta. Viidennessä luvussa esittelemme lyhyesti opettajille tarjolla olevia tukimahdollisuuksia ja materiaaleja. Tämän jälkeen esittelemme tutkimustehtävämme sekä tutkimuskysymykset luvussa kuusi. Luvussa seitsemän kerromme tutkimuksemme toteutuksesta; sen lähestymistavasta, aineistonkeruusta ja analysoinnista, tutkimukseen osallistujista sekä eettisistä ratkaisuksista. Kahdeksannessa luvussa esitellään tutkimuksemme tulokset. Lopuksi yhdeksännessä luvussa tarkastellaan tutkimustuloksia ja esitellään johtopäätöksiä. Lisäksi pohditaan tutkimuksen luotettavuutta ja esitellään jatkotutkimushaasteet.

2 OHJELMOINTI

Ohjelmointia käyttävät elämässään kaikenlaiset ihmiset, kuten ohjelmoinnin asiantuntijat työssään ja tavalliset ihmiset arkisten toimintojensa mahdollistajana. Ohjelmointi on ihmisen toimintaa, jossa hän luo, muuntaa tai pidentää jonkin järjestelmän toimintoja. Ohjelmointia voi lähestyä teknisesti rakennellen sekä tieteellisesti laskelmoiden, eikä kumpaakaan osa-aluetta voi ohittaa ohjelmoidessa. (Van-Roy & Haridi 2004, xv.) Ohjelmointi tarkoittaa yksinkertaisuudessaan käskyjen antamista tietyn toiminnan saavuttamiseksi. Useimmiten nämä käskyt annetaan jollekin digitaaliselle laitteelle, kuten tietokoneelle, mutta ohjelmoinnin kaltaista toimintaa löytyy jokapäiväisessä arjen toiminnassa. Ohjelmoitava laite pystyy prosessoimaan sille annettuja käskyjä sille opettujen tietojen perusteella. (Hyvönen, Lappalainen & Lakanen, 2013, 2.) Käytännössä laite voi olla esimerkiksi ihminen, koska ihmisen tiedon prosessointitapa muistuttaa tietokonetta. Ihmiset saattavat olla tietyissä asioissa tietokoneita etevämpiä, kuten kuvien tulkinnessa ja jäsentelyssä, mutta tietokone pystyy seuraamaan ohjeita nopeammin kuin ihminen ja toiminaan suuremman tietomäärän kanssa. (Wing 2008, 3719.) Ihmisen toimintaan liittyy myös tiettyjä inhimillisten erheiden ja vapaan tahdon ominaisuuksia. Laite, esimerkiksi tietokone, ei itse pysty varsinaisesti luomaan mitään ilman, että sille selitetään miten. Ohjeet on myös oltava tarkkoja ja kuvaavia, koska tietokone ei voi itse tehdä päätöksiä tai soveltaa tietoaan. Kerran opittuaan tietokone pystyy kuitenkin toisintamaan oppimansa asian täsmälleen samalla tavalla niin monta kertaa, kun on tarpeellista. (Hyvönen, ym. 2013, 2.)

Kun ohjelmointi oli vielä uusi asia peruskoulussa, täytyi sen mukaanottoa opetukseen tarkastella rakentavasti, koska siitä koituisi pakoltakin muutoksia opetukseen. Papert (1980, 23) toteaa, että senhetkisen opetusmallin mukaan oppilaille oli usein taipumus ajatella, että jokin asia meni joko oikein tai väärin. Ohjelmoinnin maailmassa ajatellaan taas ennemmin niin, että jos jokin ei toimi, kuinka siitä voitaisiin kehittää toimivampi. Ajatusmallina tällainen virheiden

etsintä ja korjaus on kehittävämpää kuin jaottelu oikeaan ja väärään. Hän myös kirjoittaa, että opettelemalla ajattelemaan kuin tietokone, pystyy toimimaan tietokoneiden ympäristössä. Tällaisen mekaanisen ajattelutaidon oppimisessa on hyötyä myös esimerkiksi lukemaan oppimisessa ja laskemissa. Ohjelmoijan täytyy saada jokin toimimaan ohjeidensa avulla, jolloin tapahtuu myös oman työskentelyn reflektointia. (Papert 1980, 26-27.)

Ohjelmoija pystyy kehittämään ja opettamaan tietokoneelle uusia ohjelmia vanhoja malleja hyväksikäyttäen ja käyttämällä ongelmanratkaisutaitojaan uusien keinojen keksimiseen. Ohjelmointiin voi myös sisältää muuttujia ja koodin paloja, joilla voidaan mahdollistaa monikäyttöisyys. (Liukas & Mykkänen 2016, 16-18.) Ohjelmoinnissa onkin suurelta osin kyse siitä, että ymmärtää ohjelmoinnin peruseräitä ja pystyy ajattelemaan ohjelmoinnin velvoittamalla tavalla. Ohjelmointiprosessia on hankala täysin automatisoida, koska ohjelmoijan luovuudella on oma osansa prosessissa. (Luostarinen & Peltomaa, 2016, 102.) Tietokone on ohjelmoitu toimimaan tietyllä tavalla, ja ohjelmoijan on pystyttävä toimimaan näiden säädösten ja mahdollisuuksien rajoissa. Ohjelmointiin kuuluu paljon järjestelmällistä suunnittelua. Lappalainen & Viitanen (2012, 5) toteavatkin koodin kirjoittamisen olevan ohjelmoinnin näkyvin osa, mutta samalla helpoin ja mekaanisin.

2.1 Ohjelmointiprosessin eteneminen

Ennen kuin ohjelmakoodia päästään kirjoittamaan, tarvitaan paljon suunnittelua ja mallien laatimista. Suunnittelu ja käsitteellistäminen voivat olla työlästä ja aikaa vievää, mutta tarpeellista hyvän ohjelman luomisessa. Ilman ongelmanratkaisua ja kokeilua ei voi saada aikaan edistyneempää lopputulosta. (Van-Roy & Haridi 2004, xviii.) Seuraavaksi esitetään muutamia tapoja lähteä etenemään ohjelmakoodin suunnittelussa. Lappalainen & Viitanen (2012, 5-6) ovat yksinkertaisuudessaan kuvanneet ohjelmointiprosessin etenemistä seuraavalla tavalla: ensin saadaan tehtävä. Sen jälkeen tehtävä tarkennetaan ja hahmotellaan tarvittavia toimintoja. Seuraavaksi ohjelman toimintoja ja

rakenteita suunnitellaan. Sen jälkeen tarkennetaan yksityiskohtaisten algoritmien suunnittelulla. Tämän jälkeen valitaan ohjelmointityökalut. Sen jälkeen algoritmeja sommitellaan valitulle työkalulle. Seuraavaksi tulee ohjelmakoodin kirjoittaminen, jonka jälkeen päästään testailemaan ohjelmaa. Kun ohjelma ollaan testailtu, voidaan se ottaa käyttöön ja miettiä, miten ohjelmaa ylläpidetään.

Toisen ajattelutavan esittää Heinonen (2008), jonka mukaan ensin on ymmärrettävä kyseessä oleva ongelma, joka on ohjelmakoodin suunnittelun kannalta erittäin tärkeää. Tämän jälkeen lähdetään tekemään suunnitelmaa ongelman ratkaisuun algoritmisella suunnittelulla ja testailulla. Sen jälkeen päästään ohjelman kirjoitusvaiheeseen, jonka jälkeen tulee viimeinen testaus- ja virheidenetsintävaihe. Tämän tavan yhteydessä on mainittu myös ohjelmoinnin ongelmien rönsyilevä luonne, josta johtuen ongelmia pilkotaan yleensä pienemmiksi palasiksi. Tätä tapaa kutsutaan *top-down* menetelmäksi, jolloin isosta kokonaisuudesta lähdetään osittelemaan ongelmaa pienemmiksi ja tarkemmiksi kokonaisuusiksi. Tässä tapahtuu ongelman muuttaminen algoritmiselle tasolle, jolloin se on siirrettävissä tietokoneelle. (Heinonen 2008, 3.)

Yksi uudempi tapa lähestyä ohjelmointia on ketterät menetelmät, *Agile Software Development*, jonka mukaan lähestytään ohjelmointia pienemmissä palasissa ja kokonaisvaltaisemmin. Tässä menetelmässä painotetaan arvoina ensinnäkin yhteistyötä ja tekijää työkalujen ja prosessin sijaan. Myös toimivaa ohjelmaa arvostetaan enemmän kuin sitä, että saataisiin prosessi dokumentoitua mahdollisimman tarkasti. Prosessissa mahdollistetaan joustavuus vastaamalla muutoksiin orgaanisen suunnitelman seuraamisen sijaan. (Lappalainen ja Viitanen 2012, 5-6.) Teknologian muuttuvaisuus, käyttöhelppous ja vuorovaikutus on siis otettu paremmin huomioon tässä tavassa toimia. Nämä mallit ovat suunniteltu ohjelmoijille, joten niiden soveltuvuutta peruskouluun oppilaiden käyttöön on vaikea sanoa. Lappalaisen ja Viitanen (2012) sekä Heinosen (2008) tavat tarjoavat vaihteittaiset ohjeet ohjelmakoodin tekemiseen. Osa vaiheista muistuttaa hyvin paljon muidenkin oppiaineiden tehtävänratkaisumallia, joten niissä olisi tärkeää painottaa juuri ohjelmoinnissa tärkeitä vaiheita, kuten esimerkiksi testausta ja

kokeilua. Työkalujen valinta ja järjestelmän toiminnan ymmärtäminen edellyttää ohjelmoinnin tuntemusta. Ketterät menetelmät tuovat liikkumavaraa ja sosiaalista tarttumapintaa ohjelmointiin, mitä voisi hyvinkin soveltaa koulumaailmassa.

2.2 Ohjelmointikieliet

Ihmiset ovat luoneet tietokoneiden ohjelmointiin useita kieliä. Kaikki nämä kielet noudattavat samoja periaatteita; niihin kuuluu oma sanasto ja kielioppisäännöt. Ohjelmointikielen käyttäjät ovat myös kehitelleet muita kielenkäyttöä helpottavia ilmaisutapoja tehdäkseen kielestä toimivampaa. (Liukas & Mykkänen 2016, 16-18.) Ihmiset voivat käyttää keskinäiseen kommunikointiinsa esimerkiksi englantia. Ihmisen ja laitteen välinen kommunikointi tapahtuu vuorostaan esimerkiksi Python-kielillä. Eri ohjelmointikielillä saattaa olla enemmän tai vähemmän yhtäläisyyksiä toisiinsa (Luostarinen & Peltomaa 2016, 101). Ohjelmointikieliet ovat kehittyneet nopealla tahdilla niiden luomisesta lähtien. Ohjelmointikieliä käyttäessään ihminen kirjoittaa koneella koodia. Ajan myötä ohjelmoijat ovat keksineet parempia tapoja tehdä koodia; he ovat voineet yhdistellä ja soveltaa toisia koodinpaloja. Osaltaan myös koko ajan kehittyvä teknologia on mahdollistanut uusia tapoja tehdä koodia. (Liukas & Mykkänen 2016, 16-18.) Nykyään on tarjolla esimerkiksi jo valmiista koodinpalasista koostuvia ohjelmoinnin visuaalisia ohjelmointiympäristöjä kuten Scratch, joissa käyttäjän ei itse tarvitse varsinaisesti kirjoittaa koodia. (Meerbaum-Salant 2013, 241) Tällaiset ympäristöt ovat erityisen toimivia opetuskäytössä, koska oppilaiden ei tarvitse osata esimerkiksi Python-kieltä. Tällöin he pystyvät opettelemaan ohjelmointia kokeilemalla, samalla nähden visuaalisesti ja auditiivisesti mitä kokeilut saavat aikaan. Tätä kuvaa Van-Roy ja Haridi (2004) yhtenä tapana opettaa ohjelmointia. Tällä menetelmällä oppilaat pääsevät tehokkaasti havainnoimaan interaktiivisesti suoraan ohjelman toimintaa konkreettisilla laitteilla ja ohjelmilla. Oppilaat saavat suoraa palautetta ohjelman

toiminnasta ja tämä nopeuttaa prosessin ymmärrystä. (Van-Roy ja Haridi 2004, xviii-xix.)

2.3 Ohjelmoinnin sanastoa

Tutkittaessa ohjelmoinnin opetusta, on tärkeää ymmärtää ohjelmoinnin peruseriaatteita ja ymmärtää siihen liittyviä käsitteitä. Seuraavaksi esittelemme etenkin alakoulussa hyödyllistä käsitteistöä peruskoulun opettajille suunnattujen oppaiden pohjalta.

Tietokone toimii ohjelmoinnissa työn prosessoijana ja työskentelyalustana. Se vastaanottaa tietoa, tekee mitä käsketään ja näyttää lopputuloksen jollakin päätteellä. Ohjelmoijan vastuulle jää se, että tieto, jonka hän antaa koneelle on yhteensopiva koneen kanssa. (Liukas & Mykkänen 2016, 20.) *Algoritmi* tarkoittaa mahdollisimman tarkkaa kuvausta toimenpidesarjasta, jota jonkun toiminnan tekemiseen tarvitsee. Algoritmissa toiminta on purettu osiin, joita seuraamalla tietokonekin pystyy toteuttamaan halutun toiminnan. Yksikäsitteisyys on tarpeellista, koska tietokone ei osaa itse päätellä asioita kontekstista. Monitulkinnaisuus ohjeissa voi johtaa siihen, että tietokone ei pysty suorittamaan sitä ja toimenpidesarja ei toteudu. (Luostarinen & Peltomaa, 2016, 104.)

Peräkkäisyys tarkoittaa algoritmien lukemista aina peräkkäin. Ensin tehdään vaihe 1, sitten vaihe 2. jne. Ellei tietokone ole saanut muuta käskyä, käskyt luetaan ylhäältä alaspäin rivi kerrallaan. (Luostarinen & Peltomaa, 2016, 106.) Rivit ollaan myös numeroitu koodissa ja tätä ominaisuutta, eli *listaa*, voidaan hyödyntää tiedon organisoinnissa (Liukas & Mykkänen 2016, 21).

Muuttujat ovat nimettyjä paikkoja koodissa, joihin voi tallentaa esimerkiksi numeroita tai tekstiä. Näissä kohdissa sijaitsevaa tietoa voi muuttaa tarpeen mukaan. (Luostarinen & Peltomaa, 2016, 106.) Usein muuttujilla on myös toisiinsa liittyviä käskytoimintoja, eli muuttamalla yhtä muuttujaa, toinen muuttuja reagoi siihen. *Ehdollisuutta* hyödyntäen voi tapahtumalle mahdollistaa eri vaihtoehtoja. Jos-niin-muuten -ehdollisuutta käyttäen koodin lopputulos voi

olla erilainen riippuen syötetystä tiedosta. (Luostarinen & Peltomaa, 2016, 107.) *Silmukalla* voi laittaa jonkun koodin osan toistamaan itseään halutun määrän. Määrä voi olla tarkka lukumäärä tai toisto voi jatkua niin kauan, kun on tarpeen. (Liukas & Mykkänen 2016, 22; Luostarinen & Peltomaa, 2016, 107.) Toistojen määrä voi olla loputon.

Funktio on osa koodissa, jota voi käyttää useisiin eri toimintoihin. Se toimii pohjimmiltaan samoin kaikissa tilanteissa, mutta sen muuttujia voi vaihdella. Luostarinen ja Peltomaa (2016, 109) käyttävät esimerkissään kahvin ja teen valmistusta. Operaatio on osittain sama kummankin juoman valmistuksissa, jolloin näitä yhtäläisyyksiä voi hyödyntää. Tällöin ei tarvitse luoda useasti samanlaista koodia. Funktioita voi olla koodissa niin paljon kuin on tarve. Liukas ja Mykkänen (2016, 21) toteavat, että funktioiden käyttö on silloin kätevää, kun sen itse tekemiseen menisi enemmän aikaa kuin siihen, että tietokone tekee sen puolestasi. Funktioita voi kehittää ja yhdistellä muihin komentoihin.

2.4 Robotiikka ja sen käyttö peruskoulussa

Robotiikka on yksi ohjelmointiin liittyvä ala, johon on tarjolla paljon peruskouluun soveltuvaa materiaalia. Tästä johtuen robotiikka nousee usein paljon esille peruskoulun puolella ohjelmoinnista puhuttaessa. Robotiikka tarkoittaa ihmisen suunnittelemaa, rakentamaa ja ylläpitämiä laitteita, joiden toiminta määräytyy sillä, kuinka ne on ohjelmoitu. Robotiksi lasketaan tietynlaisen rakenteen omaava mekaaninen laite, joka voidaan uudelleen ohjelmoida (Suomen Robotiikkayhdistys 1999, 13). Robotiikan ymmärtämiseen ja osaamiseen kuuluu esimerkiksi mekaniikan, anturitekniikan, ohjelmoinnin, robottien kalibroinnin ja turvallisuuden huomioinnin taitoja. Yleensä robotiikan alalla eri ihmiset erikoistuvat huolehtimaan eri osista koko prosessissa. Suurin osa robotiikasta on suunniteltu teollisen tuotannon tarpeisiin (Suomen Robotiikkayhdistys 1999, 4). Kuitenkin nykypäivänä robotiikka tulee enemmän ja enemmän kohdistumaan myös henkilökohtaisiin robotteihin (Benitti 2012, 978).

Robottiikan ala kasvaa nopeaa vauhtia ja markkinoille tulee koko ajan uusia robotiikan tuotteita. Robotiikalla on paljon tarjottavaa koulutuksenkin alalle, mutta on vähän tutkimusta siitä, onko robotiikan opiskelusta tai hyödyntämisestä koulussa ollut hyötyä tulevaisuuden taitojen oppimiselle. (Benitti 2012, 978.) Benitti (2012) kuitenkin tutki, että robottien käytössä opetuksessa on hyötyä teknologian, tekniikan ja matematiikan aihealueiden oppimisessa. Itseohjautuvuus oppimisessa oli kehittynyt päästessä kokeilemaan roboteilla ohjelmointia. Erilaisia taitoja, kuten ajattelu, ongelmanratkaisu ja sosiaaliset taidot, ei kuitenkaan pystytty osoittamaan opituksi robottien avulla, koska tuloksissa oli puolesta ja vastaan vastauksia. (Benitti 2012, 986.)

Peruskoulussa usein robotiikka keskittyy robottien ohjelmointiin ja niiden kasaamiseen. Usein robotiikka kuvastaa tällaisessa opetuksessa passiivista työkalua tai lopputuotoksen havainnollistamista. Oppilaiden tulee siis saada robotti toimimaan tavalla, jota siltä toivotaan. Tämä lähestymistapa robotiikkaan peruskoulussa on aika kapea verrattuna koko robotiikan alaan. Useampi oppilas voisi kiinnostua robotiikan eri lähestymistavoista, jos robotiikan opettamisessa otettaisiin laajemmin huomioon kaikki robotiikan kenttään liittyvät asiat. Robotiikkaa voisi lähestyä esimerkiksi tarinankerronnallisesti tai interaktiivisena taideteoksena. (Benitti 2012, 978-979.) Tällöin robotiikkaa hyödynnettäisiin ilmaisussa ja osana isompaa kokonaisuutta eikä pelkkänä toimintasarjana.

2.5 Algoritminen ajattelu

Usein ohjelmointia lähestytään harjoittelemalla laitteiden käyttöä ja opettelemalla erilaisia tekniikoita. Opetellessa saatetaan keskittyä vain yhteen ohjelmointikieleen ja sen toimintoihin. Jotta ohjelmoinnista saadaan kaikki irti, on ohjelmoinnin oltava joustavaa. Tähän päästään, kun ymmärretään ohjelmoinnin käsitteitä ja ideologiaa niin, että niitä voidaan hyödyntää eri konsepteissa (Van-Roy & Haridi 2004, xvi). Laitteiden käytön harjoittelun lisäksi tulisikin siis myös harjoittaa ajattelua, joka tukee ohjelmoinnin oppimista ja hyödyntämistä. Tähän liittyy useita termejä, kuten algoritminen ajattelu,

laskennallinen ajattelu ja ohjelmoinnillinen ajattelu. Teknologian kehittymisen ja muuttuvan luonteen takia myös siihen liittyvät termit ovat hyvin muuttuvaisia. Myös koulumaailmassa teknologiaan liittyvät aihealueet ja termit ovat hyvin tuoreita ja muutokselle alttiita, ja englanninkielisen aineiston kohdalla on vielä käännöstyön aiheuttamat ongelmat. Tästä johtuen tässä tutkimuksessa käytetään englanninkielen termejä *computational thinking* ja *algorithmic thinking* synonyymeinä ja käytämme käsitettä algoritmisen ajattelu, koska se on tutuin alakoulun ympäristössä.

Algoritmisen ajattelu tarkoittaa kykyä ymmärtää kuinka tietokonetta pystyy käskemään tekemään eri asioita sekä ymmärtää kuinka tietotekniikka toimii (Koodiaapinen 2015). Tarkemmin sanottuna algoritmisen ajattelun hallinta tarkoittaa kykyä analysoida ja tarkasti määrittää ongelmaa, löytää ongelmasta alkeellisia perusosia, joista voi ryhtyä muodostamaan algoritmia, ongelman normaalien ja erityisien tilanteiden huomioimista ja taitoa kehittää algoritmia tehokkaammaksi (Futschek 2006, 160). Algoritmisen ajattelu siis ohjaa ongelman osiin purkamista ja siihen kuuluu tapauskohtaisesti tiettyjen toistuvien sääntöjen hakemista ja yksiselitteisien käskyjen muodostamista, eli algoritmin luomista (Edu.fi, 2016). Algoritmista ajattelua, niin kuin esimerkiksi luovaa ajattelua, voi olla hankala lähteä opettamaan yksinkertaisin keinoin. Algoritmisen ajattelun harjoittamisessa ongelmien ratkonta on tehokkainta. Ongelmat täytyvät olla helppoja ymmärtää, mutta tarpeeksi haastavia, jotta oppijoilla on liikkumavaraa ja tilaa luovuudelle. Ohjelmointikieltä ei tarvita alussa, vaan ongelmia lähdetään ratkomaan visualisointia ja toiminnallisuutta hyväksi käyttäen. (Futschek 2006, 160.) Ilman tietoteknistä näkökulmaa algoritmista ajattelua voi kehittää erilaisten säännönmukaisten ohjeiden kautta. Esimerkiksi virkkausohjeiden, nuottien ja ajo-ohjeiden tekeminen ja noudattaminen edellyttävät algoritmista ajattelua. (Edu.fi, 2016.) Joitakin tutkijoita on kiinnostanut tuoda algoritmisen ajattelu pois tietokoneilta, kuten ryhmää tutkijoita Uudessa-Seelannissa. He ovat kehitelleet *the computer science unplugged* aktiviteetteja, joissa harjoitellaan ohjelmointia peruskouluissa käyttämällä tietokoneiden sijaan esimerkiksi liikkumista tai kortteja. (Burke &

Kafai 2014, 8.) Kun oppilaat harjoittavat algoritmista ajatteluaan esimerkiksi pelien luomisessa, tulee siitä heille merkityksellisempää. Luodessaan peliä oppilaat suunnittelevat järjestelmää ja ratkaisevat kuinka saada omat ajatukset siihen muotoon, että tietokone ymmärtää ne. (Burke & Kafai 2014, 8.) Oppilaiden tulee siis ymmärtää tietokoneen ja ihmisen toiminnan eroavaisuuksia.

Algoritmisen ajattelun lähikäsitteenä tunnetaan myös ohjelmoinnillinen ajattelu sekä automatisointiajattelua, joissa eri yhteyksissä on hieman eri vivahteita ja painotuksia. Wing (2008, 2720) esittelee, että ohjelmoinnillisen ajattelun, *computational thinking*, harjoittamisen tulisi aloittaa mahdollisimman varhain, johtuen siitä, että tätä taitoa tarvitaan ympäröivässä maailmassa. Useilla eri tieteenaloilla tätä ajattelua tarvitaan myös korkeamman asteen koulutuksen oppilaitoksissa sekä tulevaisuuden työelämässä. Tällöin peruskoulun tulee myös tukea tämän ajattelutaidon kehittymistä. Lapset törmäävät ohjelmoinnillisen ajattelun käsitteistöön, kuten loputtomuuden käsite ja toisto, joten näitä on hyvä myös sanallistaa. (Wing 2008, 2720-2721.)

Ohjelmoinnillisessa ajattelussa liikutaan paljon abstraktilla tasolla, ja pyritään luomaan yleiskäsitteitä ongelmasta pelkistämällä sitä. Kun ongelma ollaan saatu abstrahoimalla jäsenneiltyä, voidaan miettiä, miten työnjako olisi parasta toteuttaa ihmisen ja tietokoneen välillä. Ohjelmoinnillisessa ajattelussa tulisi huomioida, että ihmisen ajattelu on erilaista kuin tietokoneen. Yleensä ohjelmointiprosessissa ihminen suorittaa työnteosta luovan, tulkinnallisen osan sekä ohjeistuksen. Tietokone puolestaan suorittaa tiedon massaprosessointia sekä automaationa tapahtuvaa toistoa. (Wing 2008, 3717-3720.) Michaelson kuvaa ohjelmoinnillisen ajattelun rakentuvan ongelman osiin hajottamisesta, abstrahoinnista, toistuvista malleista ja algoritmeista. Tällaiseen ajatteluun ei ole valmista reseptiä, vaan on otettava huomioon rönsyily, päällekkäisyys ja osien vuorovaikutus toisiinsa. Ongelman ratkaisuun päästäkseen tarvitaan ohjelmoinnin lisäksi tarpeeksi informaatiota ja tietoa aiheesta. (Michaelson 2015, 63.)

Tedre & Denning (2016) muistuttavat, että ohjelmoinnillisen ajattelun käytössä tulisi ottaa huomioon muutamia asioita. Nyt, kun ohjelmointi on osa

uutta opetussuunnitelmaa, olisi hyvä huomioida, ettei se ole uusi ilmiö. Samanlaista ajattelua ollaan kehitelty tieteellisessä keskustelussa ja kokeiltu kouluissakin jo vuosikymmeniä sitten. Näistä saatuja oppeja ei tulisi unohtaa. Tedre & Denning (2016) muistuttavat myös, että ohjelmoinnillinen ajattelu on vain yksi tapa ajatella, eikä välttämättä se paras. Myöskään pelkkä tieto ei takaa oppimista, joten käytännön oppimisen mahdollistaminen ohjelmoinnissa on tärkeää. (Tedre & Denning 2016, 125.)

Nedre & Denning (2016) nostavat esille myös sen, että ohjelmoinnillisen ajattelun hyötyä ongelmanratkaisussa on tutkittu hyvin vähän. Tämän ajattelutavan hyötyjä tulisi siis miettiä maltillisesti. Ohjelmointia on kokeiltu kouluissa aikaisemminkin, mutta kokeilusta luovuttiin. Ohjelmointi kouluissa on hieman erilaista kuin ammattilaisten parissa, mutta ohjelmoinnillisen ajattelun tärkeimpiä asioita, kuten iteraatiota, ei silti tulisi sivuuttaa. Tulisi myös muistaa, että ohjelmointi on enemmänkin suunnittelua, kuin vain käskysarjojen antamista. (Tedre & Denning 2016, 126.)

3 OHJELMOINTI PERUSKOULUSSA

3.1 Ohjelmointi peruskoulun opetussuunnitelmassa

Valtakunnallisessa opetussuunnitelmassa ohjelmointi on sisällytetty laaja-alaisiin osaamistavoitteisiin tieto- ja viestintäteknologian (L5) alle. "*Laaja-alaisella osaamisella tarkoitetaan tietojen, taitojen, arvojen, asenteiden ja tahdon muodostamaa kokonaisuutta. Osaaminen tarkoittaa myös kykyä käyttää tietoja ja taitoja tilanteen edellyttämällä tavalla*" (POPS 2014, 20). Näin ollen ohjelmointi ei ole oma oppiaineensa, vaan se sisällytetään osaksi muita oppiaineita. Lisäksi ohjelmointia on sisällytetty matematiikan sekä teknisen työn osa-alueisiin, joten usein ohjelmointiin käytettävät tunnit ovat nimenomaan matematiikan tai teknisen työn tunteja. Seuraavaksi esitetään mitä ohjelmoinnin osa-alueita opetussuunnitelma sisällyttää eri vuosiluokille peruskoulussa. Lisäksi tarkastellaan, kuinka ohjelmoinnin opetus eroaa opetussuunnitelman tasolla muista maista ja tarkempaan käsittelyyn on valittu Englannin ohjelmoinnin opetussuunnitelma.

3.2 Ohjelmointi valtakunnallisessa opetussuunnitelmassa

Peruskoulun ensimmäisillä luokilla ohjelmointi on hyvin toiminnallisella tasolla, jolloin oppilaat pääsevät itse liikkumaan ja toimimaan tietokoneina. Opetussuunnitelma esittää, että 1-2 luokilla tutustutaan ohjelmoinnin alkeisiin vaiheittaisia toimintaohjeita laatimalla, ja niitä kokeilemalla. (POPS 2014, 129.) Tämä tarkoittaa käytännössä sitä, että oppilas antaa yksiselitteisiä ja tarkkoja käskyjä toiselle oppilaalle tai esimerkiksi opettajalle, joka pyrkii noudattamaan saamansa käskyt mahdollisimman tarkasti. Luokilla 1-2 on tavoitteena herätellä ohjelmoinnin perusajatusta leikkien ja pelien avulla. Ensimmäisinä kouluvuosina ohjelmoinnin pääpaino ei siis ole vielä tietokoneissa, mutta tällekin ikäryhmälle löytyy sopivia sovelluksia ja verkkosivustoja (Liukas & Mykkänen 2016, 86).

Luokilla 3-6 perehdytään tarkemmin tekemiseen, joka on lähempänä ohjelmointia. Voimassa olevan opetussuunnitelman (2014, 157) mukaan näillä luokilla rohkaistaan oppilaita kokeilemaan eri tieto- ja viestintäteknologian osa-alueita, yksin ja yhdessä toisten oppilaiden kanssa. Oppilas saa ohjelmoinnin avulla kokemuksen siitä, kuinka riippuvainen teknologia on ihmisten tekemistä valinnoista, eikä tietokone itsessään ilman käskyjä osaa tehdä mitään. Varsinaista ohjelmoinnin tekstikieltä ei 3-6 luokilla oteta vielä käyttöön, vaan ohjelmointia harjoitellaan erilaisten visuaalisten ohjelmointiympäristöjen avulla. (POPS 2014, 235.) Kuudennen vuosiluokan päätteeksi matematiikan osalta ohjelmoinnille on annettu osaamistavoite arvosanan kahdeksan saamiseksi. Tämä tavoite (T14) on osata ohjelmoida toimiva ohjelma graafisessa ohjelmointiympäristössä. (POPS 2014, 239.)

Luokilla 7-9 oppilaita rohkaistaan oma-aloitteiseen toimintaan tieto- ja viestintäteknologian suhteen (POPS 2014, 284). Matematiikan opetuksen tavoitteen T20 mukaan tavoitteena on ohjata oppilaita syventämään omaa algoritmista ajattelua sekä soveltamaan ohjelmointia erilaisissa ongelmanratkaisutilanteissa (POPS 2014, 375). Ohjelmointia harjoitetaan edelleen osana eri oppiaineita, mutta useissa kouluissa on mahdollista ottaa tietotekniikan valinnaisaine, jossa ohjelmoinnillisia taitoja on mahdollista kehittää. Yläkoulussa voidaan ottaa käyttöön jo jokin yksinkertainen ohjelmointikieli, kuten Python, JavaScript tai Ruby (Liukas & Mykkänen 2016, 100).

3.3 Ohjelmointi Englannin opetussuunnitelmassa verrattuna Suomen opetussuunnitelmaan

Ohjelmointi otettiin osaksi opetusta ennen Suomea esimerkiksi Virossa, Englannissa ja Australiassa. Vuonna 2012 Virossa aloitettiin opetusministeriön rahoittama ohjelmoinnin projekti, joka kattaa koko maan. Suomen vähäinen teknologian hyödyntäminen opetuksessa ennen vuoden 2014 opetussuunnitelman voimaantuloa oli huolestuttavaa, joten asialle oli tehtävä

jotain. (Malmberg 2013.) Samaa mietittiin myös Englannissa ennen opetussuunnitelman muutosta vuonna 2013.

Englannin opetussuunnitelmassa tietojenkäsittely, computing, on oma oppiaineensa ja sitä opiskellaan koko peruskoulu-aika, viisivuotiaista 16-vuotiaiksi (Department of education 2013, 7). Lähtökohdiltaan Englannissa ohjelmoinnin opetus eroaa siis merkittävästi Suomen ohjelmoinnin opetuksesta, jossa tieto- ja viestintäteknologia ei ole oma oppiaineensa, vaan kulkee mukana muiden oppiaineiden kanssa.

Englannissa tietojenkäsittelyn tavoitteena on opiskella algoritmista ajattelua, computational thinking, ja luovaa lähestymistapaa maailman ymmärtämiseen ja muuttamiseen. Oppilaat tutustuvat tieto- ja viestintäteknologiaan, digitaalisten järjestelmien toimintaan ja ohjelmointiin. Oppilaille opetetaan myös digilukutaitoa arjessa pärjäämiseen ja varmistetaan perustaidot työelämään tieto- ja viestintäteknologia-aitojen osalta. (Department of Education 2013, 178.) Sisällöllisesti tämä oppiaine kuitenkin sisältää paljon samoja elementtejä, kuin Suomen opetussuunnitelmassa tieto- ja viestintäteknologia. Esimerkiksi algoritmista ajattelua ja luovuuden kehittymistä korostetaan molemmissa opetussuunnitelmissa.

Englannin koulujärjestelmän ensimmäisellä tasolla, 5-7-vuotiaiden oppilaiden tulisi ymmärtää mitä ovat algoritmit, luoda ja testata yksinkertaisia ohjelmia, ennakoida loogisesti ohjelmien toimintaa, järjestellä tietoa digitaalisesti, tunnistaa viestintäteknologian käyttötilanteita koulun ulkopuolella ja käyttää teknologiaa turvallisesti ja nettietiketin mukaan. Vastaavasti Suomen opetussuunnitelmassa 1-2 luokilla ohjelmointiin tutustutaan hyvin toiminnallisesti. Näillä luokilla herätellään algoritmista ajattelua vaiheittaisten käskyjen tarkoilla antamisella ja niiden noudattamisella. Tavoitteissa ilmenee siis merkittävä tasoero Englannin ja Suomen opetussuunnitelmien välillä.

Englannin opetussuunnitelmassa toisella tasolla, 7-11-vuotiaiden oppilaiden tulisi osata ohjelmoida tarkoituksenmukaisesti, käyttää ohjelmoinnin perustoimintoja, selittää loogisen järjestyksen avulla algoritmien ja ohjelmien

toimintaa, ymmärtää tietokoneen tietoverkkoja, kuten internetiä, osata hyödyntää tiedonetsinnän taitoja tehokkaasti, soveltaa ja integroida tietotekniikan eri osa-alueita toisiinsa ja käyttää teknologiaa turvallisesti ja nettietiketin mukaan. (Department of Education 2013, 179.) Suomessa taas luokilla 3-6 pääpainona on rohkaista oppilaita toteuttamaan tieto- ja viestintäteknologian avulla ideoitaan sekä itsenäisesti, että ryhmässä. Näillä luokilla toiminnallisuus väistyy sivummalle ja oppilaat suunnittelevat ja toteuttavat ohjelmia graafisessa ohjelmointiympäristössä. Luokilla 3-6 on edelleen tärkeää oppilaan kannustaminen ja kiinnostuksen viriäminen ohjelmointia kohtaan. Suomen opetussuunnitelmassa oppilaiden kannustaminen ohjelmointiin on siis merkittävämmässä asemassa kuin Englannin opetussuunnitelmassa, jossa korostetaan enemmän opittavia taitoja ja tavoitteita.

Niin kuin yllä ilmenee, Englannin opetussuunnitelmassa tietojenkäsittelyn tavoitteet ovat tarkemmin laadittuja ja menevät pidemmälle kuin Suomen kouluissa. Tätä selittää jo se, että Englannissa tunteja on varattu huomattavasti enemmän pelkän tietojenkäsittelyn opettamiseen, koska se on oma oppiaineensa.

3.4 Miten ohjelmoinnin opetus alakoulussa on perusteltu?

Perusopetuksen opetussuunnitelman (POPS 2014) perusteet julkaistiin vuonna 2014 ja se otettiin käyttöön alakouluissa 1.8.2016. Uuden opetussuunnitelman myötä ohjelmoinnin opetus otettiin osaksi perusopetusta. Ohjelmoinnin tavoitteet alakoulussa ovat kuitenkin ohjelmoinnin alkeissa, eikä pitkälle meneviä ohjelmoinnin sanastoja ja menetelmiä välttämättä tarvitse käyttää alakoulu aikana. Valtakunnallisen opetussuunnitelman (POPS 2014, 20) mukaan ohjelmoinnin opetus on laaja-alaisesti muuta opetusta tukevaa ja eheyttää eri aiheita yhteen. Ohjelmoinnin opiskelun alkuvaiheessa tärkeämpää on se, mitä ohjelmoinnin kautta voidaan oppia, kuin se, että opitaan ohjelmoimaan (Luostarinen & Peltomaa 2016, 102).

Aikaisemman opetussuunnitelman pohjalta oli haasteellista saada selville yhteistä linjaa siitä, kuinka paljon silloin kutsuttua tietotekniikkaa tulisi opetuksessa opettaa. Uusi opetussuunnitelma mahdollisesti tasoittaa koulujen tilannetta asettamalla tarkempia tavoitteita. (Opetushallitus 2009, 16.) Ohjelmoinnin lisäämistä opetussuunnitelmaan on perusteltu monilta eri tahoilta ja se on nostanut esiin paljon keskustelua. Tätä on perusteltu esimerkiksi opetusministerin toimesta muun muassa nuorten tieto- ja viestintäteknologian taitojen parantamisella. Lisäksi ohjelmoinnin opetus peruskoulussa on keino turvata suomalaisten vahva tietotekniikan osaaminen myös jatkossa. (Rydman 2014.) Uuden opetussuunnitelman luomisvaiheessa pidetyssä asiantuntijaseminaarissa nostettiin myös esille palveluyhteiskunnan työelämän tarpeet, jatko-opiskelu alalla ja matemaattisen ajattelun yleinen kehittäminen ohjelmoinnin opetuksen avulla (Opetushallitus 2010, 6). Suomessa arvostetaan laadukasta perusopetusta; siihen panostetaan paljon ja sitä pyritään kehittämään jatkuvasti. Nimenomaan korkeatasoinen perusopetus on yksi Suomen ylpeydenaihe kansainvälisellä tasolla. (Ekonoja 2014, 26.) Tieto- ja viestintäteknologian osaamisen tarve on laajasti tiedossa, joten näiden taitojen kehittäminen on yksi tärkeä tavoite myös perusopetuksessa.

Niin kuin aiemmin on jo todettu, tieto- ja viestintäteknologian käyttö yleistyy jatkuvasti osana opetusta. Norrenan (2015, 70) mukaan teknologian hyödyntäminen on kuitenkin edelleen melko yksipuolista, eikä se tuo sinällään pedagogista lisäarvoa opetukselle. Norrena (2015, 70) esittää, että tieto- ja viestintäteknologia tuo pedagogista lisäarvoa opetukselle silloin, kun oppilas itse tuottaa oppimistehtäviä sen avulla. Tällöin tieto- ja viestintäteknologian yksipuolinen käyttö tarkoittaa sitä, että opettaja edestä näyttää jotakin tieto- ja viestintäteknologiaa hyödyntäen ja oppilaat ovat ulkopuolisia seuraajia. Näin ollen ohjelmoinnilla on suuri potentiaali saada oppilaat aidosti mukaan uudistuvan teknologian käyttöön. Norrenan (2015, 234) mukaan tieto- ja viestintäteknologian hyödyntäminen voidaan jakaa perustaitoja kehittävään hyödyntämiseen sekä tiedon rakentamista tukevaan hyödyntämiseen. Tavoitteena on, että teknologian hyödyntämisen tarkoitusperä suuntautuisi

ennemmin tiedon rakentamiseen ja ongelmanratkaisuun, eikä vain perustaitojen kehittämiseen (Norrena 2015, 235).

Ohjelmointi kehittää monia tärkeitä taitoja, kuten loogista päättelykykyä, ongelmanratkaisutaitoja sekä luovuutta (Koodiaapinen 2017). Tästä johtuen tieto- ja viestintäteknologian taitojen kehittämistä ohjelmoinnin avulla voidaan syventää entisestään. Majaranta (2002, 173) esittää, että olennaista ohjelmoinnin opetuksessa on nimenomaan ohjelmointitehtävien mahdollisuus kehittää oppilaiden ongelmanratkaisutaitoja sekä loogista ja matemaattista ajattelua.

Mykkänen ja Liukas (2016, 53) esittävät, että ohjelmoijista on suuri työpula Suomessa. Osaavia tietotekniikan asiantuntijoita on harvassa, joten alan tietoisuuden lisääminen jo alakoulussa on merkittävää tulevaisuuden kannalta. Oppilailla on oikeus oppia ohjelmointia ja ymmärtää sen peruseräitä. Tämä on etenkin tyttöjen osalta tärkeää, sillä tietotekniikan alalla vain noin 23% on naisia. (Mykkänen & Liukas 2016, 53.)

Luostarinen ja Peltomaa (2016, 102) perustelevat ohjelmoinnin lisäämistä osaksi opetussuunnitelmaa nykyajan vaatimuksilla. Kaikkiällä ympäröivässä yhteiskunnassa on erilaisia ohjelmoituja laitteita ja sovelluksia, mutta harva kuitenkaan ymmärtää näiden toiminnan peruseräitä. Luostarinen ja Peltomaa (2016, 102) esittävät, että tärkeämpää on kehittää ohjelmoinnillista ajattelua ja ymmärtää sen peruseräitä, kuin osata kirjoittaa jotakin koodikieltä.

Tieto- ja viestintäteknologian opetuskäyttöä on myös kritisoitu, jonka alle opetussuunnitelman mukaan myös ohjelmointi kuuluu. Sitä on puollettu tulevaisuuden taitojen kannalta ja usein se rinnastetaan tulevaisuuden työllistymisen ehdoksi. Tulevaisuuden teknologiaa ei kuitenkaan voida ennustaa, eikä tiedetä, ovatko tuolloin ohjelmointi tai tietokoneet tulevaisuuden teknologiaa. Lisäksi teknologian suoranaista yhteyttä oppimisen kehitykselle ei ole todettu. (Norrena, 2013, 55.) On varmaa, että teknologia kehittyy jatkuvasti ja myös koulussa on hyvä pysyä sen perässä. Tämä tarkoittaa jatkuvaa kehittämistä ja muutosta myös kouluissa.

Tainan (2015) mielestä ohjelmoinnin opetuksessa ensisijaisen tärkeää on opettajien motivaatio opettaa ohjelmointia. Jos opettajaa ei kiinnosta opettaa ohjelmointia, eikä hän halua itse oppia sitä, opettaja saattaa ahdistua tilanteesta ja oppilaat turhautua. Tällöin ohjelmoinnista voi tulla välttämätön paha, joka vie vain aikaa muiden oppiaineiden opetuksesta. Taina kertoo ymmärtävänsä, miksi ohjelmointia on järkevää opettaa peruskoulussa, mutta hän on myös huolissaan siitä, että kouluopetuksessa ei saavuteta selkeää teoreettista pohjaa, jolloin asia saattaa jäädä palkkien mekaaniseksi yhdistelyksi sekä irrallisten tietojen opetteluksi.

Norrenan (2013, 56) mukaan koulun vaikutusvalta rajoittuu siihen, että se saa kannustettua opettajia käyttämään opetusteknologiaa. Opettaja itse saa päättää opetuksestaan ja sen sisällöistä, kunhan opetus on opetussuunnitelman raamien sisällä. Nimenomaan opettajan asenne ja mielenkiinto ovat avainasemassa tieto- ja viestintäteknologian ja näin ollen myös ohjelmoinnin vakiinnuttamiseksi osaksi opetusta. (Norrena, 2013, 56.)

4 AIEMPAA TUTKIMUSTA OHJELMOINNIN OPETUKSESTA

4.1 Tutkimusta Suomessa

Ohjelmoinnin opettamista alakoulussa on tutkittu Suomessa vielä melko vähän, sillä aihe on maailmanlaajuisestikin vielä melko tuore. Kehittyvän teknologian hyödyntämistä opetuksessa on kuitenkin tutkittu ja näissä tutkimuksissa on viitteitä myös ohjelmoinnista. Digiajan peruskoulu -hanke selvitti digitalisaation tämänhetkistä tilannetta peruskouluissa ja aineistossa tulee kuuluviin niin rehtorien, opettajien, kuin oppilaiden näkemys perusopetuksen digitalisaatiosta. (Kaarakainen ym. 2017.) Tutkimuksessa kävi ilmi, että kouluissa on tehty digistrategioita, mutta niiden toteutuminen ja kehittäminen ovat vielä työn alla. Opettajien ja rehtoreiden näkemykset koulun yhteisistä tieto- ja viestintäteknologian hyödyntämisen tavoitteista erosivat merkittävästi ja rehtorit ajattelivat digitalisaation edenneen pidemmälle kouluissa kuin opettajat. (Kaarakainen ym. 2017, 16-19.)

Tutkimuksen mukaan opettajat ovat suhteellisen tyytyväisiä koulun tarjoamien laitteiden, ohjelmien ja verkkoyhteyden yleistoimivuuteen. Lisäksi opettajien omat yleiset digitaidot ovat kehittyneet ja järjestettäviin täydennyskoulutuksiin on oltu melko tyytyväisiä. (Kaarakainen ym. 2017, 21.) Tutkimuksessa huomautetaan, että monien eri selvitysten mukaan oppilaat ovat tottuneet käyttämään monipuolisesti erilaisia digitaalisia ympäristöjä vapaa-ajallaan. (Kaarakainen ym. 2017, 22.) Lähtökohdat digitalisaation hyödyntämiseen opetuksessa ovat siis melko hyvät, mutta tutkimuksessa saatiin kuitenkin selville, että näitä oppilaan digitaalisia taitoja ei vielä hyödynnetä parhaalla mahdollisella tavalla, vaan teknologian hyödyntäminen opetuksessa rajoittuu usein opettajalähtöisiin tilanteisiin. Tällöin oppilaat eivät itse pääse testaamaan ja omaksumaan digitalisaatiota yhtenä oppimistyökaluna.

Tutkimuksemme kannalta tärkeimpiä tuloksia kyseisestä tutkimuksesta nousi esiin opettajien ohjelmointitaidoista sekä sen opettamisen määrästä.

Opettajien ohjelmointitaidot ovat heikot ja vain noin 20% opettajista on kokeillut ohjelmointia oppilaiden kanssa. Myös oppilaiden ohjelmointitaidot olivat melko heikot, mutta moni oppilaista oli kuitenkin kokeillut koodaamista tai robottien ohjaamista. (Karakainen, ym. 2017, 39.) Ohjelmointi on lisätty osaksi opetusta uuden opetussuunnitelman myötä porrastetusti eri ikäluokittain, joka tuli ilmi myös tutkimustuloksissa: 2. luokan oppilaat olivat kokeilleet ohjelmointia huomattavasti enemmän kuin 8. luokan oppilaat. On otettava kuitenkin huomioon, että voimassa oleva opetussuunnitelma oli aineistonkeruuvaiheessa ollut voimassa vasta noin puolivuotta. Kyseisessä tutkimuksessa ei selvitetty opettajien motivaatiota tai yleistä suhtautumista ohjelmoinnin opettamista kohtaan, jota tutkimuksessamme selvitetään.

Niin kuin aiemmin mainittiin, ohjelmoinnin opetuksesta on Suomessa tehty hyvin vähän tutkimusta. Aiheesta löytyy kuitenkin pro-gradu tutkielmia, kuten Oulun yliopistolla määrällinen pro-gradu tutkielma opettajien asenteita ohjelmoinnin opetusta kohtaan. Aineisto kerättiin vuonna 2015 syksyllä verkkokyselynä. Tällöin uusi opetussuunnitelma ei ollut vielä rantautunut kouluihin, mutta perusopetuksen opetussuunnitelman perusteet oltiin julkaistu ja opettajilla oli tiedossa, että ohjelmointi on lisätty opetussuunnitelmaan. Osallistujina ohjelmointikurssille oli vapaaehtoisesti osallistuvat opettajat. (Karvonen & Laukka 2016, 23, 49.) Osa kysytyistä ohjelmointikielistä olivat suurelle osalle vastaajista tuttuja, mutta opettajat eivät uskoneet pystyvänsä opettamaan niitä. Scratchia opetuksessaan oli käyttänyt viimeisen kolmen vuoden aikana 21% opettajista. Opettajien itsevarmuus ongelmien ratkomisessa oli hyvä, varsinkin jos heillä oli tarpeeksi aikaa ongelman ratkomiseen tai jos he saisivat tukea. (Karvonen & Laukka 2016, 44-46.)

Samassa tutkimuksessa kysyttiin opettajien näkemyksiä ohjelmoinnin roolista perusopetuksessa. Suurin osa oli jollakin tasolla samaa mieltä siitä, että ohjelmoinnin opetusta tarvitaan peruskoulussa. Kysyttäessä työpaikalla vallitsevasta ilmapiiristä, useampi vastasi sen olevan positiivinen, mutta kolmasosa ei pystynyt vastaamaan kysymykseen. (Karvonen & Laukka 2016, 47.) Opettajat eivät ajatelleet ohjelmoinnista olevan vahvasti hyötyä heidän

ammattitaidolleen, mutta mielsivät ohjelmoinnin opettamisen omaksumisen enemmän helpoksi kuin hankalaksi. Yleinen suhtautuminen ohjelmointia kohtaan oli positiivinen, eikä se aiheuttanut opettajille suurta ahdistusta. On huomioitava, että tutkimukseen osallistujat saattoivat edustaa teknologiamyönteisiä opettajia tutkimusasetelman myötä. (Karvonen & Laukka 2016, 49-50.)

Keski-Suomessa tehdyssä pro-gradu tutkielmassa tutkittiin uuden opetussuunnitelman tieto- ja viestintäteknologian tavoitteiden toteutumista kouluissa. Tutkimuksessa haastateltiin rehtoreita ja havainnoitiin kouluissa tapahtuvaa opetusta. Ohjelmoinnin opetus kouluissa vaihteli yhden oppiaineen osa-alueista koko koulun laajuisiin projekteihin. (Marttala 2017, 78.) Suurin osa opetuksesta tapahtui kuitenkin matematiikan tai tietotekniikan oppiaineen sisällä. Myöskään yhteistä opetuksen suunnitelmaa ei löytynyt kahdella kolmasosasta kouluista. Ohjelmoinnin opetukseen käytettävä aika vaihteli 1-6 vuosiluokilla kahdesta tunnista kahteenkymmeneen, lisääntyen yleensä iän myötä. (Marttala 2017, 58-60.) Tutkimukseen osallistuneiden koulujen suosituimpia ohjelmistoja olivat Koodaustunti ja Scratch tai Scratch junior. Myös ohjelmoitavat robotit olivat olleet käytössä. (Marttala, 62.) Ohjelmoinnin opetus miellettiin haastavaksi ja jotkut rehtorit totesivat, että ohjelmoinnin opetus oli heikolla tasolla. Yhdistäminen muihin oppiaineisiin tuntui haastavalta. Valmiit opetusmateriaalit olivat laajasti käytössä ja ne toivat tarpeellista tukea opettajille, mutta samalla pitivät ohjelmoinnin erillään muista aiheista. (Marttala, 78.) Oppilaiden tasoerot tulivat selkeästi esiin ohjelmointia opettaessa. Itsenäinen työskentely, opettajan taitotaso ja oppilaiden jo vapaa-ajalla hankittu ennakkotieto aiheutti sen, että taitavat oppilaat suoriutuivat tehtävistä liian nopeasti, kun taas heikommat eivät päässeet alkua pidemmälle. (Marttala, 73.)

4.2 Esimerkkejä kansainvälisistä tutkimuksista ohjelmoinnin opetuksesta

Kansainvälistä tutkimusta ohjelmoinnin opetuksesta ja ohjelmoinnin oppimisesta löytyy melko kattavasti. Seuraavaksi esitellään esimerkkejä kansainvälisistä tutkimuksista.

4.2.1 Algoritminen ajattelu arjen ongelmissa

Valtakunnallisessa opetussuunnitelmassa ohjelmoinnin osalta korostetaan algoritmisen ajattelun kehittymistä. Standl (2017) on tutkinut, kuinka algoritmista ajattelua voisi yhdistää arkipäivän ongelmiin koulussa. Hän muodosti ongelmanratkaisuun ohjelmoinnin perusteella viisi vaihetta: ongelman ymmärrys, käsitteellistäminen, osiin hajottaminen, suunnittelu ja testaus. Oppilailta tämän ajattelumallin toteutuminen vaatii itsevarmuutta, sitkeyttä, monitulkinnallisuuden ja avoimeksi jäävien ongelmien sietoa sekä vuorovaikutustaitoja. Tutkimuksessa tutkittiin kahta yläluokkaa Yhdysvalloissa. (Standl 2017, 183-184.) Tutkimustuloksissa selvisi, että oppilaat onnistuivat muodostamaan ratkaisuja ongelmiin tällä tavalla, mutta heillä oli vaikeuksia abstrahoinnin, osiin jaottelun ja arvioinnin kanssa. Käsitteellistämistä ja osiin jaottelua käytettiin osana esimerkiksi ongelman ymmärtämistä ja suunnittelua, mutta käsitteenä ne olivat oppilaille vaikea ymmärtää osana ongelmanratkaisuprosessia. Tämän takia ongelmanratkaisuvaiheita ehdotettiin muutettavaksi niin, että käsitteellistäminen ja osiin jaottelu yhdistettiin jo ensimmäiseen, ymmärryksen vaiheeseen. (Standl 2017, 187-188.) Oppilaiden asenteita tämän tekniikan käyttöön tiedusteltiin myös. Tutkimuksessa selvisi, että oppilaiden asenne oli muuttunut hieman myönteisemmäksi ja he ymmärsivät tätä prosessia paremmin, mutta he eivät luultavastikaan ottaisi sitä osaksi arkeaan. (Standl 2017, 190.)

4.2.2 Ohjelmointi ja ongelmanratkaisu

Niin kuin aiemmin on mainittu, ohjelmoinnilla nähdään olevan positiivinen vaikutus oppilaan ongelmanratkaisutaitoihin. Kalelioglu (2015, 200) on tutkinut, opettaako ohjelmointi reflektiivistä ajattelua ongelmanratkaisun yhteydessä. Yksi kriittisen ajattelun taidoista on oman ajattelun reflektointi, analyysi ja arviointi. Nämä ilmenevät ongelmanratkaisuprosessissa, esimerkiksi kyselyn, järkeilyn ja arvioinnin muodossa. Kalelioglu (2015, 201) näkee koodin kirjoittamisen eräänlaisena ongelmanratkaisuprosessina. Tämän ajatuksen jakaa Heinonen (2008, 3-4) joka vertaa ohjelmointiprosessin vaiheita ongelmanratkaisuprosessiin; kummassakin ensin pyritään ymmärtämään ongelma, sitten suunnittelemaan, sen jälkeen toteuttamaan, jonka jälkeen prosessin toimivuutta arvioidaan.

Kalelioglu toteutti tutkimuksen kymmenvuotiaille oppilaille, jotka osallistuivat opettajansa johdolla Code.org -ohjelmointisivustolla löytyvään 20 tunnin ohjelmointikurssiin. Tutkimuksessa tultiin siihen tulokseen, että oppilaiden reflektiivinen ajattelu/ongelmanratkaisutaidot eivät kehittyneet merkityksellisesti. Oppilaiden motivaatio ja mielenkiinto ohjelmointia kohtaan kasvoi kuitenkin paljon ja he tiedostivat sen hyödyn esimerkiksi matemaattisessa ja geometrisen ajattelussa. Vaikka tutkimuksessa ei havaittu näiden ajattelun taitojen kehittyneen, oppilaiden motivaation kasvattamisella saattaa olla isompi merkitys alakoululaisten ohjelmoinnin oppimiselle. Suomen valtakunnallisessa opetussuunnitelmassa yksi ohjelmoinnin opetustavoitteista onkin tutustuttaa oppilaita ohjelmoinnin maailmaan, jolloin mielenkiinnon herääminen ohjelmointia kohtaan on itsessään saavutus. (POPS 2014, 157.) Lisäksi oppilaat olivat ymmärtäneet ohjelmoinnin kasvattavan heidän matemaattista ajatteluaan, joten he olivat löytäneet yhteyksiä laajemmissa kokonaisuuksissa.

4.2.3 Ohjelmointi ja motivaatio

Opetussuunnitelmassa ohjelmoinnin osalta nousee vahvasti oppilaiden kannustaminen ja ohjelmointiin aiheena motivointi. Tällöin ohjelmoinnin

opetuksesta olisi tärkeää tehdä oppilaita motivoivaa ja rohkaista oppilaita itse kokeilemaan. Quahbi, Kaddari, Darhmaoui, Elachqar & Lahmine (2014) tutkivat ohjelmoinnin oppimisen mielekkyyttä yliopistotasolla. Hän vertasi Scratch-ohjelmointiympäristön ja Pascal-ohjelmointikielen vaikutusta oppimisen mielekkyyteen. Aiempi tutkimus osoitti, opiskelijoiden mielestä ohjelmointi on tylsää ja hankalaa, joten Quahbi ym. (2014) käyttivät pelillisyyttä motivaatiokeinona. Tutkimuksen tuloksista paljastui, että Scratch vaikutti hyvin paljon opiskelijoiden motivaatioon. Vain 15% piti Scratchia tylsänä tapana opiskella ohjelmointia, kun taas 79% piti Pascalia tylsänä. Oppimisen motivaatio näyttäytyi esimerkiksi siinä, että 90% Scratchia käyttäneistä kokeili tehdä pelejä itse, kun Pascalia käyttäneistä vain 38% kokeili tehdä itse. Tutkimuksesta nousi esiin myös se, että vain Scratchia käyttäneet opiskelijat keskustelivat ja jakoivat ajatuksia tunnin jälkeen, toisista ryhmistä tämä yhteinen jakaminen puuttui. (Quahbi ym. 2014, 1480-1481.) Vaikka kyseinen tutkimus on tehty yliopistotasolla, saa siitä tärkeää informaatiota ohjelmoinnin oppimisesta ja motivaatiosta.

Yhdysvalloissa on huomattu, että lukiossa tarjottavien tietotekniikan kurssien määrä oli vähentynyt, vaikka työmarkkinat ovat kyseisellä alalla yksi nopeimmin kasvavista aloista. (Burke & Kafai 2014, 10-11.) Kaikista korkeimman asteen koulutukseen hakijoista yli puolet, 56%, on Yhdysvalloissa naisia. Kaikista tietotekniikan alan kokeisiin hakijoista naisia oli kuitenkin vain 21%, eli noin viidennes. Yhdysvaltojen lukion tyttöjä oltiin haastateltu, miksei tietotekniikan ala houkuta heitä. Tyttöjen vastaus oli ollut heidän mukaansa alaan mielleltävässä luotaantyöntävässä nörttileimassa.

4.2.4 Ohjelmoinnin opettaminen opettajien näkökulmasta

Seuraavaksi on koottu muutamia tutkimuksia opettajien näkemyksistä ohjelmoinnista ja ohjelmoinnin opettamisesta. Balttian ja pohjaismaiden välisessä tutkimuksessa (Pears, Dagiene & Jasute 2017, 147) tutkittiin opettajien asenteita ohjelmointia kohtaa. Suomen, Ruotsin ja Latvian opettajista 78 % oli sitä mieltä, että heidän työympäristössään oli ohjelmoinnin tukemista edistäviä resursseja

tarpeeksi. Heistä suurin osa kuitenkin opetti myös yläluokille tieto- ja viestintäteknologiaa tai tietotekniikkaa, jolloin lähtökohdat ovat voineet olla erilaiset muihin luokanopettajiin verrattuna.

Saman tutkimuksen yhteydessä kerättiin tietoa myös Viron, Latvian, Suomen, Ruotsin ja Liettuan opettajien näkemyksistä ohjelmoinnin tärkeydestä ja tietoa siitä, mikä motivoi opettajia opettamaan ohjelmointia. Yleisesti oltiin sitä mieltä, että ohjelmointi kuuluu opetussuunnitelmaan ja se on tärkeää. Kuitenkin oltiin huolissaan oppilaiden kehitystason mukaisesta valmiudesta ymmärtää ohjelmoinnin käsitteistöä ja valmiin materiaalin puutteesta ohjelmoinnin eheyttämisessä, esimerkiksi kuinka yhdistää algoritmista ajattelua kuvaamataidon tunnille. Motivoiviksi tekijöiksi opettajat nimesivät sisäsyntyisiä, ammatillisia ja yhteiskunnallisia tekijöitä. Opettajia motivoi myös oppimisen tavoitteet, joita olivat oppilaiden kognitiivinen kehitys, oppilaiden voimaannuttamien ja heidän tiedon rikastuttaminen, oppilaiden tulevaisuusnäkyvät sekä työpaikkataidot. (Pears ym. 2017, 149.)

Fisher, Byrne & Tangney (2016) tutki opettajien koulutautumista ja oppimista järjestämällä opettajille työpajan. Työpajassa opettajat pääsivät opiskelemaan ohjelmointia käytännössä *Bridge21* menetelmään nojautuen. Tutkimuksessa opettajat kokivat ryhmätyöskentelyn tärkeäksi osaksi oppimista. Yhdessä tekeminen, yhdessä ongelmien ratkominen ja työtehtävien jakaminen ryhmänä motivoi opettajia saamaan projektinsa valmiiksi. Yhdessä oppiminen myös antoi opettajille itsevarmuutta kokeilla ja näin lisätä ymmärrystään ohjelmoinnista. Myös tiedon jakaminen koettiin tärkeäksi, koska silloin myös tiedostettiin eri tavalla oman osaamisen taso. (Fisher ym.2016, 276-277.)

Toisessa tutkimuksessa Makkia, O'Nealb, Cottena & Rikarda (2018) tutkivat, mitkä asiat vaikuttivat estävästi opettajien teknologian käyttöön opetuksessa. He jakoivat esteet kolmelle tasolle, joita olivat 1. resurssien puute, 2. opettajien motivaatio ja asenteet sekä 3. opettajien ymmärryksen ja osaamisen taso tieto- ja viestintäteknologian opetuksessa. Tutkittavat opettajat opettivat Yhdysvaltojen viidennettä tai kuudennettä luokkaa. Tutkimuksessa selvisi, että ensimmäisen tason esteitä oli paljon. Opettajien itsevarmuus ja tuttavallisuus

teknologian kanssa vaikutti positiivisesti heidän teknologian käyttöönsä luokkahuoneessa. Kun kolmannen tason esteisiin keskityttiin ja kehitettiin opettajien taitoja, saatiin positiivisesti edistettyä toisen tason motivaatio ja asennekysymyksiä. (Makkia ym. 2018, 91-92.)

Israel, Pearson, Tapia, Wherfel & Reese (2014, 265-266) tutkimuksessa tutkittiin opettajien mielipiteitä ohjelmoinnista peruskoulussa. Tutkimukseen osallistuvaa koulun henkilökuntaa koulutettiin ohjelmoinnin osalta: apulaisrehtori, kirjastonhoitaja ja tvv-vastaava koulutettiin ohjelmoinnin asiantuntijoiksi ja toimivat opettajien tukena, vastaavasti opettajat osallistuivat kesän aikana ohjelmointikurssille. Yksi tutkimuskysymyksistä koski opettajien tukimahdollisuuksia. Opettajat ilmoittivat olevansa halukkaita oppimaan ohjelmointia, jos saivat siihen asiantuntijoilta apua. Opettajat olivat kuitenkin haluttomia suunnittelemaan ohjelmoinnin opetusta, ennen ajan tuomaa kokemuksen turvaa. Ohjelmoinnin opetuksen suunnitteluun menneen ajan opettajat ottivat ylpeyden aiheena, oman ja yhteisön ammattitaidon kehittämisenä. Opettajat olivat sitä mieltä, että oli hyvä, että koko koulun henkilökunta oli yhtä aikaa mukana ohjelmoinnin opetuksen kehittämisessä. Asiantuntijat toivat turvallisuuden tuntua kirjastonhoitajan osalta tuttuudesta ja apulaisrehtorin osalta hallintopuolen asemasta. Opettajat kokivat oman taidon kehittymisen tärkeänä opetussuunnitelman tavoitteiden kannalta. Samalla myös opettajat olivat huolissaan siitä, onko ohjelmoinnille ja muille opetussuunnitelman uusille aihealueille riittävästi aikaa. Opettajat saivat kuitenkin hallinnon puolelta ymmärrystä siitä, että alussa opetussuunnitelman muutokset voivat aiheuttaa sotkua ja jännitystä. (Israel ym. 2014, 269-271.)

Tutkimuksessa (Israel ym. 2014) kysyttiin myös, mitä haasteita ohjelmoinnin lisäys opetussuunnitelmaan toi. Yksi näistä oli teknologian puute. Opettajat usein pystyivät kuitenkin kiertämään tai korvaamaan laitteen tai palvelun puutteet. Heitä ohjattiin myös hakemaan rahoitusta puuttuville laitteille. Uudet laitteet toivat myös käyttöönottoon, säilytykseen ja ylläpitoon liittyviä ongelmia, joihin etsittiin kuitenkin yhteistyössä ratkaisuja. Toinen haaste oli opettajien omat huonot ohjelmointitaidot. Alun jälkeen opettajat kuitenkin

hyväksyivät sen, ettei aika ja resurssit mitenkään riitä siihen, että heistä tulisi alan ammattilaisia. He myös huomioivat oppilaiden taidot ja nopean oppimisen, ja hyödynsivät heidän osaamistaan apuopettajina. Kolmas haaste oli oppilaiden eriarvoinen ekonominen asema ja oppimisvaikeudet, mutta opettajat pyrkivät ratkomaan näitä ongelmia positiivisuuden kautta. Neljäs haaste oli ajan rajallisuus. Opettajilla oli huoli siitä, että ohjelmointi on muilta aineilta pois ja näin ollen vaikuttaisi negatiivisesti valtakunnallisten testien tuloksiin. Tähän opettajat hakivat apua ohjelmoinnin eheyttämisestä osaksi muita oppiaineita, lisäkoulutuksen mahdollisuudesta ja asiantuntijoiden hyödyntämisestä. Lisäksi opettajat pyysivät myös lisää joustoa opetussuunnitelmaan. Kaiken kaikkiaan opettajilla oli myönteinen suhtautuminen ongelmiin ja niihin haettiin yhdessä ratkaisua. Opettajien taitojen karttuessa heidän usko omiin taitoihinsa myös kasvoi, ja he pystyivät hyödyntämään tätä uutta tietoa ongelmien ratkonnassa. (Israle ym. 2014, 271-273.)

5 MILLAISTA TUKEA OPETTAJILLE ON TARJOLLA?

Suomalaisilla opettajilla suuri huoli ohjelmoinnin opettamisen suhteen on ollut vähäiset täydennyskoulutukset, toimivien opetusmallien sekä digitaalisten oppimateriaalien puute (Malmberg 2013). Tässä luvussa kerromme millaista tukea ja materiaaleja ohjelmoinnin opetukseen opettajille on tarjolla.

5.1 Täydennyskoulutukset ja koulun tuki

Hallituksen järjestämien opetuksen ja koulutuksen kärkihankkeiden päätavoitteena on opettajien perus- ja täydennyskoulutusten jatkuva kehittäminen ja uudistaminen (Kaarakainen ym. 2017, 7). Kaarakaisen ym. (2017, 55) tutkimustulokset osoittavat, että opettajat ovat ajatelleet saamansa täydennyskoulutuksen melko hyödyllisiksi. Nämä täydennyskoulutukset ovat kuitenkin usein painottuneet vain laitteiden tekniseen peruskäyttöön jolloin esimerkiksi erilaiset sovellukset ja niiden käyttöön ottaminen, ohjelmointi ja sen opettaminen ovat jääneet koulutuksissa melko vähäisiksi, vaikka tarve olisi suuri. OAJ:n tekemän selvityksen (Hietikko, ym. 2016, 16-17) mukaan opettajat ovat saaneet tv-täydennyskoulutusta, mutta opettajien mielestä etenkin digitaalisiin oppimateriaaleihin ja digitaalisiin oppimisympäristöihin liittyvät täydennyskoulutukset eivät ole vastannut odotuksia määrältään ja/tai laadultaan.

Monipuolisten opettajan osaamista päivittävien täydennyskoulutusten lisäksi on tärkeää, että opettaja saa tarpeeksi tukea päivittäisessä työympäristössään. Näin ollen kollegoiden tuki on erityisessä asemassa, jolloin opettajat jakavat osaamistaan ja oppivat yhdessä uutta. Monissa kouluissa toimiikin digitalisaation osalta tutoropettaja, joka mahdollisesti järjestää koulutuksia ja auttaa tarpeen tullen. (Kaarakainen ym. 2017, 54-57.)

5.2 Millaisia opetusmateriaaleja ja -ympäristöjä opettajat käyttävät ohjelmoinnin opetuksessa?

Digiajan peruskoulu -hankkeen tutkimustulosten mukaan (Kaarakainen ym. 2017, 45) suurin osa opettajista käyttää helposti verkosta saatavia digimateriaaleja. Digitaalisten oppimateriaalien luova ja monipuolinen hyödyntäminen taasen oli melko harvassa. Monet valmiit ohjelmointiympäristöt ovat sinällään hyviä, mutta jos ohjelmoinnin opetus rajoittuu ainoastaan niihin, voi ohjelmoinnista tulla muusta opetuksesta irrallinen aihe ja oppiminen jää melko pinnalliseksi (Luostarinen & Peltomaa 2016, 110). Toimivien ohjelmointiympäristöjen löytäminen vaatii opettajan laajaa tutustumista ympäristöihin sekä suunnittelua, kuinka ohjelmoinnin saisi eheytettyä järkevästi osaksi muuta opetusta.

Esimerkiksi internetsivusto Koodiaapinen (2017) järjestää peruskoulun opettajille verkkokursseja ohjelmoinnista ja algoritmisesta ajattelusta. Kurssilla käsitellään algoritmista ajattelua, sekä ohjelmoinnin peruskäsitteitä. Lisäksi kurssi sisältää harjoituksia omalle ikäryhmälle ja pohdintaa, kuinka ohjelmointia voi opettaa motivoivalla ja kiinnostavalla tavalla sekä ideoidaan kuinka ohjelmointia voisi integroida eri oppiaineisiin.

Opettajille löytyy myös paljon kirjallisia oppaita, joissa käsitellään, miksi ohjelmointia opetetaan koulussa, kuinka sitä voisi opettaa, peruskäsitteitä ja muita ohjeita ohjelmoinnin aloittamiseen ja algoritmisen ajattelun kehittämiseen. Tällaisia oppaita ovat muun muassa Koodi2016 (Mykkänen & Liukas 2016) sekä Reseptit opsin käyttöön (Luostarinen & Peltomaa 2016).

Ohjelmoinnin opetuksen avuksi on kehitetty koulumaailmaan hyvin sopivia visuaalisia *drag and drop* ohjelmointiympäristöjä, joista esimerkkejä ovat Scratch ja Code.org. Nämä ohjelmat käyttävät hyödykseen valmiita koodinpaloja, joita ohjelmoija pystyy raahaaman ja yhdistelemään toisiin paloihin ohjelmointipalvelussa. Code.org palvelu tarjoaa käyttäjilleen luentoja, materiaaleja ja tehtäviä, joilla oppilaat voivat oppia ohjelmoinnista ja harjoitella ohjelmoinnin perustaitoja (Kalelioglu 2015, 201-202.). Scratchissa yhdistellään koodinpaloja, joilla ohjataan näytöllä näkyviä hahmoja. (Meerbaum-Salant 2013,

241). Scratchin pääpainona on luoda omaa sisältöä, eikä niinkään tarjota valmista materiaalia. Ympäristöstä on kuitenkin löydettävissä tutoriaaleja ja vinkkejä. Näitä Scratchin tuotoksia, kuten pelejä ja interaktiivisia tarinoita, voidaan jakaa, samalla antaen toisille mahdollisuuden oppia, käyttää ja soveltaa julkaisijan koodia. Scratchiin liittyy siis vahvasti yhteisöllinen tekeminen ja oppiminen. (Quahbi ym. 2014, 1480).

Ohjelmoinnin opetuskäyttöön on kehitetty erilaisia robotteja, joista tunnetuimpia esitellään seuraavaksi. Yksi helpoimmista robottimalleista aloittaa on Bee-bot ja Blue-bot robotit, joista löytyy liikkumiseen tarvittavia peruskomentoja. Bee-bot robottia ohjelmoidaan sen selästä löytyvin nuolinäppäimin, Blue-bot robottia voi ohjelmoida bluetooth-yhteydellä. (Terrapin 2016.) Micro:bit on pienikokoinen tietokone, jolla voi ohjelmoida koodilohkoilla, Javascriptillä ja Pythonilla. Laite yhdistetään USB-kaapelilla tietokoneeseen, jossa sitä käytetään tietokoneen nettiselaimella. Laitteella voi ohjelmoida esimerkiksi pelejä laitteesta löytyviä valoja tai sensoreita hyödyntäen yksinkertaisia. (Micro:bit Educational Foundation 2016.) Robbo on paketti, johon kuuluu ohjain ja ohjaimella ohjattava moottoreilla liikkuva alusta. Robbosta voi suunnitella sellaisen robotin kuin itse haluaa ja se ohjelmoidaan tietokoneella Robbon omalla sovelluksella. Robbosta löytyy moottorin lisäksi erilaisia sensoreita sekä muita yksinkertaisia toimintoja. (Robbo Europe) Lego-robotit ovat erilaisista palikoista rakenneltavia robotteja, joita ohjelmoidaan niiden omalla graafisella ohjelmointiympäristöllä. Lego-roboteihinkin kuuluu erilaisia sensoreita ja liikkumismahdollisuus, muistia ja muita mahdollisuuksia soveltamiseen. (The LEGO group 2018.) Sphero robotti on visuaalisin keinoin, koodipalkein, JavaScriptillä tai kauko-ohjaimella liikuteltava pallo, johon on mahdollista yhdistää myös lisättyä todellisuutta (Sphero 2017).

6 TUTKIMUSTEHTÄVÄ JA TUTKIMUSKYSYMYKSET

Uuden opetussuunnitelman myötä koulumaailma on muutoksen alla, joten on mielestämme tärkeää tutkia, miten näitä muutoksia on otettu vastaan. Keskitymme tässä tutkimuksessa laaja-alaisten tieto- ja viestintäteknologian piiriin kuuluvaan ohjelmoinnin opetukseen ja siihen liittyviin ajatuksiin. Ohjelmointi aiheena tuntuu olevan usealle opettajalle uusi, koska se ei ole aikaisemmin kuulunut peruskoulun oppimäärään. Tässä tutkimuksessa halutaan saada selville, mikä tilanne on oikeasti opettajien keskuudessa uuden opetettavan aiheen tiimoilta. Aihetta on tutkittu Suomessa erittäin vähän, joten tutkimuksen tekeminen on perusteltua. Seuraavaksi esittelemme tutkimuksemme neljä tutkimuskysymystä.

1. Millaisia käsityksiä alakoulun opettajilla on ohjelmoinnin opettamisesta alakoulussa?

Ensimmäinen tutkimuskysymyksemme koskee opettajien mielipiteitä, näkemyksiä ja kokemuksia ohjelmoinnin opettamista kohtaan. Mielestämme on tärkeää saada selville, mitä opettajat itse ovat asiasta mieltä ja mitä ajatuksia se heissä herätti. Opettajien asenne voi kertoa jotain siitä, miten ja miten paljon ohjelmoinnin opetusta otetaan mukaan opetukseen. Käsitykset ja kokemukset taas kertovat, miten ohjelmoinnin opetuksen sisäistäminen on opettajilta luonnistunut.

2. Millaista tukea alakoulun opettajille on tarjottu ohjelmoinnin opetukseen?

Toisella tutkimuskysymyksellä halusimme saada selville koulujen resurssitilannetta ohjelmoinnin kohdalta. Tutkimuksessa selvitetään, millainen on koulujen tämän hetkinen materiaalitilanne, tuntevatko opettajat saaneensa tarpeeksi koulutusta ohjelmoinnin opetukseen sekä millaista

koulutusta opettajat pitävät hyödyllisimpänä. Koulutukset tuovat opettajille mahdollisuuden kehittää taitojaan ja oppimateriaalit ovat tärkeitä opettajille heidän opetuksen suunnittelussaan. Jos opettajat eivät koe saavansa tarpeeksi tukea näiden osalta, on tuloksissa havaittavissa tärkeä viesti ohjelmoinnin opetuksen kehittämiseen.

3. Miten alakoulun opettajat eheyttävät ohjelmointia muuhun opetukseen?

Kolmas kysymys käsittelee sitä, miten opettajat ovat pystyneet ottamaan ohjelmoinnin mukaan muuhun opetukseen ja eheyttävätkö he ohjelmointia osaksi muita oppiaineita, vai käsitelläänkö sitä erillisenä aihekokonaisuutena. Ohjelmointi on osa laaja-alaisia oppimiskokonaisuuksia, joten sen opetukseen ei ole annettu tarkkoja raameja. Laaja-alaisen kokonaisuuden asema myös velvoittaisi eheyttämään ohjelmointia muihinkin aineisiin, joten on mielenkiintoista tietää, jääkö ohjelmointi pelkästään matematiikan ja käsitöiden piiriin.

4. Millainen merkitys alakoulun opettajien mielestä ohjelmoinnilla on koulussa?

Neljäs kysymys käsittelee ohjelmoinnin roolia ja tärkeyttä opetussuunnitelmassa. Halusimme kuulla opettajien perusteluja sille, kuuluuko ohjelmointi alakouluun ja mitä hyötyjä ja haittoja sen opetuksella on. Jos ohjelmointi ei ole tärkeää opettajien mielestä, mitä se kertoo kouluissa vallitsevista arvoista ja niiden eroavaisuuksista opetussuunnitelmaan arvoihin. Jos taas on, on mielenkiintoista tietää, millä tavoin pedagogiikan asiantuntijat, opettajat perustelevat sen tärkeyttä.

7 TUTKIMUKSEN TOTEUTTAMINEN

7.1 Tutkimuksen lähestymistapa

Tämä tutkimus tehdään laadullisilla menetelmillä, sillä haluamme ymmärtää tutkimaamme aiheita ja syventyä opettajien näkemyksiin ohjelmoinnista. Tutkimuksessa tarkastellaan eri ihmisten omista näkökulmista tutkittavaa ilmiötä, jolloin ilmiö rakentuu henkilöstä riippuen omalla tavallaan. Laadullinen tutkimus antaa aiheen tutkimiselle myös joustavat raamit, jolloin tutkimustulokset saattavat muokkautua haastateltavien mukaan. (Tuomi & Sarajärvi 2018, 72-75.)

Tutkimuksen toteutimme laadullisen sisällönanalyysin keinoin sen joustavuuden takia. Mietimme aluksi, että tekisimme fenomenografisen tutkimuksen, jonka avulla olisi pystynyt käsittelemään opettajien eriäviä käsityksiä hyvin monipuolisesti, mutta teoriataustan hyödyntäminen poissulki tämän tavan (Kettunen & Tynjälä 2018, 2; Huusko & Paloniemi 2006, 164). Ohjelmointi aiheena oli tutkimuksen alussa meillekin melko vieras ja ollaksemme valmiita haastattelemaan aiheesta pintaa syvemmälle, teoreettisen viitekehyksen kirjoittaminen oli ensisijaisen tärkeää. Tähän laadullinen sisällönanalyysi tarjosi mahdollisuuden.

Laadullinen sisällönanalyysi on tekstianalyysiä, jossa tarkoituksena on tiivistää jo valmista tekstiä. Tutkittavasta ilmiöstä pyritään muodostamaan kokonaiskuva tekstiä eritellen, tiivistäen sekä samankaltaisuuksia ja eroavaisuuksia etsien. (Tuomi & Sarajärvi 2009, 105-110.) Ilmiötä tutkimalla pyritään kuvailemaan, analysoimaan ja ymmärtämään paremmin itse ilmiötä. Aineistoa analysoidessa tulokset pyritään esittämään selkeästi ja ymmärrettävästi niin, ettei aineiston informaatio muutu. (Eskola & Suoranta 2008, 137.)

Tässä tutkimuksessa sisällönanalyysin analysointimenetelmänä käytetään teemoittelua, sillä tutkimuksessa halutaan kuvailla monipuolisesti, mitä käsityksiä opettajilla on ohjelmoinnin opettamisesta. Aineistossa toistuvien

aiheiden lukumäärällä ei tässä tutkimuksessa ole merkitystä, koska jokaisella opettajalla on oma näkemyksensä aiheesta. Tutkimuksessa ei myöskään yritetä muodostaa tietyntylaisia näkemyksiä omaavia opettajatyyppejä. Tärkeintä on saada opettajien ääni kuuluviin ja avata heidän ajatuksiaan. (Tuomi & Sarajärvi 2009, 93-94.) Aineisto jaotellaan osiin ja ryhmitellään aineistosta nouseviin teemoihin tutkimuskysymysten avulla. Opettajien puheesta etsitään teemoja kuvaavia selityksiä ja näkemyksiä.

Aikaisemmin tehtyjen tutkimusten ja teorian pohjalta oli haastavaa valita lähestymistapaa. Valitsimme aineistolähtöisen lähestymistavan, jota hyödynnetään silloin, kun aikaisempaa tutkimusta ei ole paljon tai se ei ole eheää. Tällöin tutkimuksen kategoriat muodostuvat aineiston pohjalta, eivätkä ne ole ennalta määrättyjä. (Elo & Kyngäs 2008, 109.) Tutkimus sisältää kuitenkin myös teoriaohjaavan lähestymistavan elementtejä teoreettisen viitekehyksen keräämisen takia. Teoreettinen viitekehys ja aikaisemmat tutkimukset eivät kuitenkaan vaikuta aineiston keräämisvaiheessa, mutta niitä hyödynnetään tutkimuskysymysten ja haastattelurungon muodostamisessa. Teoriaa ei siis yritetä todistaa, vaan se toimii tukena uuden tiedon luomisessa. (Tuomi & Sarajärvi 2009, 93-94.) Sisällön analyysin tekemisen voidaan jakaa kolmeen työvaiheeseen; valmisteluun, järjestelyyn ja raportointiin. Kokonainen haastattelu nähdään toimivana tapana laadullisen sisällönanalyysin aineiston keruumenetelmänä. Päätimme keskittyä lähes pelkästään siihen, mitä haastateltavat sanoivat. Esimerkiksi eleet ja äänenpainot jätimme huomioimatta, ellei se ollut todella tärkeää sen suhteen, mitä haastateltava sanoi. (Elo & Kyngäs 2008, 109.)

7.2 Aineiston keruu ja analysointi

7.2.1 Haastatteluaineiston keruumenetelmänä

Aineisto kerättiin haastattelemalla yhden Etelä-Suomalaisen kaupungin koulujen opettajia. Haastattelupyynnöt lähetettiin valtaosalle tämän kaupungin kouluista yhteistyössä Innokas-verkoston (Innokas.fi 2018) kanssa. Verkoston

kautta löysimme suurimman osan haastateltavista. Tutkimuksessa haastateltiin seitsemää luokanopettajaa neljästä eri koulusta puolistrukturoidun haastattelun mukaisesti ja kaikki haastattelut nauhoitettiin. Haastattelun vahvuus aineiston keruumenetelmänä on sen joustavuus. Haastattelussa haastattelijä ja haastateltava ovat suorassa vuorovaikutuksessa keskenään, joten väärinymmärrysten mahdollisuus on pienempi, sillä haastateltava voi suoraan kysyä tarkennusta keskusteltavasta aiheesta. Haastattelu mahdollistaa myös haastateltavalle näkemystensä esittämisen mahdollisimman vapaasti. (Tuomi & Sarajärvi, 2018, 85.)

Puolistrukturoidun haastattelun mukaisesti kaikissa haastatteluissa oli sama haastattelurunko, mutta lisäkysymykset sallittiin ja kysymysten järjestys saattoi vaihdella (Hirsjärvi & Hurme, 2001, 47–48). Haastattelurunko toimi itsellemme tietynlaisena tukena, jotta emme unohtaisi kysyä tutkimukseemme liittyen jotakin olennaista. Puolistrukturoidussa haastattelussa on tärkeää, että haastattelu ei etene tarkasti tietyn kysymyslistan mukaisesti, vaan pureudutaan teemoihin yksi kerrallaan vapaassa järjestyksessä, jolloin haastattelusta saadaan rennompia ja keskustelunomaisia. (Hirsjärvi & Hurme 2009, 46.)

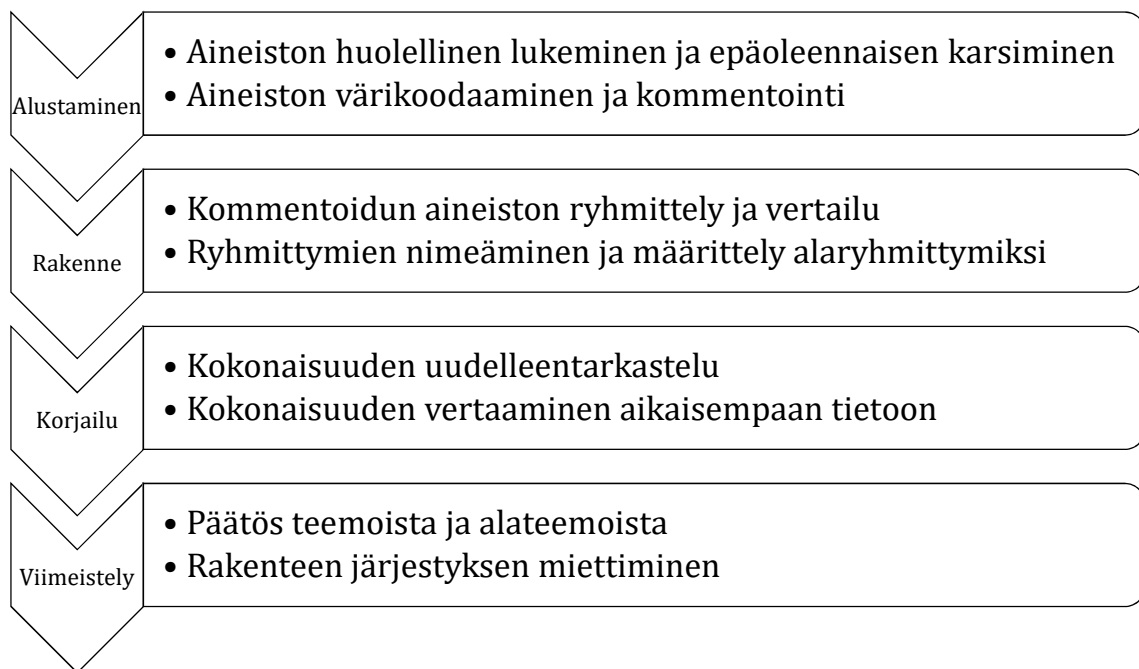
Haastateltaessa keskityimme hyvään laadulliseen haastatteluun, eli otimme aktiivisen kuuntelijan rooliin. Keskityimme parhaan mukaan siihen, mitä haastateltava sanoo. Näin pystyimme ohjaamaan haastateltavaa haastattelurungon ulkopuolisilla lisäkysymyksillä ja saavuttamaan tarvittavan tiedon. Tiedostimme myös sen, että haastateltava voi hämääntyä, jos kysymykset ovat liian vaikeita tai eivät sovi siihen, mistä ollaan juuri puhuttu. (Ahonen 1994, 136-137.)

Ennen haastatteluja anoimme tutkimusluvan (LIITE 2.) kyseisen kaupungin kasvatus- ja opetustoimelta. Tutkimuslupahakemukseen liitettiin tutkimussuunnitelma, ohjaajan hyväksyntä, haastateltaville lähetettävä informaatiokirje, tutkimuskysymykset ja haastattelurunko sekä tutkimuslupasuostumus. Tutkimuslupaa hakiessa tutkimus oli vasta alkuvaiheissa, joten lopullinen tutkimus on tarkentunut siitä, miten hakuvaiheessa sitä oltiin esitelty.

7.2.2 Laadullinen sisällönanalyysi aineiston analysointimenetelmänä

Tutkimusta lähdettiin analysoimaan laadullisen sisällönanalyysin keinoin. Kuvaamme analyysiprosessiamme alla olevassa kuviossa (KUVIO 1.) Lähdimme analysoimaan aineistoa Vaismoradin, Jonesin, Turusen ja Snelgroven (2016) tutkimuksen perusteella.

KUVIO 1. Teemojen ja alateemojen muodostus tässä tutkimuksessa



Ennakkovalmistautumisen ja haastattelujen pitämisen jälkeen aineistoa lähdettiin alustamaan käymällä sitä läpi useita kertoja. Tämä tehtiin, jotta ymmärrettäisiin esimerkiksi haastateltavien taustaa, näkemyksiä ja itsevarmuutta ohjelmointia kohtaan. Tutkijoiden tulkinnat ja ymmärrys aineistostaan on tärkeää. (Elo & Kyngäs 2008, 109-110.) Tämän jälkeen aineiston litterointi aloitettiin haastatteluaineiston auki kirjoittamisella kirjainkoolla 12, jolloin kirjoitettua aineistoa tuli 69 sivua.

Tämän jälkeen aineistoa käytiin läpi, jättäen pois kaikki tutkimuksen kannalta epäoleennaiset asiat. Aineistoa analysoidessa on tärkeää huomioida, mikä on tutkimuksen kannalta oleellista ja mikä ei. Tutkija saattaa aineiston paljoudessa sokaistua sille, mitä kyseisessä tutkimuksessa halutaan tutkia. (Tuomi & Sarajarvi 2018, 104.) Tästä syystä palasimme takaisin

tutkimuskysymyksiin, jotta pystyimme keskittymään tutkimukselle olennaisiin asioihin. Epäolennaisten asioiden poistamisen jälkeen aineistoa käytiin läpi uudestaan aineistoa kommentoiden ja värikoodaten.

Tällöin muodostettiin tutkimuksen kannalta olennaisia alustavia ryhmittelyjä. Tässä vaiheessa on tärkeää vertailla, nimetä, täsmentää ja määritellä, miksi mikäkin kommentoitu aineiston pala menee minnekin (Vaismorad ym. 2016, 105). Teemat siis muokkaantuvat analysoinnin edetessä ja suurten ryhmittelyjen sisällä kommentit järjestetään uudestaan pienemmiksi kokoonpanoiksi, alateemoiksi. Tämä prosessi tarkoittaa sitä, että aineisto pilkottiin pienempiin osiin erilaisten aihealueiden mukaisesti, joka mahdollistaa tuloksissa nousevien tiettyjen teemojen vertailun. (Tuomi & Sarajärvi 2018. 105-107.) Tässä vaiheessa teemat ja alateemat eivät kuitenkaan ole vielä valmiita, vaan muutokselle alttiita.

Ennen lopullista rakenteen muodostusta oli aika uudelleen tarkistella aineistoamme, etäännyttää itseämme siitä ja verrata löytämiämme tuloksia aikaisempiin. Vasta, kun teemat vaikuttavat järkeviltä, voidaan lopullinen päätös teemoista ja alateemoista tehdä. Sen jälkeen tulosten esittämisen rakenne ja järjestys on viimeinen vaihe teemojen raportoinnissa. (Vaismorad ym. 2016, 106-107.)

Aineistosta nousi vahvasti esiin viisi teemaa. Tutkimuskysymyksiämme sivuavia teemoja muodostui neljä. Näiden lisäksi muodostui vielä yksi teema, ohjelmoinnin haasteet, haastatteluissa useasti mainittujen ongelmien pohjalta. Teemojen alle muodostui kolmesta viiteen alateemaa, joissa käydään läpi teemoihin liittyviä tekijöitä. Teemoista luotiin visuaalisia kuvioita, jotta niiden sisältöä olisi helpompi ymmärtää.

7.3 Tutkimukseen osallistujat

Tutkimuksen osallistujat olivat kaikki yhden Etelä-Suomessa sijaitsevan kaupungin kouluissa työskenteleviä opettajia. Haastatteluaineisto koostuu seitsemästä luokanopettajasta, joista kahdella oli myös erityisluokanopettajan

koulutus ja he toimivat erityisluokanopettajan tehtävissä. Kaikki haastateltavista olivat naisia. Yksi haastatteluista käytiin englannin kielellä ja raporttia varten kyseisen haastattelun lainaukset käännettiin suomen kielelle tulos lukuun. Kolmella haastateltavista oli alle kymmenen vuotta työkokemusta, muilla kokemusta oli 10-25 vuotta. Tulososiossa suorissa lainauksissa käytämme haastateltavista lyhenteitä H1, H2, H3, H4, H5, H6 ja H7 selkeyden vuoksi.

Kolmella haastateltavalla oli koulussaan jonkintasoinen tvv-vastaavan tehtävä, joista yksi kertoi olevansa myös Innokas-verkoston ohjelmointikouluttaja. Lopuilla haastateltavista ei ollut koulussaan tvv-vastaavan roolia. Noin puolet vastaajista tiesi ohjelmoinnista jollain tasolla ennen opetussuunnitelman muutosta, mutta kuitenkin mielipide ohjelmoinnista vaihteli alkuun laidasta laitaan. Kaikki opettajat ilmaisivat omissa kouluissaan olevan melko hyvät materiaalit ohjelmoinnin opetukseen. Esimerkiksi Bee-bot ja Lego robottien käyttömahdollisuuden opetuksessaan ilmoitti viisi opettajaa. Muita yhdessä tai useammassa haastattelussa mainittuja robotteja olivat Bluebot, Pro-bot ja Sphere. Tablettien käyttö oli mahdollista ainakin viiden opettajan koulussa. Tietokoneella käytettävän Scratch ohjelmointiympäristön käytön mainitsi suurin osa opettajista, Code.org sivuston käytön kaksi.

7.4 Eettiset ratkaisut

Tutkimuksen informaatiokirjeessä painotettiin tutkimuksen luottamuksellisuutta. Tutkimustulokset on kirjattu niin, ettei vastaajien henkilöllisyyttä pysty tunnistamaan. Tuloksia kirjoittaessa tutkimukseen vastaajista on kerrottu neutraaliin sävyyn. Kielteisiä ilmaisuja vältetään, jotta tutkittavaa vastaajaryhmää ei leimata negatiivisesti. (Vilkkä 2007, 165-166.) Olemme tutkijoina pitäneet mielessä, että olemme osa tutkittavaa tieteenalaa ja kunnioitamme lähteitä viitatessamme niihin.

Haastattelun ollessa aineistonkeruumenetelmä, haastattelijan on otettava huomioon oma intersubjektiivisuutensa. Haastattelijan on siis tiedostettava lähtökohtansa pystyäkseen haastattelutilanteessa arvioimaan niiden vaikutusta

haastateltavan sanomisiin. (Ahonen 1994, 136.) Haastateltavan tulee pystyä luottamaan haastattelijaan. Yksi tapa saavuttaa tämä on pitämällä haastattelua keskusteluna. Haastateltava voi vaieta tai yrittää vastata mahdollisimman "oikeasti" jos hän kokee haastattelun kuulusteluksi. (Ahonen 1994, 137.) Tutkimuksen eettisyyteen ja luotettavuuteen liittyen tutkimuksessa on tärkeä tuoda esille oikeat lähteet ja ylipäättään käyttää paljon luotettavia lähteitä, joihin viitataan oikeaoppisesti. (Tuomi & Sarajärvi 2009, 132.)

8 TULOKSET

Aineistosta nousi esiin viisi teemaa (Kuvio 2.): ohjelmoinnin opetus alakoulussa, koulutus ja tuki, kouluyhteisön asenne ohjelmointia kohtaan, ohjelmoinnin merkitys osana opetusta, sekä ohjelmoinnin opetuksen haasteet. Tässä luvussa avataan näitä teemoja alateemojen avulla ja esitellään niitä käyttäen apuna haastatelluista nostettuja lainauksia.

KUVIO 2. Valitut viisi teemaa

Teema	
8.1 Ohjelmoinnin opetus alakoulussa	Tässä teemassa esitellään, kuinka ohjelmointi näkyy koulussa. Kaikilla haastateltavilla oli omat tyyliinsä opettaa ohjelmointia ja vähimmäismääränä pidettiin perusasioiden opettamista. Ohjelmoinnin rooli osana muita oppiaineita jakoi mielipiteitä ja osa opettajista eheytti ohjelmointia paljon muihin oppiaineisiin, toiset ei lainkaan.
8.2 Koulutus ja tuki	Haastateltavat kertoivat millaista koulutusta ovat saaneet ohjelmoinnin osalta, millaista koulutusta on tarjolla sekä millaista muuta tukea ovat saaneet ohjelmoinnin opettamiseksi. Lisäksi tässä teemassa esitellään millaista tukea ja koulutusta haastateltavat pitävät kaikkein hyödyllisimpänä ohjelmoinnin osalta.
8.3 Kouluyhteisön asenne ohjelmointia kohtaan	Tämä teema käsittelee opettajien, oppilaiden ja työyhteisön asennetta ohjelmointia kohtaan opettajien näkökulmasta. Opettajat perustelivat omaa motivaatioansa opettaa ohjelmointia oppilaiden oppimisen ilolla. Oma kiinnostus kehittää itseään oli iso vaikuttaja ohjelmointiin perehtymisessä. Lisäksi opettajien näkemyksiä heidän koulussaan vallitsevasta ilmapiiristä ohjelmointia kohtaan esitellään tässä teemassa.
8.4 Ohjelmoinnin merkitys osana opetusta	Tässä teemassa käsitellään millaisena opettajat näkevät ohjelmoinnin merkityksen osana opetussuunnitelmaa. Haastateltavien näkemykset erosivat siinä, kuinka tärkeää ohjelmoinnin opettaminen on, mutta jokaisen mielestä sillä on kuitenkin jonkinlainen paikka koulumaailmassa. Ohjelmoinnin avulla opittavista taidoista nostettiin esiin muun muassa looginen ajattelu, luovuus sekä järjestelmällisyys.
8.5 Ohjelmoinnin opetuksen haasteet	Haasteissa nousi esille resursseihin liittyviä rajoitteita, kuten laitteisiin liittyvät ongelmat, rahan riittävyys ja ajan rajallisuus. Ajankäyttöön liittyi

vahvasti myös teknologian muuttuvan luonteen ongelmia. Henkilökohtaiset asennekysymykset ja huoli oppilaiden oppimisesta nousivat myös esiin.

8.1 Ohjelmoinnin opetus alakoulussa

Tämä teema sisältää opettajien kertomuksia omaan ohjelmointiopetukseen liittyvistä järjestelyistä (KUVIO 3). Opettajilla oli erilaisia tyylejä opettaa ohjelmointia ja sen opetuksen määrä vaihteli paljon. Vähimmäismääränä pidettiin perusasioiden opettamista. Jos ohjelmoinnin opetusta haluttiin viedä pidemmälle, tarvittiin siinä usein omaa harrastuneisuutta tai tukihenkilöitä. Ohjelmoinnin rooli muiden oppiaineiden joukossa jakoi mielipiteitä opetuksen ajan ja eheytyksen näkökulmista. Oppilaiden osaamista arvostettiin ohjelmoinnin opetuksen tukena.

KUVIO 3. Ohjelmoinnin opetukseen liittyviä alateemoja.

Ohjelmoinnin opetus alakoulussa

Opettajan oma opetus	Ohjelmoinnin opetuksen eheyttäminen	Oppilaiden rooli ohjelmoinnin opetuksessa
-------------------------	---	--

8.1.1 Opettajan oma opetus

Moni haastateltava kertoi, että esimerkiksi toiminnallinen ohjelmointi on yleinen ohjelmoinnin muoto oppilaiden kanssa. Oppilaat ovat esimerkiksi itse olleet

“ihmisrobotteja”, joille annetaan käskyjä ja niitä pyritään noudattamaan mahdollisimman tarkasti. Lisäksi erilaiset robotit ilmenivät ohjelmoinnin opetuksessa suosituiksi, kuten Bee-botit, Blue-botit, ja Lego-robotit. Tällaisten erilaisten ohjelmoinnin opiskeluun tarkoitettujen robottien käytöstä opetuksessa on tutkittu olevan hyötyä teknologian, tekniikan ja matematiikan aihealueiden oppimisessa (Benitti 2012, 986).

H2: Joskus tehtiin ruutupaperilla, että oppilaat on selät vastakkain ja ollen joku yllin tunnin aikana.. Yhdistetään, he on piirtänyt sinne jonkun, asiaan liittyvät kuvan ja sitten, tota toiselle selittänyt mitä sinne piirtää. Tai sitten tälle että, ikäänkuin ihmistä on ohjelmointu, eli enkun tunneilla harjoiteltu näitä eteenpäin taaksepäin oikealle vasemmalle näitä niiku koulun tiloissa, he on kulkenut ja toisia neuvonaa. ja sitten ollen käytetty code.orgia, sitä ohjelmaa. Ja sit vähän siis näitä bluebotteja tai niiku beebotteja.

Aineistosta nousi esiin, että alakoulussa ohjelmointi keskittyy usein oppilaslähtöisyyteen itse tekemisen ja suunnittelun myötä. Opetustapoja esitettiin olevan hyvin erilaisia ja ohjelmoinnin opetusta pidettiin yleisesti mielekkäänä aiheena alakoulussa. Ohjelmoinnin opetustavalla ollaan tutkittu olevan suurtakin merkitystä oppimisen kannalta. Aiemman tutkimuksen mukaan oppilaat ovat yleensä mieltäneet ohjelmoinnin opiskelun tylsäksi ja hankalaksi, yhdessä tutkimusryhmässä oppilaiden mielenkiinto kasvoi, kun käytettiin pelillisempää oppimisympäristöä (Quahbi ym. 2014, 1480-1481). Osa haastateltavista kertoi, että ohjelmointi on luokassa ennemminkin toimintatapa ja osana jokapäiväistä arkea. Standl (2017, 183-184) on tutkimuksessaan todennut, että oppilaat pystyvät hyödyntämään koulutilanteissa ohjatusti algoritmista ajattelua hyödykseen. Mutta hänen tutkimuksensa perusteella oppilaat eivät ottaneet sitä osaksi arkeaan (Standl 2017, 190). Ohjelmointia on pyritty ujuttamaan arjen toimintoihin niin, että kynnys ohjelmointiin on hyvin pieni. Esimerkiksi yhdessä luokassa viikkojärjestelmä oli rakennettu ohjelmoimalla.

H4: Tietyllä tavalla niiku se ymmärrys niiku, et se ohjelmointi rakentuu niin niinku tietyllä tavalla ruohonjuuri tasolle, sinne elämän perusasioihin kii, et se ei oo pelkästään joku sähkölaite tai joku laite, mikä saadaan rakennettua tai muodostettua.

Osa haastateltavista oli osallistunut erilaisiin hankkeisiin, joissa ohjelmointi on ollut vahvasti läsnä. Näissä hankkeissa esimerkiksi Lego-robotit ollaan otettu

tarkempaan käsittelyyn ja yritetty laajentaa oppilaiden ymmärrystä ohjelmoinnista. Lisäksi aineistosta ilmeni, että ohjelmointia on saattanut olla tietyn jakson aikana enemmän, esimerkiksi käsitöissä, ja joissakin jaksoissa se on jäänyt vähemmälle tai sitä ei olla opetettu ollenkaan. Monet oppilaat ovat ohjelmoinnista hyvin kiinnostuneita ja saattavat tehdä sitä vapaa-ajallaan ja koulussa tarjottavilla lyhyillä tauoilla. Joissakin kouluissa laitteita saa ottaa myös kotiin mukaan, jolloin oppilas voi rauhassa omalla ajalla kehittää taitojaan. Osa haastateltavista kertoi, että ohjelmoinnin opetus on vahvasti matematiikan kirjojen varassa. Joitakin ohjelmointipelejä ja esimerkiksi Blue-bot robottia ollaan saatettu kokeilla, mutta opettajan oman ohjelmoinnin opetuksen pääpaino on kirjoissa. Näissä luokissa on käynyt koulun tv-vastaava pitämässä oppitunteja ohjelmoinnista.

H6: Okei, matikassa viidennen luokan kirjassa on kaksi kappaletta ohjelmointia ja tuota, minä käytin sitä sen omaa materiaalia. Tietysti minä olen oppinut kuinka sitä tehdään, mutta siinä se, joten seuraan mitä joku toinen on jo tehnyt. Tässä koulussa on muutama opettaja, jotka ovat erikoistuneita digitaaliseen opettamiseen ja yksi on tullut opettamaan minun luokalleni ohjelmointipelejä.

Ohjelmointiin liittyvien tehtävien keksimisen helppoudessa ilmeni eroja. Aineistosta nousi esiin näkökulma, että 5-6 luokille on vaikeampaa keksiä ohjelmointiin liittyviä tehtäviä kuin 1-4 luokille, sillä opetettavat aihealueet ohjelmoinnista ovat opettajallekin haastavia ja vaatii opettajan laajempaa tutustumista omalla ajalla. Yksi haastateltavista oli kuitenkin sitä mieltä, että 5-6 luokille on paljon valmiita materiaaleja, joita voi käyttää opetuksessa. Näille luokille on myös tarjolla enemmän ohjelmointiin liittyviä kisoja, joihin luokka voi osallistua. Lisäksi mainittiin, että ohjelmoinnin opetus painottuu enemmän vanhempiin oppilaisiin, jolloin ollaan päästy syvemmälle ohjelmoinnin ymmärryksessä. Etenkin alakoulun ensimmäisillä luokilla ohjelmointia lähestytään yksittäisten leikkien kautta, mutta 5-6 luokkalaisten kanssa pystyy tekemään laajempia projekteja.

H4: Mie luulen et myöskin se, että mikä on aina millokkii sit. Meiki koulul on paljo erilaisii teemoja, mitkä vie niitä asioita eteepäi, ni se painottuu kuitekii tonne isommille oppilaille. Et pienemmil oppilailla se on ehkä enemmi leikin kautta, sitte täs kolme-nelosilla se on ehkä enemmä sit siel niiku arjessa olevaa juttua ja sit sitä

viedää eteepäi viis-kutosella mis lähetää siit miettimää ehkä syvällisemmi ja enemmi sitä ohjelmointia.

Ohjelmoinnin opetuksen määrässä ilmeni suuria eroja. Joissakin luokissa ohjelmointi kulki jokapäiväisessä arjessa mukana, osassa ohjelmointia oli viikoittain ja jossain ohjelmointi oli harvemmin osana opetusta. Ohjelmoinnin vähäinen hyödyntäminen ei ollut kiinni laitteiden puutteesta, sillä monilla kouluilla laitteita oli monipuolisesti ja osa haastateltavista kertoi, että varauslistat ovat melko tyhjillään.

H1: Et mie luulen et kaikilla opettajilla on kyllä tiedos tavallaa näit, et mitä se suunnillee on ja mitä välineitä on, et kuhan rohkastus ottaa käyttöö niitä.

Esille nostettiin ennemmin ajan puute ja sen puitteissa asioiden tärkeysjärjestykseen asettaminen. Esimerkiksi matematiikan tunneilla ensisijaisena tulee opittavat perusasia ja ohjelmointi on lisänä, jos aikaa jää. Ohjelmointi ilmeni kouluissa olevan kuitenkin suosittu valinnaisaine.

8.1.2 Ohjelmoinnin opetuksen eheyttäminen

Aineiston mukaan opettajat eheyttävät ohjelmointia osaksi muita oppiaineita hyvin vaihtelevasti. Haastateltavista suurin osa ajatteli, että ohjelmoinnilla olisi hyvät mahdollisuudet eheyttää muihin oppiaineisiin, mutta todellisuudessa toteutus jää hyvin vähäiseksi ja monet opettajat pitävät ohjelmointia omana osaluueena opetuksessa. Ohjelmointi on sen verran uusi aihe alakoulussa, niin oppilaille kuin opettajillekin. Tästä johtuen ohjelmoinnin opetus on monilta osin vielä ohjelmoinnin idean hakemista ja perusasioiden opettelua, johon ei olla haluttu sekoittaa muita aiheita. Usein ohjelmointi sisällytetään matematiikan oppisisältöihin ja osittain myös käsitöihin. Etenkin matematiikan oppisisältöihin haastateltavat näkivät sen sopivan parhaiten. Näihin oppiaineisiin ohjelmointia on sisällytetty myös valtakunnallisessa opetussuunnitelmassa (POPS 2014). Pääpaino ohjelmoinnin opetuksessa on opetussuunnitelman mukaan laaja-alaisissa oppimistavoitteissa, jolloin sitä tulisi eheyttää muuhun opetukseen (POPS 2014, 20). Osa haastateltavista mainitsi, että kun ohjelmoinnin käytötapa

tulee enemmän tutuksi, voisivat he kuvitella laajentavansa ohjelmoinnin eheyttämistä myös muihin oppiaineisiin.

H1: Sit muissa oppiaineissa jos tota... Kuhan keksii niitä tapoja niiku hyödyntää, nii kyl varmaa tulee vielä enemmän sitä käyttöä ja hyötyä siitä sit välineestä, mut kestää aikaa enne ku saa sen omaksuttuu semmoseks, niiku osaks, sitä muuta opetusta.

Jotkut haastateltavista eheyttivät ohjelmointia muihin oppiaineisiin, kuten englantiin, äidinkieleen, ympäristöoppiin ja liikuntaan. Esimerkiksi englannin tunnilla oppilaat harjoittelivat englannin sanoja ohjelmoinnin keinoin. Lisäksi kuten aiemmin on mainittu, joillakin opettajista ohjelmointi kulkee mukana arjessa ja sitä on eheytetty jollain tavalla kaikkeen. Ohjelmoinnin eheyttämisestä muihin oppiaineisiin nousi esiin myös näkökulma siitä, että etenkin ylemmillä luokilla aihealueet menevät sen verran haastaviksi, että on pidettävä tarkasti huolta siitä, että muutakin oppimista tapahtuu. Näin ollen ajateltiin, että jos ohjelmointia eheytetään muihin oppiaineisiin alakoulun viimeisillä luokilla, näiden oppiaineiden sisällöt saattavat jäädä pinnallisiksi ja liian yksinkertaisiksi. Tämän takia ohjelmointi on ollut helpompaa pitää omana aihealueena.

H2: Just enkkuu tai sit on ollu yllissä ja jotai matikkajuttuu. Et mun mielestä se on sillee et sitä pystyy aika hyvin niiku käyttää eri oppiaineissa myös hyödyks sitte. Et ehkä nyt ku on vähä isompia oppilaita, vitosia tosiaa nii sit pitää aina miettii se et siinä ois tällasta niiku oppimista tosi tarkasti, että se tuo omat haasteensa.

8.1.3 Oppilaiden rooli ohjelmoinnin opetuksen toteutuksessa

Moni haastateltavista kertoi, että pitävät oppimista yhdessä oppilaiden kanssa mahdollisuutena. Heidän mukaansa nykyajan lapset ovat hyvin kiinnostuneita teknologiasta ja osaavat käyttää sitä luonnostaan. Osa haastateltavista selitti, että oppilaat saattavat olla taitavampia laitteiden käytössä kuin opettaja. Myös Kaarakaisen ym. (2017, 22) tutkimuksessa esitetään, että laajojen selvitysten mukaan oppilaat osaavat käyttää monipuolisesti kehittyvää teknologiaa vapaa-ajallaan. Tässä oppilaiden osaamisen hyödyntämiselle myös opetuksessa on suuri mahdollisuus (Vahtivuori-Hänninen ym. 2014). Oppilas voi opettaa myös opettajaa ja opettaja voi oppia itse oppilaidensa kanssa, jolloin tapahtuu molemminpuolista oppimista.

H7: Mun mielestä se onki aika hauskaa että ne on taitavampia ja se on niiku viehättävää, että siinä tapahtuu tällästä molemminpuolista oppimista.

Moni haastateltavista luotti oppilaiden taitoihin ohjelmoinnin parissa ja antoivat heille tietyllä tavalla vastuuta ja mahdollisuuden ohjata omaa oppimistaan.

8.2 Koulutus ja tuki

Tämä teema käsittelee ohjelmoinnin koulutusta ja tukea (KUVIO 4). Haastateltavat kertoivat millaista koulutusta ovat saaneet ohjelmoinnin osalta, millaista koulutusta on tarjolla sekä millaista muuta tukea ovat saaneet ohjelmoinnin opettamiseksi. Lisäksi käsitellään, millaista tukea ja koulutusta haastateltavat pitävät kaikkein hyödyllisimpänä ohjelmoinnin osalta.

KUVIO 4. Opettajien koulutus- ja tuki mahdollisuuksia

Koulutus ja tuki

Tarjolla olevat koulutukset	Tutor-toiminta	Kollegoiden ja ystävien tuki	Opettajien tukitoiveet
-----------------------------	----------------	------------------------------	------------------------

8.2.1 Tarjolla olevat koulutukset

Aineistosta nousi esiin, että ohjelmointiin liittyviä koulutuksia on melko hyvin tarjolla ja niihin ollaan oltu melko tyytyväisiä. Tämä on selvinnyt myös valtakunnallisella tasolla Kaarakaisen ym. (2017, 56) tekemässä tutkimuksessa, jossa ilmeni, että valtaosa opettajista on mieltänyt saamansa

digitäydennyskoulutuksen jokseenkin hyödylliseksi. Eroavaisuuksia ilmeni kuitenkin työajan puitteissa saatavasta täydennyskoulutuksesta. Myös tässä tutkimuksessa moni haastateltavista mainitsi, että työajan sisällä on mahdollista käydä vain tietty määrä täydennyskoulutuksia, jolloin opettaja päättää oman mielenkiintonsa mukaan, mihin koulutuksiin haluaa osallistua. Koulutuksia järjestetään myös työajan ulkopuolella, mutta ne eivät ole olleet haastateltavien mukaan kovin suosittuja.

H1: Muillaki on ollu tarjolla, koulutuksii, mut se on tietysti vähä, hmm, semmone valinnan paikka, että mihin menee. Meillä on yks sijaispäivä vuodessa, minkä saa käyttää koulutukseen, mut sit aina minkä kukakin kokee tarpeelliseks. nii siihen tietysti menee.

Koulutukset ohjelmoinnista olivat suosittuja, kun uusi opetussuunnitelma oli juuri rantautunut kouluihin, sillä kouluissa rehtorit huolehtivat siitä, että jokaisella opettajalla on jonkinlainen koulutus ohjelmoinnista. Ohjelmointikoulutusten suosio on joidenkin haastateltavien mukaan sittemmin hieman laskenut, sillä työajan sisällä olevia koulutuksia on vain tietty määrä ja opettajat haluavat osallistua koulutuksiin oman kiinnostuksen kohteidensa mukaan.

H7: Kyl niitä siin kohtaa ku se suurin vouhkaaminen tästä asiasta alko, nii on koulutuksii. Ja kyllä miun mielestä on jotai perustason koulutuksen, opastusta, on saanu. Kaikki halukkaat. Tänä vuonna ei ole, mut se oli oikeestaa niiku viime vuoden ja toissa vuoden juttuja.

Aiemmassa tutkimuksessa on käynyt ilmi, että opettajat ovat innokkaita oppimaan ohjelmointia, jos siihen saadaan asiantuntijoiden apua. Niin sanotusti kylmiltään opettajat eivät kuitenkaan halunneet lähteä suunnittelemaan ja opettamaan ohjelmointia, ennen kuin ovat saaneet siihen varmuutta. (Israel, ym. 2014, 265-266.) Haastatteluista nousi esiin, että koulutusta ja materiaaleja ovat tarjonneet esimerkiksi eri verkostot. Verkostot tarjoavat valmiita malleja, joita voi ottaa osaksi opetusta, ja verkostot myös vuokraavat kouluille laitteita käyttöön. Usein kuitenkin tästä yhteistyöstä opettajan pitää kirjoittaa esimerkiksi blogi-kirjoitus, jonka osa opettajista kokee työläänä. Osa haastateltavista kuitenkin suosii nimenomaan verkostojen tarjoamaa tukea niiden helpouden takia.

H4: Jos sie lähet johokii mukaa ni se yleensä edellyttää siult jonku blogipäivityksen tai jonku arvioinnin sen suhtee et mitä sie teet, jotta sie pääset niiku käyttämään vaik ilmasiks jotakii.. Et niis aina velvotetaa jollai tavalla niiku jotakin.. Ja sit moni kokee et se on taas sellast ylimäärästä.

8.2.2 Tutor-toiminta

Kaikkien haastateltavien kouluissa toimi tutoropettaja tai tvv-vastaava, jolta pystyi tarvittaessa pyytämään apua. Osa haastateltavista toimi itse koulullaan tutoropettajana. Osa tutoropettajista olivat pitäneet myös koulutuksia tieto- ja viestintäteknologiasta ja ohjelmoinnista koululla, joihin opettajat saivat itse toivoa, mitä ne pitäisivät sisällään. Tutoropettajan apua pidettiin ensisijaisen tärkeänä. Tutoropettajien tärkeys on todettu myös aiemmassa tutkimuksessa ja suurimmassa osassa kouluja toimiikin tvv-vastaava tai tutoropettaja (Kaarakainen ym. 2017, 54-57).

H1: Me olla koululle saatu sellasia tutorointitunteja ja opettajat sai niiku toivoo että minkälaista tutorointia haluais. Nii syksyllä oli tvv-ryhmä ja keväällä on vielä toine niiku tvv- ryhmä, missä on siis saanu toivoo mitä käydää läpi.

Tutoropettajat ovat pyrkinet kannustamaan ja innostamaan opettajia kokeilemaan ohjelmointia osana opetustaan. Koulutus on esimerkiksi tarjonnut eväitä siihen, kuinka aloittaa ohjelmointi ja mitä se on, sekä helppoja valmiita ideoita oppitunneille. Yksi haastateltavista oli itse tehnyt opettajien vapaaseen käyttöön kansion ohjelmoinnista, joka sisältää valmiita tehtäviä.

H4: Mie oon järjestäny ite ensin. Mie pidin tosi monta koulutusta nyt itseasiassa semmosia pajoja opettajille oon pitäny, et hyö on päässy kaattomaan et minkälaista se on ja mist kannattaa alottaa ja mikä on sellasta tosi matalan kynnyksen juttuja.

Lisäksi joissakin kouluissa toimii tutoroppilaita, jotka saavat perehdytyksen tieto- ja viestintäteknologiasta ja opettavat sitten muita oppilaita. Tämä miellettiin hyväksi tukimuodoksi, sillä oppilaiden osallisuus on viety pitkälle, ja lisäksi se vähentää opettajan työtaakkaa.

H3: Sit meillä on niitä tvv-oppilaita, näitä tvv-vastaavia, opettaa niiku oppilaita, mikä on ollu tosi hyvä, ku sit ne on tosi taitavia. et tosi, iso etu.

8.2.3 Kollegoiden ja ystävien tuki

Yhdeksi tärkeäksi tueksi ohjelmoinnin kannalta mainittiin kollegoiden ja ystävien tuki. Yhdellä haastateltavista oli esimerkiksi ystäviä, jotka opiskelevat luokanopettajiksi parhaillaan ja hän oli saanut heiltä tukea ja vinkkejä ohjelmoinnin opettamisesta. Opettajat toivoisivat enemmän tiedon jakamista työkavereiden kesken, joka etenkin isossa koulussa miellettiin haastavaksi.

H2: Isossa koulussa just se tiedon ja taidon jakaminen opettajien keskuudessa voisi olla vielä sellasta, tärkeämmässä roolissa et se on niiku ei kukaa yksinään ottas yksinää asiasta selvää vaa yhdessä niitä tai että jaettas sitä tietoa.

Opettajien yhdessä oppimisen tarvetta tieto- ja viestintäteknologian suhteen on todettu aikaisemmissakin tutkimuksissa. Esimerkiksi Kaarakaisen ym. (2017, 56) mukaan opettajat jakavat tietoa tieto- ja viestintäteknologian osalta. Lisäksi Fisher ym. (2016, 276-277) saivat tutkimuksessaan selville, että opettajat ajattelevat yhteisopettajuuden olevan tärkeää työyhteisössä. Yksi haastateltavista esitti, että opetussuunnitelma rohkaisee tekemään yhteistyötä erilaisten tahojen kanssa. Tämä mahdollistaa sen, ettei opettajan tarvitse olla joka alan asiantuntija, vaan perustaitojen hallussapito voi olla riittävää.

H7: Miun mielestä tää uus opetussuunnitelma entistä enemmän oikee houkuttaa tekemää enemmän yhteistyötä muitten kanssa, monien tahojen kanssa, ei tarttee joka alan taitaja itse ollakaa.

Kouluun on saattanut tulla vierailijoita esimerkiksi yliopistolta tai erilaisista yrityksistä, jotka tarjoavat asiantuntijuuttaan myös koulumaailmaan. Yhdessä koulussa esimerkiksi yliopiston opiskelijat olivat tulleet koululle vetämään erilaisia ohjelmointipelejä.

8.2.4 Opettajien tukitoiveet

Koulutusten suhteen nousi monia eri mielipiteitä. Osa haastateltavista toivoi, että koulutuksissa käytäisiin perusasioita ohjelmoinnista, ja osa toivoi valmiita ideoita opetukseen. Ohjelmoinnin perusasioiden käyminen nostaisi opettajan itsevarmuutta opettaa ohjelmointia. Kynnys ohjelmoinnin opetukseen miellettiin pienemmäksi, kun opettajan ei tarvitse itsenäisesti tutustua laitteisiin, vaan joku asiantuntija tulisi auttamaan alkuun laitteiden haltuunotossa. Lisäksi

koulutuksia voisi olla eri taitotasoille, joista opettajat voisivat valita itselle sopivan.

H1: ollaa yritetty löytää sellasii keinoja, että, sen luokan opettajan ei tarvii ite tietää, vaan joku muu tulee, ja kertoo mitä niillä voi tehdä, nii et se kynnys ois niiku matalmpi ottaa ehkä iteki käyttöön, tai sit kattoo vaa ku muut tekee.

Aineistosta nousi esille myös ajatus helposta sivustosta, jota joku asiantuntija päivittäisi aktiivisesti. Tällainen sivusto mahdollistaisi ohjelmointia matalalla kynnyksellä, jolloin opettaja voisi sivustolta valita omaan opetukseensa sopivimmat harjoitukset.

H4: Sit ku mie aattelin just vaik tälläst ohjelmointiakii, ihmiset aika paljo niissäkii menee sellasii tosi huippujuttuihin, mitä jaetaan, ni siinäki niinku se et sais sitä ruohonjuuritaso-levitystä ja sais sellasta tehtäväpankkia tai jotain sivustoa tai jotain vastaavaa mist ois helppo käydä niinku kattoos.

Tällaisia sivustoja on jo valmiina ja ne ovat osittain saavuttaneet opettajien suosion. Näihin kuitenkin toivottiin panostettavan entistä enemmän. Digitaalisten verkkosivustojen suosio ohjelmoinnin opetuksen tukena on näkynyt myös valtakunnallisesti. (Kaarakainen ym. 2017, 45.)

8.3 Kouluyhteisön asenne ohjelmointia kohtaan

Tässä teemassa käsitellään monipuolisesti opettajien näkemyksiä ohjelmoinnista alakoulussa (KUVIO 5). Kouluyhteisöllä tarkoitetaan opettajaa itseään, oppilaita ja muuta henkilökuntaa. Opettajat perustelivat omaa motivaatioansa opettaa ohjelmointia oppilaiden oppimisen ilolla. Oma kiinnostus kehittää itseään oli iso vaikuttaja ohjelmoinnin opettelussa. Tässä teemassa esitellään myös opettajien mielipiteitä heidän koulussaan vallitsevasta ilmapiiristä ohjelmointia kohtaan.

KUVIO 5. Opettajien näkemyksiä kouluyhteisön asenteisiin vaikuttavista tekijöistä, mukaan lukien heidän omat asenteensa

Kouluyhteisön asenne

Oppilaiden innostus ohjelmointia kohtaan	Oma kiinnostus ja osaaminen	Itsensä kehittäminen	Työyhteisö
--	-----------------------------	----------------------	------------

8.3.1 Oppilaiden innostus ohjelmointia kohtaan

Aineistosta nousi monelta opettajalta havainto siitä, että heitä itseään motivoi ohjelmoinnin opetuksessa se, kuinka paljon se innostaa oppilaita. Eräs opettaja kertoi, kuinka hienoa hänestä oli nähdä, että pieni osa päivästä ohjelmointia saattoi nostaa oppilaan oppimotivaatiota koulupäivän aikana.

H2: Ehkä sellane toiminnallisuus ja sellanen näkee oppilaassa sellasta oppimisen iloa. Hyvinki jollakin huomaa että se voi ehkä pelastaa oppilaan päivän. Siis tällanen yksi pieni hetki nii koulupäivä ei ollukaa niin tylsä vaa siitä tuliski iha niiku jees päivä.

Yleinen mielipide oli se, että oppilaista ohjelmointi on mielenkiintoista ja he pitävät siitä. Tähän nähtiin vaikuttavan tosin opittavan asian sopiva taitotaso ja oppilaan omat mielenkiinnon kohteet. Ohjelmointia pidettiin hyvänä, hauskana vaihteluna opetuksessa. Ohjelmointi tarjosi oppilaille myös onnistumisen kokemuksia.

H5: Mite kivaa niil oppilail on ja miten niinku pitkäjänteisyyttä ja muuta ne on saanu siit tavallaan siihe et ku niitäh työstetään pitkä aika et saadaan se kisahommeli niinku kuntoo.

Yksi haastateltavista mainitsi erityislasten mahdollisuuden saada onnistumisen kokemuksia, sillä ohjelmointi tarjoaa monenlaisia oppikokemuksia. Kyseinen opettaja kertoi myös, että ohjelmointi tuntuu olevan sellainen aihe, jossa monet erityislapsset ovat erityisen hyviä.

H5: Et nehän on niinku monet nokkelampii ja siis monelle erityislapsille saattaa ollakkii se erityisosaaminen siellä alalla. Tai jossain muussa näissä jutuissa, et ne pääsee loistamaan.

Opettajat kertoivat, miten oppilaiden innostuminen sai heille itsellekin hyvän olon aikaiseksi. Varsinkin jos opettaja sai jonkun oppilaan kiinnostumaan ohjelmoinnista niin, että oppilas lähti vapaa-ajallaan tutustumaan aiheeseen. Tietenkään kaikkia oppilaita aihe ei kiinnostanut yhtä paljon ja joidenkin oppilaiden kiinnostus teknologiaa kohtaan oli hyvin vähäistä. Silloin kun ohjelmointi meni liian haastavaksi, huomattiin kuitenkin että oppilaiden motivaatio laski eikä jaksettu samalla tavalla ponnistella eteenpäin. Kalelioglu (2015, 200) mukaan motivaatio on lapsille tärkeää myös teknologiaan liittyvässä oppimisessa. Suurin osa haastateltavista ei nähnyt sukupuolten välillä eroa kiinnostuksen määrässä, mutta esimerkiksi yksi opettaja sanoi haluavansa näyttää mallia tytöille ohjelmoinnista kiinnostuneena naisopettajana. Jotkut haastateltavista kertoivat ohjelmoinnilla olevan miesvaltaisen alan leima.

H4: Kyl mie koen sen niinku tietyl taval et mie haluun tarjota tämmöst tytöil niinku ajatusta siit et mie niinku oon naisopettajana kiinnostunu tästä ja tykkään tehdä näitä asioita, niin se malli niinku heillekki, et se ei oo pelkästään niinko miesten..

Yhdysvalloissa on aikaisemmin tutkittu, että tyttöjä kiinnostaa vähemmän ohjelmoinnin opetus (Kalelioglu 2015, 204-209 ja Burke & Kafai 2014, 10-119). Tässä tutkimuksessa haastatellut opettajat eivät sanoneet tytöillä ja pojilla olevan suurta eroa mielenkiinnossa ohjelmointia kohtaan. Osassa haastatteluista kerrottiin, että yleisellä tasolla pojilla saattaa olla teknisempi tapa ajatella, ja että tyttöjä tulisi kannustaa ohjelmoinnin opetukseen.

H7: Mutta kyllähä se niiku käytännössä näyttää siltä että pojat on enemmän tekniikan jutuista kiinnostuneita. Pääsääntöisesti. Mut ei näy se nii selkeesti tossa miun luokalla.

Stereotypistä ajattelua oli hyvin vähän, eikä se käytännössä näyttänyt vaikuttavan opetusoloissa. Tyttöjen ja poikien taidoissa ei oltu myöskään

havaittu suuria eroja, työtävät vain saattoivat olla erilaisia. Tytöt saattoivat olla hieman tarkempia tekemisessään, kun taas pojat kokeilivat rohkeammin ja mahdollisesti erheenkin kautta.

8.3.2 Oma kiinnostus ja osaaminen

Joitakin opettajia ohjelmointiin perehtyminen kiinnosti enemmän kuin muita, näitä opettajia motivoi usein se, mitä taitoja ohjelmoinnin avulla pystyi oppilaille opettamaan. Osa laitto taas muut opittavat aiheet itselleen tärkeämmäksi. Jotkut opettajista sanoivat myös suoraan, että uuden opetussuunnitelman pakko motivoi heitä opettamaan ohjelmointia, sillä se on yksi koulun vaatimista taidoista. Ohjelmointiin tutustuminen perusteita pidemmälle ei kiinnostanut joitakin henkilökohtaisella tasolla tai sitten opettajilla oli omia erityisalvoja, joiden kehittämisen eteenpäin he mielsivät tärkeämmäksi, kuin ohjelmointiin syvemmälle tasolle perehtymisen. Opettajan mielenkiinto on siis vahvassa asemassa siinä, kuinka paljon ja miten ohjelmointia alakoulussa opetetaan, sillä hän saa itse päättää omasta opetuksestaan, kunhan opetus on opetussuunnitelman mukaista. Tähän koulu pystyy vaikuttamaan kannustamalla opettajia käyttämään opetusteknologiaa. (Norrena, 2013, 56) Osa opettajista luotti koulun tvv-vastaavaan tai tutoreihin ohjelmoinnin opetuksen järjestämisessä, kuitenkin huolehtien että itsekin on jollain tasolla mukana.

H7: Kyllä miä varmistan sen että asiat mistä miä puhun, on oikea kielistä lähinnä. Mutta sitte, täytyy vaa varmistaa että on joku jolta kysyy apua ja joka voi pitää tarvittaessa pitää sellasta selkeätä opetustuokioita nii siihe löytyy kyllä apua meiltä. Ja useesti tehäänki nii että sit joku toinen opettaa. Jos joku vaikeempi käsitteistö on tai vaikka. Mut ei niin paljo kiinnosta että jaksasin kauheesti kouluttatua tässä asiassa. Mulla on ihan nyt tonttia tässä omassa asiassa (musiikki).

Opettajien omat asenteet ohjelmoinnin opetusta kohtaan olivat hyvin vaihtelevia. Alkuun ohjelmoinnin mukaan ottaminen opetukseen oli tuntunut usean haastateltavan mielestä pelottavalta. Ajan kuluessa pelko oli kuitenkin hellittänyt, kun tajuttiin, että ohjelmoinnillisia piirteitä on ollut opetuksessa aiemminkin ja alakoulussa ohjelmoinnin opettavat sisällöt ovat melko helposti omaksuttavia. Osassa haastatteluista painotettiin myös sitä, että on hyvä

pysähtyä välillä miettimään mitä ohjelmointi on, ja miksi sitä opetetaan koulussa. Tällöin saattoi nähdä läpi ohjelmoinnin, ja tajuta, että ohjelmoinnin avulla harjoitetaan vaikkapa ajattelun taitoja, jotka eivät tunnu olevan enää niin teknisiä tai vieraita aihealueita kouluun.

H3: Mun mielestä siitä on tehty kauheen iso peikko, ku toisaalta sit ku ei se oo mun mielestä nii vaikeeta tai jos kattoo jotai matikan kirjaaki nii sinne on vähä niiku sillee piilotettu. Niiku että se vähän sillee välillä tuntuu että kaikki ei pysähdy siihe miettimää, ett kyl te ootte ehkä jo aikasemminki tehny tätä jotaki, että ei olla vaa painotettu nyt tätä.

Oma näkemys osaamisensa tasosta kuitenkin vaikutti rohkeuteen opettaa ohjelmointia. Jos opettaja ei tuntenut olevansa tarpeeksi taitava ohjelmoinnin opetukseen, ei hän mielellään sitä lähtenyt opettamaan.

8.3.3 Itsensä kehittäminen

Omaa mielenkiintoa oppia motivoi lähinnä itsensä kehittämisen halu. Tähän vaikutti vahvasti siis se, halusiko opettaja oppia enemmän ohjelmoinnista. Yksi opettaja ilmaisi haluavansa oppia lisää ja uudistaa itseään. Toinen koki olevansa ohjelmoinnin saralla "aallonharjalla" pysymällä kehityksen mukana. Nämä opettajat mieltävät siis motivaationsa ohjelmoinnin opetukseen tulevan sisältäpäin, omaa ammattiaan kehittävinä tekijöinä, niin kuin Pears ym. (2017, 149) tutkimuksessa tuli myös esille. Myös oma tuttavapiiri vaikutti omaan kiinnostukseen. Haastatteluista nousi esiin, että ohjelmointia voi oppia helposti itsekkin, jos tuttavapiirissä pyöriteltiin ohjelmointia aiheena. Lisäksi, jos yleisellä tasolla itseä kiinnosti kehittyvä teknologia, oli helpompi omaksua ohjelmoinnillisia elementtejä ja pysyä kehityksen mukana. Jos opettaja näki ohjelmoinnin opetuksen oppimisessa itsensä kehittämisen mahdollisuuden, ei opettelua koettu lisätyönä.

H5: No ei se nyt niinku, mä en ehkä osaa ajatella sitä, vaik joutuuki perehtyy siihee. Se on mun mielest taas yks tapa rikastuttaa sitä opiskelua ja oppimista, itekki oppii sit uusii juttui.

Tämä todettiin myös Israelin ym. (2104, 269-271) tekemässä tutkimuksessa, jossa opettajat kokivat ohjelmoinnin opetteluun käytetyn ajan ylpeästi oman ja työyhteisön ammattitaidon kehittämiseen käytettynä aikana. Työyhteisön

yhteinen panos toi opettajille turvaa. Aineistossa kuitenkin nousi esille se, etteivät kaikki nähneet ohjelmoinnin opetuksen opiskelua itselleen tärkeäksi. Silloin opettajat eivät kokeneet tärkeäksi kehittää itseään ohjelmoinnin saralta vaan luotti toisiin.

H6: Minä luotan tvv-vastaaviin, joilla on muutama tunti viikossa aikaa auttaa meitä näissä asioissa. Joten luotan heihin enemmän kuin opiskelen itse.

8.3.4 Työyhteisön ilmapiiri

Haastateltavien kokemus työyhteisönsä ohjelmoinnin vastaanotosta olivat päällisin puolin positiivisia. Moni haastateltavista kertoi, että työyhteisön jäsenet ymmärtävät, miksi ohjelmointi on otettu osaksi opetussuunnitelmaa.

H6: Mutta en oo kenenkään kuullu sanovan ei, en aio tehdä sitä, tai mitään, et en pidä siitä, tai miksi me tehdään tätä. Minä luulen, että kaikki ymmärtävät.

Alkuun ohjelmointi tuntui hankalalta ja vieraalta, ja herätti jopa hieman vastustusta. Ajan kanssa mielipide on kuitenkin muuttunut myönteisemmäksi. Joissakin kouluissa tunnelma saattoi kuitenkin edelleen olla negatiivissävytteistä. Uusi opetussuunnitelman muutos oli yleisellä tasolla herättänyt keskustelua, mutta ohjelmoinnin osuus tästä oli aika pieni.

H3: Mä mietin et onks se niiku ohjelmointi itsessään. Et ei se, en minä oo ikinä aatellu et se ois sieltä just se niiku juttu, vaa yleensäki nyt tää uus opetussuunnitelma on, et se aika paljo on niiku, puhututtaa kyl niiku.

Ohjelmointi lasketaan usein tieto- ja viestintäteknologian aihealueisiin ja aineistosta nousi esiin myös epävarmuutta sen osalta. Osalla opettajista ei omien sanojensa mukaan ollut valmiuksia hyödyntää teknologiaa opetuskäytössä, jolloin ohjelmoinnin opetuskin oli jäänyt vähäiseksi. Vaikka Kaarakainen, ym. (2017, 21, 39) mukaan opettajien digitaidot ovat lähiaikoina kehittyneet, opettajien ohjelmointitaidot ovat valtakunnallisella tasolla heikot. Ohjelmoinnin opiskelua ei mielletty taakaksi, jos itsellä oli sisäistä motivaatiota opiskella ohjelmoinnin opettamista. Koulun painottamalla arvoilla oli osansa. Hyödyksi luettiin myös ylipäättään tieto- ja viestintäteknologian käyttö opetuksessa, jolloin tekniset laitteet ja materiaalit olivat tutumpia opettajille. Jos opettaja oli saanut

monipuolista ohjausta ohjelmoinnin opetukseen ja saanut onnistumisen kokemuksia, oli sitä helpompi lähteä soveltamaan omaan opetukseen.

H4: Niin, muutaki siinä samalla, et siin on voinu tulla se tunne et mitä tähä viel pitä's ruveta opettelemaan ite lisää. Ja sit jos toisaalt kuiteki jos ajatellaa uuden opetussuunnitelman niitä niiku isompia raameja, muutoksia ni sit ohjelmointi on kuitenkin aika pien pala sit siellä tietyl tavalla. Mut et se jotenki korostuu tosi voimakkaasti.

Työyhteisössä on myös noussut huolta siitä, että ohjelmointi lisää opettajan työmäärää, ja että opettajan pitää perehtyä aiheeseen omalla ajallaan.

8.4 Ohjelmoinnin merkitys osana opetusta

Tässä teemassa käsittelemme millaisena opettajat näkevät ohjelmoinnin merkityksen osana opetussuunnitelmaa (KUVIO 6). Onko ohjelmoinnin opetus tärkeää? Sopiiko ohjelmointi koulumaailmaan? Millaisia taitoja ohjelmointi kehittää opettajien mielestä? Näihin kysymyksiin pyrittiin löytämään vastauksia.

KUVIO 6. Ohjelmoinnin merkityksestä nousseita alateemoja

Ohjelmoinnin merkitys osana opetusta

Ohjelmoinnin sopivuus koulumaailmaan	Tulevaisuuden tarpeet	Ohjelmoinnin kautta opittavat taidot
--------------------------------------	-----------------------	--------------------------------------

8.4.1 Ohjelmoinnin sopivuus koulumaailmaan

Aineistosta kävi ilmi, että kaikkien haastateltavien mielestä ohjelmoinnilla on paikkansa koulumaailmassa. Eriäväisyyksiä nousi kuitenkin siitä, kuinka suuri tämä paikka on ja kuinka tärkeää ohjelmoinnin opetus on. Myös Karvosen ja Laukan (2016, 47) Pro-gradu tutkielmassa suurimman osan, 74 %, mielestä ohjelmointia on tärkeää opettaa alakoulussa.

Aineistosta nousi esille, että ohjelmointi voi opettaa monia erilaisia taitoja ja lisäksi oppilaat pitävät siitä. Toisaalta nousi myös näkökulma, että koulun arki on niin hektistä ja kiireistä, että muut opittavat asiat menevät usein ohjelmoinnin edelle. Aika on rajallista ja opettajan pitää valita mitä pitää tärkeimpinä asioina oppia, jotta oppiminen ei jäisi pinnalliseksi yksittäisten asioiden haalimiseksi. Tämä riippuu siis merkittävästi opettajan kiinnostuksen kohteista. Ne opettajat, joita aihe kiinnostaa, tarjoavat sitä enemmän opetuksessaan kuin ne, joita aihe ei kiinnosta tai jotka ovat hyvin alkutaipaleilla ohjelmoinnin suhteen. Uusi opetussuunnitelma antaa opettajalle mahdollisuuden, että hän saa itse valita tiettyjen raamien mukaan, mitä opettaa. Aineiston mukaan ohjelmointi sopii alakouluun, jos sille olisi aikaa.

H1: No, tää on ehkä tämmöne ikuisuuskyseminen ku on aika paljo kaikkee tärkeätä mitä vois hyvä näitten oppia, ja missä välissä ehtii sen kaiken käydä, et se olis oppilaille hyödyllistä nii...keskustelua herättävää kyllä.

Osa haastateltavista oli sitä mieltä, että ohjelmointi on mukava lisämauste koulun arkeen, mutta ei korostaisi sen merkitystä liikaa. Tätä verrattiin esimerkiksi siihen, että matematiikan kirjassa ohjelmointi on kirjan viimeisillä sivuilla, joita usein käydään läpi, jos ylimääräistä aikaa jää. Aineistosta nousi esiin myös näkökulma siitä, että vaikka lapset tykkäävät ohjelmoinnista; se on hauskaa ja monipuolista, oppilaiden on kuitenkin tarkoitus oppia monia tärkeämpiä asioita, joilta ohjelmointi saattaa viedä ajan. Osa haastateltavista näki koulussa olevan tärkeämpiäkin asioita oppia kuin ohjelmointi. Tässä yhteydessä ohjelmointi miellettiin perinteisiä tapoja pois sulkevaksi tavaksi opiskella.

H3: Siin on sellasii vähä asioita, mitä mä en välttämättä ehkä tarvii töissäni. Tai just sitä et ei kaikki oo vaa nii, ei kaiken tarvii vaa olla nii hauskaa ja kivaa ja siis tämmöstä, tavallaa juttui voi käyttää niiku sillee enemmänki lisänä. Et kuitenkin se,

et se idea on se et lapset oppii. Et kyl mie niiku, se on iha hyvä juttu, tulee nii monipuolisuutta ja.. menetelmiä ja sisältöihinki ja sillee mut että.. Ja onhan se tulevaisuutta, et kyl mä sen tajuan, mut ei se minusta mikään kaikista tärkein asia oo.

Laaja-alaisten osaamistavoitteiden merkitys opetuksessa nostettiin esiin siinä, että niiden opetus pitäisi tapahtua muiden aineiden sivussa ilman, että ne pois sulkisivat mitään. Tällöin miellettiin, ettei ohjelmointi ollut pois miltään muulta aiheelta, vaikka ajankäyttö saattoikin huolestuttaa. Tämä huoli ajasta leikkasi kuitenkin läpi kaiken koulussa tapahtuvan.

8.4.2 Tulevaisuuden tarpeet

Ohjelmoinnin merkitystä alakoulussa perusteltiin monissa haastatteluissa sillä, että se on vahvasti osa tulevaisuutta sekä nyky-yhteiskunnan tarpeilla. Haastateltavien mukaan nykyajan lapsille tieto- ja viestintäteknologian käyttö on hyvin luontaista, ja usein he osaavat käyttää erilaisia laitteita paremmin kuin opettaja. Ohjelmointi on tietyllä tavalla osana oppilaiden maailmaa. Osa haastateltavista esitti, että lasten kanssa olisi tärkeää pohtia, millaisia ammatteja tulevaisuudessa saattaa olla ja miksi tiettyjä aineita opetetaan. Pears ym. (2017) on nostanut tutkimuksessaan esiin sen, että opettajia motivoi ohjelmoinnin opettamisessa oppilaiden tulevaisuusnäkökymät ja työpaikkataidot (Pears ym. 2017, 149). Oppilaiden tulevaisuuteen linkittyi siis teknologian lisääntyvä rooli työpaikoillakin. Koulun olisi siis hyvä ohjata oppilaita työelämää varten ja moni haastateltavista ajatteli, että ohjelmointi on merkittävä osa tulevaisuuden ammatteja.

H2: Ehkä niikun tulevaisuuttaki varten ku ei oikee tiiä minkälaisii ammatteja tulevaisuudessa on niiku, nii vähä ehkä sitäki niiku et mitä ne nyt noitte oppilaitten kanssa yleisestiki mietitää aina et miks on tällasii jotenki tiettyjä oppiaineita. Vähä sitä että täällä alakoulussaki kaikkee ku ei vielä tiedä mitä teistä tulee ja minkälaisia ammatteja, nii kyllä mä koen sen sillee tietyllä tavalla. Yksi tärkeä osa, muiden joukossa.

Aineistosta nousi esiin, että ohjelmointi saattaa sivuuttaa joitakin prosesseja ja näin ollen jotkut ammatit tulevat katoamaan. Ihmisten olisi hyvä ymmärtää ohjelmointia jollakin tasolla, vaikka heistä ei itsestään tulisikaan ohjelmoijia. Teknologian rooli yhteiskunnassa kasvaa jatkuvasti ja nyky-yhteiskunta

tarvitsee ohjelmoinnin osaajia, jotka toimivat laitteiden rinnalla. Näitä taitoja oppilaiden olisi siis hyvä oppia.

H6: Minun mielestä se on tärkeää, koska se on maailman tapa toimia, että teknologia kasvaa kasvamistaan, ja sen käyttäminen, joten se on tärkeää lapsille omata tällaisia taitoja joten... Tulevaisuudessa he tarvitsevat niitä.

Yhdessä haastattelussa esitettiin, että ohjelmointi kiinnostaa oppilaita ja avaa heille maailmaa, mitä tämänhetkisille opettajille ei ole itselleen koulussa tarjottu. Yksi haastateltavista mainitsi myös, että ohjelmointi on globaali ilmiö, josta emme voi jäädä suomalaisina jälkeen.

H4: Ja sit se, että, et mil taval se vaikuttaa niin niinku mei koko maapalloo kaikel tapaa ja myö ei voida suomalaisina jäädä ilman sitä ku sit se on tuol niinku jos aatellaa vaik Aasian puolta, nii mite se on voimakkaast siellä.

Globalisaation mukana on tullut paine pysyä mukana maailman kehityksessä, ja yhä useampi yhtiö toimii monessa maassa. Tämän takia useassa työpaikassa käytetään erilaisia teknologioita informaation tuottamiseen, liikuttamiseen ja säilyttämiseen.

8.4.3 Ohjelmoinnin kautta opittavat taidot

Haastatteluista nousi monia erilaisia taitoja, joita ohjelmoinnin avulla voidaan oppia. Yksi opettajista sanoi, että ohjelmointiin liittyvissä aiheissa mahdollisuudet olivat loputtomia. Osa haastateltavista mainitsi heti tiettyjä taitoja ja osa joutui miettimään hetken, ennen kuin osasi sanoa, millaisia taitoja ohjelmointi voi opettaa. Monista haastatteluista nousi esiin ongelmanratkaisutaidot ja siihen liittyvät elementit, kuten syy-seuraus-suhteen ymmärtäminen. Lisäksi mainittiin havainnointikyvyn kehittyminen, ohjeiden täsmällinen antaminen ja noudattaminen sekä prosessin etenemisen vaiheet.

H2: Hyvin vahvasti mä niiku yritän niiku aina miettiä niiku et siinä on tällane opettamisen myös näkökulma. Niiku yleensäkin tää ohjelmointi paljon niiku kehittää oppilaiden kykyä havainnoida ja kaikkea ongelmanratkasuu.

Aineistosta ilmeni, että ohjelmointi voi kehittää myös oppilaan sosiaalisia- ja ryhmätyötaitoja. Näissä tilanteissa oppilailla on jokin yhteinen määränpää, jota he tavoittelevat, kuten robottien rakentaminen, pelit ja kisat. Oppilaat joutuvat

tekemään kompromisseja ja tekemään tiivistä yhteistyötä, jotta prosessi saadaan päämäärään.

H5: No yhteistyötaitoja, jos tekee just jotain robojuttua eikä valmistellakaan jotain esitystyylisiä ja sitten toki sellasta matemaattista ajattelua myös.

Näihin prosesseihin liittyy myös pitkäjänteisyys, sillä ohjelmointiin liittyy monia työvaiheita. Esimerkiksi Lappalainen ja Viitanen (2012, 5-6) kuvailevat ohjelman luomiseen liittyvän useita eri vaiheita, joissa ohjelmanluontiin liittyviä ongelmia tarkastellaan.

Osa opettajista sanoi että, ohjelmointi kehittää oppilaan ajattelun taitoja ja loogista päättelykykyä. Ajattelun taitojen kehittymistä avattiin sillä, että oppilas oppii tiettyjen kaavojen avaamista ja luomista, jossa asiat tapahtuvat tietyssä järjestyksessä ja tahdissa. Oppilaan olisi hyvä ymmärtää, että omalla tavallaan kaikki perustuu tähän ajatukseen, kuten esimerkiksi hampaiden pesu. Lisäksi aineistosta ilmeni, että ajattelun taitojen tärkein asia oppia olisi se, että oppilas ymmärtäisi ohjelmoinnin perusajatuksen; ohjelmointi on käskyjen antamista ja noudattamista. Ilman näitä ei tapahdu mitään, eikä laite toimi, jos joku ei ole sitä ohjelmoinut. Tätä ajatusta korostaa myös valtakunnallinen opetussuunnitelma (POPS 2014, 157).

H5: Et jotenki niinku se on mun mielest ehkä se tärkein pointti, mitä ne oppilaat tajuis, että se ohjelmointi on sitä ja koodaaminen on sitä et annat käskyjä jotka sit se laite tai mikä tahansa toteuttaa et se ei... tai ilman sitä ei tapahdu mitään. Et siel on joku koodannu.

Aineistossa mainittiin, että ohjelmointi kehittää oppilaan käsitystä hänen omista taidoista. Lisäksi ymmärrys oman kiinnostuksen merkityksestä ja ikätason vaikutuksesta oppimiseen kasvavat. Lisäksi esille nousivat oppilaan itseohjautuvuuden ja itsesäätelyn kehittyminen ohjelmoinnin avulla. Ohjelmoinnilla ei aina ole valmiita tiettyjä ratkaisuja, joita lähdetään etsimään ja kokeilemaan, jolloin oppilas joutuu käyttämään myös luovuuttaan.

H4: Ja sit tietyl tavall niinku semmone luovuus ja innovatiivisuus lähtis niinku sitä kautta syntymään, että niinku ei välttämättä oo valmiita ratkaisuja, vaan lähetään ettimään niitä, niinku kokeilemalla tietyl tavalla.

Opettajat näkivät siis ohjelmoinnin kautta pystyvän oppimaan erilaisia taitoja ja ohjelmoinnin opettavan oppilaille esimerkiksi vastuuta ja luovuutta.

8.5 Ohjelmoinnin opetuksen haasteet

Tässä teemassa käsitellään opettajien puheista nousseita ohjelmointiin liittyviä haasteita (KUVIO 7). Haasteissa nousi esille resursseihin liittyviä rajoitteita, kuten laitteisiin liittyvät ongelmat, rahan riittävyys ja ajan rajallisuus. Ajankäyttöön liittyi vahvasti myös teknologian muuttuvan luonteen ongelmia. Henkilökohtaiset asennekysymykset ja huoli oppilaiden oppimisesta nousivat myös esiin.

KUVIO 7. Erilaisia haasteita, joita opettajat mainitsivat ohjelmointiin liittyen.

Ohjelmoinnin opetuksen haasteet

Aika	Taloudelliset resurssit ja laitteiden ylläpito	Teknologian muuttuva luonne	Asennehaasteet	Oppimisen arviointi

8.5.1 Aika

Ohjelmoinnin opetuksessa aika nousi suurimpana haasteena esiin. Ohjelmointi on uusi aihe opetustyössä monille opettajille, joten uuden opiskeluun ja asioista selvittämiseen kuuluva aika huoletti opettajia. Jos opettaja ei tuntunut pystyvänsä irtoamaan luokasta tai hänellä ei ollut kontakteja ohjelmointiin liittyen, oli vaikeaa lähteä tutustumaan siihen. Koulupäivät koettiin kiireisiksi, joten uuden asian sisältäminen normaaliin koulurytmiin koettiin hankalaksi, vaikka se nähtäisiinkin tärkeäksi.

H6: "Koulussa on aina niin kiire, niin hankala tunkea sinne mitään, oli se sitten kuinka tärkeää. Eli, ei tarpeeksi aikaa"

Moni haastateltava tiedosti, että ohjelmointi ei ole sen vaikeampaa kuin mikään muukaan aihe. Aineistossa mainittiin myös se, että mitä enemmän ohjelmointiin tutustuu, sitä helpommaksi sen käyttö tulee. Mutta ajan löytäminen ja aloittaminen hidastavat tätä prosessia, johon kuuluu esimerkiksi perusasioihin paneutuminen ja oppimateriaalien etsintä. Tähän prosessiin kuuluu myös laitteisiin tutustuminen, joka vie oman aikansa. Laitteisiin liittyy myös resurssien puolesta ongelma, että laitteita ei välttämättä riitä kaikille luokille tai koululle omia. Tästä syntyy laitteiden varaukseen liittyviä hankaluuksia suunnittelun puolelta. Osa opettajista ei tuntunut pystyvänsä suunnittelemaan opetustaan niin, että voisivat varata laitteet tarpeeksi ajoissa tunnilleen. Myös toisten tekemät varaukset ja laitteiden lainassa oleminen voivat rajoittaa omien tuntien suunnittelua.

H4: Liikuntaa pidetään tärkeänä ja on niinku sovittu et välitunnilla ei pelata puhelimella ja siestat on sellast vapaa-aikaa, mis ei käytetä mitää laitteita. Ni sit tietyl taval niinku se, että mikä vois olla sit se kolikon kääntöpuoli niille oppilaille, jotka intohimosesti sitä (ohjelmointia) tekee ja haluais tehdä.

Ohjelmoinnin ja muiden koulussa opettavien aiheiden arvottaminen ajan suhteen oli ongelmallista. Esimerkiksi liikunta ja ohjelmointi saattoivat mennä arvoineen ja ajankäytännöllisesti välitunneilla päällekkäin. Moni lapsi liikkuu alle liikuntasuosituksen, joten välitunneilla olisi hyvä liikkua. Välitunnit taas olisivat hyvää aikaa oppilaille tutustua omatoimisesti koulun laitteilla ohjelmointiin, mutta ohjelmoidessa usein ollaan paikallaan ruudun tai laitteen äärellä. Ajallisesti oltiin myös huolissaan, miten kaikkia aineita keretään opettamaan kouluvuoden aikana, että jääkö jotain tärkeämpää pois ohjelmoinnin takia.

Israel ym. (2014) nostivat tutkimuksessaan esiin myös aikahaasteen. Suomessa valtakunnalliset testit eivät ole tärkeässä roolissa niin kuin Yhdysvalloissa, mutta silti aika mielletään tässäkin tutkimuksessa haasteeksi. Tähän eräs opettaja tarjosikin ajatusmallia siitä, että koska ohjelmointi kuuluu

laaja-alaisiin opintoihin, pitäisin sen opetuksen soljua muiden oppiaineiden mukana niin kuin muidenkin laaja-alaisten kokonaisuuksien.

8.5.2 Taloudelliset resurssit ja laitteiden ylläpito

Ajan lisäksi raha nähtiin isona haasteena ohjelmoinnin opetuksessa. Haastateltavien mukaan ohjelmointiin tarvitaan paljon materiaaleja, joista suuri osa on kalliita. Kaikilla kouluilla ei ole mahdollista hankkia ohjelmoinnin monipuolistamiseen tarvittavia välineitä ja jotkut opettajat olisivat kaivanneet enemmän materiaaleja projektien toteuttamiseen. Rahan vähyyden takia kaikille luokille ei välttämättä riitä tarpeeksi laitteita, joka omalla tavallaan hidastaa oppimista. Ohjelmoinnin opetuksen monipuolisuus voi kärsiä, jos on varaa hankkia vain tietynlaisia laitteita.

H4: Et jos haluais toteuttaa lasten kans jotaki niinku isompaa ni siihe tarvitaa ain sitä rahaa. Sama niinku mein just näitten robotiikkalelujen niin ne osat ei oo mitää ilmasia. Ja sit se että jos on niinku se jonkilaine ajatus et mitä voidaa vaik 45 oppilaille ostaa, ni sit mei oli esimerkiks pakko jättää kodille patteri.

Raha nähtiin ongelmana myös kouluttautumisessa. Jos opettajalla ei ollut mahdollisuutta lähteä työajallaan täydennyskoulutuksiin ohjelmoinnin saralta, koulutus jäi käymättä. Jos opettajalle oli tarjottu mahdollisuus vain yhteen koulutuspäivään, voi olla, että hän valitsi jonkun toisen aihealueen. Joillakin kouluilla oli annettu mahdollisuus käydä koulutuksissa niin, että rinnakkaisluokan opettaja pitää molempien luokkien tunnit, mutta näissä tilanteissa ei haluttu omalla ylimääräisellä kouluttautumis päivällä siirtää lisätyötä kollegoille.

Rahan riittävyyden lisäksi laitteita kohtaan nousi myös ekologinen huoli. Elektronisten laitteiden käyttöikä ei ole pitkä ja laitteiden käyttöjärjestelmät vanhenevat myös nopeasti.

H1: Joku Ipadinki käyttöikä, ei oo mikää, kauhee montaa vuotta koulukäytössä. Nytki meill on ihan läjäpäin Ipadeja, nyt niit alkaa tipahtamaa pois, ett mistä myös saahaan rahotusta sille, että saadaan tavallaa pidetty konekanta hyvänä, ja sit vähä tällane ekologinenkii kysymys, et jos iha hirveesti aina hankitaa ja sit ne kestää jonki aikaa, ja sit ne poistuu käytöstä.

Laitteisiin on siis saatettu laittaa paljon rahaa, mutta muutaman vuoden päästä koululla täytyy miettiä, miten laitteisto saadaan päivitettyä ajan tasalle. Opettajat korostivat tarkan harkinnan tärkeyttä laitteiden hankinnassa. Laitteiden vanhenemisen lisäksi nostettiin esille myös rikkoontumisen ynnä muun käytöstä johtuvan uhka. Jonkun täytyy pystyä ylläpitämään ja huoltamaan laitteistoa, johon menee oma aikansa. Yhteiskäyttö tuo myös omat haasteensa, kun kukaan ei ole täysin vastuussa laitteiden kunnosta vaihtumien välissä. Vaikka tunti olisi hyvin suunniteltu ja laitteet varattu ajallaan, on ikävä yllätys, jos vaikka laitteiden akut ovat tyhjä tuntiin alussa.

H4: Se kunnossapito ja huolehtimine ku ne on yhteiskäytössä ni seki oon aika sellane haasteellinen asia. Et sit ku tota.. ite ku rupeet pitää tuntia ja aatteelet et nyt mie käyn hakemassa ni sielt puuttuu puolist laturit ja ne ei oo ladattu ja sit pyörität et mihi on puolet palloist kadonnu ja niinku sellasta.. mikä turhauttaa sit itteesä.

Näitä samoja haasteita oltiin nostettu esiin myös aikaisemmissa tutkimuksissa. Ensimmäisenä haasteena todettiin usein siis materiaalien ja rahan määrä. Tämä nousi esiin myös Israel ym. (2014, 271-273), tekemässä tutkimuksessa, jossa teknologian puute nähtiin ongelmaksi ja Makkia ym. (2018, 91-92) tekemässä tutkimuksessa, jossa he nimesivät resurssien puutteen ensisijaiseksi esteeksi ohjelmoinnin opetuksessa. Ilman resursseja ohjelmointi koettiin haastavaksi, joten materiaalien huolehtiminen kuntoon oli ensisijaista ohjelmoinnin opetuksen aloittamiseksi. Tässä tutkimuksessa kouluilla materiaalien ongelmat kuitenkin olivat jollain tasolla kunnossa, kaikilta kouluilta löytyi mahdollisuuksia ohjelmoinnin perustarvikkeiden käyttöön.

8.5.3 Teknologian muuttuva luonne

Haastateltavien mielestä yksi iso haaste oli oman tiedon päivittäminen ajan tasalla. Tähän kuului niin uusien sovellusten ja opetustapojen selvittäminen kuin laitehallintakin.

H2: "Ja sit siun pitäs, et jotta sie pystyt ittees kehittämään ni aina pitää ottaa aika iso harppaus eteenpäin niinkun. Ja tietyl taval just tää laitehallinta"

Taitojen ylläpitäminen jo itsessään koettiin haastavana. Jos itse ei tasaiseen tahtiin nähnyt vaivaa ja harjoittanut ohjelmointia, taidot unohtuivat. Uuden tiedon kerääminen ja itsensä kehittäminen vievät aikaa ja velvoittaa omaa

aktiivisuutta. Osa haastateltavista sanoi, että vaikka käytti monia eri väyliä tiedon keräämiseen, esimerkiksi sosiaalisen median kanavia, saattoi silti syntyä tunne, ettei vielääkään ole perillä siitä, mitä kaikkea tapahtuu. Aineistossa todettiin, että jo kahden vuoden välinpitämättömyys saattaa aiheuttaa sen, että ei oikein tiedä mitä ohjelmoinnin saralla nyt tapahtuu. Koelherin & Mishran (2008, 6) mukaan sovelluksia kehitetään jatkuvasti ja niiden käyttötarkoitukset eivät ole aina vakiintuneita tai itsestään selviä. Usein sovellusten sisältö tai käyttötavat eivät edes täysin avaudu tavallisille ihmisille. Tästä syystä ne saattavat olla joillekin vaikeasti lähestyttäviä, ja niiden kehityksen perässä liikkuminen vaatii ahkeruutta, valmiutta muutokseen ja uskallusta kokeilla. (Koelher & Mishra 2008, 6.)

Aineistossa oli myös havaittavissa turhautumista sen takia, että koulu ei ole kehityksessä ajallaan. Nähtiin, että ulkomaailma kehittyy nopeampaa tahtia kuin koulu, että kouluun asiat tulevat vähän jäljestäpäin. Uusia, ajan tasalla olevia oppimateriaalejakin saattoi olla tarjolla, mutta ne piti itse etsiä ja moni näistä eivät olleet suomen tai englannin kielellä.

8.5.4 Asennehaasteet

Haasteisiin kuului luonnollisesti myös opettajien asenteet. Opettajien mielestä ohjelmointiin tutustuminen saattoi olla hankalaa, jos oma ennakoasenne oli huono. Nähtiin tärkeäksi, että opettaja tiedosti ohjelmoinnin opetuksen syyt alakoulussa, sillä se helpotti omaa motivointia.

H4: No kyl se varmaa sellaselle jol ei oo mitään kontaktipintaa siihen ni on et ennakko asenne saattaa olla vähä semmone vääränlaina ja ei ehkä oikeesti sit tajua et kuin helpost voi alottaa. Sit niinku ettei sen tarvii heti olla jollai ihmekielel käydä jotain tekstipohjast tuottamal vaan se voi olla tosi yksinkertainen juttu ja sekin on jo sitä kohti menemistä.

Opettajilla saattoi olla ohjelmoinnista sellainen kuva, että se on erittäin vaikeaa, vaikka alakoulussa ohjelmointi pysyy melko helpoissa kokonaisuuksissa. Opettajien ikä mainittiin myös ohjelmointiin perehtymisen haasteena. Näkemys oli, että vanhemmille opettajille saattoi olla vaikeampaa opetella ohjelmointia ja että he tarvitsisivat enemmän koulutusta. Syinä tähän kerrottiin vanhempien

opettajien vaikeus tarttua ohjelmointiin aiheena. Digiajan peruskoulu -selvityksessä testattiin opettajien ICT-taitoja. Tässä testissä nuoret opettajat pärjäsivät kaikilla osa-alueilla paremmin, kuin iäkkäämmät opettajat. (Kaarakainen ym. 2017, 30.) Samassa selvityksessä ilmeni, että nuoret opettajat ovat saaneet enemmän digitäydennyskoulutusta ja iäkkäämmät opettajat toivoivat enemmän koulutusta (Kaarakainen ym. 2017 58).

H3: Et sit moni ehkä aattelee, tai sit monilla saattaa olla ehkä enempi sitte, oisko joillaki vanhemmilla opettajilla enemmän semmone, et niiku vaikee tarttua siihe. Ku ite kuitenkin jo sillee, tai sillee aika tuore koulutus.

Asenteeseen liittyi siis myös se, että ohjelmoinnin saralla kokematon opettaja ei tuntenut itseään itsevarmaksi opettaa muita. Oman osaamisen puute koettiin haasteeksi. Tähän liittyi pelko, että opettaa asioita väärin, jos ei tiennyt tarpeeksi aiheesta. Vaikka opettajat tiedostivat, ettei ohjelmointi ole niin vaikeaa kuin miltä se kuulostaa, opettaja ei tuntenut itseään päteväksi opettaa sitä ilman hyvää aiheentuntemusta.

H6: *“Jos oppisin, jos tuntisin itseni tosi itsevarmaksi ohjelmoinnin osalta, voisinkin, mutta siltä ei tunnu nyt.”*

Yksi iso haaste oli siis opettajien oma osaaminen, joka nostettiin esille tämän tutkimuksen aineistossa. Jos opettaja ei uskonut omiin taitoihinsa, kynnys lähteä opettamaan oli korkea. Opettajat tiedostivat tämän ja kuten Makkia ym. (2018, 91-92) toteavat, kouluttamalla opettajia ohjelmoinnissa, opettajien asenteessa opettaa ohjelmointia nähtiin positiivista muutosta. Israel ym. (2014, 271-273) tutkimuksessa todettiin myös se, että mitä enemmän opettaja luotti taitoihinsa, sitä pätevämmiltä heistä tuntui opettaa sitä.

8.5.5 Oppimisen arviointi

Kouluissa oli huoli siitä, miten oppiminen varmistetaan ohjelmoinnin kautta. Ohjelmointia eheyttäessä eri oppiaineisiin oli opettajien mielestä tärkeää varmistaa, että oppilailta tapahtui oppimista myös eheyttävän oppiaineen osalta.

H2: *“Vitos-kutosilla alkaa olla eri oppiaineiden sisällöt niin vaativaa, et miten nihi, ei pysty yhdistämään tätä ohjelmointia, vaikka nii”*

Haasteena nähtiin yhdistää ohjelmoinnin aihealueet muiden oppiaineiden ainesisältöihin. Tavoitteena tässä tilanteessa olisi tietenkin, että molempien aiheiden osalta tapahtuisi syvällistä oppimista, jotta oppiminen ei jäisi pinnalliseksi. Taina (2015) esittää, että ohjelmoinnin opetuksen haasteena on se, että jos ohjelmoinnissa ei päästä pintaa syvemmälle, käsiteltävä asia jää helposti yksittäisten aiheiden ulkoa opetteluksi, eikä syvällistä oppimista tapahdu. Osa haastateltavista ei nähnyt tätä suurena haasteena, vaan perusteli, että ohjelmointia voi linkittää arkisiin toimintoihin, jolloin ohjelmoinnillinen ajattelu kehittyy jatkuvasti.

H5: Et tota, on se sit niinku et mihin sen haluu linkittää ja sit toisaaalta laaja-alasestiha pitäis olla opiskelutaitone, eihä sillo pitäs mistään mennä mitään pois, vaan nehä pitäis olla ajateltu että se juttu niinku liittyy kaikkeen tietyl tapaa

Ohjelmoinnin ei siis pitäisikään olla mikään erillinen kokonaisuus, vaan kuulua osana laajempiin, elämässä tarvittaviin taitoihin.

8.6 Tulosten koonti

Aineistosta nousi esiin viisi teemaa: ohjelmoinnin opetus alakoulussa, koulutus ja tuki, kouluyhteisön asenne ohjelmointia kohtaan, ohjelmoinnin merkitys osana opetusta, sekä ohjelmoinnin opetuksen haasteet. Seuraavaksi kootaan lyhyesti keskeiset tutkimustulokset.

Ohjelmointia opetetaan määrällisesti melko vaihtelevasti, riippuen opettajan kiinnostuksesta. Ohjelmoinnin opetuksen vähimmäismääränä ajateltiin olevan sen perusteet. Noin puolet kokivat haastavaksi ohjelmoinnin eheyttämisen osaksi muita oppiaineita ja he pitivät ohjelmoinnin mieluummin omana opetettavana aiheena.

Aineiston mukaan koulutuksia ohjelmoinnista on hyvin tarjolla, mutta kouluajan sisällä olevia koulutuspäiviä on rajallisesti. Tällöin opettajan täytyy tehdä valinta, minkä koulutuksen kokee itselleen kaikkein tarpeellisimmaksi. Tässä valinnassa ohjelmointikoulutukset jäivät usein välistä. Koulun tvt-vastaavien tukea pidettiin ensisijaisen tärkeänä tukimuotona.

Ohjelmoinnin opetuksessa opettajia motivoi oppilaiden innostus aihetta kohtaan. Yleinen mielipide oli, että oppilaista ohjelmointi on mielenkiintoista ja he pitävät siitä. Opettajan oma kiinnostus ohjelmointia kohtaan oli vahvasti yhteydessä siihen, kuinka paljon hän ohjelmointia opettaa. Mielenkiintoa oppia motivoi lähinnä itsensä kehittämisen halu.

Aineistosta kävi ilmi, että kaikkien haastateltavien mielestä ohjelmoinnilla on paikkansa koulumaailmassa. Osa opettajista oli kuitenkin sitä mieltä, että ohjelmoinnin opetus saattaa viedä aikaa joltakin tärkeämmältä aiheelta oppia. Tällöin ohjelmoinnin rooli saattoi olla hauskana lisänä koulun arjessa. Ohjelmoinnin opetusta perusteltiin nyky-yhteiskunnan tarpeilla sekä sen kautta opittavilla taidoilla.

Ohjelmoinnin opetuksen vahvimpana haasteena miellettiin, mistä löytää aikaa ohjelmoinnille koulun hektisessä arjessa. Lisäksi mainittiin, että ohjelmoinnin opetus saattaa lisätä opettajien työmäärää. Ohjelmoinnin opetuksen yhdeksi haasteeksi miellettiin raha; monet ohjelmointiin liittyvät laitteet ovat kalliita, eikä kouluilla ole varaa niihin. Opettajat olivat myös huolissaan siitä, että teknologian kehittyessä heidän on vaikea pysyä siinä mukana. Lisäksi ohjelmoinnin kautta arviointi koettiin haastavaksi.

9 POHDINTA

9.1 Tutkimustulosten tarkastelu ja johtopäätökset

Tässä tutkimuksessa oli tarkoituksena selvittää opettajien käsityksiä ohjelmoinnin opettamisesta alakoulussa. Esioletuksemme oli, että opettajat suhtautuvat ohjelmoinnin opettamiseen melko kriittisesti ja jopa vastahakoisesti. Tutkimuksessa halusimme selvittää, onko tämä ennakkokäsitys oikea ja jos on, mitä syitä negatiiviselle asenteelle on. Halusimme tietää, onko ohjelmoinnista tarjottu riittävää koulutusta ja tukea, ja näin ollen onko opettajilla valmiuksia opettaa ohjelmointia. Lisäksi meitä kiinnosti, kuinka ohjelmoinnin opetus on kouluissa järjestetty, lisääkö ohjelmoinnin opetus merkittävästi opettajien työmäärää sekä kiinnostaako ohjelmoinnin opetus opettajia.

Tutkimukseen osallistujat suhtautuivat ohjelmointiin omasta ennakkoletuksestamme poiketen kuitenkin melko myönteisesti. Jokainen opettaja osasi perustella, miksi ohjelmointi on otettu osaksi opetusta ja olivat sitä mieltä, että ohjelmoinnilla on paikka koulussa. Osa haastateltavista kuitenkin oli sitä mieltä, että ohjelmoinnin opettaminen ei ole kovinkaan tärkeää. Tutkimustuloksia tarkastellessa on otettava huomioon, että haastateltavat on hankittu yhteistyössä Innokas-verkoston (Innokas.fi 2018) kanssa, joten haastatteluun on saattanut osallistua tieto- ja viestintäteknologiaan painottuneempia opettajia. Lisäksi kolme haastateltavaa toimi oman koulunsa tvv-vastaavana tai ohjelmointikouluttajana, joka on voinut ohjata haastateltavien näkemyksiä. Pears, ym. (2017,147) tutkimuksessa esimerkiksi opettajista suurin osa koki materiaalin määrän tieto- ja viestintäteknologian puolella riittäväksi, mutta tilasto on saattanut vääristyä sen takia, että suurin osa vastaajista oli myös tieto- ja viestintäteknologian tai tietotekniikan aineen opettajia. Tutkimustulosten mukaan ohjelmoinnin opetus on pitkälti opettajan omasta kiinnostuksesta ja asenteesta riippuvainen.

Ohjelmoinnin opetus kouluissa tapahtui monilta osin samankaltaisesti toiminnallisesti ja helppoja robotteja hyödyntäen. Tässä oli huomattavissa se, että

uutena aiheena ohjelmoinnin opetukseen oli luultavastikin tarjottu samanlaisia koulutuspaketteja ja materiaalia tietynlaiseksi starttipaketiksi. Ne opettajat, joilla oli omaa kiinnostusta alaa kohtaan, olivat selvittäneet huomattavasti enemmän ja syvemmälle ohjelmoinnin opetustapoja, jolloin ohjelmoinnin opetus oli monipuolisempaa. Tutkimuksessa oli huomattavissa se, että opettajat luottivat tv-tv-vastaaviinsa ohjelmointiin liittyvissä asioissa, mutta samoin yksi haastateltavista totesi, että hänen erikoisalaansa oli joku muu aihe. Tältä opettajalta olisi varmasti saanut apua hänen omaan erikoisalaansa liittyvissä asioissa. Näin ollen ei ole tarpeellista, että kaikki opettajat pyrkisivät ohjelmoinnin mestareiksi, vaan vain muutama. Tällainen ajattelu asiantuntijoista ehkä säästääkin opettajien aikaa ja mahdollistaa erikoistumisen paremmin eri aiheisiin. Tällaisesta tilanteesta hyödyn irti ottaminen riippuu kuitenkin paljon opettajien välisen tiedon vaihtumisesta. Osa opettajista olisikin toivonut tätä tietojen jakamista enemmän kollegoiden kanssa.

Aika tulee nykyisin useasti vastaan yleisessä keskustelussa, kun puhutaan koulusta. Se nostettiin myös tässä tutkimuksessa yhdeksi haasteeksi ohjelmoinnin opetuksessa. Jotkut opettajat olivat huolissaan siitä, riittääkö aika kaiken tärkeän oppimiseen, jos opetussuunnitelmaan lisätään uutta. Ne, jotka olivat kokeneempia ohjelmoinnin kanssa, eivät kokeneet aikaa kuitenkaan suureksi haasteeksi, sillä he pystyivät paremmin soveltamaan ohjelmointia osaksi muita oppiaineita ja pitämään sitä osana arjen toimintoja. Ohjelmoinnin osalta heikompiteitoisilla opettajilla oli selvästi ajatus siitä, että ohjelmointi olisi helpompi pitää omana aiheenaan erikseen muista, sillä ohjelmoinnin eheyttäminen osaksi muita oppiaineita nähtiin aikaa vieväksi prosessiksi. Tällöin saattaa syntyä tunne, että ohjelmointituntia varten täytyi jättää joku toinen tunti pois. Jos ohjelmointi oli vieras opettajalle, se saatettiin nähdä myös ristiriidassa perinteisten oppimistapojen kanssa. Suurin osa opettajista kuitenkin kertoi, että he ovat ainakin ideoineet, kuinka he voisivat vaikka kielen tai ympäristöopintunnilla käyttää ohjelmointia hyväksi.

Tästä herää kysymys toisesta tutkimuksesta nousseesta aiheesta, eli siitä, miten voidaan varmistaa, että oppimista tapahtuu. Yksi opettaja mielsi

ohjelmoinnin pelkäksi tavaksi oppia, eikä hän mieltänyt ohjelmoinnilla itsellään olevan hirveästi asiasisältöä mitä piti opettaa. Toinen opettaja taas oli huolissaan siitä, miten hän pystyy varmistamaan, että tunnilla opitaan esimerkiksi kielten tunnin aiheesta, että ohjelmoinnin asiasisällöistä. Myös opittava taito saattoi olla hankala arvioida, esimerkiksi miten oppilaan algoritminen ajattelu on kehittynyt. Haastatteluista noussutta ajattelun taitojen kehittämisen toteutumista ei ehkä pysty mittaamaan yhtä suoraviivaisesti, kuin esimerkiksi kertolaskujen osaamista. Tällöin täytyy olla tarkkana siitä, että ohjelmointia opetetaan tavoitteellisesti, eikä käytetä sitä vain hauskana vaihteluna.

Oppilaiden mukaanotto ja vastuuttaminen ohjelmoinnin opetuksessa kuulostaa mielenkiintoiselta ja uuden opetussuunnitelman huomioivalta tavalta opiskella. Ohjelmoinnin opetuksessa oppilailla tuntuu olevan automaattisesti aktiivisen osallistujan rooli, jolloin he ovat mukana suunnittelussa, luonnissa ja mahdollisesti myös arvioinnissakin. Muissa aihealueissa kuin tieto- ja viestintäteknologiassa ei ole ehkä havaittavissa samanlaista luottoa siihen, että oppilaat oppivat nopeammin tai osaavat paremmin kuin opettajat. Tämä johtuu ehkä siitä, että opettajat ajattelevat oppilaiden kasvaneen teknologian parissa ja käyttävät sitä luonnollisemmin kuin opettajat. Tähän liittyy esimerkiksi termi diginatiivi, jollaiseksi Valtonen (2011, 14-16) kuvaa ihmistä, joka oppii löydösten kautta, osaa käyttää sosiaalista mediaa työvälineenään ja ymmärtää hypertekstin luonnetta. Kuitenkin hänen tutkimuksessaan diginatiiveiksi luokiteltavat opiskelijat eivät yltäneet näissä taidoissa tasolle, jota heiltä olisi odotettu. Tästä voidaan päätellä, että kaikki eivät luontaisesti ehkä osakaan käyttää teknologiaa hyödykseen, vaikka ovatkin kasvaneet teknologian ympäröimänä. Ei siis pidä ohjelmoinninkaan osalta unohtaa, että opettajien tehtäviin kuuluu ohjata oppilaiden oppimista.

Opettajien kertomista ohjelmoinnin kautta opittavista taidoista moni oli sellaisia, mitä pystyi oppimaan muidenkin aineiden kautta, kuten ryhmätyötaidot ja pitkäjänteisyys. Yhden opettajan kielentämät ajattelun taidot ohjelmoinnin opetuksen pääpainona kuvasivat hyvin algoritmisen ajattelun huomioimista. Tosin usea opettaja totesi, ettei ohjelmointiin liittyvä ajattelu ollut

koulumaailmassa uutta, vaan siitä käytetään nyt uutta nimitystä. Opettajat kuvailivat ohjelmoinnin kautta opittaviin taitoihin ohjelmoinnin peruskäsitteisiin kuuluvia algoritmin käsitettä, peräkkäisyyttä ja ehdollisuutta (Luostarinen & Peltomaa, 2016, 106-107; Liukas & Mykkänen 2016, 21). Opettajat eivät käyttäneet taidoista näitä käsitteitä, josta huomaa sen, että kielentäminen on iso osa ohjelmointia. Kielentäminen olisi tärkeää myös Wingin (2008, 2720-2721) mielestä, koska silloin lapset oppivat ohjelmointia paremmin jo alakoulussa, eikä se tulisi heille uutena asiana myöhemmin elämässä.

Ohjelmoinnissa on paljon asioita, jotka arkikielessä ovat tuttuja, ja joita ei heti yhdistä ohjelmointiin. Algoritmit tulivat esiin ohjeiden annossa. Peräkkäisyyttä selitettiin sillä, että asiat tapahtuvat tietyssä järjestyksessä, eli yleensä peräjälkeen. Ehdollisuus nousi esiin taas syy-seuraus suhteen luonteessa. Niin kuin ihmiset toiminnassa, myös koneiden ohjelmoinnissa tietystä tapahtumasta seuraa tietty reaktio. Esimerkiksi silmukan ja funktion mukaanotto vastaavanlaisen maanläheisen ajattelun kautta ei olisi enää kovin hankalaa, jotka periaatteessa tarkoittavat vain saman tapahtuman toistoa ja erilaisia variaatiota toistoista. (Luostarinen & Peltomaa, 2016; Liukas & Mykkänen 2016.) Kaikki opettajat kuvailivat jollain tasolla tähän algoritmiseen ajatteluun liittyviä ajatusmalleja, joskaan he eivät käyttäneet niistä ohjelmoinnin sanastoa. Tämä saattoi johtua siitä, että niitä on helpompi ymmärtää puhekielellä, mutta on myös mahdollista, että opettajat eivät olleet täysin perillä ohjelmoinnin käsitteistä, joita tarvitaan varsinaisessa ohjelmakoodin rakentamisessa. Se, tarvitseeko näitä tietääkään alakoulun oppimäärään peilaten, riippuu hyvin paljon opettajasta itsestään. Opetussuunnitelman mukaan oppilaan tulisi kuitenkin osata ohjelmoida kuudennen luokan päätyttyä toimiva ohjelma graafisessa ohjelmointiympäristössä (T14) (POPS 2014, 239). Tällöin herää kysymys, pystyykö oppilas ohjelmoimaan, jos ei tiedä tarpeeksi ohjelmoinnin perusteista. Esimerkiksi ohjelmointiin vahvasti liittyviä funktioita ja muuttujia opetellaan matikan tunneilla läpi perusopetuksen, mutta toisaalta, kuinka moni oppilas ymmärtää, mitä funktiolla oikeasti tarkoitetaan.

Osa taidoista, joita opettajat nimesivät ohjelmoinnin kautta opittavaksi, kuten sosiaaliset taidot, eivät liittyneet pelkästään ohjelmointiin, vaan olivat taitoja, joita pystyi opettamaan missä vain aineessa. Tällöin on hyvä tiedostaa mitä opetuksella on tarkoitus opettaa, ettei mennä vain tietokoneilla puuhailuun, kuten eräs haastateltava ilmaisi. Niin kuin kaikessa oppimisessa, myös ohjelmoinnissa on huolehdittava siitä, että opiskellaan oikeita asioita. Ohjelmoinnin kohdalla nämä taidot ovat algoritmisen ajattelun taidot ja ohjelmoinnin perusteiden hallinta. On esimerkiksi tutkittu, että vain 10% Scratchiin ladatuista ohjelmista käyttää hyväkseen esimerkiksi toistuvuutta tai muuttujia, ohjelmoinnin perusominaisuuksia (Meerbaum-Salant, Armoni & Ben-Ari 2013, 240). Peruskoulun opetussuunnitelmassa ohjelmoinnin opetuksessa tosin riittää pysytellä aika maanläheisellä tasolla, mutta oppilaiden tulisi kuitenkin osata ohjelmoida jollain tasolla alakoulun jälkeen. Meerbaum-Salant ym. (2013, 261-262) mainitseekin oman tutkimuksensa jälkeen, että Scratch toimii hyvin ohjelmoinnin opetuksessa, kunhan opetus on hyvin suunniteltua.

Tyttöjen ja poikien rooli ohjelmoinnissa niin kuin muissakin aineissa pitäisi olla tasa-arvoinen opetussuunnitelmaankin nojautuen (POPS 2014, 18), mutta silti ohjelmoinnin osalla opettajien puheista oli havaittavissa tietynlaisia ennakko-oletuksia. Tärkeää heidän puheissaan oli kuitenkin se, että he itse tiedostivat, että he eivät tue tällaista ajattelua, ja että he eivät ole sitä omissa oppilaissaan huomanneet. Usean muunkin oppiaineen kohdalla on mediassa keskustelua sukupuoleen liittyviä oletuksia, kuten matikassa ja kielissä, joihin pyritään vaikuttamaan tasa-arvoistavasti. Toivon mukaan näitä oletuksia saadaan häivytettyä nykyisen opetussuunnitelman myötä, jotta oppilaiden sukupuoli ei vaikuttaisi heidän opiskelumotivaatioonsa tai jatko-opinto suunnitelmiin. Tutkimuksemme perusteella onneksi sukupuolten välillä ei nähty eroa oppilaiden innostuksella ohjelmointia kohtaan, vaan kaikki pitivät sitä mielekkäänä aiheena.

Teknologia on haastava ala sen jatkuvan kehittymisen myötä. Koulumaailmassa tämä aiheuttaa ongelmia laitteiston uusimisessa ja opettajien kouluttamisessa eli ajan löytämisessä. Jälkimmäisessä myös opettajien

mielenkiinnon puuttuminen voi tietenkin olla esteenä. Ajan ja rahan haasteet ovat varmaan aika samankaltaisia kaikilla aloilla eikä koulussakaan keskity vain teknologiahankkeisiin.

Uutena asiana koulussa ohjelmoinnin mukaanoton tuloksia ja sudenkuoppia voi olla hankala arvioida. Vaikuttaisi siltä, että maissa, jossa tietotekniikka on ollut oma aineensa, sen taso on päällisin puolin parempi kouluissa kuin niissä, joissa se ei ole ollut oma aineensa (Burke & Kafai 2014, 115-116). Opetuksen taso ja painotukset tietenkin vaihtelevat paljon myös maittain. Ohjelmointikin on saattanut lähivuosien aikana muuttua paljon laitteiston uudistumisen takia, Suomessakin hyvin tunnetun Scratchin ensimmäinen versio on julkaistu 2000-luvun puolella. Useassa maassa ohjelmointi on kuitenkin otettu vasta lähiaikoina mukaan opetukseen. Niissäkään maissa, missä tietotekniikka on ollut oma aineensa, ohjelmointia ei olla opetettu varmaan niin, kuin sitä nykyään haluttaisiin opettaa alakoulun puolella. Ohjelmoinnin mukaanotosta opetussuunnitelmaan ei siis ole vielä paljon tutkimusta, varsinkaan pitkäkestoista tutkimusta, jossa päästäisiin tutkimaan alakoulussa opettajien taitojen hyötyä esimerkiksi jatko-opintovaiheessa. Tässä vaiheessa ohjelmoinnin opetusta koskeva teoria ja opettajien näkemykset ovat paljolti havaintoja, joiden tueksi ei ole paljoa tutkimustuloksia pitkällä aikavälillä.

9.2 Tutkimuksen luotettavuus ja jatkotutkimuksen mahdollisuudet

9.2.1 Tutkimuksen rajoitukset ja luotettavuus

Tutkimusta tehdessä tutkimukseen liittyy aina joitakin rajoitteita ja haasteita, jotka on hyvä ottaa huomioon tutkimustuloksia tulkittaessa. Tässä tutkimuksessa ei ole ollut tarkoituksena kerätä yleistettäviä faktoja, vaan päästä syvemmälle opettajien ajatuksiin ohjelmoinnin opettamisesta alakoulussa. Seitsemän haastattelun aineisto on suhteellisen pieni ja haastattelut pysyivät ajallisesti melko lyhyinä. Aihe on itsessään sellainen, ettei haastatteluissa lähdetty paljoa rönsyilemään ohi aiheen, vaan pysyttiin melko tiiviisti itse

aiheessa, jolloin lähes kaikki materiaali pystyttiin hyödyntämään. Aineiston keruu rajattiin yhteen tiettyyn kaupunkiin ja lisäksi aihe on hyvin tuore koulumaailmassa, joka rajasi haastateltavien määrää. Tällä otannalla saimme kuitenkin monipuolisia tuloksia ilmiöstä. Rajasimme aiheemme tarkoituksella suhteellisen kapeaksi, jotta ilmiössä päästäisiin pintaa syvemmälle.

Tutkimuksessa tehtiin yhteistyötä Innokas-verkoston (Innokas.fi 2018) kanssa. Tutkimuksen luotettavuuden kannalta on otettava huomioon, että tutkimukseen osallistuneet haastateltavat on valittu yhdessä Innokas-verkoston kanssa, jolloin haastatteluihin on saattanut osallistua valtaosasta poikkeavia opettajia. Tuloksia tarkastellessa on myös hyvä ymmärtää, että tutkimus perustuu haastatteluihin eikä havainnointiin, joten tutkijat eivät ole voineet todentaa käytännön havaintojensa kautta haastatteluissa esiin nousseita asioita.

Tutkijatriangulaatio vahvisti tutkimuksesta nousevien näkökulmien huomioimista ja rikasti ajatustenvaihtoa, kun tutkijoita oli kaksi (Tuomi ja Sarajärvi, 2018, 168). Riski omille tulkinnoille ja mielipiteiden nousemiselle oli pienempi, muttei olematon. Kirjoitusprosessin aikana kirjoitimme yhdessä ja erikseen, tekstejä vaihdellen, muokaten ja kommentoiden, jolloin molempien ääni on kuuluvissa koko tutkimuksen ajan. Tutkijatriangulaatio oli vahvasti läsnä myös haastattelutilanteissa. Yhtä haastattelua lukuunottaamatta kaikki haastattelut suoritettiin niin, että molemmat tutkijat olivat paikalla. Tämä mahdollisti sen, että toinen otti vastuun kysymysten esittäjänä ja toinen esitti lisäkysymyksiä ja kirjasi mahdollisia muistiinpanoja.

Haastattelutilanteissa vältimme keskustelun ohjaamista tiettyyn suuntaan ja toimimme tilanteessa aktiivisina kuuntelijoina. Aineistoa analysoidessamme keskustelimme jatkuvasti ja pyrimme välttämään omien tulkintojen nousemista tuloksiin. Haastatteluiden osalta tutkimuksen luotettavuudesta on otettava huomioon tutkijoiden melko vähäinen kokemus haastattelutilanteista. Huomasimme, että ensimmäisissä haastatteluissa jännitimme melko paljon, jolloin osa mahdollisista lisäkysymyksistä jäi kysymättä. Viimeisimmissä haastatteluissa tilanne oli jo paljon luontevampi ja keskustelunomaisempi.

9.2.2 Tulosten hyödyntäminen ja jatkotutkimushaasteet

Yhteistyö Innokas-verkoston kanssa mahdollistaa tutkimustulosten hyödyntämisen myös jatkossa. Tutkimuksessa pyritään saamaan opettajien ääni kuuluviin, jolloin mahdollisesti tarvittavaa täydennyskoulutusta ja tukea pystyttäisiin järjestämään ja kehittämään ohjelmoinnin osalta eteenpäin. Lisäksi tutkimustulokset ovat hyödynnettävissä alakoulun opettajien keskuudessa, jotka pohtivat opetuksessaan samoja asioita, kuin tutkimuksessamme.

Tutkimuksen haastattelut tehtiin vain yhden kaupungin sisällä mutta ohjelmoinnin opetus koskettaa jokaista suomalaista luokanopettajaa, joten tutkimus on tarpeellinen koko Suomessa. Toivottavaa olisi, että tämä tutkimus lisäisi jatkotutkimusta aiheesta. Luonnollinen tapa tehdä jatkotutkimusta olisi haastatella lisää luokanopettajia ympäri Suomea. Tämä mahdollistaisi ohjelmoinnin opetuksen kehittämisen koko maassa. Kaupungeittain saattaa olla suuriakin eroja ohjelmoinnin opetuksen osalta. Tässä kaupungissa ohjelmoinnin opetukseen on saattanut vaikuttaa vahvasti kaupungissa aktiivisesti toimiva Innokas-verkosto sekä Mediakeskus. Tällaisia tukimuotoja ei välttämättä ole yhtä vahvasti muissa kaupungeissa, tai vastaavasti koulun ulkopuolista tukea voi olla vielä enemmän tarjolla. Tutkimukseen voisi olla hyvä valikoida opettajia, joilla ei ole koulussa tvt-vastaavan tai -kouluttajan roolia.

Lisäksi olisi mielenkiintoista haastatella samoja opettajia esimerkiksi kaksi vuotta myöhemmin, jolloin pystyisi vertailemaan millaista muutosta ohjelmoinnin opetuksen suhteen on tapahtunut. Yksi jatkotutkimusmahdollisuus olisi seurata kyseisten opettajien ohjelmoinnin opetusta, jolloin aiheesta oltaisiin päästy entistä syvemmälle ja saatu selville vastaavatko haastateltavien sanat tekoja.

LÄHTEET

- Ahonen, S. 1994. Fenomenografinen tutkimus. Teoksessa Laadullisen tutkimuksen työtapoja. (Toim.) L. Syrjälä, S. Ahonen, E. Syrjäläinen & S. Saari. Helsinki: Kirjayhtymä
- Benitti, F., B., V., 2012. Exploring educational potential of robotics in schools: A systematic review. *Computers & Education*, 58, 978-988
- Burke & Kafai 2014. Connected code: why children need to learn programming. The Cambridge, Massachusetts; London, England, MIT press
- Department of Education. 2013. National curriculum in England: design and technology programmes of study. Saatavana sähköisesti: <https://www.gov.uk/government/publications/national-curriculum-in-england-design-and-technology-programmes-of-study>
- Edu. (15.09.2016). Ohjelmoinnin opetukseen liittyvien käsitteiden määritelmät peruskoulun opettajille. Saatavilla WWW-muodossa <URL: http://www.edu.fi/materiaaleja_ja_tyotapoja/tvt_opetuksessa/ohjelmoi/nti/kasitteet Luettu 20.1.2018
- Elo, S. & Kyngäs, H. 2008. The Qualitive Content Analisis Process. *Journal of Advanced Nursing* 62(1), 107-115.
- Ekonoja, A. 2014. Oppimateriaalien kehittäminen, hyödyntäminen ja rooli tieto- ja viestintätekniiikan opetuksessa. Jyväskylä University Printing House. Jyväskylä
- Eskola, Jari & Suoranta, Juha 1996. Johdatus laadulliseen tutkimukseen. Vastapaino, Tampere.
- Eurodice, 2012. Key Data on Education in Europe. European Comission Saatavilla WWW-muodossa <URL: http://eacea.ec.europa.eu/education/eurydice/documents/key_data_series/134EN.pdf
- Fisher, L., Byrne, J. & Tangney, B. 2016. Teacher Experiences of Learning Computing using a 21st century Model of Computer Science Continuing

- Professional Development. *Proceedings of the 8th International Conference on Computer Supported Education*, 2, 273-280
- Heinonen, M. 2008. Johdatus ohjelmointiin. Saatavilla WWW-muodossa <URL: <http://ohjelmointi.medianurkka.com/wp-content/uploads/2010/04/ohjelmointi.pdf> Luettu 6.6.2018
- Hietikko, P., Ilves, V. & Salo, J. 2016. Askelmerkit digiloikkaan. OAJ:n julkaisusarja 3, 1-46.
- Hirsjärvi, S. & Hurme, H. 2001. Tutkimushaastattelu. Helsinki: Yliopistopaino.
- Hirsjärvi, S. & Hurme, H. 2009. Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Helsinki: Gaudeamus. University Press.
- Huusko, M. & Paloniemi, S., 2006. Fenomenografia laadullisena tutkimussuuntauksena kasvatustieteissä. *Kasvatus*, 2, 162-173
- Hyvönen, M., Lappalainen V., & Lakanen, A. 2013. Ohjelmointi 1: C#. Jyväskylä. Jyväskylän yliopistopaino.
- Israel, M., Pearson, J., Tapia, T., Wherfel, Q. & Reese, G., 2015. Supporting all learners in school-wide computational thinking: A Cross-case qualitative analysis. *Computers & Education*, 82, 263-279
- Innokas Network. (13.8.2018). *Innokas.fi*. Saatavilla WWW-muodossa <URL: <https://www.innokas.fi/> Luettu 13.8.2018
- Koodiaapinen. (27.1.2018). *Koodiaapinen.fi*. (IT-kouluttajat ry) Saatavilla WWW-muodossa <URL: <http://koodiaapinen.fi/> Luettu 27.1.2018
- Karakainen, M., Karakainen, S., Tanhua-Piiroinen, E., Viteli, J., Syvänen, A. & Kivinen, A. 2017. Digiajan peruskoulu 2017 – Tilanearvio ja toimenpidesuosituksset. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 72/2017.
- Kalelioglu, F. 2105. A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behaviour*, 52, 200-210
- Karvonen, V-P. & Laukka P., 2016. Suomalaisten opettajien asenteita ja valmiuksia ohjelmoinnin opetukseen. *Kasvatustieteen pro gradu-tutkielma*. Oulun yliopisto. Opettajankoulutuslaitos. Saatavilla WWW-muodossa <URL: <http://jultika.oulu.fi/files/nbnfioulu-201602121186.pdf>

- Kettunen, J. & Tynjälä, P., 2018. Applying phenomenography in guidance and counselling research. *British Journal of Guidance & Counselling*, 46(1), 1-11
- Koelher, M. J. & Mishra, P., 2008. Introducing TPCK. Teoksessa AACTE (toim.) Handbook of Technological Pedagogical Content Knowledge (TPCK) for education. New York: Routledge. 3-32
- Lappalainen, V. & Viitanen, S., 2012. Ohjelmointi 2. Jyväskylä. Jyväskylän yliopistopaino.
- Liukas, L. & Mykkänen, J., 2014. Koodi 2016- Ensiapua ohjelmoinnin opettamiseen peruskoulussa. Helsinki, Lönnberg Print
- Luostarinen, A. & Peltomaa, I., 2016. Reseptit OPSin käyttöön: opettajan opas työssä onnistumiseen. PS-Kustannus, Jyväskylä
- Majaranta, P. (2002). Ohjelmointi on helppoa - lapsikin sen osaa. Teoksessa P. Hietala & S. Ovaska (Toim.) Lasten Käyttöliittymät. Tampere: Tampereen yliopiston tietojenkäsittelytieteiden laitos.
- Makkia, T., O'Neal L. J., Cottena S. R. & Rikarda, R.V., 2018. When first-order barriers are high: A comparison of second- and third-order barriers to classroom computing integration. *Computers & education*, 120, 90-97
- Malmberg, M. 2013. Suomen koulujen it-opetus Jordanian tasolla - uutta mallia haetaan Virosta. Yle Uutiset. Saatavilla WWW-muodossa <URL: <https://yle.fi/uutiset/3-6885099> Luettu 4.2.2018
- Marttala, L. 2017. Tietotekniikan valtakunnallisten oppisisältöjen toteutuminen Keski-Suomen peruskoulujen opetuskäytänteistä. *Tietotekniikan Pro gradu-tutkielma*. Jyväskylän yliopisto. Informaatioteknologian tiedekunta Saatavilla WWW-muodossa <URL: <https://jyx.jyu.fi/dspace/bitstream/handle/123456789/55992/URN%3aNBN%3afi%3ajyu-201711234351.pdf?sequence=1>
- Meerbaum-Salant, O., Armoni, M. & Ben-Ari, M., 2013. Learning computer science concepts with scratch. *Computer Science Education*, 23 (3) 239-264
- Micro:bit Educational Foundation, 2016. Micro:bit.org. Saatavilla WWW-muodossa <URL: <http://microbit.org/guide/> Luettu 6.6.2018

- Norrena, J. 2015. Innostava koulun muutos - Opas laaja-alaisen osaamisen opetukseen. Ps-kustannus. Jyväskylä.
- Norrena, J. 2013, Opettaja tulevaisuuden taitojen edistäjänä - "Jos haluat opettaa noita taitoja, sinun on ensin hallittava ne itse". Väitöskirja. Jyväskylän yliopisto. Informaatioteknologian tiedekunta.
- Opetushallitus. 2009. Näkökulmia perusopetuksen tavoitteisiin ja tuntijakoon. Opetushallituksen muistio perusopetuksen yleisten tavoitteiden ja tuntijaon uudistustyöryhmälle. Saatavilla WWW-muodossa <URL: http://www.oph.fi/download/118604_Tuntijako.pdf
- Opetushallitus. 2010. Tieto- ja viestintätekniiikan opetuskäyttö ja tietotekniikan opetus. Opetushallituksen muistio perusopetuksen yleisten tavoitteiden ja tuntijaon uudistustyöryhmälle. Saatavilla WWW-muodossa <URL: http://www.oph.fi/download/124371_TVT_muistio_tuntijakoa_varten.pdf
- Pears, A., Dagiene, V. & Jasute, E., 2017. Baltic and Nordic K-12 Teacher Perspectives on Computational Thinking and Computing. Teoksessa *Informatics in Schools: Focus on Learning Programming* (toim.) Dagiene, V. & Hellas, A. Switzerland, Springer Nature
- POP 2014 = Perusopetuksen opetussuunnitelman perusteet 2014. Helsinki: Opetushallitus. Saatavilla WWW-muodossa <URL: http://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf
- Quahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A & Lahmine, S. 2014. Learning Basic Programming concepts by creating games with Scratch programming environment. *Procedia-Social and behavioiral sciences*, 191, 1479-1482
- Robbo Europe. Aii Corporation Oy. Saatavilla WWW-muodossa <URL: <https://www.robbo.world/> Luettu 6.6.2018
- Rydman, A. 2014. Ministeri Kiuru: Ohjelmointi peruskoulun opetussuunnitelman perusteisiin. Verkkouutiset. Saatavilla WWW-muodossa <URL: <https://www.verkkouutiset.fi/ministeri-kiuru>

- ohjelmointi-peruskoulun-opetussuunnitelman-perusteisiin-14614/ Viitattu 27.1.2018
- Sphero 2017. Sphero edu. Saatavilla WWW-muodossa <URL: <https://www.sphero.com/education> Luettu 6.6.2018
- Standl, B., 2017. Solving Everyday Challenges in a Computational Way of Thinking. Teoksessa *Informatics in Schools: Focus on Learning Programming* (toim.) Dagiene, V. & Hellas, A. Switzerland, Springer Nature
- Suomen Robotiikkayhdistys oy, 1999. Robotiikka. (Toim) Kuivanen. R. Vantaa, Tummavuoden kirjapaino (vai kustantaja Talentum oyj/Metallitekniikka?)Taina, J. 2015 Peruskoulun ohjelmointiopetus. Matematiikkalehti Solmu. Saatavilla WWW-muodossa <URL: <https://matematiikkalehtisolmu.fi/2015/2/ohjelmointiopetus.pdf> Luettu 20.1.2018
- Taina, J. 2015 Peruskoulun ohjelmointiopetus. Matematiikkalehti Solmu. Saatavilla WWW-muodossa <URL: <https://matematiikkalehtisolmu.fi/2015/2/ohjelmointiopetus.pdf> Luettu 20.1.2018
- Tedre, M. & Denning, P., J. 2016. The Long Quest for Computational Thinking. *Proceedings of the 16th Koli Calling Conference on Computing Education Research*, 120-129.
- Terrapin 2016. Bee-bot.us. Saatavilla WWW-muodossa <URL: <https://www.bee-bot.us/bee-bot.html> Luettu 6.6.2018
- The LEGO group 2018. Mindstorms. Saatavilla WWW-muodossa <URL: <https://www.lego.com/en-us/mindstorms> Luettu 6.6.2018
- Tuomi, J. & Sarajärvi, A. 2009. Laadullinen tutkimus ja sisällönanalyysi. Helsinki. Kustannusosakeyhtiö Tammi
- Tuomi, J. & Sarajärvi, A. 2018. Laadullinen tutkimus ja sisällönanalyysi. Helsinki. Kustannusosakeyhtiö Tammi
- Vahtivuori-Hänninen, S., Halinen I., Lavonen, J. & Lipponen, L. A. (2014). National core curriculum for basic education (2014) and technology as an integrated tool for learning. Teoksessa H. Niemi, J. Multisilta, L. Lipponen

- & M. Vivitsou (eds.) *Finnish Innovations & Technologies in Schools*. Rotterdam: Sense publishers, 21–32.
- Valtonen, T., Pontinen, S., Kukkonen, J., Dillon, P., Väisänen, P. & Hacklin, S. 2011. Confronting the technological pedagogical knowledge of Finnish Net Generation student teachers. *Technology, Pedagogy and Education* 20(1), 3-18.
- Vaismoradi, M., Jones, J., Turunen, H., & Snelgrove, S. 2016. Theme development in qualitative content analysis and thematic analysis. *Journal of Nursing Education and Practice*, 6(5), 101-110.
- Vilkka, H. 2007. *Tutki ja mittaa- määrällisen tutkimuksen perusteet*. Jyväskylä: Gummerus.
- Vänskä, O. 2012. Viron kunnianhimoinen tavoite: kaikki 7-vuotiaat aloittavat ohjelmointiopinnot. Tivi, koulutus. Saatavilla WWW-muodossa <URL: <https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/79573/perusopetuksen%20oppimisymp%C3%A4rist%C3%B6jen%20digitalisaation%20nykytilanne.pdf?sequence=1>
- Wing, J., M. 2008. Computational thinking and thinking about computing. *Philosophical transactions of the royal society*. 366, 3717-3725.
- Van-Roy, P. & Haridi, S. 2004. *Concepts. Techniques and Models of Computer Programming*. Massachusetts, the MIT Press
- Futschek, G. 2006. Algorithmic Thinking: The Key for Understanding Computer Science. *ISSEP 2006, LNCS 4226*, 159-168.
- Michaelson, G. 2015. Teaching Programming with Computational and Informational Thinking. *Journal of Pedagogic Development* 5(1), 51-65.
- Papert, S. 1980. *Children, Computers and Powerful Ideas*. New York, Basic Books.

LIITTEET

LIITE 1.

Haastattelurunko

1. Mitä tiesit ohjelmoinnista ennen uutta opetussuunnitelmaa?
2. Miten opetat ohjelmointia? Millä tunneilla opetat ohjelmointia? (mitä ohjelmia, mitä laitteita, millaisia harjoitteita)
3. Miten ohjelmointi integroituu muuhun opetukseen?
4. Miksi ohjelmoinnin opetus on mielestäsi tärkeää/ei tärkeää?
5. Millaisia resursseja koulullanne on tarjota ohjelmoinnin opetukseen?
6. Mistä ja millaista tukea olet saanut ohjelmoinnin opettamiseen? (koulutukset koulun kautta, FB-ryhmät, ystävät, Itsenäinen tutustuminen jne)
7. Kerro suhtautumisestasi ohjelmoinnin opetukseen alakoulussa.
8. Mitkä asiat motivoivat sinua ohjelmoinnin opettamisessa?
9. Millaista lisäarvoa ohjelmoinnin opetus tuo? Vai onko muulta opetukselta pois?
10. Millaista keskustelua ohjelmoinnin opettaminen on herättänyt työyhteisössänne? (tuntuuko työläältä, viekö ylimääräistä aikaa?)

LIITE 2.

Viite

Opinnäytetyön tutkimuslupahakemus 12.3.2018

Tutkimusluvan myöntäminen

Myönnän luvan Anniina Pyykösen ja Jenni Makkosen haastattelututkimuksen tekemiseen opinnäytettä varten aiheesta "Kuinka ohjelmoinnin opetus on otettu vastaan luokanopettajien keskuudessa?".

Tutkimuksen tekijä ei saa käyttää tutkimuksen aikana saamiaan salassa pidettäviä tietoja henkilökunnan, oppilaan tai hänen läheistensä vahingoksi tai halventamiseksi tai sellaisten muiden etujen loukkaamiseksi, joiden suojaksi on säädetty salassapitovelvollisuus. Tutkimuksen tekijä ei saa luovuttaa salassa pidettäviä henkilötietoja sivulliselle. Tutkimuksen tekijän tulee toimittaa maksutta tutkimusraportista yksi kappale [REDACTED] kaupungin kirjaamoon.

[REDACTED]
Allekirjoitus

[REDACTED]
Nimi

vs. opetustoimenjohtaja

[REDACTED] kaupunki

LIITE 3.

TUTKIMUSLUPA

Olemme Anniina Pyykönen ja Jenni Makkonen ja tekemässä pro gradu - tutkielmaamme. Tutkielmamme käsittelee ohjelmoinnin opetuksen vastaanot luokanopettajien keskuudessa. Tutkimuksemme aihe on merkityksellinen, ko ohjelmoinnin lisääminen osaksi opetussuunnitelmaa on tuore ja opettajien käsitysten tutkiminen auttaa kehittämään opetustyötä. Tutkielmamme ohjaaja toimii Mikko Vesisenaho, mikko.vesisenaho@jyu.fi

Aineisto kerätään haastattelemalla. Haastatteluaineisto ~~äänitallennetaan~~.

Tutkimukseen osallistuminen on vapaaehtoista, ja osallistumisen voi keskeytt missä tahansa vaiheessa tutkimusta. Tutkimusaineistoa käsitellään ja säilytetään luottamuksellisesti. Tutkimusaineistoa käytetään siten, etteivät henkilöt ole yksilöinä tunnistettavissa. Tutkimusaineisto säilytetään Jyväskylän yliopiston tutkimusaineistojen hallintaohjeen mukaan.

Annamme mielellämme lisätietoa tutkimuksesta. Voitte ottaa meihin yhteyttä sähköpostitse anmapyyk@student.jyu.fi tai puhelimitse 0442892224

Pyydämme suostumustanne osallistua tutkimukseemme.

SUOSTUMUS OSALLISTUA YLLÄ KUVATTUUN TUTKIMUKSEEN

- suostun
 en suostu

Aika ja paikka

Allekirjoitus ja nimenselvennys