



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Miro Nevala

Tekoälyn rooli aloittelevan ohjelmoijan apuna ja haasteena

Opinnäytetyö

Syksy 2024

Tradenomi (AMK), Liiketalous



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Tutkinto-ohjelma: Tradenomi (AMK), Liiketalous

Tekijä: Miro Nevala

Työn nimi: Tekoälyn rooli aloittelevan ohjelmoijan apuna ja haasteena

Ohjaaja: Mikko Kulmala

Vuosi: 2024

Sivumäärä: 42

Liitteiden lukumäärä: 4

Tässä opinnäytetyössä tutkittiin tekoälyn roolia aloittelevan ohjelmoijan oppimisessa ja työskentelyssä. Työn tavoitteena oli selvittää, miten tekoäly voi tukea ohjelmoinnin oppimisprosessia, millaisia haasteita sen käytössä esiintyy ja millaisia vaikutuksia sillä on ohjelmointityöhön. Tutkimuksen tarkoituksena oli myös kartoittaa tekoälytyökalujen, kuten ChatGPT:n ja GitHub Copilotin, käytännön sovelluksia ja niiden tarjoamia mahdollisuuksia ohjelmointiprosessissa.

Tutkimus toteutettiin yhdistämällä teoreettinen tarkastelu ja laadullinen haastattelututkimus. Teoriaosuudessa analysoitiin tekoälyn teknologisia peruseriäiteitä, kuten koneoppimista ja neuroverkkoja sekä tekoälyn kehitystä ja sen nykyisiä sovelluksia ohjelmistokehityksessä. Haastattelututkimus kohdistui ohjelmointia opiskeleviin ja työssä toimiviin ohjelmoijiin, jotka jakoivat kokemuksiaan tekoälyn hyödyistä ja haasteista ohjelmoinnissa.

Tulokset osoittivat, että tekoäly tukee ohjelmoinnin oppimista tarjoamalla reaaliaikaista palautetta, helpottamalla virheiden korjaamista ja nopeuttamalla ohjelmointiprosessia. Tekoälyn koettiin myös vähentävän stressiä ja edistävän luovuutta ohjelmointitehtävissä. Samalla tunnistettiin haasteita, kuten riippuvuuden riski, tekoälyn tuottamien virheellisten koodien tarkistustarve sekä eettiset ja tietoturvaan liittyvät kysymykset.

Johtopäätöksissä todettiin, että tekoäly toimii tehokkaana apuvälineenä ohjelmoinnissa, mutta sen käyttö vaatii kriittistä arviointia ja vastuullista lähestymistapaa. Tekoälyn kehittämisessä tulisi panostaa erityisesti sen läpinäkyvyyteen ja turvallisuuteen, jotta sen potentiaali voidaan hyödyntää täysimääräisesti ohjelmointialan tarpeisiin. Tämä työ tarjoaa hyödyllistä tietoa tekoälyn roolista ohjelmointityössä ja sen kehityssuunnista.

¹ Asiasanat: tekoäly, ohjelmointi, oppiminen

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Degree programme: Bachelor of Business Administration, Business Management

Author: Miro Nevala

Title of thesis: The Role of Artificial Intelligence as an Aid and Challenge for a Beginner Programmer

Supervisor: Mikko Kulmala

Year: 2024

Number of pages: 42

Number of appendices: 4

This thesis examined the role of artificial intelligence (AI) in the learning and work of beginner programmers. The objective was to determine how AI can support the programming learning process, what challenges its use presents, and what effects it has on programming work. The study also aimed to explore the practical applications of AI tools, such as ChatGPT and GitHub Copilot, and the opportunities they provide in programming processes.

The research was conducted by combining theoretical analysis with a qualitative interview study. The theoretical section analyzed the technological fundamentals of AI, such as machine learning and neural networks, as well as the development of AI and its current applications in software development. The interview study focused on students learning programming and professional programmers who shared their experiences regarding the benefits and challenges of using AI in programming.

The results indicated that AI supports programming learning by providing real-time feedback, facilitating error correction, and accelerating the programming process. AI was also perceived to reduce stress and enhance creativity in programming tasks. However, some challenges were identified, including the risk of dependency, the need to verify AI-generated erroneous code, and ethical and data security concerns.

The conclusions highlighted that AI serves as an effective tool in programming but requires critical evaluation and responsible use. It was recommended that AI development prioritize transparency and security to fully harness its potential for the needs of the programming field. This thesis provides valuable insights into the role of AI in programming work and its future directions.

¹ Keywords: Artificial intelligence, programming, learning

SISÄLTÖ

Opinnäytetyön tiivistelmä	1
Thesis abstract	2
SISÄLTÖ	3
Kuva-, kuvio- ja taulukkoluetelo	5
1 JOHDANTO	6
2 TEKOÄLY	7
2.1 Mitä tekoäly on	7
2.1.1 Heikko tekoäly (Narrow AI)	8
2.1.2 Vahva tekoäly (AGI)	8
2.2 Tekoälyn synty	9
2.3 Tekoälyn kehityksen vaiheet	10
2.4 Tekoälyn peruseräatteen ja -teknologiat	11
2.4.1 Koneoppiminen	11
2.4.2 Luonnollisen kielen käsittely	13
2.4.3 Suuret kielimallit	13
3 TEKOÄLY OHJELMISTOKEHITYKSESSÄ	14
3.1 Tekoälyn käyttökohteet ohjelmistokehityksessä	14
3.2 Tekoälyn edut ja hyödyt aloitteleville ohjelmoijille	15
3.3 Haasteet ja riskit tekoälyn käytössä aloitteleville ohjelmoijille	16
3.4 Tekoälyn tulevaisuus ohjelmistokehityksessä	17
3.5 ChatGPT ja CoPilot	18
3.5.1 CoPilotin ja ChatGPT edut ja käytännön sovellukset	19
3.5.2 Haasteet ja eettiset näkökulmat	19
3.5.3 Tulevaisuuden näkymät	20
4 HAASTATTELUTUTKIMUS	21
4.1 Tutkimuksen tarkoitus ja tavoitteet	21
4.2 Tutkimusmenetelmät	22
4.3 Haastattelurunko ja toteutus	23

4.4	Haastattelurungon rakenne	25
5	TUTKIMUSTULOKSET	28
5.1	Tausta ja kokemus ohjelmoinnista	28
5.2	Tekoälyn käyttö ohjelmointityössä.....	28
5.3	Tekoälyn tuomat hyödyt	29
5.4	Haasteet tekoälyn käytössä	31
5.5	Tekoälyn vaikutus työprosesseihin.....	32
5.6	Tekoälyn tulevaisuus ohjelmoinnissa	32
6	POHDINTA.....	34
6.1	Tekoälyn merkitys ohjelmoijan oppimisessa ja työssä	34
6.2	Hyödyt ja mahdollisuudet	35
6.3	Haasteet ja riskit.....	35
6.4	Tekoälyn vaikutukset työprosesseihin	36
6.5	Suositukset ja parannustoiveet.....	36
7	JOHTOPÄÄTÖKSET	38
	LÄHTEET	40

Kuva-, kuvio- ja taulukkoluetelo

Kuva 1. Alan Turing (Nordal, 2024).	9
Kuva 2. ChatGPT-prompti, jolla kyselurunko luotiin.....	26

Kuvaotsikkoluettelon hakusanoja ei löytynyt.

Taulukko 1. SWOT-analyysi tekoälystä ohjelmoinnissa vahvuudet ja heikkoudet.	16
Taulukko 2. SWOT-analyysi tekoälystä ohjelmoinnissa mahdollisuudet ja uhat.	18

1 JOHDANTO

Tekoäly on noussut yhdeksi merkittävimmistä teknologioista, joka muuttaa monia aloja, mukaan lukien ohjelmistokehitystä. Tekoälyn hyödyntäminen ohjelmoinnissa on kasvanut nopeasti viime vuosina, ja sen vaikutukset näkyvät erityisesti ohjelmointiprosessien nopeutumisessa, tehokkuuden parantumisessa ja uudenlaisten työkalujen integroinnissa kehitystyöhön. Tämä opinnäytetyö tarkastelee tekoälyn roolia ohjelmoinnin tukena, erityisesti oppimisen ja työnteon kontekstissa.

Työ lähestyy aihetta sekä teoreettisesta että empiirisestä näkökulmasta. Teoriaosuudessa käsitellään tekoälyn kehitystä, sen peruseriaatteita sekä sovelluksia ohjelmistokehityksessä. Näkökulmana on tekoälyn hyödyntäminen ohjelmointiprosessin eri vaiheissa, kuten koodin generoinnissa, virheiden korjaamisessa ja dokumentoinnissa. Erityistä huomiota kiinnitetään tekoälyn kykyyn tukea oppimista ja tarjota ratkaisuja ohjelmointitehtäviin, mikä on tärkeää ohjelmoijille heidän taitotasostaan riippumatta.

Työssä hyödynnetään haastattelututkimusta, jossa selvitetään, miten ohjelmoinnin parissa työskentelevät henkilöt kokevat tekoälyn vaikutukset käytännössä. Haastattelujen avulla kartoitetaan, millä tavoin tekoälyä hyödynnetään ohjelmoinnissa, mitkä ovat sen suurimmat hyödyt ja mitkä mahdolliset haasteet tai riskit sen käytössä ilmenevät. Tutkimus antaa äänen niille, jotka käyttävät tekoälyä osana ohjelmointityötään, ja tarjoaa laajan katsauksen tekoälyn nykytilanteeseen sekä tulevaisuuden mahdollisuuksiin.

Työn tavoitteena on muodostaa kokonaiskuva tekoälyn roolista ohjelmistokehityksessä ja sen käytön vaikutuksista oppimiseen ja työnteon. Lopuksi työssä esitetään suosituksia ja näkemyksiä tekoälyn kehittämisestä sekä pohditaan sen potentiaalia vastata alan tarpeisiin vastuullisella ja innovatiivisella tavalla. Tämä opinnäytetyö toimii resurssina sekä ohjelmoinnin että tekoälyn parissa työskenteleville ja tarjoaa arvokasta tietoa tekoälyn kehittyvästä roolista ohjelmointialalla.

2 TEKOÄLY

Tekoäly eli AI (englanninkielinen termi *artificial intelligence*), on noussut suuren yleisön suosioon viimeisten vuosien aikana. Jotta voidaan ymmärtää tekoälyn roolia ohjelmoinnissa, pitää meidän ensin ymmärtää, että mitä tekoäly on, miten se toimii ja mitä sillä voi tehdä.

2.1 Mitä tekoäly on

Tekoäly on laaja tutkimusala, jonka keskeisenä tavoitteena on kehittää älykkäitä koneita. Nämä koneet pystyvät oppimaan heidän kokemuksestansa ja suoriutumaan täten erilaisista monimutkaista tehtävistä, jotka tavallisesti vaativat kognitiivisia taitoja. Esimerkkinä näistä kognitiivista taidoista on päättelykyky, ongelmanratkaisutaidot, sekä oppiminen. Nykyiset tekoälyjärjestelmät hyödyntävät koneoppimista, algoritmeja ja tietoanalytiikkaa, joiden avulla ne pystyvät jäljittelemään älyllistä toimintaa. Nykyaikaiset tekoälysovellukset ovat monipuolisia ja kehittyneitä. Ne pystyvät esimerkiksi analysoimaan suuria määriä dataa, tunnistaa ihmisten kasvot heidän kasvopiirteistään ja tehdä ennusteita todennäköisyyslaskentaan perustuen. Gulenin (2023) mukaan tekoälyn eri osa-alueisiin kuuluvat muun muassa tietokonenäkö ja robotiikka, luonnollisen kielen käsittely ja koneoppiminen.

Tekoälyllä tarkoitetaan myös ohjelmistoja, laitteita ja järjestelmiä, jotka pystyvät oppimaan ja tekemään päätöksiä lähes samalla tavalla kuin ihmiset (Marjamaa, 2019). Tekoälyn ansiosta erilaiset tekniset järjestelmät kykenevät havainnoimaan ympäristöään, käsittelemään näitä havaintoja ja ratkaisemaan itsenäisesti ongelmiaan saavuttaakseen halutun lopputulokset (Guillot, 2023).

Tekoäly voidaan erottaa kahteen eri kategoriaan, joita ovat; heikko/kapea tekoäly (*Narrow AI*) ja vahva tekoäly AGI (*Artificial General Intelligence*). Näillä kahdella kategoriolla on samoja piirteitä, mutta vahva tekoäly (AGI) on paljon edistyksellisempi ja jäljittelee jo ihmisen älykkyyttä. Vertailen seuraavaksi hieman heikkoa ja vahvaa tekoälyä.

2.1.1 Heikko tekoäly (Narrow AI)

Nummisen (2023a) mukaan heikko tekoäly viittaa järjestelmiin, jotka on suunniteltu ainoastaan suorittamaan tiettyjä ja erityisiä ilman ihmismäistä älykyyttä ja todellista tietoisuutta vaativia tehtäviä. Nämä järjestelmät eivät pysty ajattelemaan, tuntemaan tai ymmärtämään asiayhteyksiä laajasti. Niiden toiminta perustuu ennalta määrättyyn dataan, sääntöihin ja algoritmeihin (mt.). Heikko tekoäly toimii yksinkertaisesti sille asetettujen sääntöjen mukaan ja on niiden sitoma. Se ei pysty menemään sille annettujen sääntöjen yli (Maria, 2024a). Heikkoa tekoälyä hyödynnetään nykypäivänä esimerkiksi autoteollisuudessa ja terveysalalla. Esimerkkejä heikosta tekoälystä ovat sähköpostiviestin suodatus, henkilökohtaiset teknologiset avustajat (Siri, Alexa), kuvantunnistusohjelmat ja asiakaspalvelubotit. Esimerkiksi erilaiset kuvantunnistusohjelmat ja chatbotit pohjautuvat koneoppimiseen.

2.1.2 Vahva tekoäly (AGI)

Vahva tekoäly eli toiselta nimeltä yleinen tekoäly viittaa tekoälyjärjestelmiin, joilla on ihmistason tai jopa sen ylittävä älykkyys ja kyky tehdä erilaisia tehtäviä. Vahva tekoäly pystyy periaatteessa suorittamaan minkä tahansa älyllisen toiminnon, jonka ihmisen pystyy tehdä (Numminen, 2023b). Vahva tekoäly kykenee oppimaan itsenäisesti ilman ihmisen ohjausta syväoppimisen kautta. Vahva tekoäly omaa kyvyn ymmärtää ja ratkaista monimutkaisia ongelmia suurten kielimallien avulla, joka ylittää ihmisten kapasiteetin ratkoa tällaisia ongelmia.

Marian (2024b) mukaan vahva tekoäly on enemmän filosofia kuin todellinen lähestymistapa tekoälyn luomisessa. Vahva tekoäly on eräänlainen käsitys tekoälystä, jossa se rinnastetaan jopa ihmisen tasolle. Innodata (i.a.) mukaan edistyneimmät virtuaaliavustajat kuten OpenAI GPT-4 omaavat yleisen tekoälyn piirteitä, kuten kykyä ymmärtää kontekstia ja tuottaa ihmismäistä tekstiä. Ne eivät ole kuitenkaan aitoja AGI-järjestelmiä, koska niiltä puuttuu täysin ihmisen kaltainen ymmärrys (Innodata).

2.2 Tekoälyn synty

Modernin tekoälyn historia juontaa juurensa 1900-luvun puoleen väliin. Tekoälyn isänä pidetään Alan Turingia (1912–1954). Alan Turing (1912–1954) oli brittiläinen matemaatikko ja loogikko, jonka työ loi perustan modernille tietojenkäsittelytieteelle ja tekoälylle (Cope-land, 2024). Turingin panos tekoälyn kehitykselle on erityisen merkittävä hänen vuonnansa 1950 julkaisemansa artikkelin *Computing Machinery and Intelligence* myötä, jossa hän esitti kuuluisan ”Turingin testin”. Turingin testissä koneen kykyä ajatella mitataan sillä, pystyykö se jäljittelemään ihmisen älykkyyttä niin, ettei ihminen pysty erottamaan sitä toisesta ihmisestä keskustelun perusteella. Turingin (1950, s. 443) mukaan koneiden älykkyys voisi ajan myötä kehittyä niin, että 50 vuoden päästä ne voisivat jopa haastaa ihmisen kognitiivisia kykyjä. Alla kuva Alan Turingista.



Kuva 1. Alan Turing (Nordal, 2024).

Turingin työ asetti peruskysymyksen tekoälyn kehitykselle, joka on, että voiko kone ajatella? Tämä kysymys on edelleen keskeinen tekoälytutkimuksessa ja sen filosofisessa pohdinnassa. Tekoälyä tutkiessa tämä on ollut yksi pääkysymyksistä koko tekoälyn historian ajan.

2.3 Tekoälyn kehityksen vaiheet

Tekoälyn virallinen kehitys alkoi vuonna 1956 Dartmouthin konferenssissa, jossa McCarthy esittelee käsitteen "tekoäly" ja asettivat tavoitteeksi luoda koneita, jotka voisivat jäljitellä ihmisen älykkyyttä (McCorduck, 2003, s. 26). Tämä ajanjakso tunnetaan symbolisen tekoälyn aikakautena, jolloin tekoäly keskittyi loogisiin päättelyihin ja sääntöpohjaisiin järjestelmiin. Tekoälyn tutkimuksen ensimmäisiin merkittäviin saavutuksiin kuului Logic Theorist -ohjelma, joka oli ensimmäinen toimiva tekoälyohjelma, jonka kehittivät Shaw, Simon ja Newell vuonna 1956 (mp.).

1960-luvulla vallitsi suurta optimismia tekoälytutkimuksen piirissä. Tutkijat uskoivat, että rajaton laskentateho voidaan saavuttaa nopeimmilla koneilla ja suuremmalla muistimäärällä (Russell & Norvig, 2003, s. 22). Tärkeitä tekoälyohjelmia olivat *General Problem Solver* (GPS), joka ratkaisi monia loogisia ja matemaattisia ongelmia, sekä varhaiset puheentunnistusjärjestelmät, kuten Weizenbaumin kehittämä ELIZA, joka simuloi ihmisen kanssa käytävää keskustelua (mts. 262).

1970-luvulla tekoälyn kehitys hidastui merkittävästi, kun laskentatehon ja muistiresurssien rajallisuus rajoittivat monimutkaisten ongelmien ratkaisemista. Tätä aikaa kutsutaan "tekoälyn talveksi", jolloin rahoitus ja kiinnostus tekoälytutkimukseen vähenivät merkittävästi (McCorduck, s. 442).

Vaikka 1970-luvulla tekoälytutkimus oli kriisissä, 1980-luvulla kiinnostus heräsi uudelleen neuroverkkojen ja koneoppimisen kehittymisen myötä. Vuonna 1980 kaupallisille markkinoille tuli ensimmäinen asiantuntijajärjestelmä, joka suunniteltiin auttamaan tietokonejärjestelmän tilaamisessa valitsemalla komponentit automaattisesti asiakkaiden tarpeiden mukaan (Tableau i.a.).

1990-luvulla tekoälyn tutkimuksessa tapahtui merkittävä käänne, kun symbolisen tekoälyn menetelmät alkoivat menettää jalansijaa ja koneoppiminen, erityisesti neuroverkot, nousivat alalle hallitseviksi menetelmiksi. Mitchellin (1997 s. 3) mukaan tietokoneohjelma, joka oppi pelaamaan tammaa, saattoi parantaa suorituksiaan mitattuna sen kykyä voittaa tehtävälukassaan. Kone siis oppi pelaamalla ja paransi suoritustaan opitun datan perusteella. Vuonna 1997 IBM:n kehittämä tietokoneohjelma voitti shakin maailmanmestarin Garri

Kasparovin erittäin tunnetussa ottelussa, ja siitä tuli ensimmäinen ohjelma, joka onnistui päihittämään ihmisen shakkimestarin (Tableau).

2000-luvulla tekoälyn kehitys kiihtyi huomattavasti, kun laskentateho, suuret tietomäärät (big data) ja tehokkaammat oppimisalgoritmit, kuten syväoppiminen, ottivat alan haltuunsa. Tableaun, (i.a.) mukaan IBM:n kehittämä Watson-niminen luonnollisen kielen käsittelyn (NLP) tietokoneohjelma voitti televisioidussa Jeopardy-visailussa kaksi aiempaa mestaria vuonna 2011. Apple julkaisi ensimmäisen virtuaaliavustajan, Sirin, vuonna 2011 (mt.).

Nykyään tekoäly on laajentunut lukuisille eri sovellusalueille, ja sen vaikutus näkyy erityisesti suurissa kielimalleissa. Tekoäly ei enää ainoastaan tue yksittäisiä tehtäviä, vaan se pystyy luomaan monimutkaisia, ihmisen kaltaisia vastauksia ja ratkaisuja laajassa mittakaavassa.

2.4 Tekoälyn peruseriaatteet ja -teknologiat

Tekoälyn ytimessä ovat sen peruseriaatteet ja teknologiat, jotka mahdollistavat koneiden kyvyn oppia, mukautua ja suorittaa tehtäviä älykkäästi. Tekoälyn keskeinen periaate on sen kyky jäljitellä inhimillistä ajattelua ja päätöksentekoa ohjelmistojen ja algoritmien avulla.

2.4.1 Koneoppiminen

Koneoppiminen on tekoälyn tärkein osa-alue, jonka avulla tietokoneet voivat parantaa suorituskykyään kokemuksen kautta ilman erillistä ohjelmointia. Koneoppiminen on nopeasti yleistynyt teknologia, jonka avulla tietokoneiden on mahdollista oppia automaattisesti sille annettujen aiempien tietojen pohjalta (Klusaité, 2023). Koneoppiminen perustuu mallien ja algoritmien kehittämiseen. Tämä mahdollistaa datan oppimisen ja sen pohjalta päätösten ja ennusteiden tekemisen.

Koneoppiminen voidaan jakaa 4 koneoppimismalliin. Ne ovat valvottu, valvomaton, puoli-valvottu ja vahvistusoppiminen. Seuraavaksi kerrotaan jokaisesta mallista tiivistetysti.

Ohjatun tai valvotun oppimisen voi rinnasta tapaan, jolla ihmiset opettelevat asioita. Jos tarkoituksena on esiopettaa kone erottamaan kaksi asiaa toisistaan, sisällytetään tietopariin tiedot molemmista asioista. Klusaitén, (2023) mukaan koneoppimisen ansiosta tietokoneen algoritmi tunnistaa kahdesta annetusta vaihtoehdosta halutun, joka on ennalta määriteltä oikeaksi tulokseksi. Ajan myötä järjestelmä pystyy hyödyntämään keräämäänsä tietoa ja kykenee myöhemmin ennakoimaan itse vastaukset sille esitettyihin kysymyksiin halutun asia tunnistamista koskien.

Klusaité (2023) kertoo, että ohjaamaton tai valvomaton oppiminen käsittelee merkitsemätöntä tietoa. Ohjaamattoman oppimisen tarkoituksena on tunnistaa tiedoista piilotettuja malleja, rakenteita tai yhteyksiä ilman ennakkotietoa tulosten luokittelusta. Tämä menetelmä analysoi dataa itsenäisesti, etsien luonnollisia klustereita ja kuvioita, joita ei ole etukäteen merkitty tai luokiteltu (Euroopan tietoteknologien sertifiointiakatemia, 2023).

Klusaitén (2023) mukaan puolivalvottu oppiminen yhdistää piirteitä sekä valvotusta, että valvomattomasta oppimisesta, muodostaen näin hybridimallin. Hän jatkaa, että tämä lähestymistapa soveltuu erityisesti tilanteisiin, joissa saatavilla on suuri määrä jäsentämätöntä ja merkitsemätöntä dataa, mutta vain rajallinen osa datasta on merkitty ohjatun oppimisen tarpeisiin.

Klusaitén (2023) mukaan vahvistusoppiminen perustuu oppimiseen vuorovaikutteisessa ympäristössä, jossa järjestelmä kehittää toimintaansa kokemusten ja erehdysten kautta. Toisin kuin ohjatussa oppimisessä, jossa käytetään vastausavainta, vahvistusoppimisessä määritellään sallitut säännöt ja mahdolliset lopputulokset.

Neuroverkot ovat merkittävä tekoälyteknologia, joka toimii jäljitellen ihmisaivojen hermoverkkojen toimintaa informaation prosessoinnissa. Neuroverkkojen ansiosta tekoäly pystyy analysoimaan monimutkaisia tietomassoja ja oppimaan datasta. LeCunin ym. (2015, s. 437) mukaan neuroverkon perusrakenne koostuu kolmesta keskeisestä kerroksesta: syöttökerros (*input layer*), piilokerrokset (*hidden layers*) ja ulostulokerros (*output layer*). Syöttökerros vastaanottaa alkuperäisen datan, piilokerrokset käsittelevät ja oppivat datan piirteet suorittamalla monimutkaisia laskutoimituksia, ja ulostulokerros tuottaa lopullisen vastauksen, kuten luokituksen tai ennusteen.

2.4.2 Luonnollisen kielen käsittely

Tekoälyn keskeisiin periaatteisiin kuuluu myös luonnollisen kielen käsittely (NLP), jonka avulla koneet voivat ymmärtää ja tuottaa ihmisten kieltä. Tekoälykeskuksen (2024) mukaan luonnollisen kielen käsittely NLP (*Neuro-Linguistic Programming*) hyödyntää koneoppimisen, kielitieteen ja tilastollisen analyysin menetelmiä. Mallit koulutetaan suurilla teksti- ja puhedatamäärillä, jotta ne oppivat tunnistamaan kieliopillisia rakenteita, sanojen merkityksiä sekä tekstin taustalla olevia tunteita ja sävyjä.

Esimerkkejä NLP sovelluksista on chatbotit, älykkäät hakukoneet, erilaiset tekstianalyysit sekä kielenkäännössovellukset. Luonnollisen kielen käsittely kehittyy jatkuvasti, ja tulevaisuudessa voimme odottaa yhä älykkäämpiä ja luonnollisempia vuorovaikutuksia koneiden kanssa. NLP:n avulla voidaan kehittää entistä yksilöllisempiä ja tehokkaampia sovelluksia, jotka ymmärtävät käyttäjiä syvällisemmin kuin koskaan aiemmin (Tekoälykeskus, 2024).

2.4.3 Suuret kielimallit

Suuret kielimallit (*Large Language Model*, LLM) ovat perustavamalleja, jotka on koulutettu valtavilla tietomäärillä. Tämä tekee niistä kykeneviä ymmärtämään ja tuottamaan luonnollista kieltä sekä muuta sisältöä, mahdollistaen laajan kirjon erilaisia tehtäviä (IBM, i.a.).

Suuret kielimallit hyödyntävät koneoppimisen osa-aluetta nimeltä syväoppiminen ymmärtääkseen, miten merkit, sanat ja lauseet liittyvät toisiinsa. Syväoppiminen perustuu jäsentämättömän datan todennäköisyysanalyysiin, mikä lopulta mahdollistaa mallin kyvyn tunnistaa sisältöjen välisiä eroja ilman ihmisen ohjausta (Cloudfare, i.a.).

3 TEKOÄLY OHJELMISTOKEHITYKSESSÄ

Tekoälyn (AI) vaikutus ohjelmistokehitykseen on ollut huomattava, ja sen käyttö on laajentunut merkittävästi viime vuosina. AI auttaa automatisoimaan monia ohjelmistokehityksen prosesseja, parantamaan koodin laatua ja tehostamaan kehittäjien työtä. Tämän luvun tarkoituksena on tarkastella tekoälyn käyttökohteita ohjelmistokehityksessä, sen etuja ja haasteita aloitteleville ohjelmoijille sekä tekoälyn tulevaisuuden näkymiä alalla.

3.1 Tekoälyn käyttökohteet ohjelmistokehityksessä

Tekoälyn käyttökohteet ohjelmistokehityksessä ovat moninaiset. AI pystyy automatisoimaan monia kehitysprosessin osa-alueita, kuten koodin tuottamista, virheiden löytämistä ja testauksen suorittamista. Qureshin (2024) mukaan tekoälytyökalut voivat suorittaa automaattisia testitapauksia ja simuloida monimutkaisia skenaarioita koodin toiminnan varmistamiseksi. Tämä vähentää manuaalisen työn määrää ja mahdollistaa nopeamman kehitysprosessin.

Tekoäly voi myös tunnistaa koodissa toistuvia virheitä ja ehdottaa optimointeja, mikä auttaa parantamaan ohjelmiston laatua ja estämään mahdolliset virheet jo varhaisessa kehitysvaiheessa. Savaniyan (i.a.) mukaan tekoäly pystyy analysoimaan suuria tietomääriä ja antamaan kehittäjille ehdotuksia siitä, miten koodia voidaan parantaa suorituskyvyn, turvallisuuden ja skaalautuvuuden näkökulmasta. Lisäksi AI voidaan optimoida ohjelmiston dokumentointia ja refaktorointia reaaliaikaisesti, mikä tekee kehitysprosessista jatkuvasti sujuvampaa (Qureshi, 2024).

Toinen merkittävä tekoälyn käyttökohde on käyttäjäkokemuksen parantaminen. Babenko (2024) mukaan tekoälyä käytetään asiakastuen automatisoinnissa, esimerkiksi chatbottien muodossa, sekä suositusjärjestelmissä, jotka mukautuvat käyttäjän toimintaan ja tarjoavat henkilökohtaisempia kokemuksia. Tekoäly voi analysoida käyttäjien toimintaa ja auttaa kehittäjiä parantamaan ohjelmiston käytettävyyttä ja käyttäjäkokemusta analysoimalla, miten käyttäjät ovat vuorovaikutuksessa ohjelmiston kanssa (Savaniya, i.a.).

3.2 Tekoälyn edut ja hyödyt aloitteleville ohjelmoijille

Aloitteleville ohjelmoijille tekoäly tarjoaa useita merkittäviä etuja. Yksi tärkeimmistä eduista on rutiinitehtävien automatisointi, kuten virheiden havaitseminen ja koodin refaktorointi, mikä antaa aloitteleville ohjelmoijille enemmän aikaa keskittyä monimutkaisempiin ohjelmointitehtäviin (Qureshi, 2024). Tekoäly voi myös toimia mentorina tarjoamalla välitöntä palautetta ja koodiehdotuksia, mikä nopeuttaa oppimisprosessia. Tämä on erityisen hyödyllistä aloitteleville ohjelmoijille, jotka ovat vasta aloittamassa uraansa ja tarvitsevat apua oikeiden ohjelmointikäytäntöjen omaksumisessa (mt.).

Savaniyan (i.a.) mukaan tekoäly voi myös tukea oppimista reaaliaikaisella analytiikalla ja ehdotuksilla, joiden avulla aloittelevat ohjelmoijat voivat nähdä, miten heidän kirjoittamansa koodi voidaan optimoida. Tämä parantaa ohjelmoijan kognitiivisia taitoja ja tukee laajempaa oppimista. Lisäksi tekoälyn avulla voidaan vähentää virheiden määrää ja lisätä koodin laatua, mikä auttaa aloittelevia ohjelmoijia kehittämään laadukasta koodia jo varhaisessa vaiheessa (mt.).

Tein tekoälystä ohjelmistokehityksessä SWOT-analyysin kuvaamaan ja havainnollistamaan sitä. Seuraavalla sivulla taulukko, jossa kuvataan tekoälyn vahvuuksia ja heikkouksia.

Taulukko 1. SWOT-analyysi tekoälystä ohjelmoinnissa vahvuudet ja heikkoudet.

Vahvuudet	Heikkoudet
- Automatisointi: Tekoäly voi automatisoida rutiinitehtäviä, kuten koodin generointia, testausta ja virheiden korjausta, mikä säästää aikaa ja resursseja (Qureshi, 2024).	- Riippuvuus tekoälystä: Liiallinen tekoälyn käyttö voi estää ohjelmoijia oppimasta syvällisesti tarvittavia ohjelmointitaitoja (Qureshi, 2024).
- Koodin optimointi ja virheiden tunnistus: Tekoäly voi analysoida suuria tietomääriä, tunnistaa toistuvia virheitä ja parantaa koodin suorituskykyä jo kehitysvaiheessa (Savaniva, i.a.).	- Mustan laatikon ongelma: Tekoälyn päätöksenteon logiikka ei aina ole läpinäkyvä, mikä vaikeuttaa sen tuottaman koodin tarkistamista ja korjaamista (Savaniva, i.a.).
- Käyttäjäkokemuksen parantaminen: AI-pohjaiset työkalut, kuten chatbotit ja suositusjärjestelmät, voivat parantaa ohjelmistojen käyttäjäystävällisyyttä ja tarjota henkilökohtaisia kokemuksia (Babenko, 2024).	- Dataintensiivisyys: Tekoälyn käyttö vaatii suuria tietomääriä ja monimutkaisia datankäsittelymenetelmiä, mikä voi olla haastavaa erityisesti aloitteleville ohjelmoijille (Savaniva, i.a.).
- Nopeampi kehitysprosessi: Tekoäly vähentää manuaalista työtä, mikä nopeuttaa ohjelmistokehitystä ja antaa kehittäjille enemmän aikaa keskittyä luovempiin tehtäviin (Qureshi, 2024).	- Yhteensopivuusongelmat: Tekoäly voi tuottaa koodia, joka ei täysin vastaa organisaation turvallisuus- tai suorituskykyvaatimuksia (Babenko, 2024).

3.3 Haasteet ja riskit tekoälyn käytössä aloitteleville ohjelmoijille

Tekoälyn käytössä on kuitenkin myös haasteita ja riskejä, erityisesti aloitteleville ohjelmoijille. Qureshin, (2024) mukaan yksi suurimmista riskeistä on, että tekoälyyn tukeutuminen voi estää ohjelmoijia oppimasta tarvittavia ohjelmointitaitoja. Jos AI tekee suurimman osan työstä, aloittelevat ohjelmoijat eivät ehkä kehitä riittävästi omaa ongelmanratkaisu- ja

koodauskykyään. Tämä voi johtaa siihen, että ohjelmoijat eivät osaa korjata tai ymmärtää syvällisemmin koodin toimivuutta, mikä voi hidastaa heidän urakehitystään (Babenko, 2024).

Toinen merkittävä haaste on tekoälyn päätöksenteon läpinäkyväisyys. Savaniyan (i.a.) mukaan tekoälyn toimintalogiikka voi olla vaikeasti ymmärrettävissä, mikä tekee AI tuottamasta koodista haasteellisen tarkastaa ja korjata. Hän jatkaa, että tämä "mustan laatikon" ongelma voi johtaa tilanteisiin, joissa ohjelmoijat eivät pysty seuraamaan, miten ja miksi tekoäly on tehnyt tiettyjä päätöksiä, mikä voi olla kriittistä, kun koodia käytetään esimerkiksi turvallisuuskriittisissä sovelluksissa.

3.4 Tekoälyn tulevaisuus ohjelmistokehityksessä

Tekoälyn tulevaisuus ohjelmistokehityksessä näyttää lupaavalta. Savaniyan (i.a.) mukaan AI-kehittyessä sen kyky suorittaa monimutkaisempia ohjelmointitehtäviä tulee lisääntymään, ja se saattaa mahdollistaa yhä laajempien ohjelmistoprojektien automatisoinnin. Tulevaisuudessa tekoäly voi esimerkiksi tuottaa ja hallita kokonaisia ohjelmistoarkkitehtuuria ilman ihmisen välitöntä osallistumista, mikä tehostaisi kehitysprosessia merkittävästi ja nopeuttaisi ohjelmistojen markkinoille pääsyä.

Samalla kuitenkin tulee huomioida eettiset ja turvallisuushaasteet. Babenkon (2024) mukaan tekoälyn laajempi käyttöönotto vaatii selkeitä sääntely- ja hallintamekanismeja, jotta voidaan estää esimerkiksi AI-tuottamat algoritmien vinoumat ja tietoturvaongelmat. Tekoälyn on toimittava läpinäkyvästi ja vastuullisesti, jotta sen käyttö ohjelmistokehityksessä voi laajentua turvallisesti ja oikeudenmukaisesti.

Seuraavalla sivulla toinen osa SWOT-analyysistä, jossa näkyy tekoälystä ohjelmoinnissa mahdollisuudet ja uhat.

Taulukko 2. SWOT-analyysi tekoälystä ohjelmoinnissa mahdollisuudet ja uhat.

Mahdollisuudet	Uhat
- Laajemmat sovellukset: Tekoäly voi tulevaisuudessa automatisoida yhä monimutkaisempia kehitystehtäviä, kuten ohjelmistoarkkitehtuurin suunnittelun ja koko ohjelmistoprojektin hallinnan (Savaniva, i.a.).	- Eettiset ja turvallisuushaasteet: Tekoälyn lisääntynyt käyttö ohjelmistokehityksessä voi tuoda mukanaan ongelmia, kuten algoritmien vinoumat ja tietoturvariskit (Babenko, 2024).
- Mukautuvat ohjelmistot: Tulevaisuudessa AI-pohjaiset ohjelmistot voivat mukautua reaaliajassa käyttäjien tarpeisiin ja tarjota henkilökohtaisempia palveluita (BrainerHub, 2023).	- Läpinäkymättömyys ja sääntelyn puute: Tekoälyn päätöksenteon läpinäkymättömyys voi aiheuttaa riskejä, jos ohjelmistokehityksessä käytettävää tekoälyä ei säännellä riittävästi (Qureshi, 2024).
- Oppimisen tuki: Tekoäly voi toimia mentorina aloitteleville ohjelmoijille, tarjoten reaaliaikaista palautetta ja koodiehdotuksia, mikä nopeuttaa oppimista ja parantaa koodin laatua (Qureshi, 2024).	- Tekoälyn väärinkäytön riski: Tekoälyn pohjautuvia työkaluja voidaan käyttää väärin tai liiallisesti, mikä voi johtaa epäeettisiin käytäntöihin ohjelmistokehityksessä (Savaniva, i.a.).
- Yhteistyö eri osa-alueilla: AI-pohjaiset analyysityökalut voivat auttaa eri kehitystieimejä tekemään yhteistyötä tehokkaammin ja optimoimaan koko kehitysprosessia (Jurafsky & Martin, 2021).	- Henkilöstön osaamisen heikkeneminen: Jos kehittäjät tukeutuvat liikaa tekoälytyökaluihin, se voi heikentää heidän osaamistaan ja kykyään ratkaista ongelmia itsenäisesti (Qureshi, 2024).

3.5 ChatGPT ja CoPilot

ChatGPT ja GitHub Copilot ovat kaksi merkittävää tekoälytyökalua, jotka tukevat ohjelmistokehittäjiä erityisesti koodin kirjoittamisessa ja ongelmanratkaisussa. Ne molemmat käyttävät todella edistynyttä tekoälyteknologiaa, joka on lähimpänä vahvaa tekoälyä tällä hetkellä olevista tekoälysovelluksista. Coursera (2023) mukaan ChatGPT on OpenAI:n kehittämä tekoälypohjainen chatbot, joka mahdollistaa käyttäjien "keskustelun" sen kanssa luonnollista vuoropuhelua jäljittelevällä tavalla. Käyttäjä voi esittää kysymyksiä tai tehdä pyyntöjä syöttämällä ohjeita, joihin ChatGPT vastaa. Tämä intuitiivinen, helppokäyttöinen ja ilmainen työkalu on jo saavuttanut suosiota sekä perinteisten hakukoneiden

vaihtoehtona että tekoälypohjaisten tekstintuottamisen välineenä ja monissa muissa käyttötarkoituksissa (Coursera, 2023). Sen kyky ymmärtää ja tuottaa monimutkaisia tekstimuotoisia vastauksia on erityisen hyödyllinen ohjelmistokehityksessä, sillä kehittäjät voivat käyttää ChatGPT esimerkiksi teknisen tuen apuvälineenä tai kysyä neuvoja monimutkaisista koodiongelmista.

GitHub Copilot puolestaan on erityisesti ohjelmointityöhön kehitetty tekoälytyökalu, joka toimii "pariohjelmoijana" ehdottaen koodia käyttäjän kirjoittaessa. GitHub (i.a.) mukaan Copilot uudistaa kehittäjien työskentelytapoja tarjoamalla älykästä ja kontekstiin perustuvaa apua ohjelmistokehitysprosessin eri vaiheissa. GitHub jatkaa, että tekoälyn edelläkävijöiden kehittämä Copilot tukee kehittäjiä muun muassa koodin täydennyksessä ja keskusteluavustuksessa IDE-ympäristössä, sekä tarjoaa selityksiä koodille ja vastauksia dokumentaatioihin liittyviin kysymyksiin suoraan GitHubissa. Tämä mahdollistaa kehittäjille keskittymisen enemmän arvoa tuottaviin tehtäviin, luovuuteen ja työn mielekkyyteen (GitHub).

3.5.1 CoPilotin ja ChatGPT edut ja käytännön sovellukset

ChatGPT on merkittävä rooli erityisesti aloittelevien ohjelmoijien tukemisessa. Se voi selittää monimutkaisia koodin osia, tarjota koodin optimointiehdotuksia ja auttaa ymmärtämään perusohjelmointikäsitteitä yksinkertaisella tavalla. Dicksonin (2023) mukaan ChatGPT on hyvä työkalu muun muassa kielenkäännöksessä, koodin kirjoittamisessa ja korjaamisessa, ideoiden keksimisessä ja kasaamisessa sekä tekstin ja koodin muokkaamisessa.

GitHub (i.a.) mukaan mullistaa kehittäjien työskentelykokemuksen täydentämällä koodia reaaliajassa, selittämällä koodia haluttaessa sekä integroitumalla suosittuihin koodiedito-reihin.

3.5.2 Haasteet ja eettiset näkökulmat

Vaikka ChatGPT ja Copilot ovat tehokkaita työkaluja, niiden käyttöön liittyy myös haasteita. Bender ym. (2021 s. 617) huomauttavat, että kielimalleihin, kuten GPT-3, liittyy "mustan laatikon" ongelma, eli niiden päätöksenteko ei aina ole läpinäkyvää käyttäjälle. Tämä

voi olla ongelmallista etenkin tilanteissa, joissa Copilot tai ChatGPT tuottaa koodia, joka ei täytä organisaation standardeja tai ei ole täysin turvallista.

Dicksonin (2023) mukaan suuret kielimallit, kuten ChatGPT voi tuottaa kielellisesti oikeaa, mutta sisällisesti virheellistä tekstiä. ChatGPT voi luoda virheellistä tekstiä tai koodia, vaikka se näyttäisi oikealta. Dickson jatkaa, että ChatGPT data on rajoitettu sille ajalle, kun se on sinne syötetty. Näin ollen se ei osaa vastata kaikkiin uusimpiin kysymyksiin, koska sen tieto rajoittuu siihen aikakauteen, kun sille on tietoa viimeksi syötetty (mt).

Lisäksi on huomioitava, että ChatGPT ja Copilotin kaltaiset mallit on koulutettu valtavilla määrällä julkista dataa, mikä herättää kysymyksiä tekijänoikeuksista ja etiikasta. Tämä on erityisen tärkeä kysymys organisaatioille, jotka haluavat varmistaa, että niiden käyttämä koodi on vapaasti käytettävissä ilman tekijänoikeusrikkomuksia.

3.5.3 Tulevaisuuden näkymät

Tekoälytyökalujen, kuten ChatGPT ja Copilotin, odotetaan kehittyvän edelleen tulevaisuudessa. Savaniyan (i.a.) mukaan näiden työkalujen käyttökohteet tulevat laajenemaan, ja ne saattavat tukea yhä monimutkaisempia ohjelmointitehtäviä. Esimerkiksi AI-pohjainen automaatio voi tulevaisuudessa mahdollistaa koko ohjelmistoratkaisun kehittämisen vähäisellä ihmisen puuttumisella, mikä voi mullistaa ohjelmistokehityksen prosessit ja käytännöt (GitHub, i.a.).

4 HAASTATTELUTUTKIMUS

4.1 Tutkimuksen tarkoitus ja tavoitteet

Tekoälyn vaikutus ohjelmistokehitykseen on kasvanut merkittävästi viime vuosina. Tämä tutkimus keskittyi kartoittamaan tekoälyn käyttöä erityisesti haastatteluun osallistuneiden keskuudessa ja selvittämään sen vaikutuksia oppimiseen ja työskentelytapoihin. Tekoälyn käyttö ohjelmoinnin apuvälineenä on yleistynyt, ja sen rooli oppimisessa sekä työnteon tukena tarjoaa uusia mahdollisuuksia mutta tuo mukanaan myös haasteita.

Tutkimuksen tavoitteena oli saada kokonaiskuva siitä, kuinka haastatteluun osallistuneet hyödynsivät tekoälyä oppimisessaan ja työssään. Haastattelu keskittyi kolmeen keskeiseen teemaan, jotka mahdollistivat tekoälyn käytön tarkastelun oppimisen ja ohjelmointityön näkökulmasta. Ensimmäinen teema käsitteli tekoälyn käyttöä. Haastatteluissa pyrittiin ymmärtämään, millä tavoin haastateltavat hyödynsivät tekoälyä niin oppimisessaan kuin ohjelmointityössään. Keskusteluissa kartoitettiin käytettyjä tekoälytyökaluja sekä niiden roolia päivittäisessä työssä ja opiskelussa.

Toinen teema keskittyi tekoälyn hyötyihin ja haasteisiin. Haastateltavat arvioivat tekoälyn käytön merkittävimpiä etuja, kuten työn nopeuttamista, virheiden vähentämistä ja oppimisen tukemista. Samalla tarkasteltiin myös käytön tuomia haasteita, kuten tekoälyn mahdollisia luotettavuusongelmia, riippuvuuden riskiä ja muita ohjelmointityön ongelmakohtia.

Kolmas teema käsitteli tekoälyn tulevaisuuden näkymiä. Haastateltavilta kysyttiin heidän odotuksiaan tekoälyn kehityksestä ja sen vaikutuksista ohjelmointityöhön sekä oppimiseen. Keskustelut avasivat myös huolenaiheita, kuten tekoälyn eettisiä ja turvallisuuskysymyksiä sekä sen mahdollisia vaikutuksia alan käytäntöihin pitkällä aikavälillä.

Näiden teemojen avulla haastattelut tarjosivat kattavan näkemyksen tekoälyn roolista, sen vaikutuksista ja tulevaisuuden mahdollisuuksista ohjelmointityön ja oppimisen tukena.

Tärkeänä lähtökohtana oli ymmärtää tekoälyn roolia nimenomaan haastateltavien oppimiskokemuksissa. Tekoälyn käyttö tarjoaa erityisesti aloittelijoille uuden tavan oppia ja soveltaa tietoa käytännössä, mikä voi tukea heidän taitojensa kehittymistä. Samalla tutkimuksessa huomioitiin myös kokeneempien käyttäjien näkemykset, jotta saatiin kattava kuva tekoälyn käytön vaikutuksista eri tasoilla.

Tutkimus oli ajankohtainen, sillä tekoälyteknologiat, kuten ChatGPT ja GitHub Copilot, ovat vasta hiljattain yleistyneet ohjelmointityökaluina. Näiden työkalujen hyödyntäminen tarjoaa mahdollisuuden tehostaa oppimista ja työntekoa, mutta niiden pitkäaikaiset vaikutukset ohjelmointitaitoihin ovat vielä osin tuntemattomia. Näin ollen tutkimuksen tulokset voivat tarjota arvokasta tietoa tekoälyn kehittämiseen ja käyttöön liittyvien suositusten luomisessa.

4.2 Tutkimusmenetelmät

Tutkimus toteutettiin laadullisena haastattelututkimuksena, joka soveltui hyvin syvällisen ja monipuolisen ymmärryksen muodostamiseen aiheesta. Laadullinen lähestymistapa mahdollisti haastatteluun osallistuneiden yksilöllisten kokemusten ja näkemysten tarkastelun sekä niiden vaikutusten analysoinnin. Hirsjärven ym. (2009, s. 161) mukaan kvalitatiivisessa tutkimuksessa pyritään tutkimaan kohteita mahdollisimman kokonaisvaltaisesti. Haastattelut valittiin menetelmäksi, koska ne tarjosivat mahdollisuuden esittää syventäviä kysymyksiä ja mukauttaa keskustelua osallistujien vastauksien perusteella.

Hirsjärven ym. (2009, s. 205) mukaan haastattelujen etuna muihin tiedonkeruumuotoihin verrattuna on sen kyky säädellä aineistojen keruuta joustavasti tilanteeseen kuuluvalla- ja sen edellyttämällä tavalla vastaajia myötäillen. Kvalitatiivisen haastattelun etuna on myös mahdollisuus säädellä aiheiden järjestystä ja tulkita vastauksia (mts. 161). Eskolan ja Suorannan (1998, s. 208) mukaan laadullisessa tutkimuksessa on mahdollisuus kulkea paljon vapaammin edestakaisin aineiston analyysin, tutkimusmenetelmien ja tehtyjen tulkintojen välillä kuin kvantitatiivisessa tutkimuksessa.

Haastattelut toteutettiin puolistrukturoidulla menetelmällä. Tämä tarkoitti, että kysymysrakenne oli etukäteen suunniteltu, mutta haastattelun aikana oli mahdollista syventyä tarkemmin tiettyihin teemoihin osallistujien vastausten mukaan. Tämä menetelmä varmisti,

että tutkimuskysymyksiin liittyvät keskeiset teemat tulivat kattavasti käsitellyiksi, mutta antoi haastateltaville myös tilaa jakaa omia kokemuksiaan ja näkemyksiään.

Tutkimukseen osallistui kuusi ohjelmoinnin parissa toimivaa henkilöä, jotka edustivat eri taitotasoja ja kokemustaustoja. Haastateltavien joukossa oli aloittelijoita, jotka olivat käyttäneet tekoälyä opintojensa aikana, sekä kokeneempia ohjelmoijia, jotka hyödynsivät tekoälyä päivittäisessä työssään. Tämä monipuolisuus mahdollisti erilaisten näkökulmien esille tuomisen ja kokemusten vertailun taitotasosta riippumatta.

Eskolan ja Suorannan (1998, s. 60) mukaan yksi yleisimmistä kysymyksistä, joka kvalitatiivisen tutkimuksen aineistojen keruun yhteydessä esitetään, koskee aineistojen kokoa. Kuinka paljon aineistoa pitää kerätä, että tutkimus olisi edustavaa, tieteellistä ja yleistettävissä. Omassa työssäni haastattelut, joiden kesto vaihteli 30–50 minuutin välillä toi riittävän kattavan ja laajan aineiston tutkimukseen liittyen. He (s. 61) kuitenkin toteavat, että yleisesti voidaan sanoa, että laadullisessa tutkimuksessa aineistojen koolla ei ole välitöntä merkitystä tai vaikutusta tutkimuksen onnistumiseen. Olen tämän toteamuksen kanssa samaa mieltä. Laajempi tutkimuksesta saatu aineisto ei olisi laajentanut vastauksia tai tehnyt niistä laadukkaampia.

Hirsjärvi ym. (2009, s. 262) korostavat, että ennen tuloksia koskevan osuuden kirjoittamista tutkijan on syytä tarkistaa, että tutkimusongelmiin on saatu vastaus. Toteutettuani laadullisen haastattelututkimuksen ja käytyäni niistä saadut aineistot läpi huomasin, että kaikki haluamiini kysymyksiin saatiin vastaukset, eli haastattelututkimus oli onnistunut.

Laadullinen aineistonkeruu mahdollisti myös yllättävien teemojen ja havaintojen esille nousemisen. Esille nousi erilaisia asioita ja näkökulmia, joita ei varsinaisesti kysymyksiä perusteella odotettu. Tämä näyttää sen, että kvalitatiivinen haastattelututkimus oli oikea valinta haastattelujen toteuttamiseen.

4.3 Haastattelurunko ja toteutus

Haastattelu rakentui viidestä keskeisestä osiosta, jotka mahdollistivat monipuolisen tarkastelun tekoälyn käytöstä ohjelmointityössä ja sen vaikutuksista kehittäjien arkeen.

Ensimmäisessä osiossa keskityttiin haastateltavien taustatietoihin, erityisesti heidän työkokemukseensa ja ohjelmointiin liittyvään historiaansa. Tämä osio tarjosi tärkeää kontekstia tekoälyn käytölle ja auttoi ymmärtämään haastateltavien taitotasoa sekä heidän näkökulmiaan tekoälyn hyödyntämiseen.

Toisessa osiossa kartoitettiin tekoälyn käyttöä ohjelmoinnissa. Keskustelut keskittyivät käytössä oleviin tekoälytyökaluihin ja niiden rooliin ohjelmointiprosessissa. Haastatteluissa pyrittiin selvittämään, millaisiin tehtäviin tekoälyä käytettiin ja miten se integroitiin osaksi päivittäistä työskentelyä.

Kolmannessa osiossa tarkasteltiin tekoälyn tuomia hyötyjä ja sen vaikutusta työn tehokkuuteen. Haastateltavilta kysyttiin, miten tekoäly oli nopeuttanut heidän työskentelyään, vähentänyt virheitä ja mahdollisesti edistänyt uusien taitojen oppimista. Tämä osio tarjosi arvokasta tietoa tekoälyn konkreettisista vaikutuksista ohjelmointityössä.

Neljännessä osiossa keskityttiin haasteisiin ja riskeihin tekoälyn käytössä. Haastateltavat kertoivat kokemuksistaan, jotka liittyivät esimerkiksi tekoälyn mahdolliseen epäluotettavuuteen, sen luomaan riippuvuuteen tai eettisiin huolenaiheisiin. Tämä osio auttoi tunnistamaan tekoälyn käyttöön liittyviä ongelmakohtia.

Lopuksi haastattelut käsittelivät tekoälyn tulevaisuuden näkymiä ohjelmointityössä. Haastateltavat ilmaisivat odotuksiaan ja huolenaiheitaan tekoälyn kehityksestä sekä näkemyksiään sen pitkän aikavälin vaikutuksista ohjelmointiin. Tämä osio tarjosi arvokkaita näkökulmia tekoälyn potentiaalista ja mahdollisista haasteista tulevaisuudessa.

Näiden osioiden avulla haastattelurunko tarjosi kattavan ja monipuolisen lähestymistavan tekoälyn roolin, hyötyjen, haasteiden ja tulevaisuuden kartoittamiseen ohjelmistokehityksessä.

Haastattelut litteroitiin tarkasti analyysiä varten. Aineisto analysoitiin teemakohtaisesti, mikä tarkoittaa, että vastaukset luokiteltiin tutkimuskysymysten mukaisesti teemoihin. Tämä analyysitapa mahdollisti yhtenäisten havaintojen esiin tuomisen sekä erojen ja yhtäläisyyksien tarkastelun eri osallistujien välillä.

Haastattelut toteutettiin anonymisti, jotta osallistujat saattoivat jakaa näkemyksiään ja kokemuksiaan avoimesti. Tämä lisäsi tutkimuksen luotettavuutta ja toi esille syvällisiä näkemyksiä, joita ei välttämättä olisi saavutettu muilla menetelmillä.

Haastattelujen vastauksissa käytän muutamia suoria lainauksia ja merkitsen ne seuraavalla tavalla. Haastateltava 1 = H1, haastateltava 2 = H2 jne. Olen numeroinut haastateltavat haastattelujen järjestyksen perusteella, eli haastateltavaa 1 = H1 olen haastatellut ensimmäisenä ja haastateltavaa 6 = H6 olen haastatellut viimeisenä. Käytän niitä satunnaisessa järjestyksessä, mutta jokainen suora lainaus on numeroitu tarkasti. Haastateltavat 1,2 ja 4 olivat vielä ohjelmoinnin opiskelijoita eivätkä työskentele alalla. Haastateltavat 3, 5 ja 6 ovat valmistuneita ohjelmoinnin parissa työskenteleviä henkilöitä.

4.4 Haastattelurungon rakenne

Haastattelun alussa kerätään taustatietoja haastateltavalta. Ensimmäiseksi pyydetään haastateltavaa kertomaan lyhyesti omasta taustastaan. Tämän jälkeen kysytään kokemuksesta ohjelmoinnin parissa sekä siitä, kuinka kauan haastateltava on ollut mukana ohjelmointityössä.

Seuraavaksi siirrytään käsittelemään tekoälyn käyttöä ohjelmoinnissa ja työssä. Haastateltavalta tiedustellaan avoimella kysymyksellä, kuinka hän käyttää tekoälyä työssään ja ohjelmoinnissaan. Tämän jälkeen pyydetään tarkentamaan, millaisia tekoälytyökaluja tai -ratkaisuja hän hyödyntää päivittäisissä työtehtävissään. Lisäksi keskustellaan siitä, miten tekoäly helpottaa tai tehostaa hänen työtään ohjelmoinnin ja muiden tehtävien osalta.

Haastattelun kolmannessa osiossa keskitytään tekoälyn käytössä koettuihin haasteisiin ja ongelmiin. Haastateltavalta kysytään avoimesti, onko hän kohdannut haasteita tekoälyn hyödyntämisessä ohjelmointityössä. Tämän jälkeen pyydetään tarkentamaan yleisimmät ongelmat ja riskit, joihin hän on törmännyt. Lopuksi keskustellaan siitä, miten hän käsittelee tilanteita, joissa tekoäly tuottaa virheellistä tai epäluotettavaa koodia tai tietoa.

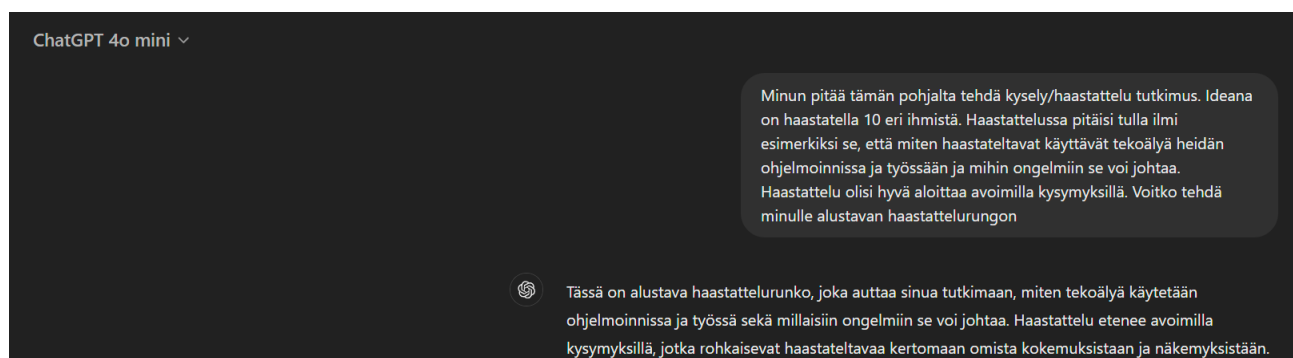
Seuraavassa osiossa pohditaan tekoälyn vaikutusta työprosesseihin. Haastateltavalta kysytään, miten hän kokee tekoälyn vaikuttaneen työprosessien laatuun ja tehokkuuteen.

Lisäksi pyydetään näkemyksiä siitä, onko tekoäly muuttanut tapaa, jolla hän suunnittelee tai toteuttaa ohjelmointiprojekteja.

Haastattelun loppupuolella siirrytään tarkastelemaan tekoälyn tulevaisuutta ja siihen liittyviä näkemyksiä. Haastateltavaa pyydetään kertomaan, millaisena hän näkee tekoälyn tulevaisuuden omassa työssään ja ohjelmoinnissa. Samalla keskustellaan hänen odotuksistaan ja huolenaiheistaan tekoälyn kehityksen suhteen. Haastateltavaa pyydetään myös arvioimaan, missä tekoäly voisi tarjota suurimmat mahdollisuudet ja mitä haasteita se voi tuoda tulevaisuudessa omalla alallaan.

Haastattelu päätetään pyytämällä haastateltavalta mahdollisia toiveita tai ehdotuksia tekoälyn kehittäjille ohjelmoinnin ja työelämän parantamiseksi. Tämä antaa haastateltavalle mahdollisuuden esittää omia näkemyksiään ja ideoitaan tekoälyn kehityksen tueksi.

Haastattelurungon rakenne on ChatGPT-tekoälybotin tuottama minun sille antamieni ohjeiden perusteella. Alhaalla kuvana löytyy annettu prompti, jonka perusteella se loi minulle haastattelurungon.



Kuva 2. ChatGPT-prompti, jolla kyselurunko luotiin.

5 TUTKIMUSTULOKSET

5.1 Tausta ja kokemus ohjelmoinnista

Haastatteluun osallistuneiden ohjelmointikokemus vaihteli huomattavasti, mikä tarjosi laajan näkökulman tutkimuksen teemaan. Kaikki haastatteluun osallistuneet olivat tutustuneet ohjelmointiin vasta opintojensa aikana, ja heidän kokemuksensa opintojen aikana painotuivat kouluprojekteihin ja oppimistehtäviin. Näissä tilanteissa tekoäly toimi tärkeänä apuvälineenä, tarjoten opastusta ja ratkaisuja, jotka auttoivat ymmärtämään ohjelmointia käytännössä. Moni haastateltava korosti, että tekoäly helpotti heidän ensimmäisiä askeliaan ohjelmoinnin parissa, sillä se nopeutti ongelmien ratkaisua ja vähensi oppimiseen liittyvää stressiä.

Kokeneimmalla osallistujilla oli takanaan yli vuosikymmenen kokemus alalta. Kokeneemmat haastateltavat korostivat, että tekoäly täydentää heidän osaamistaan erityisesti toistuvien ja rutiiniluonteisten tehtävien automatisoinnissa. Tämä vapautti heidän aikaansa monimutkaisempiin ja luovempiin tehtäviin.

Tekoälyn avulla rutiinitehtävät, kuten koodirunkojen kirjoittaminen, sujuvat nopeammin, ja voin käyttää enemmän aikaa ongelmien syvälliseen ratkaisemiseen. (H3)

Kaiken kaikkiaan haastattelut osoittivat, että tekoälyn hyötyjen kokeminen ei ole riippuvaista käyttäjän taitotasosta, vaan siitä, miten ja mihin tehtäviin työkalua käytetään.

5.2 Tekoälyn käyttö ohjelmointityössä

Haastatteluun osallistuneet käyttivät tekoälyä ohjelmointityössään monipuolisesti. Suosituimmat käyttökohteet olivat koodin generointi, virheiden korjaaminen ja uusien teknologioiden oppiminen. Nämä tehtävät ovat usein aikaa vieviä ja vaativat tarkkuutta, joten tekoälyn apu koettiin erityisen arvokkaaksi.

Eryityisesti ChatGPT ja GitHub Copilot mainittiin keskeisinä työkaluina. Kaikki osallistujat kertoivat käyttävänsä ChatGPT-tekoälybottia työskentelyn tai opiskelun tukena ja kaikki jo

työelämässä olevat haastateltavat kertoivat käyttävänsä GitHub Copilotia työskentelyn apuna. Osallistujat kertoivat, että tekoäly integroituu heidän työnkulkuunsa tarjoamalla tukea esimerkiksi funktioiden kirjoittamiseen, syntaksivirheiden tunnistamiseen ja dokumentaation luomiseen.

Tekoäly toimii kuin toisena mielenä, joka tarkistaa ja täydentää työtäni reaaliajassa. (H6)

Jokainen haastateltava kertoi käyttävänsä tekoälyä jollain tapaa erityisesti monimutkaisempien ongelmien ratkaisuun. Manuaalisesti virheen etsimiseen olisi mennyt paljon aikaa ja vaivaa, mutta tekoälyn avulla ongelma löytyi suurimmissa osissa tapauksia nopeasti ja vaivattomasti. Tässäkin ilmeni haastatteluissa poikkeuksia. Haastateltavat huomasivat, että tekoäly ei osaa etsiä tai ymmärtää kaikkia ongelmia, jolloin manuaalinen tapa korjata nämä ongelmat ovat ainoa toimiva tapa.

Tekoälyä hyödynnettiin myös uusien taitojen oppimiseen. Useat osallistujat hakivat tekoälyltä apua uuden ohjelmointikielen tai kirjaston haltuun ottamisessa. Yksi haastateltava totesi seuraavasti. Tekoäly tarjosi minulle väsymättömän tietolähteen, joka antoi nopeita ja suoria vastauksia ilman dokumentaation selailua.

(H3) Näiden ominaisuuksien ansiosta tekoäly nähtiin osana modernin ohjelmointityön perusvälineistöä.

5.3 Tekoälyn tuomat hyödyt

Hyödyistä keskeisimpiä olivat tehokkuuden lisääntyminen, oppimisen helpottuminen ja luovuuden edistäminen. Kaikki haastateltavat olivat yhtä mieltä siitä, että tekoäly nopeutti heidän työtään merkittävästi. Ohjelmointitehtävien nopeuttamisen lisäksi tekoäly auttoi vähentämään virheitä ja tarjosi selkeitä esimerkkejä, jotka olivat erityisen hyödyllisiä aloitteleville ohjelmoijille.

Myös tekoälyn kyky tuottaa boilerplate-koodia eli valmiita pohjia nousi esiin haastatteluissa. Useampi haastateltava kertoi käyttävänsä tekoälyä välillä tuottamaan valmiin pohjan, jonka päälle he lähtevät rakentamaan haluttua sovellusta tai projektia. Tämä valmiin

pohjan ominaisuus nopeuttaa ohjelmointia todella paljon vähentämällä manuaalisen koodin kirjoittamiseen kuluvaa aikaa.

Oppimisen tukemisessa tekoäly nähtiin erityisen arvokkaana. Aloittelevat ohjelmoijat kertoivat, että tekoäly vastasi heidän kysymyksiinsä yksinkertaisesti ja selkeästi, mikä teki oppimisesta sujuvampaa. Tekoälyn tarjoama välitön palaute auttoi myös ymmärtämään virheitä ja parantamaan omaa koodausta.

Aloittelijana tekoäly on toiminut opettajanani, tarjoten yksinkertaisia vastauksia monimutkaisiin kysymyksiin. (H4)

Luovuuden edistäminen oli toinen tärkeä hyöty. Haastateltavat korostivat, että tekoäly mahdollisti kynnystä kokeilla uusia ratkaisuja ja ominaisuuksia, joita he eivät olisi muuten uskaltaneet toteuttaa. Tämä näkyi erityisesti projekteissa, joissa tekoäly tarjosi uudenlaisia lähestymistapoja ongelmien ratkaisemiseen.

Eräs haastateltava kertoi tekoälyn tuoneen hänelle niin sanotusti ohjelmoinnin iloa takaisin. Hän kertoi, että nuorempana tuli tehtyä erilaisia hauskoja pieniä ohjelmia ja projekteja hovin vuoksi. Jossain vaiheessa, kun aikaa oli vähemmän käytettävissä, niin vapaa-ajan ohjelmointi jäi ja tämä harmitti haastateltavaa. Nyt kun tekoäly on nopeuttanut ohjelmointia ja vähentänyt niin sanottua manuaalista työtä koodaamisessa, niin haastateltava kertoo aloittaneensa uudestaan omien projektien ja ohjelmien tekemisen vapaa-ajallaan.

Tekoäly on tuonut takaisin ohjelmoinnin iloa, joka oli vuosien varrella päässyt katoamaan. Ohjelmoinnista oli tullut niin monimutkaista ja kynnistä, että ilo oli välillä hävinnyt dokumentaation lukemisen ja erilaisten teknisten yksityiskohtien ymmärtämisen varjoon. (H3)

Tekoälyn avulla voi luoda myös erilaisia kuvia. Tämä on uusi ja vähän tuntematon ominaisuus. Se nousi esiin yhdessä haastattelussa. Haastateltava kertoi, että on käyttänyt tekoälysovellusta, jolle loi ChatGPT:n avulla koodiohjeen ja tällä koodiohjeella tekoäly loi hänelle haluamansa logon, joka tuli käyttöön koulun eräällä kurssilla.

5.4 Haasteet tekoälyn käytössä

Tekoälyn käytössä ilmenneet haasteet liittyivät erityisesti liialliseen riippuvuuteen ja virheellisen koodin tuotantoon ja eettiset kysymykset tekoälyn käsittelemän tiedon kanssa. Useat haastateltavat korostivat, että tekoälyyn ei voi aina luottaa täysin, vaan sen tuottama sisältö vaatii tarkistamista ja muokkaamista.

Jos en tarkistanut tekoälyn tuottamaa koodia kunnolla, virheitä saattoi jäädä huomaamatta, mikä johti myöhemmin ongelmiin. (H5)

Tämä kuvaa, miten tekoäly ei ole erehtymätön, ja käyttäjän on oltava kriittinen sen tarjoamia ratkaisuja kohtaan. Myös toinen haastateltava kertoi tekoälyn tuottaneen hänelle virheellistä koodia ja tämän aiheuttaneen hänelle ongelmia.

Oletin että tekoälyn luoma koodi toimii, mutta huomasin että koodi olikin täynnä erilaisia bugeja ja tajusin jälkeinpäin, että mun piti käyttää varmaan enemmän aikaa sen tekoälyn generoiman koodin korjaamisen, kun mitä siihen olisi mennyt, jos olisin tehnyt sen koodin itse alusta. (H6)

Tällaiset vastaukset haastatteluissa näyttävät, että ei tekoäly pysty tekemään kaikkea, mitä osaava ihminen pystyy. Haastatteluissa tuli ilmi, että tekoäly on ottanut todella suuren harppauksen viimeisin puolentoista vuoden aikana ja varsinkin maksullinen ChatGPT 4 on todella edistyksellinen työkalu, mutta sekin tekee virheitä eikä siihen saa turvautua liikaa.

Haastateltavat toivat toisena tekoälyn haasteena esille huolen siitä, että aloittelevat ohjelmoijat voivat tulla liian riippuvaisiksi tekoälystä. Tämä riippuvuus saattaa heikentää heidän kykyään ratkaista ongelmia itsenäisesti. Eettiset ja tietoturvakysymykset nousivat esiin erityisesti silloin, kun tekoälyä käytettiin arkaluonteisen datan kanssa. Huolenaiheena haastatteluissa nousi esiin myös se, että miten tekoäly käsittelee sille syötettyä tietoa ja koodia.

Tekoälyn käsittelemän datan turvallisuus herättää kysymyksiä, erityisesti kun en tiedä, mihin tietoa tallennetaan ja miten sitä käytetään. (H6)

Eettiset kysymykset tekoälyn ympärillä nousi jokaisessa haastattelussa esiin ja sitä pidetään yhtenä suurimpana, ellei suurimpana ongelmana haastateltavien keskuudessa.

5.5 Tekoälyn vaikutus työprosesseihin

Tekoälyn vaikutukset työprosesseihin olivat erittäin merkittäviä. Useat haastateltavat kertoivat, että tekoäly ei ainoastaan nopeuttanut työntekoa, vaan myös mahdollisti parempien ratkaisujen löytämisen. Työtehtävät, jotka aiemmin veivät päiviä, saattoivat tekoälyn avulla valmistua tunneissa. Tämä vapautti aikaa keskittyä tärkeämpiin ja monimutkaisempiin tehtäviin.

Haastateltavat kertoivat myös, että tekoäly mahdollisti projekteissa sellaisia ratkaisuja, joita ei olisi muuten toteutettu, koska niiden kehittäminen manuaalisesti olisi ollut liian aikaa vievää.

Tekoäly on vaikuttanut työn nopeuteen merkittävästi. Se madaltaa kynnystä tehdä asioita, joita muuten en ehkä toteuttaisi aikapulan vuoksi. (H3)

Esimerkiksi käyttäjäliittymien suunnittelussa tekoäly tarjosi ehdotuksia, jotka säästivät aikaa ja lisäsivät työn laatua. Tekoäly on siis haastattelujen perusteella vaikuttanut tapaan, jolla he suunnittelevat tai toteuttavat erilaisia työprosesseja.

5.6 Tekoälyn tulevaisuus ohjelmoinnissa

Kaikki haastateltavat olivat yhtä mieltä siitä, että tekoälyn merkitys ohjelmoinnissa tulee kasvamaan tulevaisuudessa. He kuitenkin painottivat, että tekoäly ei koskaan täysin korvaa ihmisen asiantuntemusta. Tekoälyn kehittymisen myötä ohjelmoijan rooli voi muuttua, mutta ihmisen kyky tehdä kriittisiä päätöksiä ja ymmärtää kokonaisuuksia säilyy tärkeänä. Tämä näkemys vahvistaa ajatusta siitä, että tekoäly on ennen kaikkea apuväline, joka täydentää ihmisen osaamista sen sijaan, että se toimisi itsenäisesti ratkaisevana tekijänä.

Useat haastateltavat toivoivat tekoälyn kehittäjiltä parempaa selkeyttä ja tarkkuutta erityisesti virheiden tunnistamiseen ja korjaamiseen liittyen.

Olisi hyödyllistä, jos tekoäly pystyisi paremmin selittämään, miksi se ehdottaa tiettyä ratkaisua. (H2)

Tämä osoittaa, että käyttäjät kaipaavat tekoälyltä läpinäkyvyyttä ja perusteluita sen toiminnalle. Haastateltavat toteavat, että tekoälyn tulevaisuuden kehityksessä on tärkeää keskittyä sen käyttäjäystävällisyyteen, turvallisuuteen ja eettiseen toimintaan, jotta se voi lunastaa täyden potentiaalinsa ohjelmoinnin työkaluna.

6 POHDINTA

6.1 Tekoälyn merkitys ohjelmoijan oppimisessa ja työssä

Tekoäly on vakiinnuttanut paikkansa ohjelmoinnin kentällä monipuolisena ja arvokkaana työkaluna. Haastattelujen perusteella sen merkitys oppimisessa ja työssä on huomattava. Haastatteluun osallistuneille tekoäly tarjosi paitsi teknistä tukea myös jatkuvasti saatavilla olevan "opettajan", joka auttoi heitä navigoimaan ohjelmoinnin perusteiden ja käytännön työtehtävien läpi. Erityisesti aloitteleville käyttäjille tekoäly tarjosi yksinkertaisia ratkaisuja ja selkeää opastusta.

Teoriaosuudessa Qureshi (2024) kertoo, että tekoäly voi myös toimia mentorina tarjoamalla välitöntä palautetta ja koodiehdotuksia, mikä nopeuttaa oppimisprosessia. Haastattelutulokset vahvistavat tämän, sillä ne osoittivat, että tekoäly toimii erityisesti aloittelevien ohjelmoijien reaaliaikaisena teknisenä tukena ja "virtuaalisena opettajana". Myös Savaniyan (i.a.) mukaan tekoäly voi myös tukea oppimista reaaliaikaisella analytiikalla ja ehdotuksilla. Tämä vahvistaa entisestään sitä faktaa, että tekoäly toimii aloittelevan ohjelmoijan apuna ja opettajana.

Oppimisessa tekoälyn merkitys korostui erityisesti sen kyvyssä antaa välitöntä palautetta ja selkeitä ohjeita. Haastateltavat mainitsivat, että tekoäly ei pelkästään tarjoa oikeaa ratkaisua, vaan auttaa ymmärtämään, miksi tietty ratkaisu toimii. Tämä on erityisen hyödyllistä käyttäjille, jotka eivät vielä hallitse ohjelmoinnin perusteita täysin. Tämä kertoo tekoälyn potentiaalista oppimisen yksilöllistämässä ja sujuvoittamisessa.

Kokeneemmille käyttäjille tekoäly oli ennen kaikkea tehokkuuden lisääjä. Haastattelujen perusteella se automatisoi aikaa vieviä tehtäviä, kuten rutiinikoodin kirjoittamista ja virheiden etsimistä. Tämä mahdollisti resurssien siirtämisen strategisempiin ja monimutkaisempiin tehtäviin.

6.2 Hyödyt ja mahdollisuudet

Olen teoriassa tutkinut tekoälyn hyötyjä, kuten tehokkuuden kasvua ja luovuuden tukemista, ja nämä käsitykset ovat vahvistuneet haastattelujen perusteella. Haastattelut toivat esiin tekoälyn monipuoliset hyödyt, jotka liittyivät erityisesti tehokkuuden, oppimisen ja luovuuden edistämiseen. Suurimpana hyötynä osallistujat kokivat sen kyvyn säästää aikaa tehtävissä, jotka muuten olisivat olleet hitaita ja vaativia. Tätä tukee myös Qureshi (2024), jonka mukaan tekoälytyökalut vähentävät manuaalisen työn määrää ja mahdollistavat nopeamman kehitysprosessin. Esimerkiksi koodin generoinnissa tekoäly tarjosi pohjia, joita haastateltavat pystyivät muokkaamaan tarpeidensa mukaan, mikä nopeutti varsinaiseen ongelmanratkaisuun siirtymistä.

Oppimisen näkökulmasta tekoäly madalsi kynnystä aloittaa uusia ohjelmointitehtäviä. Useat haastateltavat kertoivat, että tekoäly auttoi heitä omaksumaan uusia ohjelmointikieliä ja -teknologioita nopeammin kuin perinteisin menetelmin. Tämä kuvastaa tekoälyn roolia oppimisprosessien nopeuttajana erityisesti alalla, jossa teknologiat kehittyvät nopeasti.

Luovuuden tukeminen nousi myös merkittäväksi hyödyksi. Haastateltavat kuvasivat, kuinka tekoäly madalsi kynnystä kokeilla uusia ideoita ja lähestymistapoja, joita he eivät olisi muuten uskaltaneet tai ehtineet toteuttaa. Esimerkiksi tekoäly ehdotti vaihtoehtoisia ratkaisuja, joita ohjelmoija ei olisi itse tullut ajatelleeksi. Tämä näkyi erityisesti projekteissa, joissa käyttäjät saivat uusia oivalluksia tekoälyn avulla.

Haastatteluissa esiinnoussut tekoälyn avulla luotu logo kouluprojektiin ja tapahtumaan yllätti haastatteluista tehdessä. Missään teorian lähteessä ei ole mainittu tekoälyn kykyä generoida kuvia, vaikka sillä sellainen on. Tämä tuli minulle yllätyksenä, sillä luin jonkin aikaa sitten uutisen, jossa kerrottiin henkilön voittaneen valokuvauskilpailun tekoälyn generoimalla kuvalla.

6.3 Haasteet ja riskit

Teoriassa nousi esille tekoälyn käytön haasteita, kuten sen päätöksenteon läpinäkyttömyyttä ja riippuvuuden riskin. Haastattelut toivat esiin samoja tekoälyn käyttöön liittyviä haasteita ja riskejä. Yksi merkittävimmistä ongelmista oli tekoälyn tuottaman koodin

tarkkuus ja luotettavuus. Vaikka tekoäly pystyi tarjoamaan nopeita ratkaisuja, se ei aina ottanut huomioon kaikkia mahdollisia poikkeustilanteita tai erityistarpeita. Tämä asetti käyttäjälle vastuun tarkistaa ja korjata tekoälyn tuotoksia.

Lisäksi haastatteluun osallistuneet korostivat huolta liiallisesta riippuvuudesta. Tämä nähtiin erityisen haitallisena aloitteleville käyttäjille, joiden ongelmanratkaisutaidot eivät välttämättä päässeet kehittymään itsenäisesti, jos tekoälyä käytettiin liikaa. Eettiset kysymykset, kuten datan yksityisyys ja tietoturva, herättivät myös huolta. Haastateltavat toivoivat tekoälyn kehittäjiltä lisää läpinäkyvyyttä ja selkeyttä näiden kysymysten ratkaisemiseen. Tätä vastausta tukee myös Qureshi (2024), jonka mukaan tekoälyn päätöksenteon läpinäkyväisyys voi aiheuttaa riskejä, jos ohjelmointikehityksessä käytettävää tekoälyä ei säännellä riittävästi.

6.4 Tekoälyn vaikutukset työprosesseihin

Teoriaosuudessa tarkastelin, kuinka tekoäly voi muuttaa ohjelmointiprosessien dynamiikkaa, ja havainnot haastatteluista vahvistavat tämän. Tekoäly on muuttanut ohjelmointiprosesseja merkittävästi. Haastattelut toivat esiin, että tekoäly ei ainoastaan nopeuta yksittäisiä tehtäviä, vaan muuttaa myös kokonaisvaltaisesti työskentelytapoja. Osallistujat kertoivat, että tekoälyn avulla he pystyivät keskittymään enemmän strategiaan ja luoviin tehtäviin, kun taas aikaa vievät rutiinitehtävät automatisoitiin.

Erityisen merkittävä havainto oli tekoälyn kyky alentaa kynnystä aloittaa uusia projekteja. Haastateltavat kuvasivat, kuinka tekoäly mahdollisti nopean prototyyppien luomisen ja ideoiden testaamisen. Tämä ei ainoastaan lisännyt työn tehokkuutta, vaan vaikutti myös positiivisesti motivaatioon ja luovuuteen. Tekoälyn positiivinen vaikutus motivaatioon ja luovuuteen ei noussut suoraan esiin teoriaosuudessa. Tämä voi kertoa siitä, että tekoälyä ei ole tutkittu henkisen tai psyykkisen hyvinvoinnin kannalta.

6.5 Suositukset ja parannustoiveet

Haastateltavat esittivät useita parannustoiveita tekoälyn kehittäjille. Yksi keskeinen toive oli tekoälyn läpinäkyvyyden lisääminen. Osallistujat halusivat ymmärtää paremmin, miten

tekoäly tekee päätöksensä ja miksi se ehdottaa tiettyjä ratkaisuja. Tämä parantaisi käyttäjien luottamusta tekoälyyn ja tukisi oppimista.

Toinen tärkeä parannusehdotus liittyi tekoälyn virheiden käsittelyyn. Haastateltavat toivoivat, että tekoäly voisi tunnistaa paremmin omat rajoituksensa ja tarjota vaihtoehtoisia ratkaisuja silloin, kun alkuperäinen ehdotus ei ole käyttökelpoinen. Lisäksi tietoturva ja datan yksityisyys nousivat tärkeiksi teemoiksi, joiden kehittäminen nähtiin keskeisenä käyttäjäkokemuksen parantamisessa. Babenkon (2024) mukaan tekoälyn lisääntynyt käyttö ohjelmistokehityksessä voi tuoda mukanaan ongelmia, kuten algoritmien vinoumat ja tietoturvariskit. Tämä tukee myös haastatteluista saatuja vastauksia, mutta teoriaa tutkiessa tätä asiaa ei nostettu niin suurena uhkana esiin, kuin se oikeasti on.

7 JOHTOPÄÄTÖKSET

Tekoäly on vakiinnuttanut paikkansa ohjelmoinnin alalla monipuolisena ja arvokkaana työkaluna. Haastattelujen perusteella sen käyttö tarjoaa huomattavia etuja sekä oppimiseen että työtehtävien suorittamiseen. Tutkimustulokset osoittavat, että tekoälyllä on potentiaalia muuttaa ohjelmointityön dynamiikkaa merkittävästi, erityisesti sen vaikutuksen kautta tehokkuuteen, oppimiseen ja luovuuteen. Vaikka nämä edut ovat kiistattomia, tekoälyn käytön yhteydessä on myös huomioitava siihen liittyvät haasteet, jotta sen kehittäminen ja käyttö olisi mahdollisimman hyödyllistä ja vastuullista.

Ensimmäinen keskeinen havainto liittyy työn tehokkuuteen. Haastattelut korostivat, että tekoäly auttaa ohjelmoijia vähentämään aikaa vieviä rutiinitehtäviä, kuten koodin generointia, virheiden tunnistamista ja yksinkertaisten tehtävien automatisointia. Tämä vapauttaa käyttäjien aikaa strategisempien ja luovempien tehtävien suorittamiseen. Tämä on erityisen arvokasta projekteissa, joissa aikataulut ovat tiukkoja ja tarvitaan nopeita ratkaisuja muuttuviin tilanteisiin. Tekoäly pystyy myös tarjoamaan vaihtoehtoisia lähestymistapoja monimutkaisiin ongelmiin, mikä laajentaa käyttäjän näkökulmaa ja mahdollisuuksia.

Oppimisen tukena tekoäly osoittautui erityisen arvokkaaksi haastateltaville, etenkin aloitteleville käyttäjille. Tekoäly tarjoaa reaaliaikaisia vastauksia ja selityksiä, mikä tekee oppimisprosessista nopeamman ja vähemmän turhauttavan. Tärkeä ominaisuus on sen kyky yksilöllistää oppimiskokemus tarjoamalla käyttäjän taitotason ja tarpeisiin räätälöityjä ratkaisuja. Tämä voi olla erityisen hyödyllistä ohjelmointikielissä ja teknologioissa, joissa teorian ja käytännön välinen kuilu on suuri.

Huolimatta merkittävistä hyödyistä tekoälyn käyttöön liittyy myös kriittisiä haasteita. Yksi keskeisimmistä huolenaiheista on liiallinen riippuvuus tekoälystä. Tämä nähtiin erityisen haitallisena oppimisvaiheessa, jossa ohjelmoijien on tärkeää kehittää itsenäisiä ongelmanratkaisutaitoja. Tekoälyn tuottamien ratkaisujen sokea hyväksyminen voi heikentää käyttäjän ymmärrystä siitä, miksi ratkaisu toimii. Toinen merkittävä haaste liittyy tekoälyn tuottaman koodin tarkkuuteen. Vaikka tekoäly pystyy tarjoamaan nopeita ratkaisuja, sen tuottama koodi ei aina ole virheetöntä tai kontekstiin sopivaa. Tämä asettaa käyttäjälle vastuun tarkistaa ja tarvittaessa korjata tekoälyn tuotoksia.

Eettiset kysymykset nousivat myös haastatteluissa esiin. Datan yksityisyys ja tietoturva olivat keskeisiä huolenaiheita, jotka vaikuttavat käyttäjien luottamukseen tekoälyä kohtaan. Haastateltavat toivoivat, että tekoälyn kehittäjät panostaisivat enemmän näiden kysymysten ratkaisemiseen ja tekisivät tekoälyn toiminnasta läpinäkyvämpää. Läpinäkyvyyden lisääminen koettiin erityisen tärkeäksi, jotta käyttäjät voisivat ymmärtää tekoälyn päätöksenteon perusteet ja oppia sen rinnalla.

Tekoälyn tulevaisuus ohjelmoinnissa näyttää lupaavalta, mutta se asettaa uusia vaatimuksia sekä käyttäjille että kehittäjille. Ohjelmoijien on opittava käyttämään tekoälyä kriittisesti ja ymmärrettävä sen rajoitukset, jotta he voivat hyödyntää sen tarjoamia mahdollisuuksia vastuullisesti. Kehittäjien puolestaan on parannettava tekoälyn luotettavuutta ja toiminnan läpinäkyvyyttä, jotta käyttäjien luottamus työkaluihin säilyy.

Johtopäätöksenä voidaan todeta, että tekoäly on voimakas ja hyödyllinen työkalu, mutta sen käyttö edellyttää tasapainoa. Käyttäjien on ylläpidettävä ja kehitettävä omia ohjelmointitaitojaan, vaikka tekoäly tarjoaakin nopeita ratkaisuja. Lisäksi tekoälyn kehittäjien tulee ottaa huomioon käyttäjien tarpeet ja parannustoiveet, kuten läpinäkyvyyden lisääminen, virheiden käsittelyn parantaminen ja eettisten kysymysten ratkaiseminen.

Kun käyttäjät ja kehittäjät työskentelevät yhdessä vastuullisen kehityksen ja käytön eteen, tekoäly voi saavuttaa täyden potentiaalinsa ja toimia kestävässä osana ohjelmoinnin ekosysteemiä. Näin tekoäly voi kehittyä paitsi tehokkuuden ja oppimisen työkaluksi myös innovaatioiden katalysaattoriksi, joka muuttaa ohjelmointityön mahdollisuuksia ja käytäntöjä radikaalisti.

LÄHTEET

- Babenko, I. (1.2024). *How artificial intelligence is transforming software development*. Insoftex. <https://insoftex.com/artificial-intelligence-in-software-development/>
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. <https://doi.org/10.1145/3442188.3445922>
- Cloudflare. (i.a.). *What is a large language model?* <https://www.cloudflare.com/learning/ai/what-is-large-language-model/>
- Copeland, B. J. (16.11.2024) *Alan Turing*. Britannica. <https://www.britannica.com/biography/Alan-Turing>
- Coursera, (26.3.2024) *What is Chat GPT? (And how to use it)*. <https://www.coursera.org/articles/chatgpt>
- Dickson, B. (5.6.2023) *What Is ChaGPT? A Basic Explainer*. <https://uk.pcmag.com/ai/147184/what-is-chatgpt-a-basic-explainert>
- Eskola, J., & Suoranta, J. (1998). *Johdatus laadulliseen tutkimukseen*. Vastapaino.
- Euroopan tietoteknologien sertifiointiakatemia (26.11.2023). *Mitä eroa on ohjatun ohjaamattoman ja vahvistavan oppimisen lähestymistapojen välillä?* <https://fi.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/introduction/what-is-machine-learning/what-are-the-distinctions-between-supervised-unsupervised-and-reinforcement-learning-approaches/>
- GitHub. (i.a.). *GitHub Copilot*. <https://github.com/features/copilot>
- Guillot, J. D. (20.06.2023). *Mitä tekoäly on ja mihin sitä käytetään?* Euroopan parlamentti. https://www.europarl.europa.eu/pdfs/news/expert/2020/9/story/20200827STO85804/200827STO85804_fi.pdf
- Gulen, K. (18.4.2023). *AI 101: A beginner's guide to the basics of Artificial Intelligence*. Dataconomy. <https://dataconomy.com/2023/04/18/basics-of-artificial-intelligence/>.
- Hirsjärvi, S., Remes, P., Sajavaara, P., Sinivuori, E., & Sinivuori, E. (2009). *Tutki ja kirjoita* (15. uud. p.). Tammi.
- IBM. (i.a.). *Large language models*. <https://www.ibm.com/topics/large-language-models>

- Innodata (i.a.) *Artificial general intelligence vs generative AI: which is the future?* Innodata <https://innodata.com/artificial-general-intelligence-vs-generative-ai-which-is-the-future/>
- Klusaitė, L. (20.3.2023) *Mitä on koneoppiminen?* NordVPN <https://nordvpn.com/fi/blog/koneoppiminen/>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Maria, T. (2024a) *Heikko tekoäly*. Technopedia <https://www.techopedia.com/fi/sanasto/heikko-tekoaly>
- Maria, T. (2024b) *Vahva tekoäly*. Technopedia <https://www.techopedia.com/fi/sanasto/vahva-tekoaly>
- Marjamaa, M. (19.11.2019) *Mitä on tekoäly?* Kreodi. <https://www.kreodi.fi/arkisto/artikkelit/mita-on-tekoaly.html>
- McCorduck, P. (2003) *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. A. K. Peters, Ltd. [https://monoskop.org/images/1/1e/McCorduck Pamela Machines Who Think 2nd ed .pdf](https://monoskop.org/images/1/1e/McCorduck_Pamela_Machines_Who_Think_2nd_ed.pdf)
- Mitchell, T. M. (1997) *Machine learning*. McGraw Hill. <https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf>
- Nordal, O. (i.a.) *Alan Turing*. Store Norske Leksikon https://snl.no/Alan_Turing
- Numminen, L. (2023a) *Mitä on heikko tai kapea tekoäly?* Finnishup.com <https://www.finnishup.com/mita-on-heikko-tai-kapea-tekoaly/>
- Numminen, L. (2023b) *Mitä on vahva tekoäly, eli AGI?* Finnishup.com. <https://www.finnishup.com/mika-on-vahva-tekoaly-eli/>
- OpenAI. (2023). *ChatGPT: Advancing conversational AI*. <https://openai.com>
- Qureshi, M. (9.4.2024) *AI In Software Development: Benefits + Risk + Challenges* Geeks of Kolachi <https://geeksofkolachi.com/blogs/ai-in-software-development/>
- Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3. painos). Pearson. http://repo.darmajaya.ac.id/4836/1/Stuart%20Russell%2C%20Peter%20Norvig-Artificial%20Intelligence_%20A%20Modern%20Approach-Prentice%20Hall%20%28%20PDFDrive%20%29.pdf

Tableau. (i.a.). *A brief history of artificial intelligence*. <https://www.tableau.com/data-insights/ai/history>

Tekoälykeskus. (24.5.2024). *Tekoälyn luonnollisen kielen käsittely (NLP): Miten koneet ymmärtävät meitä paremmin kuin koskaan?* <https://www.tekoalykeskus.fi/tekoalyn-luonnollisen-kielen-kasittely-nlp-miten-koneet-ymmartavat-meita-paremmiin-kuin-koskaan/>

Turing, A. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460. <https://doi.org/10.1093/mind/LIX.236.433>.

Savaniya, N. (i.a.) *AI in Software Development: Role, Benefits, Impacts & Future* BrainerHub. <https://www.brainerhub.com/blog/ai-in-software-development/>