

The Doctrine of Product Liability and Negligence Cannot Be Applied to Malware-Embedded Software

Ayup Suran Ningsih

Ayup Suran Ningsih
Faculty of Law, Universitas Negeri Semarang, Indonesia
Department of Private and Commercial Law
✉ ayuupp@mail.unnes.ac.id

TABLE of CONTENTS

INTRODUCTION	8
THE CONCEPT OF PRODUCT LIABILITY	9
ANALYSIS OF THE DOCTRINE OF PRODUCT LIABILITY AND NEGLIGENCE CANNOT BE APPLIED TO MALWARE-EMBEDDED SOFTWARE	11
CONCLUSION	19
REFERENCE	19

 10.15294/jils.v4i01.29157



Copyright © 2019 by Author(s)
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. **All writings published in this journal are personal views of the authors and do not represent the views of this journal and the author's affiliated institutions.**

Article Info

Submitted on February 2019
Approved on February 2019
Published on May 2019

Keywords:
Information Technology
Law, Product Liability,
Negligence, Malware-
Embedded Software

Abstract

Today, the development of technology is remarkable; the world has faced the industrial era 4.0 where people are now more popular to carry out various financial transactions, both the process of buying and selling and other financial transactions through digital transactions. This digital transaction is run by an information system and is provided with special software that runs it. Damage to computer devices and software can cause all kinds of damage. This damage can cause someone to experience damage or loss due to damaged hardware or software, one or more of the following legal areas can provide recovery; such as contract law; technology law; consumer protection; and product liability. This article is to examine the doctrine of product liability and negligence cannot be applied to malware-embedded software. The approach of the research method used in this article is normative juridical. The normative juridical approach is an approach carried out based on the main legal material by examining theories, concepts, legal principles and laws and regulations related to this research.

HOW TO CITE (Chicago Manual Style)

Ningsih, Ayup Suran. "The Doctrine of Product Liability and Negligence Cannot Be Applied to Malware-Embedded Software", *JILS (Journal of Indonesian Legal Studies)*, 4 (1), 2019: 7-20

INTRODUCTION

SOFTWARE is undoubtedly the driving force of the information society. There have been occasions when defects in software have had very serious consequences. The term 'safety-critical' is applied to software (and hardware) which is used in situations involving risk to life and limb. Defect in computer equipment and software can cause all manner of damage. The failure of flight control systems, nuclear power station systems and defense systems could result in major loss of life. If a person suffers loss or damage as a result of defective hardware or software, one or more of the following areas of law might provide a remedy: contract; law of negligence; negligent misstatement; or product liability (Bainbridge 2008). Liability laws designed to compensate

for harms caused by defective products may also affect innovation incentives (Galasso & Luo 2018).

Defects in software controlling financial transactions may result in economic loss and as shown above there is clearly capacity for physical damage resulting in such defects. Predictably enough, those in the software industry are concerned as to the likelihood of personal and corporate liability when undetected faults in software precipitate such events. From the point of view of potential plaintiffs, it is more a question of whether there will be recourse to any remedy if they are injured as a result of such incidents when the cause of the problem might have been neither reasonably foreseeable nor even detectable.

Liability law with respect to computer software has important implications. Potential lawsuits act as both a deterrent to software development as well as an incentive for the creation of reliable software. While other areas of tort law have been present for generations, tort law with respect to computer software is a new area of law. It is important for computer scientist to play a role in the policy-making process of this field as new laws and precedents are developed. This article is to examine the doctrine of product liability and negligence cannot be applied to malware-embedded software.

The approach of the research method used in this article is normative juridical. The normative juridical approach is carried out by analyzing and interpreting theoretical matters concerning principles, conceptions, doctrines and legal norms relating to information technology law. The normative juridical approach is an approach carried out based on the main legal material by examining theories, concepts, legal principles and laws and regulations related to this research. This approach is also known as the literature approach, which is by studying books, regulations and other documents related to this research (Irianto & Shidarta 2011).

THE CONCEPT OF PRODUCT LIABILITY

TORT law regards software as a product, rather than a service, will also play a role in determining the application of product liability to software defects. A “product” is defined under Products Liability as a “tangible personal property distributed commercially for use or consumption” (Sunghyo 2017). Unlike other component parts of a vehicle, because software is not a tangible “manufactured product,” a court might find that manufacturing defect theory is not applicable for software or algorithmic errors. While many courts have applied contract law in software related cases under the Uniform Commercial Code, software manufacturers have not been found strictly liable for software defects based on tort product liability theories.

Products liability refers to the liability of any or all parties along the chain of manufacture of any product for damage caused by that product. This includes the manufacturer of component parts (at the top of the chain), an assembling manufacturer, the wholesaler, and the retail store owner (at the bottom of the chain).

Products containing inherent defects that cause harm to a consumer (or someone to whom the product was loaned, given, etc.) of the product would be the subjects of products liability suits. While products are generally thought of as tangible personal property, products liability has stretched that definition to include intangibles (for example gas), naturals (for example pets), real estate (for example house), and writings (for example navigational charts). Products liability is derived mainly from torts law.

Types of Products Liability Claims

Products liability claims can be based on negligence, strict liability, or breach of warranty of fitness. This will typically depend on the jurisdiction within which the claim is based, due to the fact that there is no federal products liability law.

Defects That Create Liability

There are three types of product defects that incur liability in manufacturers and suppliers:

a. Design Defects

Design defects are inherent, as they exist before the product is manufactured. While, the item might serve its purpose well, it can be unreasonably dangerous to use due to a design flaw.

b. Manufacturing Defects

Manufacturing defects occur during the construction or production of the item. Only a few out of many products of the same type are flawed in this case.

c. Defects in marketing

Defects in marketing deal with improper instructions and failures to warn consumers of latent dangers in the product.

d. Strict Liability

Products Liability is generally considered a strict liability offense. With regard to products liability, a defendant is liable when the plaintiff proves that the product is defective, regardless of the defendant's intent. It is irrelevant whether the manufacturer or supplier exercised great care; if there is a defect in the product that causes harm, he or she will be liable for it.

ANALYSIS OF THE DOCTRINE OF PRODUCT LIABILITY AND NEGLIGENCE CANNOT BE APPLIED TO MALWARE-EMBEDDED SOFTWARE

1. Is Software a Product or Service?

Software is defined as a device that is part of a computer that is not in the form of hardware, which can specifically be interpreted as a computer program. In Article 1 number 14 of Law Number 11 of 2008 concerning Electronic Information and Transactions (ITE Law), computers are defined as "tools for processing electronic, magnetic, optical, or system data that carry out functions of logic, arithmetic, and storage." This software was built with the aim of running an electronic system. Electronic systems are defined in Article 1 number 5 of the ITE Law as a series of electronic devices and procedures that function to prepare, collect, process, analyze, store, display, announce, transmit, and/or distribute Electronic Information. While the definition of electronic information is given in article 1 number 1 of the ITE Law as follows:

"Electronic Information is one or a set of electronic data, including but not limited to writing, sound, images, maps, designs, and photos, electronic data interchange (EDI), electronic mail (electronic mail), telegram, telex, telecopy or the like, letters, signs, numbers, codes, processed access, symbols, or perforations that have meaning or can be understood by people who are able to understand them."

Some of the problems surrounding the categorization of software from a legal point of view have already surfaced, usually in the commercial field and relating to whether supply of software can be properly classified as supply of goods or supply of services. Computer programs frustrate the law's traditional categories; they exhibit characteristics of both concrete property and abstract knowledge (Rowland 1991). Based on Section 2(1) Consumer Protection Act 1987 requires that:

...where any damage is caused wholly or partly by a defect in a product, every person to whom subsection (2) below applies shall be liable for the damage.

The people referred to in subsection (2) are a producer, an own-brander or an importer. Product is defined in section (1) as:

Any goods or electricity and (subject to subsection (3) below) includes a product which is comprised in another product,

whether by virtue of being a component part or raw material or otherwise.

In other words, the components comprising a product are also treated as products in their own right. It is well known these days that computers consist of both hardware and software and so in broad terms these could be termed the components of the computer system. Such a naive analysis would suggest that software would attract the application of the Consumer Protection Act in the same way as any other component of a product. The simplicity of this approach may be challenged by the unique nature of software.

It is the dichotomy between the tangible and intangible nature of software which lies at the heart of the problem of applying the existing legal provisions. In relation to product liability this has become of paramount importance only relatively recently, not only because of the Consumer Protection Act, but also because it is only within this time scale that computer-controlled systems have really begun to impinge on the life of the man in the street.

If software can be classified as a product, then there will be liability if there is a defect in the software and that defect causes damage. All software errors may not give rise to defects in this sense; only the ones which could lead to damage.

Furthermore, it is also should be distinguished some different types of software. Software performs many functions and there have been attempts to distinguish certain types primarily as to whether they constitute goods or services but this discussion has also spilled over into the debate as to whether software can be regarded as a product. Software can basically be divided in to two classes; embedded software and applications software.

Embedded software is the software which is supplied with the system by the manufacturer, it is available as soon as the system is switched on and is very difficult for anyone other than the producer to change. Embedded software is nearly always firmware, a generic term for software in ROM (read only memory); it is always in the computer's memory and starts executing the program immediately. Consider a lift control system for example. It is required that the moment the system is on the program is executed and remains in operation until the system is switched off again. This is achieved by means of embedded software or firmware supplied as an integral part of the lift system.

Applications software on the other hand causes a system to perform a particular function, thus a general purpose PC can be loaded with different software packages to provide spread sheet, word processor, database functions etc. In such a general purpose computer, the amount of embedded software is minimal, just sufficient to allow the computer to read in the information supplied by the applications software. Such software is frequently 'off-the shelf' but may also be specially written to enable a general system to carry out a specific task (Rowland 1991).

2. The Nature of Software Defects

Prior to considering issues of legal liability, it might be helpful to attempt a brief analysis of the nature of the differences which exist between software and the tangible product with which society and the law are more familiar. Defects in a traditional product such as a motor car may originate in one of two ways. Design defects in a traditional product such as a motor car may originate in one of two ways. Design defects relate to some failure at the design stage, with the consequence that the failure node will be exhibited in every species of the product. A more commonplace form of defect is introduced during the production stage (Lloyd 2011).

Where software is concerned, the nature of the digital copying process is such that there can be a high degree of confidence that every copy of software will be identical. If particular copies are corrupted, the likelihood is that they will not work at all, so that any defect becomes apparent before any damage is caused. If customer should wish to establish that a copy of a word processing program which has been purchased is not of satisfactory quality, argument will have to proceed by reference to word processing programs produced by other producers and to general standards. Although the task can be accomplished, it is a significantly more onerous burden than that faced by a person claiming the existence of a production defect (Lloyd 2011).

3. Basis of Liability

Yusuf et.al (2016) emphasized that liability can arise in four basic ways, as follows:

- a. Direct
 - 1) A defective software program causes a radiology machine to malfunction, burning a patient.
 - 2) Hardware malfunction, e.g. computer catches fire
- b. Indirect
 - 1) A software produces incorrect information which feeds directly into a physical process, for example ATM dispenses notes
 - 2) Software produces incorrect information which is relied on by a human mind, for example computer-controlled traffic signals, reliance on spreadsheet calculations to build a bridge or calculate tax liability
 - 3) A bug cause a cardiology machine to produce inaccurate information. The physician relies upon the machine as being correct, and administers the wrong treatment.
- c. Negligence

Negligence is an arrangement in which liability is established only after it is shown that a producer failed to take a given level of care in producing the product. Software functions normally, but a technician uses the machine improperly, administers the wrong treatment, or misinterprets results.

Under the negligence interpretation of liability, the victim would need to prove that the manufacturer of the software failed to develop and test its product well enough to the point where it was reasonably confident that the product was safe to operate, or that the operator of the software failed to use the software correctly or grossly failed to interpret the software's finding correctly.

d. No-fault

Software functions properly and medical personnel act appropriately. However, injury occurs because of imperfections of the test or the test is not designed to find the patient's specific abnormality.

4. What is Malware?

Malicious software (malware) is any software that gives partial to full control of your computer to do whatever the malware creator wants. Malware can be a virus, worm, trojan, adware, spyware, root kit, etc. The damage done can vary from something slight as changing the author's name on a document to full control of your machine without your ability to easily find out. Most malware requires the user to initiate its operation. Some vectors of attack include attachments in e-mails, browsing a malicious website that installs software after the user clicks ok on a pop-up, and from vulnerabilities in the operating system or programs. Malware is not limited to one operating system. Malware types can be categorized as follows: viruses, worms, trojans, and backdoors seek to infect and spread themselves to create more havoc. Adware and spyware seek to embed themselves to watch what the user does and act upon that data. Root kits seek to give full access of your machine to the attacker to do what they want ([Anonym, UCLA 2014](#)).

5. Negligence

Negligence is part of an area of law known as tort. Basically, a tort is a civil wrong, independent of contract. It imposes legal liabilities on a person who has acted carelessly or unreasonably omits to do something. Under certain circumstances a person will be liable to another for failing to exercise a required duty of care. A claim in negligence does not depend on the presence of a contract, so if the person injured is someone other than the buyer, that person can still sue. The buyer also should be able to sue, but on the basis of breach of contract if the item is defective and fails to comply with implied terms such as those concerning satisfactory quality and fitness for purpose. To be able to sue in negligence, three essential ingredients must be present ([Hermana & Silfianti 2011](#)):

- a. A duty of care owed to the injured party;
- b. A breach of that duty of care; and
- c. Consequential loss (loss which is a direct and natural result of the breach of duty of care)

Negligence can be thought of as an early form of product liability and has developed over the years to its present wide scope, although this is

tempered to some extent by the growth of insurance. It is also limited, to some extent, by police considerations. This is particularly so where the loss is purely economic or the claim is in respect of nervous shock or if a professional would be exposed to an unlimited number of claims from persons other than those for whom he performed his duties.

In the other conditions, when negligence and computers analyzed, stated that computers and computer software could kill or cause serious injury; however, negligent liability does not stop at personal injury but extends to damage to property. Computer software has the potential to cause serious loss of life as well as causing economic losses. It is possible that the software developer was negligent in writing and testing of the software. The fact that an action in negligence lies without the need for a contract is important both for computer program writers and manufactures of computer equipment. If a program is licensed by a publisher, the program author could be liable in negligence even though he is not a party to the license agreement (Setiadi, Sucahyo and Hasibuan 2012).

There are limitations, however, to the scope of the law of negligence. A person writing a computer program, or company manufacturing computer equipment, will not necessarily is potentially liable to the world at large in negligence. The person/company will be liable, however, to those whom they could contemplate being adversely affected by any negligent act or omission by them. A future limiting factor is that the claimant bears the burden of proof; he has to show that the defendant was negligent and this is not always easy to do.

6. Product Liability and Software

Transactions carried out electronically are basically engagements or legal relationships carried out electronically by combining computer-based electronic system networks with communication systems, which are further facilitated by the existence of a global computer network or internet (vide Article 1 number 2 of the ITE Law).

A legal relation is a relationship between two or more parties (legal subjects) that have legal consequences (giving rise to rights and obligations) and are regulated by law. In this case the right is the authority or role that is in someone (the holder) to act on something that is the object of that right to another person. Whereas, the obligation is something that must be fulfilled or carried out by a person to obtain his rights or because he has already obtained his rights in a legal relationship.

The object of law is something that is useful, valuable, and valuable to the legal subject and can be used as the subject of legal relations. While, the legal subject is anything that can be a supporter of their rights and obligations or has legal authority (*rechtsbevoegdheid*).

In the private sphere, the legal relationship will include relations between individuals, while in the public sphere, the legal relationship will include relations between citizens and the government and relations between

fellow members of the community that are not intended for commercial purposes, which include public services and information transactions between Government organizations.

In commercial activities, transactions have a very important role. In general, the meaning of transactions is often reduced as a sale and purchase agreement between the parties that agree to it, even though in a juridical perspective, the terminology of the transaction is basically the existence of an agreement or legal relationship that occurs between the parties. The juridical meaning of transactions is basically emphasized in the material aspects of the legal relationship agreed upon by the parties, not formally legal actions. Therefore, the existence of legal provisions regarding the engagement remains binding even though there are changes in the media and changes in procedures for transactions. This is of course an exception in the context of legal relations involving immovable objects, because in that context the actions have been determined by law, that is, they must be carried out in "light" and "cash".

In the scope of civilization, especially the engagement aspect, the meaning of the transaction will refer to civilization, especially the engagement aspect, the meaning of electronic legal transactions itself will include buying and selling, licenses, insurance, leasing and other agreements born in accordance with the development of trade mechanisms in the community. In the public sphere, the legal relationship will include relations between citizens and the government and relations between fellow members of the community that are not intended for commercial purposes.

Product liability is direct civil liability (strict liability) from the business actor for losses suffered by consumers due to using the products they produce. This responsibility is applied in the event that there is no agreement (no private of contract) between business actors and consumers.

This condition as what happened in United Kingdom, that the entry into force of the product liability provisions of the Consumer Protection Act 1987 has brought about major changes in the non-contractual liability regime in the United Kingdom. The Act, which was introduced pursuant to the requirements of an EC Directive on the Approximation of the laws, regulations and administrative provisions of the member states concerning liability for defective products, serves principally to introduce a system of no fault liability in respect of certain forms of injury and damage.

A producer will incur liability only when a product is defective. To date, there has been almost no litigation concerned directly with the non-contractual liability of software producers or suppliers. It seems unlikely that this can continue. Whilst the requirement that a claimant establish negligence may be a barrier to claims based in negligence, there appears steadily increasing recognition that software is to be regarded as a product and hence will be subject to the product liability regime. Although the limitation to situations where software causes injury or damage to non-commercial property is a significant one, the ever-expanding range of software

applications must make a similar expansion in litigation a not unreasonable prospect.

7. Misrepresentation

Misrepresentation claims that the vendor fraudulently misrepresented the capabilities of the software. In order to prevail under this theory, the plaintiff must show that it was damaged because:

- a. The vendor misrepresented a material fact concerning the software, and
- b. The plaintiff justifiably relied on this misrepresentation

A fraudulent misrepresentation claim is especially threatening to software vendors because under this theory, a plaintiff may sue when it suffers damages solely to its intangible economic interest (such as business reputation), rather than personal injuries or damage to tangible personal property (Rowland & Macdonald 2005).

Section 3 of the Misrepresentation Act 1967 provides that a clause in a contract which purports to exclude or restrict liability for misrepresentation will only be effective if it satisfies the requirement of reasonableness. The burden of proof is on the person seeking to rely on the clause. If a computer salesperson claims that the computer she is selling will run a particular software package and this claim turns out to be untrue, it will be for the company selling the computer to show that any exemption clause it hopes to rely on passes the test of reasonableness. The test is laid out in section 11 of the Unfair Contract Terms Act 1977.

In a subsequent appeal to the House of Lords, the Court of Appeal's decision was affirmed. It should be noted that, by section 7 of the Unfair Contract Terms Act 1977, liability for defective products under Part I of the Consumer Protection Act 1987 cannot be excluded or limited by any contract term.

8. Professional Malpractice

In this variation on the negligence action, the software vendor is characterized as a professional and therefore is held to owe to the plaintiff not merely a duty to act reasonably, but a higher duty to use a professional standard of care, analogous to the duty required of a physician or lawyer. This theory could apply only if the provision of software is characterized as a service, rather than as a sale of product.

9. Strict Liability

If programs are viewed as a product, then strict liability may be applicable and a plaintiff would not need to prove the "absence of due care" element needed in proving negligence cases. Under strict liability claims, consumers only need to prove that there was a defect in the product under the tort law.

Strict liability is an arrangement in which a producer is held liable regardless of how much care the producer took to make the product reliable.

Under a strict liability interpretation, a person who is harmed in some way by a software failure would have the right to obtain damages either from the manufacturer of the software or the institution operating the software when the error occurred. Under current law, strict liability principles are not applicable to doctors and hospital, although strict liability is being applied more frequently these days to manufacturers of medical software.

In a modern society such as ours, where technology rapidly advances and changes and computer software is being used in more innovative situations, the “chilling” effect on technology as a result of imposing strict liability would be too great. Furthermore, the protection afforded under a warranty theory, negligence theory, and computer malpractice theory can adequately protect and compensate the consumer without preventing the innovation that is so desirable in our complex and technologically growing society. If strict liability were imposed upon computer software manufactures, society could lose out on very important and potentially lifesaving computer software (Miyaki 1992).

For strict liability to apply to the manufacturer of software, the user must have used the product in a reasonable fashion and the product must have reached the user without substantial change. If the user is injured while using the product, the user need show only that the product caused the injury, and that the product was sold in a defective or unreasonably dangerous condition. The alleged defect could be a defect in the design or manufacturing of the software, or it could simply be a failure to warn of hazards.

An important feature of the strict liability theory is that it renders legally irrelevant the issue of whether the vendor acted reasonably. By preventing the vendor from presenting exculpatory arguments, this theory in effect forces software manufacturers to guarantee the safety of their products.

The strict liability theory also has an effect on recoverable damages. If it is applied, everyone in the chain of distribution of the product may be liable for the plaintiff's damages. However, users are not generally compensated for economic loss under a strict liability theory, but only for personal injury or property damage.

As product liability and reputation may achieve in isolation the same outcomes in terms of incentives, it is clear, and in fact it seems to be widely shared idea, that they are substitutes as instruments to induce adequate behaviour.

The complementarity between product liability and reputation: product liability reduces the cost of reputational sanctions. This becomes particularly obvious when one considers the range of parameters for which trade between company and consumers can be sustained. Product liability makes it possible that market reputation allows cooperation to happen for a larger set of parameter values than market reputation alone would be able to induce to equilibrium. In other words legal, liability makes reputation more successful in ensuring trade in markets (Ganuza, Gomez, and Robles 2016).

CONCLUSION

THE DOCTRINE of product liability and negligence cannot be applied to malware-embedded software. Based on the analyses above and many theories regarding the product liability and negligence, the doctrine of product liability and negligence cannot be applied to malware-embedded software. Product liability applies to defective products though not software. The law of negligence can apply to defective hardware and software (original software by producers) but it is not easy to apply. There are many terms and conditions and also test to prove the negligence of defective software. Malware-embedded software usually is not original software by producer. If a consumer want to install a software, they should accept the terms and conditions but many consumers did not read the terms and conditions. Because of that it is hard to apply product liability and negligence to producer for malware-embedded software.

REFERENCES

- Anonym, UCLA. "How to Protect Against Malicious Software", *Online*, 2014, <http://www.seas.ucla.edu/security/malware.html>. Accessed: 25 January 2019). **Online**
- Bainbridge, David. *Introduction to Information Technology Law Sixth Edition*. United Kingdom: Pearson Longman, 2008. **Google Scholar Crossref**
- Galasso, Alberto & Luo, Hong. "How Does Product Liability Risk Affect Innovation? Evidence from Medical Implants." *CEPR Discussion Paper No. DP13036*, 2018. **Google Scholar Crossref**
- Ganuzza, Juan José. Gomez, Fernando, Robles, Marta. "Product Liability versus Reputation", *The Journal of Law, Economics, and Organization*, 32 (2), 2016: 213–241. **Google Scholar Crossref**
- Hermana, Budi & Silfianti, Widya. "Evaluating E-government Implementation by Local Government: Digital Divide in Internet Based Public Services in Indonesia", *International Journal of Business and Social Science*, 2011, 2 (3):156-161. **Google Scholar Crossref**
- Irianto, Sulistyowati & Shidarta. *Metode Penelitian Hukum Konstelasi dan Refleksi*, Jakarta: Yayasan Pustaka Obor Indonesia, 2011. **Google Scholar Crossref**
- Lloyd, Ian J. *Information Technology Law*. New York: Oxford University Press, 2011. **Google Scholar Crossref**
- Miyaki, Patrick T. "Computer Software Defects: Should Computer Software Manufacturers be Held Strictly Liable for Computer Defect?", *Computer and High Technology Law Journal* , 1992, 8 (1): 121-144. **Google Scholar Crossref**

- Rowland, Diane. *Liability for Defective Software*. London: *The Cambrian Law Review*, 1991, 78 (22): 73-92. [Google Scholar Crossref](#)
- Rowland, Diane & Macdonald, Elizabeth. *Information Technology Law Third Edition*, Australia: Cavendish Publishing, 2005. [Google Scholar Crossref](#)
- Setiadi, Farisya, Sucahyo, Yudho Giri and Hasibuan, Zainal A. “An Overview of the Development Indonesia National Cyber Security”, *International Journal of Information Technology & Computer Science*, 2012, 6 (1): 106-114. [Google Scholar Crossref](#)
- Sunghyo, Kim. “Crashed Software: Assessing Product Liability for Software Defects in Automated Vehicles.” *Duke Law & Technology Review*, 2018, 16 (1): 300-317. [Google Scholar Crossref](#)
- Zawiyah M. Yusof, et.al. *Teknologi Maklumat dan Alam Siber: Isu Etika dan Undang-undang (Information Technology and Cyber World: Ethical and Legal Issues)*. Kuala Lumpur: Pearson Education, 2016. [Online](#)