Monitoring System for Autonomous Underwater Vehicles with IDEA Algorithm

Nanang Syahroni, Risky Ageng Kharisma, Widya Andi Palupi, Djoko Santoso dan Hari Wahjuningrat Suparno

electrical engineering department
Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jalan Raya ITS Sukolilo, Surabaya 60111
nanang@pens.ac.id, hari@pens.ac.id, widyaandi09@gmail.com

Abstract. Autonomous Underwater Vehicles (AUV) is a type of underwater vehicle that has been widely developed in the world and in Indonesia. This AUV is well known for commercial use and even for military purposes. AUV is equipped with various sensors and other equipment to support activities in making observations under water. The use of these sensors can be used as parameters for actual conditions in underwater observations. In this final project, a smartphone-based application will be created to monitor telemetry data on the AUV and add a security system to the vehicle. The monitoring process can be carried out by the user to determine underwater conditions and is equipped with a data security system while on the network to ensure data security during the delivery process. The working principle of this application is that the user accesses data that has been secured on the database server using the International Data Encryption Algorithm (IDEA) algorithm for the data decryption process. The IDEA algorithm is used because it is the best and newest block-cipher algorithm and is rarely used. Telemetry data will be processed on a smartphone so that users can see or monitor activities underwater and can be used for actual analysis. From the experimental results, it can be average processing time of 0.00065 seconds, that can be concluded that the telemetry data monitoring system using a security system with IDEA algorithm can work with an used to secure and monitor telemetry data on the AUV.

Keywords: AUV, IDEA, Kriptografi, Encryption.

1. Introduction

Autonomous Underwater Vehicles (AUV) have been widely developed in the world and even in Indonesia. This underwater vehicle is well known for military use and even for commercial purposes. Underwater rides are equipped with various sensors and other equipment to support underwater observations. The observed data is in the form of telemetry data, namely telemetry data that shows the parameters of an object (object, space and natural conditions) and the measurement results can be sent to another place through the process of sending data. Thus, the use of sensors in AUV seems to have the potential to accurately solve problems when observing underwater [1].

M. U. H. Al Rasyid and M. R. Mufid (eds.), *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, Advances in Engineering Research 230,
https://doi.org/10.2991/978-94-6463-364-1_25

The monitoring system that is widely used today is to use a PC (Personal Computer) as a control medium and visual display [2]. Systems that are used for commercial purposes must be able to guarantee that the data sent and received is safe because it is confidential for each purpose that uses it. The existing system is not equipped with a security system that ensures telemetry data remains safe even though it passes through a network that allows it to be accessed by users other than those who have access to the system.

In this paper, a smartphone-based application will be created to monitor telemetry data on the AUV and add a security system to the vehicle. The monitoring process can be carried out by the user to find out about underwater conditions and is equipped with data security while on the network to ensure data security. The working principle of this application is that the user accesses data that has been secured on a database server using the IDEA algorithm for the data decryption process.

The IDEA (International Data Ecryption Algorithm) algorithm is used because it is the best and newest block-cipher algorithm and is rarely used [3]. Telemetry data will be processed on a smartphone so that users can see or monitor activities underwater and can be used for actual analysis.

2. System Implementation

2.1. Telemetry Data Monitoring System Design

In the block diagram, an overview of the application system that will be made will be explained. The following is a block diagram of a telemetry data monitoring system at AUV using a security system with the IDEA algorithm.

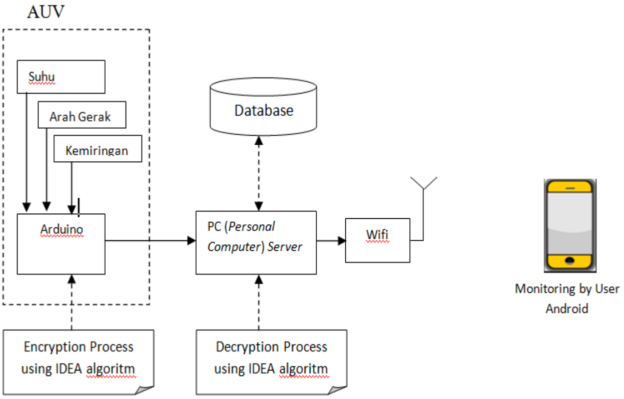

AUV
Suhu
Arah Gerak
Kemiringan
Arduino
Database
PC (Personal Computer) Server
Wifi
Monitoring by User Android
Encryption Process using IDEA algoritm
Decryption Process using IDEA algoritm

Fig. 1. System Block Diagram.

The explanation from Figure 2.1 is that activity information from sensors on the AUV is taken by Arduino and then encrypted with the IDEA method first before being sent to the PC server. On the PC server, encrypted data is stored on the database server.

The encrypted data in the database is returned as initial information through a decryption process by applying the IDEA algorithm which is placed on the PC server. Further information can be displayed on the android application to monitor every activity that is on each sensor underwater after the input key is suitable for the data decryption process.

2.2. Decryption System Design with the IDEA Method

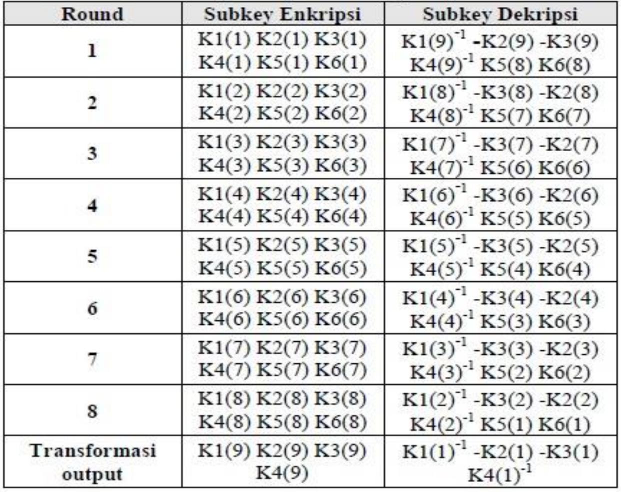
IDEA operates on 64-bit plaintext blocks and the key length is 128 bits. The same algorithm is used for encryption and decryption. Like other encryption algorithms, IDEA uses confusion and diffusion. In contrast to DES which uses permutations and substitutions for confusion and diffusion. IDEA uses the following incompatible algebraic operations:

1. XOR.
2. Addition modulo 216.
3. Multiplication modulo 216 + 1 (this operation replaces the S-box or S-Box).

The IDEA algorithm uses multiplication modulo 216 + 1 with the following considerations, Multiplication by zero always results in zero and has no inversion. Multiplication modulo n also has no inversion if the numbers being multiplied are not prime relative to n. While cryptographic algorithms require operations that have inversions. The number 65537 (216 + 1) is a prime number. Therefore, the multiplication operation modulo (216 + 1) in the IDEA algorithm has an inversion. If we construct a multiplication table for the numbers from 1 to 65536, then each row and column contains each number only once.

In IDEA, for the multiplication operation, a 16-bit number consisting of all zeros is considered the number 65536, while any other number remains the unsigned number it represents. The IDEA algorithm can be divided into three major parts, namely key generation, encryption and decryption algorithms.

Table 1. Decryption Subkey Rules.



Round	Subkey Enkripsi	Subkey Dekripsi

1	K1(1) K2(1) K3(1)
	K4(1) K5(1) K6(1)
		$K1(9)^{-1}$ -K2(9) -K3(9)
		$K4(9)^{-1}$ K5(8) K6(8)

2	K1(2) K2(2) K3(2)
	K4(2) K5(2) K6(2)
		$K1(8)^{-1}$ -K3(8) -K2(8)
		$K4(8)^{-1}$ K5(7) K6(7)

3	K1(3) K2(3) K3(3)
	K4(3) K5(3) K6(3)
		$K1(7)^{-1}$ -K3(7) -K2(7)
		$K4(7)^{-1}$ K5(6) K6(6)

4	K1(4) K2(4) K3(4)
	K4(4) K5(4) K6(4)
		$K1(6)^{-1}$ -K3(6) -K2(6)
		$K4(6)^{-1}$ K5(5) K6(5)

5	K1(5) K2(5) K3(5)
	K4(5) K5(5) K6(5)
		$K1(5)^{-1}$ -K3(5) -K2(5)
		$K4(5)^{-1}$ K5(4) K6(4)

6	K1(6) K2(6) K3(6)
	K4(6) K5(6) K6(6)
		$K1(4)^{-1}$ -K3(4) -K2(4)
		$K4(4)^{-1}$ K5(3) K6(3)

7	K1(7) K2(7) K3(7)
	K4(7) K5(7) K6(7)
		$K1(3)^{-1}$ -K3(3) -K2(3)
		$K4(3)^{-1}$ K5(2) K6(2)

8	K1(8) K2(8) K3(8)
	K4(8) K5(8) K6(8)
		$K1(2)^{-1}$ -K3(2) -K2(2)
		$K4(2)^{-1}$ K5(1) K6(1)

Transformasi output	K1(9) K2(9) K3(9)
	K4(9)
		$K1(1)^{-1}$ -K2(1) -K3(1)
		$K4(1)^{-1}$

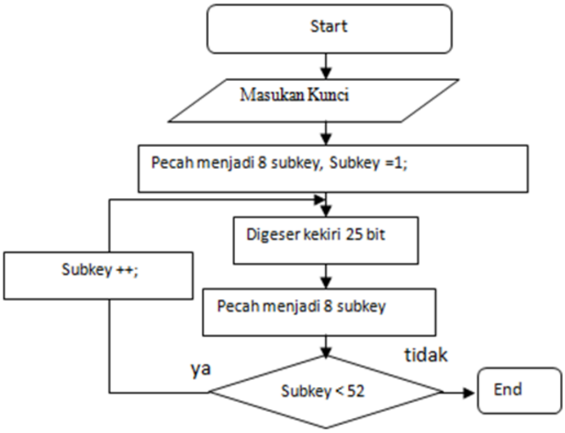


Fig. 2. IDEA Decryption Key Formation Flowchart.

Decryption Key Generation

The algorithm used is like the encryption process but 52 subkeys are derived from 52 encryption subkeys. On each decryption subkey is one of the addition or multiplication inverses that corresponds to the encryption subkey. Table 1 describes the decryption key used from deriving the encryption key.

KD-1 in the table is the inverse multiplication modulo (216 + 1) of KE, where KE.KD-1 = 1. Meanwhile –KD is the inverse of addition modulo 216 of KE, where KE.KD -1=0. The following is a flowchart for the formation of the decryption key.

IDEA Decryption System

The decryption process is exactly the same as the encryption process. The difference lies only in the rules of the subkey. The order of the subkeys is reversed by the encryption process and the subkeys are inverted. The subkey in the output transformation step in the encryption process is inverted and used as a subkey in round 1 of the decryption process. Subkeys in round 8 are inverted and used as subkeys in rounds 1 and 2 of the decryption process. And so on as shown in figure 3 below.

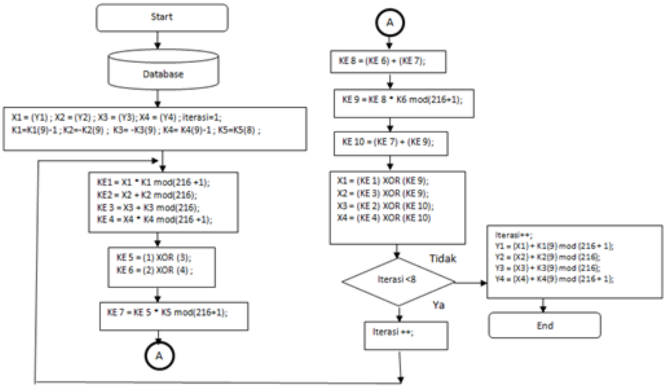

Start
Database
X1 = (Y1) ; X2 = (Y2) ; X3 = (Y3); X4 = (Y4) ; iterasi=1;
K1=K1(9)-1 ; K2=-K2(9) ; K3= -K3(9) ; K4= K4(9)-1 ; K5=K5(8) ;
KE1 = X1 * K1 mod(216 +1);
KE2 = X2 + K2 mod(216);
KE 3 = X3 + K3 mod(216);
KE 4 = X4 * K4 mod(216 +1);
KE 5 = (1) XOR (3);
KE 6 = (2) XOR (4) ;
KE 7 = KE 5 * K5 mod(216+1);
A
A
KE 8 = (KE 6) + (KE 7);
KE 9 = KE 8 * K6 mod(216+1);
KE 10 = (KE 7) + (KE 9);
X1 = (KE 1) XOR (KE 9);
X2 = (KE 3) XOR (KE 9);
X3 = (KE 2) XOR (KE 10);
X4 = (KE 4) XOR (KE 10)
Iterasi <8
Tidak
Ya
Iterasi ++;
Iterasi++;
Y1 = (X1) + K1(9) mod (216 + 1);
Y2 = (X2) + K2(9) mod (216);
Y3 = (X3) + K3(9) mod (216);
Y4 = (X4) + K4(9) mod (216 + 1);
End

Fig. 3. Decryption System Flowchart.

The following is the sequence of the decryption system from plaintext to plaintext in the IDEA algorithm:

1. Multiply X1 by K1 mod (216 + 1).
2. Add X2 with K2 mod 216.
3. Add X3 with K3 mod 216.
4. Multiply X4 by K4 mod (216 + 1).
5. XOR the results from steps 1 and 3.
6. XOR the results from steps 2 and 4.
7. Multiply the result from step 5 by the K5 mod (216 + 1).
8. Add the results from step 6 and 7 mod 216.
9. Multiply the result from step 8 by the K6 mod (216 + 1).
10. Add the results from step 7 and 9.
11. XOR the results from steps 1 and 9.
12. XOR the results from step 3 and 9.
13. XOR the results from step 2 and 10.
14. XOR the results from step 4 and 10.

The decryption process uses the same algorithm as the encryption process but the 52 key subblocks used are derived from 52 encryption key subblocks. In this case we will take the inverse of the addition operation by mod 216 and the multiplication mod 216 +1, depending on the operation made in the encryption phase. Each decryption subkey is one of the corresponding addition or multiplication inverses to the encryption subkey.

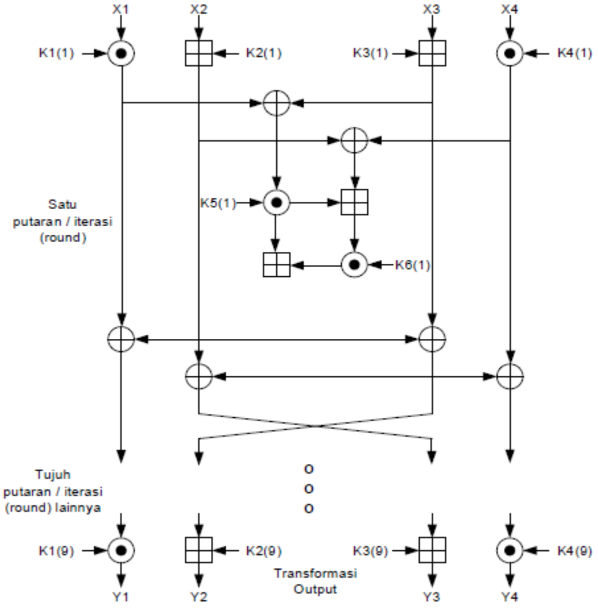


Fig. 4. IDEA Decryption Diagram.

In this paper, the data taken is in the form of telemetry data obtained from hardware prototypes in the form of temperature, tilt, and direction sensors which are processed using Arduino. The data obtained from each sensor is processed by Arduino and then encrypted before entering the database server. The data is then sent by Arduino with the addition of the character "A" as a marker of the beginning of data transmission, "B" as a marker of the end of data transmission and the character "#" as a separator as shown in figure 5.

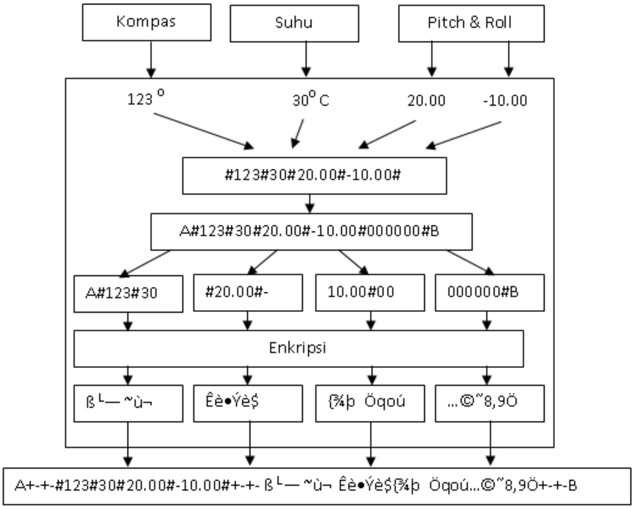
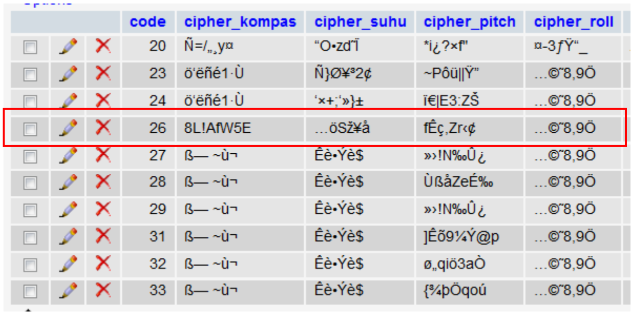


Fig. 5. Encryption Process Diagram.

The data will then be encrypted using the IDEA algorithm after being divided into 4 sub-blocks of 8 characters each. The encryption process is carried out on the four sub-blocks and then sent via the USB port on the database server. Figure 2.6 shows the format of data stored on the database server. Each ciphertext is stored in four different fields.



			code	cipher_kompas	cipher_suhu	cipher_pitch	cipher_roll
□	✎	✕	20	Ñ=/„¸y¤	"O•zd˜Ï	*i¿?×f"	¤-3ƒŸ"_
□	✎	✕	23	ö'ëñé1·Ù	Ñ}Ø¥ª2¢	~Pôü||Ÿ"	…©˜8,9Ö
□	✎	✕	24	ö'ëñé1·Ù	'×+;'»}±	ï€|E3:ZŠ	…©˜8,9Ö
□	✎	✕	26	8L!AfW5E	…öSž¥å	fÊç,Zr‹¢	…©˜8,9Ö
□	✎	✕	27	ß— ~ù¬	Êè•Ýè$	»›!N‰Û¿	…©˜8,9Ö
□	✎	✕	28	ß— ~ù¬	Êè•Ýè$	ÙßåZeÉ‰	…©˜8,9Ö
□	✎	✕	29	ß— ~ù¬	Êè•Ýè$	»›!N‰Û¿	…©˜8,9Ö
□	✎	✕	31	ß— ~ù¬	Êè•Ýè$	]Êõ9¼Ý@p	…©˜8,9Ö
□	✎	✕	32	ß— ~ù¬	Êè•Ýè$	ø„qiö3aÒ	…©˜8,9Ö
□	✎	✕	33	ß— ~ù¬	Êè•Ýè$	{¾þÖqoú	…©˜8,9Ö

Fig. 6. Telemetry Data on the Database Server.

The telemetry data to be processed is obtained from the database server which is stored in the form of ciphertext. Telemetry data processing is carried out according to the process in figure 7 below.

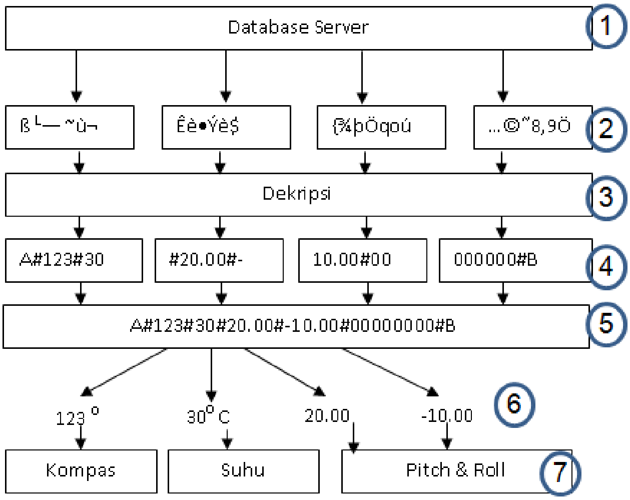


Fig. 7. Decryption Process Diagram.

From the diagram above it can be seen that the decryption process is divided into 7 stages. The key used in this diagram is "cryptography-idea" with the following explanation:

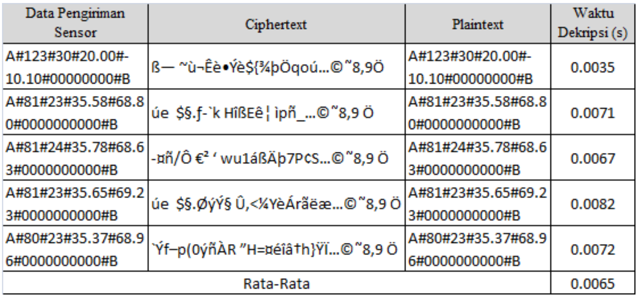
1. The data to be processed comes from the database server in the form of ciphertext from the results of the previous encryption process data transmission.
2. Ciphertext results of the encryption process are stored into 4 parts.
3. The decryption process is carried out in each section.

4. The results of the decryption process are obtained from each part. Each part contains 8 characters of plaintext.
5. Plaintext combined into one part.
6. Telemetry data information is obtained by separating data between characters A and B and each data is delimited by the character "#".
7. The results of compass, temperature, pitch, and roll information are displayed on the Android platform.

The above process is carried out repeatedly with a duration of 2 seconds to update data retrieval from the server side and display it on the monitor system on Android.

3. Results And Discussion

Plaintext in IDEA will be converted to 64 bit binary or to 8 ciphertext characters. The average time that can be used for the encryption process is 0.0208 seconds while the time needed for the decryption process is 0.0065 seconds.



Data Pengiriman Sensor	Ciphertext	Plaintext	Waktu Dekripsi (s)
A#123#30#20.00#-10.10#00000000#B	ß— ˜ù¬Êè•Ýè${¾þÖqoú…©˜8,9Ö	A#123#30#20.00#-10.10#00000000#B	0.0035
A#81#23#35.58#68.80#0000000000#B	úe $§.ƒ-`k HîßEê¦ ìpñ_…©˜8,9 Ö	A#81#23#35.58#68.80#0000000000#B	0.0071
A#81#24#35.78#68.63#0000000000#B	-¤ñ/Ô €² ‘ wu1áßÄþ7P¢S…©˜8,9 Ö	A#81#24#35.78#68.63#0000000000#B	0.0067
A#81#23#35.65#69.23#0000000000#B	úe $§.ØýÝ§ Û,<¼YèÁrãëæ…©˜8,9 Ö	A#81#23#35.65#69.23#0000000000#B	0.0082
A#80#23#35.37#68.96#0000000000#B	`Ýf–p(0ýñÀR ”H=¤éîâ†h}ŸÏ…©˜8,9 Ö	A#80#23#35.37#68.96#0000000000#B	0.0072
Rata-Rata			0.0065

Fig. 8. Decryption Process Test Results.

From the data obtained, IDEA's decryption system is very capable of being implemented on Android smartphones in securing a data transmission, with ciphertext data in the form of Unicode characters, namely characters with the Ascii range 0 - 255 found on the server can be decrypted into plaintext multiples of 8 or 32 characters. The decryption process will be successful when the decryption key matches or matches the key during the encryption process.

In future research it is hoped that the system can be applied to AUV to monitor underwater conditions so that the observed underwater activity data is not easily intercepted by irresponsible parties and can be added to a security system with a more complex algorithm method than the previous method.

4. Conclusion

From the test results and analysis of experimental data, it can be concluded that the telemetry data monitoring system using a security system with the IDEA algorithm can work with an average processing time of 0.00065 seconds. With a value of 0.0065, this monitoring system can be used to secure and monitor telemetry data on the AUV.

References

[1] David Bingham: The Application of Autonomous Underwater Vehicle (AUV) Technology in the oil Industry, International Journal FIG XXII International Congress, (2002).

[2] Wanbin Wang, Peter W. Tse, and Jay Lee.: Remote machine maintenance system through Internet and mobile communication, International Journal Advanced Manufactured Technology, (2007).

[3] Kolidya Yuli.: Implementasi Metode Kriptografi IDEA pada Priority Dealer untuk Layanan Pemesanan dan Laporan Penjualan Handphone Berbasis Web, Politeknik Elektronika Negeri Surabaya, Surabaya, (2012).

[4] Evalin Marta Damayanti Sihombing.: Pembangunan aplikasi sistem informasi dosen politeknik telkom pada smartphone berbasis android, Politeknik Telkom Bandung, Bandung, (2011).

[5] Becik Gati A.: Enkripsi SMS (Short Message Service) pada Telepon Selular Berbasis Android , Politeknik Elektronika Negeri Surabaya, Surabaya, (2014).

[6] Bruce Schneir.: Applied Cryptography : Protocols, Algorithms and source code in C, 2nd edition edition, John Wiley & Sons Publishers, Oak Park, pp. 21-22, (1996).

[7] Ariyus, Dony.: Pengantar Ilmu Kriptografi. Yogyakarta: Andi Offset. (2008).

[8] Wisnu Tanaya K.: Aplikasi Monitoring Pekerjaan Event Organizer Berbasis Smartphone, Politeknik Elektronika Negeri Surabaya, Surabaya, (2014).

[9] Dicky Abdillah.: Rancang Bangun Sistem Monitoring Untuk AUV (Autonomous Underwater Vehicle), Politeknik Elektronika Negeri Surabaya, Surabaya, (2015).

[10] Metha Puspa.: Rancang Bangun Akuisisi Data Telemetri Pada Wahana Tanpa Awak Bawah Air (Autonomous Underwater Vehicle) Berbasis Android, Politeknik Elektronika Negeri Surabaya, Surabaya, (2015).