e-ISSN: 2798-9593
p-ISSN: 2798-9836

LOFIAN

Jurnal Teknologi Informasi dan Komunikasi

Volume 1, Nomor 1, September 2021

Program Studi Teknik Infromatika
AMIK Medan Business Polytechnic
Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia – 20155
https://ejournal.amikmbp.ac.id/index.php/lofian/

e-ISSN: 2798-9593
p-ISSN: 2798-9836

LOFIAN:
Jurna**l** Tekn**o**logi In**f**orma**s**i d**a**n Komu**n**ikasi
Volume 1, Nomor 1, September 2021

TIM PENGELOLA

ALAMAT REDAKSI
AMIK Medan Business Polytechnic (MBP)
Jalan Jamin Ginting No. 285-287, Padang Bulan, Medan Baru,
Kota Medan, Sumatera Utara, Indonesia - 20155
Email: lofian@amikmbp.ac.id

e-ISSN: 2798-9593
p-ISSN: 2798-9836

LOFIAN
Jurna**l** Tekn**o**logi In**f**orma**s**i d**a**n Komu**n**ikasi
Volume 1, Nomor 1, September 2021

Daftar Isi

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 1-6
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

Perancangan Simulator Interlock Protocol saat Serangan Man-in-the-Middle-Attack pada Kriptografi Kunci Publik RSA

Maranata Pasaribu[1], Marice Hotnauli Simbolon[2]

[1,2]*AMIK Medan Business Polytechnic*
Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia - 20155

[1]*maranata@amikmbp.ac.id*, [2]*simbolonice@gmail.com*

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

Abstrak

Dalam proses transmisi data, walaupun data telah dienkripsi, namun terdapat kemungkinan bahwa data tersebut dapat dimiliki oleh orang lain. Salah satu kemungkinan tersebut adalah dengan terjadinya penyadapan media komunikasi yang digunakan oleh kedua orang yang sedang berkomunikasi tersebut. Problema man-in-the-middle-attack ini dapat dicegah dengan menggunakan interlock protocol. Algoritma inti dari protokol ini yaitu mengirimkan 2 bagian pesan terenkripsi. Bagian pertama dapat berupa hasil dari fungsi hash satu arah (one way hash function) dari pesan tersebut dan bagian kedua berupa pesan terenkripsi itu sendiri. Penelitian ini akan merancang perangkat lunak simulasi yang dapat menjelaskan proses kerja dari man-in-the-middle-attack dalam menyadap dan mengubah pesan, menjelaskan proses kerja interlock protocol untuk mengatasi problema Man-In-The-Middle-Attack. Setelah menyelesaikan perangkat lunak simulasi, peneliti menarik kesimpulan: Dengan menggunakan interlock protocol, walaupun kunci publik pihak penerima dan pengirim didapatkan dan diganti oleh penyadap, tetapi penyadap tidak dapat menjalankan prosedur man-in-the-middle-attack untuk melihat dan mengubah pesan. Hal ini dikarenakan pesan terenkripsi terbagi menjadi dua bagian pada variasi pertama dan terdapat fungsi hash untuk memverifikasi keaslian pesan pada variasi kedua.

Kata Kunci: Kriptografi, Rivest-Shamir-Adleman, Protokol Interlock, Enkripsi, Dekripsi.

1. Pendahuluan

Dalam proses transmisi data, walaupun data telah dienkripsi, namun terdapat kemungkinan bahwa data tersebut dapat dimiliki oleh orang lain. Salah satu kemungkinan tersebut adalah dengan terjadinya penyadapan media komunikasi yang digunakan oleh kedua orang yang sedang berkomunikasi tersebut. Hal inilah yang disebut dengan man-in-the-middle-attack. Dalam keadaan ini, orang yang menyadap berada di antara kedua orang yang sedang berkomunikasi. Data-data yang dikirimkan oleh orang yang sedang berkomunikasi satu sama lain selalu melalui orang yang menyadap tersebut, sehingga orang yang menyadap tersebut dapat mengetahui semua informasi yang dikirimkan satu sama lain. Keadaan ini muncul karena kedua orang yang sedang berkomunikasi tersebut tidak dapat mem-verifikasi status dari orang yang berkomunikasi dengannya tersebut, dengan mengambil asumsi bahwa proses penyadapan tersebut tidak menyebabkan gangguan dalam jaringan.

Problema man-in-the-middle-attack ini dapat dicegah dengan menggunakan interlock protocol. Algoritma inti dari protokol ini yaitu mengirimkan 2 bagian pesan terenkripsi. Bagian pertama dapat berupa hasil dari fungsi hash satu arah (one way hash function) dari pesan tersebut dan bagian kedua berupa pesan terenkripsi itu sendiri. Hal ini menyebabkan orang yang menyadap tersebut tidak dapat mendekripsi pesan pertama dengan menggunakan kunci privatnya. Ia hanya dapat membuat sebuah pesan baru dan mengirimkannya kepada orang yang akan menerima pesan tersebut.

Maka dari pemapanan di atas, peneliti akan merancang perangkat lunak simulasi yang dapat menjelaskan proses kerja dari man-in-the-middle-attack dalam menyadap dan mengubah pesan, menjelaskan proses kerja interlock protocol untuk mengatasi problema Man-In-The-Middle-Attack, menampilkan algoritma dari sistem kriptografi kunci publik metode RSA. Dengan harapan perangkat lunak simulasi yang dibangun dapat digunakan sebagai fasilitas pendukung dalam proses belajar mengajar terutama untuk mata kuliah Kriptografi.

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi e-ISSN: 2798-9593
Vol. 1, No. 1, September 2021, Halaman 1-6 p-ISSN: 2798-9836
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

2. Landasan Teori

Dalam penelitian ini, peneliti akan memaparkan beberapa landasan teori yang digunakan dalam penelitian ini, antara lain:

2.1. Kriptografi

Kriptografi (Cryptography) berasal dari bahasa Yunani yaitu dari kata 'crypto' dan 'graphia' yang berarti penulisan rahasia. Kriptografi adalah suatu ilmu yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut cryptology. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah.

Menurut Stalling, ada beberapa tuntutan yang terkait dengan isu keamanan data, yaitu:

1. *Confidentiality*. Menjamin bahwa data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja.
2. *Authentication*. Baik pada saat mengirim atau menerima informasi, kedua belah pihak perlu mengetahui bahwa pengirim dari pesan tersebut adalah orang yang sebenarnya seperti yang diklaim.
3. *Integrity*. Tuntutan ini berhubungan dengan jaminan setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya, dan ditambahkan.
4. *Nonrepudiation*. Mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan/informasi. Jika sebuah pesan dikirim, penerima dapat membuktikan bahwa pesan tersebut memang dikirim oleh pengirim yang tertera. Sebaliknya, jika sebuah pesan diterima, pengirim dapat membuktikan bahwa pesannya telah diterima oleh pihak yang ditujunya.
5. *Access Control*. Membatasi sumber-sumber data hanya kepada orang-orang tertentu.
6. *Availability*. Jika diperlukan setiap saat semua informasi pada sistem komputer harus tersedia bagi semua pihak yang berhak atas informasi tersebut.

Dari keenam aspek keamanan data tersebut, empat diantaranya dapat diatasi dengan menggunakan cryptography yaitu confidentiality, integrity, authentication, dan nonrepudiation.

2.2. Algoritma Rivest-Shamir-Adleman (RSA)

RSA merupakan salah satu teknik enkripsi dan dekripsi dengan menggunakan dua buah kunci. Kunci-kunci tersebut diperoleh dari hasil perhitungan eksponensial, perkalian, pembagian, penjumlahan dan pengurangan. Perhitungan dilakukan terhadap dua buah bilangan prima.

Walaupun RSA cenderung aman bukan berarti tidak bisa dilakukan "attack" terhadap enkripsinya. Didukung perkembangan hardware komputer yang semakin cepat maka semakin terbuka kemungkinan memecahkan enkripsi RSA. Pada tahun 1977 Rivest, Shamir dan Adleman mempublikasikan tantangan memecahkan enkripsi RSA yang memakai 129 digit bilangan bulat. Tantangan ini diharapkan bisa bertahan dari "attack" untuk waktu yang lama. Tetapi pada tahun 1994 tantangan ini dipecahkan dengan menggunakan komputer yang kekuatan komputasinya berimbang dengan komputer untuk membuat film animasi "Toy Story" (kumpulan, 87 unit komputer dual prosesor, 30 unit komputer empat procesor, 100Mhz SPARCstation).

Secara garis besar, algoritma kunci publik RSA dapat dijabarkan sebagai berikut:

```
Key generation:

1. Hasilkan dua buah integer prima besar, p dan q.
   Untuk memperoleh tingkat keamanan yang tinggi
   pilih p  dan q yang berukuran  besar, misalnya
   1024 bit.
2. Hitung m = (p-1)*(q-1).
3. Hitung n = p*q
4. Pilih e yg relatif prima terhadap m.
   e relatif prima thd m artinya faktor pembagi
   terbesar keduanya adalah 1, secara matematis
   disebut gcd(e,m) = 1. Untuk mencarinya dapat
   digunakan algoritma Euclid.
5. Cari d, sehingga e*d = 1 mod (m), atau
   d = e^-1 * mod (m). Untuk mencarinya, dapat
   digunakan algoritma extended Euclid.
6. Kunci publik : e, n
   Kunci private : d, n
```

Gbr. 1. Key Generation

```
Public key encryption & decryption


B mengenkripsi message M untuk A

Yg harus dilakukan B :
  1. Ambil kunci publik A yg otentik (n, e)
  2. Representasikan message sbg integer M dalam
     interval [0,n-1]
  3. Hitung C = M ^ e (mod n)
  4. Kirim C ke A

Untuk mendekripsi, A melakukan :
  Gunakan kunci pribadi d untuk menghasilkan M =
  C ^ d  (mod n)
```

Gbr. 2. Public Key Encrytion & Description

2.3. Interlock Protocol

Problema man-in-the-middle-attack dapat atasi dengan menggunakan interlock protocol. Interlock

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 1-6
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

protocol ini diciptakan oleh Ron Rivest dan Adi Shamir. Algoritma inti dari protokol ini yaitu protokol ini mengirimkan 2 bagian pesan terenkripsi.

Bagian pertama dapat berupa hasil dari fungsi hash satu arah (one way hash function) dari pesan tersebut dan bagian kedua berupa pesan terenkripsi itu sendiri.

Hal ini menyebabkan orang yang menyadap tersebut tidak dapat mendekripsi pesan pertama dengan menggunakan kunci privatnya. Ia hanya dapat membuat sebuah pesan baru dan mengirimkannya kepada orang yang akan menerima pesan tersebut.

3. Metodologi Penelitian

Adapun lokasi penelitian ini berada transmisi data yang di enkripsi antar pengirim dan penerima serta juga di pihak ketiga yang melakukan penyadapan.

3.1. Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi lapangan

   Degan metode ini peneliti mengamati bagaimana data ditransmisikan dari pengirim dan penerima serta penyadap.
2. Studi Kepustakaan

   Dengan melakukan studi pustaka, peneliti mendapatkan data-data yang bersifat teori ilmiah yang dipergunakan sebagai dasar dalam melakukan penulisan dan analisa terhadap kendala-kendala yang ada sehingga kendala tersebut dapat diselesaikan dengan baik.

3.2. Langkah dalam pembuatan perangkat lunak

Terdapat serangkaian langkah-langkah yang dilakukan secara terencana dan sistematis guna mendapatkan penecahan masalah atau menjawab pertanyaan-pertanyaan dari penelitian.

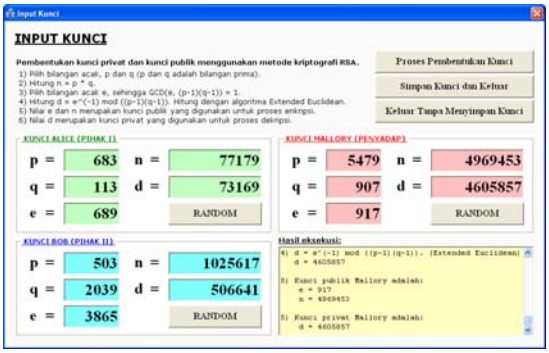
1. Membaca dan mempelajari buku-buku Kriptogafi terutama yang berhubungan dengan man-in-the-middle-attack ini dan interlock protocol.
2. Mempelajari teknik-teknik dasar pemrograman.
3. Mempelajari proses kerja dari problema man-in-the-middle-attack ini dan proses pencegahannya dengan menggunakan interlock protocol.
4. Mempelajari proses kerja dari algoritma RSA dan fungsi SHA-1 serta algoritma-algoritma pendukung yang digunakan.
5. Merancang algoritma dari RSA dan fungsi SHA-1 serta algoritma-algoritma pendukung lainnya.
6. Merancang interface dari perangkat lunak simulasi.
7. Merancang perangkat lunak simulasi yang mampu menjelaskan prosedur kerja dari problema man-in-the-middle-attack dan solusi untuk mengatasinya dengan menggunakan interlock protocol.
8. Menguji perangkat lunak dan memperbaiki kesalahan yang timbul.

4. Hasil dan Pembahasan

Penelitian ini selanjutnya mendapatkan hasil dan akan dibahas sebagai berikut:

4.1. Hasil

Misalkan, diambil contoh input kunci seperti terlihat pada Gbr. 3. berikut.



Gbr. 3. Contoh input kunci

Hasil eksekusi proses pembentukan kunci adalah sebagai berikut:

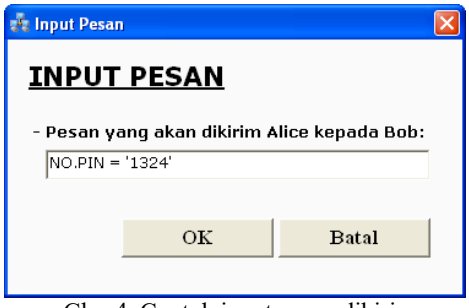
1. Perhitungan Kunci Alice
   - p = 683, q = 113
   - n = p * q
     n = 683 * 113
     n = 77179
   - e = 689, GCD(e,(p-1)(q-1)) = 1.
   - d = e^(-1) mod ((p-1)(q-1)). (Extended Euclidean)
     d = 73169
   - Kunci publik Alice adalah:
     e = 689
     n = 77179
   - Kunci privat Alice adalah:
     d = 73169
2. Perhitungan Kunci Bob

e-ISSN: 2798-9593
p-ISSN: 2798-9836

- p = 503, q = 2039
- n = p * q
  n = 503 * 2039
  n = 1025617
- e = 3865, GCD(e,(p-1)(q-1)) = 1.
- d = e^(-1) mod ((p-1)(q-1)). (Extended Euclidean)
  d = 506641
- Kunci publik Bob adalah:
  e = 3865
  n = 1025617
- Kunci privat Bob adalah:
  d = 506641

3. Perhitungan Kunci Mallory
   - p = 5479, q = 907
   - n = p * q
     n = 5479 * 907
     n = 4969453
   - e = 917, GCD(e,(p-1)(q-1)) = 1.
   - d = e^(-1) mod ((p-1)(q-1)). (Extended Euclidean)
     d = 4605857
   - Kunci publik Mallory adalah:
     e = 917
     n = 4969453
   - Kunci privat Mallory adalah:
     d = 4605857

Misalkan, Alice mengirimkan pesan kepada Bob dan Mallory mengubah pesan Alice. Contoh input pesan terlihat pada Gbr. 4 dan 5. berikut.



Gbr. 4. Contoh input pesan dikirim

Gbr 5. Contoh input pesan (ubah) dikirim

Tampilan proses simulasi terlihat pada Gbr. 6. berikut.

Gbr. 6. Contoh tampilan Form Utama

Hasil eksekusi proses man-in-the-middle-attack adalah sebagai berikut:

1. ALICE mengirimkan pesan kepada BOB dan dienkripsi dengan menggunakan kunci publik "BOB"

Pesan = 'NO.PIN = '1324"

Ubah pesan menjadi biner:
01001110010011110010111001010000010010010100111000100000001111010010000000010011100110001001100100111001100100011010000100111
Ubah setiap 3 bit biner menjadi bentuk desimal:
234474562404451610036440116304631443204 7

Masukkan setiap 4 digit desimal (m) ke fungsi enkripsi: c = (m ^ e) mod n

(Gunakan kunci publik Mallory)
c = (2344 ^ 917) mod 4969453 = 2401144
c = (7456 ^ 917) mod 4969453 = 1679420
c = (2404 ^ 917) mod 4969453 = 2967120
c = (4516 ^ 917) mod 4969453 = 2054777

Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

c = (1003 ^ 917) mod 4969453 = 4491255
c = (6440 ^ 917) mod 4969453 = 4898261
c = (1163 ^ 917) mod 4969453 = 166858
c = (0463 ^ 917) mod 4969453 = 3875803
c = (1443 ^ 917) mod 4969453 = 4764414
c = (2047 ^ 917) mod 4969453 = 2067413

Hasil enkripsi:
2401144 1679420 2967120 2054777 4491255 4898261 166858 3875803 4764414 2067413

2. Mallory mendekripsi pesan ALICE dengan menggunakan kunci publiknya

Cipher Text = '2401144 1679420 2967120 2054777 4491255 4898261 166858 3875803 4764414 2067413'

Masukkan cipher text (c) ke fungsi dekripsi: m = (c ^ d) mod n

(Gunakan kunci privat Mallory)
m = (2401144 ^ 4605857) mod 4969453 = 2344
m = (1679420 ^ 4605857) mod 4969453 = 7456
m = (2967120 ^ 4605857) mod 4969453 = 2404
m = (2054777 ^ 4605857) mod 4969453 = 4516
m = (4491255 ^ 4605857) mod 4969453 = 1003
m = (4898261 ^ 4605857) mod 4969453 = 6440
m = (166858 ^ 4605857) mod 4969453 = 1163
m = (3875803 ^ 4605857) mod 4969453 = 0463
m = (4764414 ^ 4605857) mod 4969453 = 1443
m = (2067413 ^ 4605857) mod 4969453 = 2047

Hasil dekripsi:
23447456240445161003644011630463144320 47

Ubah pesan menjadi biner:
01001110010011110010111001010000010010010100111100001000000011110100100000000100111001100010011001100110010001101000010 0111

Ubah setiap 8 bit biner ke bentuk ascii:
NO.PIN = '1324'

3. Mallory mengubah pesan dan mengenkripsi pesan samaran dengan menggunakan kunci publik BOB

Pesan = 'NO.PIN = '9999"

Ubah pesan menjadi biner:
01001110010011110010111001010000010010010100111100001000000011110100100000000100111001110010011100100111001001001111

Ubah setiap 3 bit biner menjadi bentuk desimal:
23447456240445161003644011634471162344 47

Masukkan setiap 4 digit desimal (m) ke fungsi enkripsi: c = (m ^ e) mod n

(Gunakan kunci publik Mallory)
c = (2344 ^ 3865) mod 1025617 = 814513
c = (7456 ^ 3865) mod 1025617 = 615339
c = (2404 ^ 3865) mod 1025617 = 566072
c = (4516 ^ 3865) mod 1025617 = 783295
c = (1003 ^ 3865) mod 1025617 = 750546
c = (6440 ^ 3865) mod 1025617 = 879723
c = (1163 ^ 3865) mod 1025617 = 457933
c = (4471 ^ 3865) mod 1025617 = 359231
c = (1623 ^ 3865) mod 1025617 = 93145
c = (4447 ^ 3865) mod 1025617 = 1020355

Hasil enkripsi:
814513 615339 566072 783295 750546 879723 457933 359231 93145 1020355

4. BOB mendekripsi pesan dengan menggunakan kunci privatnya

Cipher Text = '814513 615339 566072 783295 750546 879723 457933 359231 93145 1020355'

Masukkan cipher text (c) ke fungsi dekripsi: m = (c ^ d) mod n

(Gunakan kunci privat Mallory)
m = (814513 ^ 506641) mod 1025617 = 2344
m = (615339 ^ 506641) mod 1025617 = 7456
m = (566072 ^ 506641) mod 1025617 = 2404
m = (783295 ^ 506641) mod 1025617 = 4516
m = (750546 ^ 506641) mod 1025617 = 1003
m = (879723 ^ 506641) mod 1025617 = 6440
m = (457933 ^ 506641) mod 1025617 = 1163
m = (359231 ^ 506641) mod 1025617 = 4471
m = (93145 ^ 506641) mod 1025617 = 1623
m = (1020355 ^ 506641) mod 1025617 = 4447

Hasil dekripsi:
23447456240445161003644011634471162344 47

Ubah pesan menjadi biner:
01001110010011110010111001010000010010010100111100001000000011110100100000000100111001110010011100100111001001001111

Ubah setiap 8 bit biner ke bentuk ascii:
NO.PIN = '9999'

(PROSEDUR PENYADAPAN DAN PENYAMARAN PESAN OLEH MALLORY BERHASIL)

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi e-ISSN: 2798-9593
Vol. 1, No. 1, September 2021, Halaman 1-6 p-ISSN: 2798-9836
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

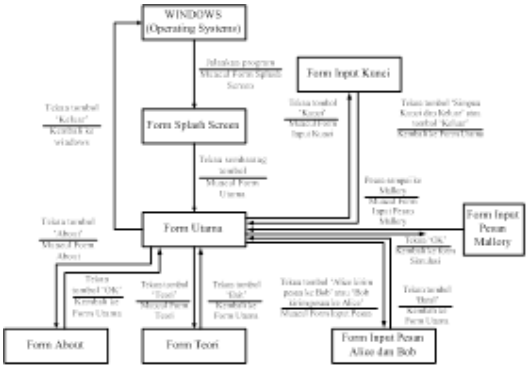
4.2. Pembahasan

Pada bagian ini, akan dibahas mengenai bagaimana alur kerja perangkat lunak dan proses-proses yang terjadi. Masing-masing pembahasan akan dibahas dalam sub bab berikut ini.

1. Alur Kerja

Di asumsikan terdapat dua pihak yang sedang berkomunikasi (Alice dan Bob) dan satu pihak sebagai penyadap (Mallory) yang berada di tengah-tengah saluran komunikasi. Sebelum proses simulasi dijalankan, user harus meng-input kunci privat dan publik Alice, Bob dan Mallory pada form Input Kunci. Input kunci juga dapat dihasilkan secara acak oleh komputer menggunakan random number generator. Setelah itu, ditampilkan proses simulasi penyadapan dan penukaran kunci yang dilakukan oleh Mallory terhadap Bob dan Alice.

Selanjutnya, user dapat meng-input sendiri pesan yang akan dikirim Bob kepada Alice atau sebaliknya dan pesan samaran yang dibuat oleh Mallory. Dalam proses simulasi ini, user juga dapat memilih untuk menggunakan opsi untuk menggunakan interlock protocol atau tidak. Mengatasi masalah man-in-the-middle-attack dengan interlock protocol ini terbagi lagi menjadi dua cara, yaitu: pisahkan hasil enkripsi menjadi dua bagian atau gunakan fungsi hash sebagai bagian pertama dan pesan terenkripsi menjadi bagian kedua. Kedua cara ini akan mengatasi tindakan penyamaran pesan oleh Mallory. Alur kerja perangkat lunak dapat digambarkan dalam bentuk state transition diagram (STD), seperti terlihat pada Gbr. 7.



Gbr. 7. State Transition Diagram (STD) Perangkat Lunak

5. Kesimpulan dan Saran

Dari hasil penelitian dan pembahasannya, maka dapat disimpulkan dan saran.

5.1. Kesimpulan

Setelah menyelesaikan perangkat lunak simulasi pemanfaatan metode interlock protocol untuk mengatasi man-in-the-middle-attack, peneliti menarik kesimpulan sebagai berikut:

1. Perangkat lunak mensimulasikan proses kerja man-in-the-middle-attack sebagai salah satu bentuk penyerangan terhadap metode kriptografi publik dan proses kerja interlock protocol untuk mengatasinya, sehingga perangkat lunak dapat digunakan untuk mendukung proses belajar mengajar, terutama dalam mata kuliah Kriptografi.
2. Dengan menggunakan interlock protocol, walaupun kunci publik pihak penerima dan pengirim didapatkan dan diganti oleh penyadap, tetapi penyadap tidak dapat menjalankan prosedur man-in-the-middle-attack untuk melihat dan mengubah pesan. Hal ini dikarenakan pesan terenkripsi terbagi menjadi dua bagian pada variasi pertama dan terdapat fungsi hash untuk memverifikasi keaslian pesan pada variasi kedua.

5.2. Saran

Peneliti ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak ini yaitu:

1. Perangkat lunak ini dapat dikembangkan dengan menambahkan algoritma kunci publik lainnya, seperti: metode Rabin, ElGamal dan LUC.
2. Perangkat lunak dapat dikembangkan dengan menambahkan fitur multimedia, yaitu dengan menambahkan animasi yang lebih baik dan suara yang mendukung proses simulasi.

Referensi

[1] Schneier, B, 1996, Applied Crytography, Second Edition, John Willey and Sons Inc. Canada.
[2] Kurniawan, J., 2004, Kriptografi, Keamanan Internet dan Jaringan Komunikasi, Informatika, Bandung.
[3] Stallings, W., Cryptography and Network Security Third Edition, Prentice Hall.
[4] Putar, R., 2005, The Best Source Code Visual Basic, PT. Elex Media Komputindo, Jakarta.
[5] Suryokusumo.A, 2001, Microsoft Visual Basic 6.0, PT. Elex Media Komputindo, Jakarta.
[6] Novian.A, 2004, Panduan MS. Visual Basic 6, Andi, Yogyakarta.
[7] Supardi.Y, 2006, Microsoft Visual Basic 6.0 Untuk Segala Tingkat, PT. Elex Media Komputindo, Jakarta.
[8] en.wilkipedia.org/wiki/man in the middle attack
[9] www.computerhope.com/jargon/m/mitma.htm
[10] www.cs.umu.se/education/examina/rapporter/mattiasericsson.pdf
[11] www.ouah.org/mitmbrief.htm
[12] www.cs.steven.edu/~swetzel/publications/mim.pdf

LOFIAN: Jurna**l** Tekn**o**logi In**f**orma**s**i d**a**n Komu**n**ikasi
Vol. 1, No. 1, September 2021, Halaman 7-11
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Simulasi Pengontrolan Kesalahan Transmisi Komunikasi Data dengan Menggunakan Metode Automatic Repeat Request (ARQ)

Beny Irawan

Institut Kesehatan Medistra Lubuk Pakam
Jl. Sudirman No. 38, Lubuk Pakam, Kab. Deli Serdang, Sumatera Utara, Indonesia - 20512

benyirawan@medistra.ac.id

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

Abstrak

Dalam komunikasi data, pengontrolan kesalahan berkaitan dengan mekanisme untuk mendeteksi dan memperbaiki kesalahan yang terjadi pada pentransmisian frame. Data dikirim sebagai deretan frame, frame tiba sesuai dengan perintah yang sama saat dikirim, dan masing-masing frame yang ditransmisikan mengalami perubahan dan sejumlah variabel penundaan sebelum mencapai receiver (penerima). Kesalahan yang mungkin terjadi adalah hilangnya frame (sehingga frame gagal diterima receiver) dan rusaknya frame (frame diakui telah tiba, namun beberapa bit mengalami kesalahan, sehingga dibuang oleh receiver). peneliti tertarik untuk merancang suatu perangkat lunak yang mampu mensimulasikan proses kerja metode ARQ dalam mengontrol kesalahan pada proses pengiriman data. Metode Selective-Reject ARQ merupakan metode yang paling efisien dari sisi waktu proses, karena metode tersebut mengirimkan beberapa frame sekaligus tanpa harus menunggu balasan dari receiver terlebih dahulu (keunggulan dibandingkan dengan metode Stop-and-Wait ARQ) dan frame yang dikirim ulang hanyalah frame yang rusak / salah (keunggulan dibandingkan dengan metode Go-back-N ARQ). Simulasi metode ARQ di dalam perangkat lunak memberikan gambaran secara umum prosedur yang dilakukan untuk mengontrol kesalahan pada saat pengiriman frame antar perangkat keras.

Kata Kunci: Automatic Repeat Request, Komunikasi Data, Frame, Perangkat Lunak.

1. Pendahuluan

Dalam komunikasi data, pengontrolan kesalahan berkaitan dengan mekanisme untuk mendeteksi dan memperbaiki kesalahan yang terjadi pada pentransmisian frame. Data dikirim sebagai deretan frame, frame tiba sesuai dengan perintah yang sama saat dikirim, dan masing-masing frame yang ditransmisikan mengalami perubahan dan sejumlah variabel penundaan sebelum mencapai receiver (penerima). Kesalahan yang mungkin terjadi adalah hilangnya frame (sehingga frame gagal diterima receiver) dan rusaknya frame (frame diakui telah tiba, namun beberapa bit mengalami kesalahan, sehingga dibuang oleh receiver).

Secara umum, teknik yang paling umum untuk mengontrol kesalahan adalah pendeteksian kesalahan, balasan positif (mengembalikan balasan positif untuk frame bebas-kesalahan yang diterima dengan baik), retransmisi setelah waktunya habis (sumber melakukan retransmisi frame yang belum dibalas setelah beberapa saat tertentu) dan balasan negatif dan retransmisi (mengembalikan balasan negatif kepada frame yang dideteksi mengalami kesalahan. Sumber melakukan retransmisi terhadap frame yang rusak). Mekanisme pengontrolan kesalahan ini disebut sebagai Automatic Repeat Request (ARQ). Efek ARQ ini adalah mengubah jalur data yang tidak andal menjadi andal. Tiga metode ARQ yang sudah distandarisasikan adalah Stop-and-Wait ARQ, Go-Back-N ARQ dan Selective-Reject ARQ.

Berdasarkan penjelasan diatas, peneliti tertarik untuk merancang suatu perangkat lunak yang mampu mensimulasikan proses kerja metode ARQ dalam mengontrol kesalahan pada proses pengiriman data. Perangkat lunak yang dirancang akan dibatasi dengan Metode ARQ yang disimulasikan dibatasi 3 buah, yaitu Stop-and-Wait ARQ, Go-Back-N ARQ dan Selective-Reject ARQ, Satuan yang digunakan di dalam perangkat lunak adalah tick yaitu yang merupakan satuan waktu terkecil di dalam computer, Skala waktu perangkat lunak dapat diatur, Perangkat lunak mencatat setiap proses yang terjadi ke dalam sebuah log/history, Pada Go-Back-N ARQ dan Selective-Reject ARQ perangkat lunak juga mensimulasikan flow control jendela penggeseran baik dari perspektif pengirim maupun dari perspektif penerima, dan Sistem pengiriman frame akan dimatikan apabila terjadi ‘time-out‘ sebanyak 10 kali berturut-turut.

Dalam proses transmisi data, walaupun data telah dienkripsi, namun terdapat kemungkinan bahwa data tersebut dapat dimiliki oleh orang lain. Salah satu kemungkinan tersebut adalah dengan terjadinya penyadapan media komunikasi yang digunakan oleh kedua orang yang sedang berkomunikasi tersebut. Hal inilah yang disebut dengan man-in-the-middle-attack. Dalam keadaan ini, orang yang menyadap berada di antara kedua orang yang sedang berkomunikasi. Data-data yang dikirimkan oleh orang yang sedang berkomunikasi satu sama lain selalu melalui orang yang menyadap tersebut, sehingga orang yang menyadap tersebut dapat mengetahui semua informasi yang dikirimkan satu sama lain. Keadaan ini muncul karena kedua orang yang sedang berkomunikasi tersebut tidak dapat mem-verifikasi status dari orang yang berkomunikasi dengannya tersebut, dengan mengambil asumsi bahwa proses penyadapan tersebut tidak menyebabkan gangguan dalam jaringan.

2. Landasan Teori

Dalam penelitian ini, peneliti akan memaparkan beberapa landasan teori yang digunakan dalam penelitian ini, antara lain:

2.1. Jaringan Komunikasi Data

Seringkali, sangatlah tidak praktis apabila dua perangkat komunikasi dihubungkan secara langsung, dari ujung ke ujung. Berikut merupakan contoh kemungkinan-kemungkinan yang terjadi:

1. Bila perangkat-perangkatnya merupakan bagian yang saling jauh terpisah, misalnya berada pada jarak ribuan kilometer, tentunya akan memakan biaya yang sangat banyak sekali untuk menyambung dan menghubungkannya.
2. Terdapat serangkaian perangkat, masing-masing membutuhkan jaringan untuk menghubungkan satu sama lain pada waktu-waktu yang berbeda. Sebagai contoh, seluruh telepon di dunia serta semua terminal dan komputer dimiliki oleh satu perusahaan yang sama. Kecuali dalam hal-hal tertentu, misalnya untuk beberapa alat-alat yang jumlahnya terbatas, sangatlah tidak praktis bila harus menyediakan kabel untuk menghubungkan masing-masing bagian perangkat tersebut.

2.2. Data Link Control

Karena kemungkinan bisa terjadi kesalahan pada transmisi, serta karena receiver data perlu mengatur rate terhadap data yang diterimanya, perlu untuk membuat lapisan kontrol pada setiap perangkat komunikasi yang menyediakan fungsi seperti flow control, pendeteksian kesalahan dan kontrol kesalahan. Lapisan kontrol ini disebut data link control protocol.

Beberapa persyaratan dan tujuan komunikasi data efektif diantara dua station pentransmisi dan penerima yang dihubungkan secara langsung, yakni:

1. Sinkronisasi Frame

Blok data dalam jumlah besar akan dipecah-pecah oleh sumber menjadi blok-blok yang lebih kecil yang disebut frame. Permulaan dan ujung setiap frame harus nampak jelas. Hal ini dilakukan karena:

a. Ukuran penyangga receiver terbatas.
b. Blok data dalam jumlah besar dapat menyebabkan transmisi menjadi lebih lama, akibatnya dimungkinkan terjadinya kesalahan lebih besar, sehingga mengharuskan dilakukannya transmisi ulang keseluruhan frame. Dengan frame yang lebih kecil, kesalahan bisa terdeteksi lebih cepat, dan data yang harus ditransmisikan ulang juga lebih sedikit.
c. Pada media yang dipakai bersama, seperti LAN, biasanya tidak dikehendaki satu station menempati media dalam waktu yang panjang, karena bisa menyebabkan penundaan yang lama pada station-station pengirim lain.

2. Flow Control
   Station pengirim tidak boleh mengirim frame pada rate yang lebih cepat dibanding rate station penerima dalam menerima frame-frame tersebut.
3. Pengontrolan Kesalahan
   Kesalahan-kesalahan bit diakibatkan oleh sistem transmisi yang harus diperbaiki.
4. Pengalamatan
   Pada jalur multipoin, seperti Local Area Network (LAN), identitas dua station yang berkomunikasi harus ditentukan dengan jelas.
5. Kontrol data pada jalur yang sama
   Biasanya tidak diharapkan memiliki jalur komunikasi yang terpisah secara fisik untuk mengontrol informasi. Karenanya, receiver harus mampu membedakan informasi kontrol dari data yang sedang ditransmisikan.
6. Manajemen Jalur
   Permulaan, pemeliharaan dan penghentian pertukaran data memerlukan koordinasi dan kerjasama yang baik di antara station. Karena itu diperlukan suatu prosedur manajemen untuk pertukaran ini.

2.3. Tick

Tick sering juga disebut dengan clock tick atau cycle. Tick merupakan unit waktu terkecil yang dikenal oleh komputer. Semakin cepat clock tick atau

e-ISSN: 2798-9593
p-ISSN: 2798-9836
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

cycle, maka semakin banyak instruksi yang dapat dijalankan CPU dalam satu detik. Kecepatan clock tick diekspresikan dalam megahertz atau gigahertz. Setiap instruksi yang akan dijalankan oleh komputer memerlukan sejumlah clock tick, tetapi adakalanya beberapa instruksi dapat dieksekusi dalam satu clock tick pada komputer yang cepat.

Di dalam program simulasi, tick merupakan satuan waktu khusus yang digunakan sebagai iterasi antar proses atau mekanisme simulasi pada komputer.

2.4. Simulasi

Simulasi adalah proses merancang model dari suatu sistem yang sebenarnya, mengadakan percobaan-percobaan terhadap model tersebut dan mengevaluasi hasil percobaan tersebut. Adapun manfaat dari simulasi adalah sebagai berikut:

1. Menjelaskan kelakuan sistem.
2. Menirukan bekerjanya suatu sistem melalui melalui suatu model.
3. Memecahkan suatu persoalan matematik dengan analisis numerik.
4. Mempelajari dinamika suatu sistem.
5. Memberikan suatu deskripsi perilaku sistem dalam perkembangan sejalan dengan bertambahnya waktu.
6. Membangun teori atau hipotesa yang mempertanggungjawabkan kelakuan dari sistem yang diamati.
7. Meramalkan kelakuan sistem yang akan datang yaitu pengaruh yang dihasilkan oleh perubahan-perubahan sistem atau perubahan operasinya.

3. Metodologi Penelitian

Adapun lokasi penelitian ini berada transmisi data yang antar pengirim dan penerima.

3.1. Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi lapangan
   Degan metode ini peneliti mengamati bagaimana data dikirim sebagai deretan frame, frame tiba sesuai dengan perintah yang sama saat dikirim, dan masing – masing frame yang ditransmisikan mengalami perubahan dan sejumlah variabel penundaan sebelum mencapai receiver (penerima).
2. Studi Kepustakaan
   Dengan melakukan studi pustaka, peneliti mendapatkan data-data yang bersifat teori ilmiah yang dipergunakan sebagai dasar dalam melakukan penulisan dan analisa terhadap kendala-kendala yang ada sehingga kendala tersebut dapat diselesaikan dengan baik.

3.2. Langkah dalam pembuatan perangkat lunak

Terdapat serangkaian langkah-langkah yang dilakukan secara terencana dan sistematis guna mendapatkan penecahan masalah atau menjawab pertanyaan-pertanyaan dari penelitian.

1. Membaca dan mempelajari buku-buku yang berhubungan dengan metode ARQ.
2. Mempelajari prosedur pengontrolan kesalahan pada Stop-and-Wait ARQ, Go-Back-N ARQ dan Selective-Reject ARQ.
3. Mempelajari teknik-teknik dasar pemrograman.
4. Merancang interface untuk perangkat lunak simulasi.
5. Merancang perangkat lunak simulasi pengontrolan kesalahan dengan metode ARQ.
6. Melakukan pengujian dan pengetesan terhadap perangkat lunak hasil rancangan.

4. Hasil dan Pembahasan

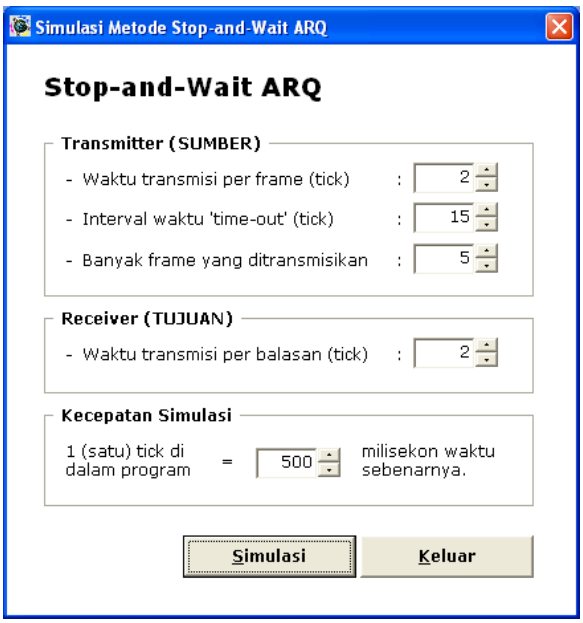
Penelitian ini selanjutnya mendapatkan hasil dan akan dibahas sebagai berikut:

4.1. Hasil

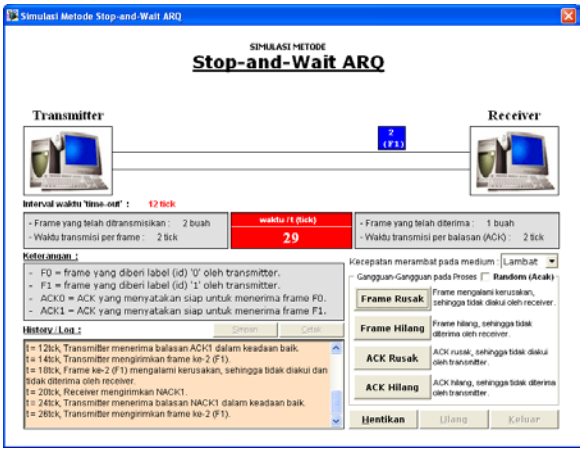
Sebagai contoh, peneliti meng-input data sebagai berikut:

1. Metode yang disimulasikan adalah metode Stop-and-Wait ARQ.
2. Waktu transmisi per frame = 2 tick.
3. Interval waktu time-out = 15 tick.
4. Banyak frame yang ditransmisikan = 5 buah.
5. Waktu transmisi per balasan = 2 tick.
6. 1 tick di dalam program = 500 milisekon waktu sebenarnya.

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 7-11
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836



Gbr. 1. Form Input Metode Stop-and-Wait ARQ



Gbr. 2. Form Simulasi Metode Stop-and-Wait ARQ

History / log yang berhasil dicatat adalah sebagai berikut:

1. Simulasi Metode Stop-And-Wait ARQ

- Waktu transmisi per frame = 2 tick.
- Waktu transmisi per balasan (ACK) = 2 tick.
- Interval waktu 'time-out' = 15 tick.
- Banyak frame yang ditransmisikan = 5 buah.

2. History / Log:

t = 2tck, Transmitter mengirimkan frame ke-1 (F0).
t = 6tck, Receiver menerima frame ke-1 (F0) dalam keadaan baik.
t = 8tck, Receiver mengirimkan ACK1.
t = 12tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 14tck, Transmitter mengirimkan frame ke-2 (F1).
t = 18tck, Frame ke-2 (F1) mengalami kerusakan, sehingga tidak diakui dan tidak diterima oleh receiver.
t = 20tck, Receiver mengirimkan NACK1.
t = 24tck, Transmitter menerima balasan NACK1 dalam keadaan baik.
t = 26tck, Transmitter mengirimkan frame ke-2 (F1).
t = 30tck, Receiver menerima frame ke-2 (F1) dalam keadaan baik.
t = 32tck, Receiver mengirimkan ACK0.
t = 36tck, ACK0 mengalami kerusakan, sehingga tidak diakui dan tidak diterima oleh transmitter.
t = 41tck, Waktu pada pencatat waktu di transmitter habis (time-out). Transmitter kembali mengirimkan frame ke-2 (F1).
t = 43tck, Transmitter mengirimkan frame ke-2 (F1).
t = 47tck, Receiver menerima frame ke-2 (F1). Frame ini sebelumnya telah diterima, sehingga dibuang oleh receiver.
t = 49tck, Receiver mengirimkan ACK0.
t = 53tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 55tck, Transmitter mengirimkan frame ke-3 (F0).
t = 59tck, Receiver menerima frame ke-3 (F0) dalam keadaan baik.
t = 61tck, Receiver mengirimkan ACK1.
t = 65tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 67tck, Transmitter mengirimkan frame ke-4 (F1).
t = 71tck, Receiver menerima frame ke-4 (F1) dalam keadaan baik.
t = 73tck, Receiver mengirimkan ACK0.
t = 77tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 79tck, Transmitter mengirimkan frame ke-5 (F0).
t = 83tck, Receiver menerima frame ke-5 (F0) dalam keadaan baik.
t = 85tck, Receiver mengirimkan ACK1.
t = 89tck, Transmitter menerima balasan ACK1 dalam keadaan baik.

4.2. Pembahasan

Perangkat lunak simulasi ini memiliki persyaratan sebagai berikut:

1. Perangkat lunak menerima input berupa: waktu transmisi per frame, waktu transmisi per balasan, interval waktu time-out, banyak frame yang akan ditransmisikan dan kecepatan simulasi.
2. Untuk metode Go-Back-N ARQ dan Selective-Reject ARQ yang menggunakan flowcontrol jendela penggeseran, urutan nomor bit dan ukuran jendela dapat di-input.
3. Perangkat lunak harus mampu mensimulasikan prosedur pengontrolan kesalahan dengan metode

e-ISSN: 2798-9593
p-ISSN: 2798-9836
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

Stop-and-Wait ARQ, Go-Back-N ARQ dan Selective-Reject ARQ.
4. Di tengah proses simulasi, user dapat membuat gangguan berupa frame atau balasan yang sedang dikirim mengalami kerusakan atau hilang dan melihat tindakan yang akan dilakukan oleh transmitter atau receiver dalam menghadapi gangguan yang terjadi sesuai dengan metode ARQ yang sedang disimulasikan.
5. Perangkat lunak mencatat semua kejadian ke dalam sebuah log. Log dapat disimpan dalam file berformat text atau dicetak.
6. Kecepatan merambat frame pada medium dapat diatur di tengah proses simulasi.
7. User dapat menghentikan (pause) dan melanjutkan (resume) proses simulasi. Ini dimaksudkan agar user dapat memahami kejadian yang telah / sedang terjadi secara bertahap.

Dengan perangkat lunak simulasi ini, diharapkan prosedur pengontrolan kesalahan dengan metode ARQ tidak akan sulit dipahami lagi, karena masing-masing metode disimulasikan dengan jelas.

5. Kesimpulan dan Saran

Dari hasil penelitian dan pembahasannya, maka dapat disimpulkan dan saran.

5.1. Kesimpulan

Setelah menyelesaikan perangkat lunak simulasi pengontrolan kesalahan dengan metode ARQ, peneliti menarik kesimpulan sebagai berikut:
1. Metode Selective-Reject ARQ merupakan metode yang paling efisien dari sisi waktu proses, karena metode tersebut mengirimkan beberapa frame sekaligus tanpa harus menunggu balasan dari receiver terlebih dahulu (keunggulan dibandingkan dengan metode Stop-and-Wait ARQ) dan frame yang dikirim ulang hanyalah frame yang rusak / salah (keunggulan dibandingkan dengan metode Go-back-N ARQ).
2. Walaupun lebih efisien dalam sisi waktu proses, metode Selective-Reject ARQ memerlukan algoritma yang lebih kompleks dibandingkan dengan metode Go-back-N ARQ, karena metode ini memerlukan algoritma penyisipan frame yang salah pada urutan yang benar (hal ini disebabkan karena hanya frame yang salah yang dikirim ulang). Oleh karena itu, metode Go-back-N ARQ lebih banyak dipakai dibandingkan dengan metode Selective-Reject ARQ.
3. Simulasi metode ARQ di dalam perangkat lunak memberikan gambaran secara umum prosedur yang dilakukan untuk mengontrol kesalahan pada saat pengiriman frame antar perangkat keras.

5.2. Saran

Peneliti ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak simulasi ini yaitu:
1. Perangkat lunak dapat dikembangkan dengan menambahkan fasilitas suara atau gambar yang lebih menarik, sehingga diharapkan proses yang disimulasikan akan lebih mudah dimengerti.
2. Untuk mendapatkan animasi yang lebih baik, animasi proses simulasi dapat dibangun dengan menggunakan aplikasi Macromedia Flash.
3. Dipertimbangkan untuk menambah animasi antrian paket pada metode Stop-and-Wait ARQ.

Referensi

[1] Green, D.C., Komunikasi Data, Andi, Yokyakarta, 2000.
[2] William Stallings, Dasar-dasar Komunikasi Data, Salemba Teknika, 2003.
[3] Bertsekas, D., dan Gallagher, Data Network, Prentice Hall, 1992.
[4] Abraham Silberschatz, dan James L. Peterson, Operating Systems and Concepts, Addison-Wesley Publishing Company, Inc., June 1988.
[5] Hadi, Rahadian, Pemrograman Microsoft Visual Basic 6.0, PT. Elex Media Komputindo, Jakarta, 2001.
[6] Rahmat Putar, The Best Source Code Visual Basic, PT. Elex Media Komputindo, 2005.
[7] Hadi, Rahadian, Pemrograman Microsoft Visual Basic, PT. Elex Media Komputindo, Jakarta, 2001.
[8] http://www.webopedia.com/TERM/C/clock_tick.html
[9] http://www.webopedia.com/TERM/C/clock_speed.html

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 12-16
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

Analisis Preboot Execution Environment Server Linux dengan Algoritma First Come First Serve

Erwin Daniel Sitanggang

AMIK Medan Business Polytechnic
Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia - 20155

rwins@amikmbp.ac.id

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

Abstrak

Preboot Execusion Environment Server merupakan layanan jaringan computer yang terintegrasi dengan system operasi linux. Layanan ini merupakan suatu workstation/mesin yang dapat beroperasi tanpa adanya dukungan media penyimpanan (storage/disk) local. Hal ini tentu saja memberi manfaat dalam perkembangan teknologi jaringan computer dalam berbagai aspek. Dalam penerapannya dengan menggunakan perangkat server maupun client yang sama pada distro linux CentOS 7 dan Debian 7 dengan memperhatikan aspek efesiensi seperti waktu akses, jumlah akses dan kecepatan akses untuk meminimalkan penggunaan sumber daya dapat disimpulkan bahwa semakin banyak client maka burst time yang akan digunakan client semakin sedikit. Untuk menanggulangi permasalahan tersebut diterapkan algortima penjadwalan FSCS sehingga kecepatan masing-masing client tetap stabil. Dan hasil diperoleh perbandingan kinerja dari kedua distro linux tersebut bahwa Debian lebih unggul bila melayani client yang sedikit dan CentOS unggul dari segala aspek jika jumlah client semakin banyak.

Kata Kunci: Preboot Execution Environment, Open Source, Jaringan Komputer, First Come First Serve, Distro Linux.

1. Pendahuluan

Teknologi infromasi dan komunikasi saat ini berkembang dengan sangat pesat dan terus meningkat dalam waktu yang singkat terutama system operasi open source berbasis linux atau yang disebut dengan distro linux. Hal ini memberi manfaat dalam perkembangan teknologi jaringan computer yang meliputi aspek salah satunya adalah Preboot Execution Environment server. Dengan integrasi yang dimiliki antar komponen yang terhubung dengan jaringan, Preboot Execution Environment server akan mampu menyediakan fungsi dan fitur berkualitas, handal, cepat dan aman sesuai dengan system yang dibutuhkan.

Penerapan Preboot Execution Environment server dengan tetap memperhatikan aspek efesiensi seperti waktu akses, jumlah akses dan kecepatan akses., sehingga sebuah server dituntut untuk meminimalisir penggunaan sumber daya. Namun maksud efesiensi dengan meminimalisir sumber daya bukan berarti mengurangi kualitas performa. Ketika sebuah performasi jaringan computer terganggu, efek yang ditimbulkan akan sangat beragam.

Manfaat utama dari menggunakan Preboot Execution Environment server adalah mengurangi konsumsi sumber daya dan tempat. Namun bagaimana dengan performa dan kualitas services ditinjau dari waktu, jumlah dan kecepatan akses terhadap penerapannya diberbagai varian distro linux.

2. Tinjauan Pustaka

Dalam penelitian ini, pada dasarnya peneliti membandingkan dari penelitian-penelitian terdahulu yang relevan.

Meninjau dari penelitian Widagdo (2009) yang berjudul Penerapan Quality of Service (QoS) menggunakan Traffic Shaping pada Jaringan Diskless PXE Linux, peneliti membahas tentang implementasi dan analisis Quality of Service (QoS) jaringan komputer diskless menggunakan PXE Linux untuk mendapatkan jaringan yang efisien, hemat sumber daya dan biaya ditinjau dari segi kecepatan dan keutuhan data. Tapi pada penelitian ini memiliki beberapa kelemahan, ytiu kurangnya kelas-kelas untuk kualitas layanan Quality of Services (QoS) pada jaringan diskless sehingga tidak terdapat pilihan untuk kualitas layanan yang bisa disesuaikan dengan kebutuhan dalam jaringan tersebut.

Dalam penelitian Hidayat (2012) dengan judul Implementasi dan Analisa Redundansi dan High Availability dalan Server untuk Diskless Thin Client berbasis Storage Area Network membahas tentang Storage Area Network (SAN) yang merupakan metode

Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

dengan kecepatan tinggi yang cocok untuk server diskless yang memiliki redundansi dan ketersediaan yang tinggi bagi client yang terhubung ke server. Peneliti menyarankan untuk pengembangan ke level perusahaan dalam penerapan diskless harus mengikuti standar menggunakan network yang berkecepatan tinggi.

Penelitian dari Kurniadi (2012) dengan judul Pemanfaatan PXE Untuk Mengatasi Gagal Booting PC-Client dan Akses Image Sistem Operasi/File di Infrastruktur Jaringan membahas bagaimana pemanfaatan PXE untuk mengatasi gagal booting PC-Client dan akses image sistem operasi/file di infrastruktur jaringan. Dari pembuatan system ini terdapat beberapa kelemahan seperti kurangnya penerapan WOL SSHFS sebagai pengganti NFS, karena teknologi ini memiliki tingkat keamanan yang lebih mutakhir.

Dari peninjauan penelitian-penelitian yang telah dipaparkan diatas, peneliti akan memaparkan beberapa landasan teori yang digunakan dalam penelitian ini, antara lain:

2.1. Preboot Execution Environment (PXE) Server

Menurut Husni (2014) menjelaskan bahwa PXE server adalah suatu workstation/mesin yang dapat beroperasi tanpa adanya dukungan media penyimpanan (storage/disk) local.

2.2. Jaringan Komputer

Sebuah jaringan komputer paling sedikit terdiri dari dua komputer yang saling terhuung dengan sebuah media sehingga komputer-komputer tersebut dapat saling berbagi resource dan saling berkomunikasi. Namun kenyataannya sebuah jaringan komputer biasanya terdiri dari banyak komputer (lebih dari dua). Semua network berbasis pada konsep pembagian (sharing).

Jaringan komputer muncul dari adanya kebutuhan untuk berbagi data di antara para pengguna. Komputer memiliki kemampuan dalam memproduksi beberapa jenis informasi yang berupa data, sphearsheet atau grafik.

2.3. First Come First Serve (FCFS)

Pertama datang, pertama dilayani (First Come First Serve/First In First Out (FIFO)), tidak peduli burst timenya panjang atau pendek. Bila sebuah proses yang sedang dikerjakan maka akan diselesaikan dulu dan baru kemudian proses berikutnya dilayani (Pangera et al, 2005).

Secara harafiah queue dapat diartikan sebagai artian yaitu merupakan data dimana penambahan data (elemen) hanya melalui satu sisi yaitu depan (head) dan penghapusan data (elemen) hanya melalui sisi belakang (tail). Sifat ini sering dengan FIFO yaitu data yang masuk pertama akan keluar pertama juga data yang terakhir masuk akan terakhir keluar (Utami et al, 2007).

Operasi-operasi standar pada queue adalah:

1. Membuat queue atau inilisiasi.
2. Mengecek apakan queue penuh atau full.
3. Mengecek apakah queue kosong atau empty.
4. Memasukkan elemen ke dalam queue atau InQueue.
5. Menghapus elemen dari queue atau DeQueue.

3. Metodologi Penelitian

Penelitian ini dilakukan di jaringan Local Area Network (LAN) laboratorium jaringan komputer dengan topologi jaringan yang digunakan adalah star. Pemilihan topologi star karena dianggap cukup mudah untuk diimplementasi dan dikelola karena komputer-komputer terhubung melalui kabel secara terpusat ke dalam sebuah Switch/Hub.

3.1. Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi lapangan
   Peneliti terjun kelapangan untuk mendapatkan data dengan wawancara dan pengamatan langsung dengan pihak yang berwenang dan peneliti mendapatkan data berupa laboratorium yang dapat digunakan beserta jumlah computer yang ada dan computer spesifikasi computer yang dapat digunakan sebagai server.
2. Studi Kepustakaan
   Dengan melakukan studi pustaka, peneliti mendapatkan data-data yang bersifat teori ilmiah yang dipergunakan sebagai dasar dalam melakukan penulisan dan analisa terhadap kendala-kendala yang ada sehingga kendala tersebut dapat diselesaikan dengan baik.

3.2. Metode Analisa Data

Adapun metode yang digunakan dalam menganalisis data dalam penelitian adalah:

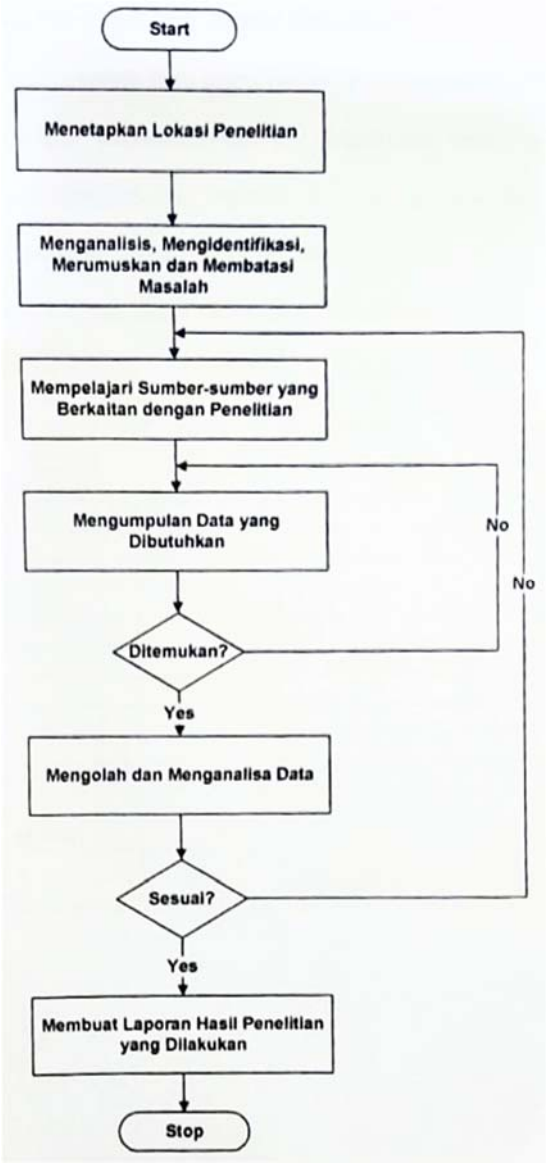
1. Metode Deskriptif, mengumpulkan, merumuskan, mengklasifikasikan, menganalisa dan menyimpulkan sehingga data tersebut dapat

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 12-16
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

memberikan gambaran yang jelas tentang masalah yang diteliti.
2. Metode Deduktif, menganalisa data dengan cara mengambil kesimpulan berdasarkan teori yang telah diterima sebagai suatu kebenaran umum mengenai fakta yang diamati.

3.3. Langkah dan Diagram Alir Penelitian

Terdapat serangkaian langkah-langkah yang dilakukan secara terencana dan sistematis guna mendapatkan penecahan masalah atau menjawab pertanyaan-pertanyaan dari penelitian.



Gbr. 1. Langkah dan Diagram Alir Penelitian

4. Hasil dan Pembahasan

Penelitian ini selanjutnya mendapatkan hasil dan akan dibahas sebagai berikut:

4.1. Hasil

Hasil penerapan Preboot Execution Environment Server pada sistem operasi linux yang memiliki kestabilan tinggi dapat meningkatkan efisiensi waktu akses, jumlah akses dan kecepatan akses. Namun sistem operasi yang stabil saja tidak cukup, untuk itulah perlu diterapkan algoritma antrian FCFS untuk menguji beberapa jumlah user yang dapat dilayani user sehingga mempermudah manajemen user guna menciptakan waktu, jumlah, kecepatan akses yang efesien.

4.2. Pembahasan

Pembahasan dilakukan untuk menganalisa, merancang, menerapkan dan menguji system sehingga output berdasarkan waktu, jumlah akses, kecepatan akses kinerja Preboot Execution Environment Server.
1. Analisis
   Proses akhir dari analisis adalah pelaporan rincian dari berbagai komponen atau elemen system yang dibutuhkan. Berbagai elemen atau komponen tersebut mencakup:

Tabel 1
Spesifikasi Sistem

Sistem	Keterangan
Preboot Execution Environment	Jaringan Diskless
Dynamic Host configuration Protokol	Layanan Internet Akses
FTP	Transfer File antar Jaringan

Tabel 2
Spesifikasi Software

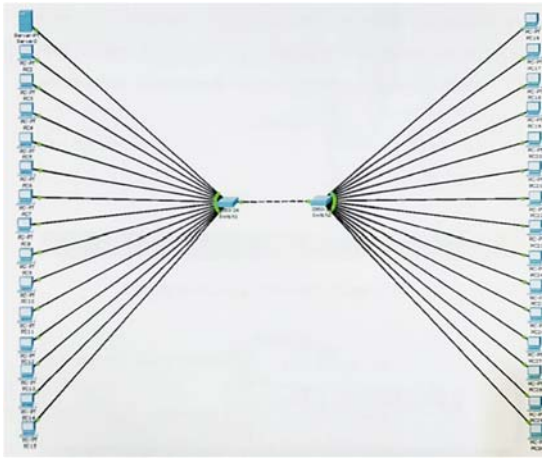
No.	Software	Keterangan
1	CentOS 7	Sistem Operasi PXE Server
2	Debian 7	Sistem Operasi PXE Server
3	CloneZilla	Backup & Restore System

Tabel 3
Spesifikasi Perangkat Keras

No.	Perangkat	Jumlah	Spesifikasi Unit
1	Server	1	Intel Core 2 Duo 1.80 GHz Hardisk STA 150 GB Momeory RAM 1 GB Monitor LCD
2	Client	40	Intel Core 2 Duo 1.80 GHz Hardisk STA 320 GB Momeory RAM 1 GB Monitor LCD

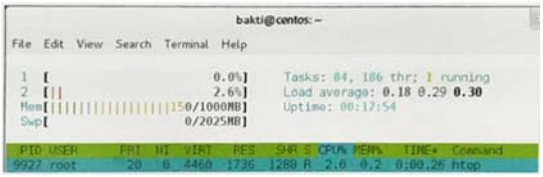
LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 12-16
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

2. Rancangan Jaringan PXE Server
   Pada topologi star yang dibangun, setiap computer langsung terhubung dengan Switch/Hub yang menggunakan media transmisi kabel UTP dengan menggunakan konektor RJ-45 sehingga membentuk jaringan LAN dengan topologi sebagai berikut:



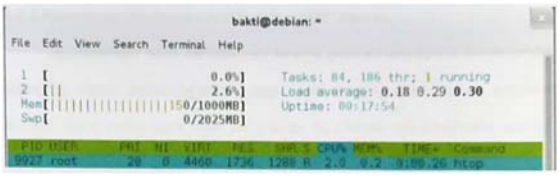
Gbr. 2. Topologi Jaringan

3. Hasil Analisis Penerapan PXE Server
   Hal yang diuji pada penerapan PXE Server adakah kestabilan server untuk menangani beberapa client daoam sebuah jaringan. Pada keadaan ideal seluruh client sedang tidak digunakan, server hanya membutuhkan sedikit resource untuk menjaga agar system tetap stabil.

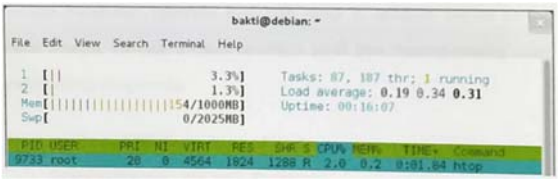


Gbr. 3. Keadaan CPU Usage CentOS (0 Client)

Gbr. 4. Keadaan CPU Usage CentOS (1 Client)



Gbr. 5. Keadaan CPU Usage Debian (0 Client)



Gbr. 6. Keadaan CPU Usage Debian (1 Client)

Jika dua buah client yang melakukan request makan kecepatan transfer yang tadinya digunakan seutuhnya untuk satu client, kini terbagi menjadi dua dan begitu pula berlaku kelipatannya.

Tabel 4
Perbandingan Kecepatan Transfer

No.	Jumlah Client	Fast Ethernet (100 Mbps) CentOS 7	Fast Ethernet (100 Mbps) Debian 7
1	1	620,97 MB/min	678,90 MB/min
2	2	535,06 MB/min	540,38 MB/min
3	3	464,79 MB/min	450,76 MB/min
4	4	285,44 MB/min	282,61 MB/min
5	5	162,37 MB/min	140,08 MB/min

4. Penerapan Algoritma FCFS dalam PXE Server
   Berdasarkan hasil pengukuran kinerja PXE Server di varian system operasi linux dapat disimpulkan bahwa semakin banyak client maka burst time yang akan digunakan client semakin sedikit. Untuk menanggulangi permasalahan tersebut perlu diterapkan algoritma penjadwalan FCSC sehingga kecepatan masing-masing client tetap stabil. Setelah melakukan beberapa pengujian maka ditetapkan bahwa jumlah client dibatasi menjadi 5 client sehingga burst time masing-masing client dapat terbagi dengan rata.

5. Perbandingan Kinerja CentOS 7 dan Debian 7
   Hasil ini didapatkan berdasarkan beberapa kali pengujian dengan spesifikasi server dan client yang sama.

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Gbr 7. CentOS7 dan Debian 7

Dari gambar diatas, hasil yang diperoleh adalah Debian lebih unggul dari pada CentOS apabila melayani sedikit user/client. Apabila jumlah client semakin banyak, CentOS lebih unggul dari pada Debian dari segala aspek.

5. Kesimpulan dan Saran

Dari hasil penelitian dan pembahasannya, maka dapat disimpulkan dan saran.

5.1. Kesimpulan

Algoritma FCFS membatasi jumlah antrian user sehingga server tidak down/crash menghadapi banyaknya jumlah permintaan client sehingga dapat diputuskan beberapa jumlah setiap antrian.

Debian mempunyai efisiensi penggunaan waktu dan kecepatan akses yang lebih baik dibandingkan dengan CentOS apabila jumlah user/client sedikit. Sebaliknya, CentOS mempunyai efisiensi penggunaan waktu dan kecepatan akses yang lebih baik dari pada Debian apabila jumlah user/client sangat banyak.

5.2. Saran

Untuk pengembangan sistem jaringan Preboot Execution Environment server ini, maka penulis menyarankan beberapa hal sebagai berikut:

1. Adanya pengembangan pada fitur yang disediakan untuk client dapat di klarifikasikan dan spesifik sesuai dengan kebutuhan pengguna agar efisiensi penggunaan jaringan dapat lebih optimal.
2. Pengamatan dan pengelolaan lalu intas dan kebijakan dalam jaringan Preboot Execution Environment Server, sehingga proses komunikasi antar jaringan diskless dapat mendukung komunikasi yang lebih kompleks dan dapat dikombinasikan dengan konsep komunikasi lain namun tetap dapat terjaga kualitas layanannya.

Referensi

[1] Widagdo, K. 2012. Penerapan Quality of Service (QoS) menggunakan Traffic Shaping pada Jaringan Diskless PXE Linux. Jakarta: UIN Syarif Hidayatullah Fakultas Sains dan Teknologi.
[2] Hidayat, F. 2012. Implementasi dan Analisa Redundansi dan High Avaibility dalam Server untuk Diskless Thin Client berbasis Storage Area Network. Depok: Universitas Indonesia Fakultas Teknik.
[3] Kurniadi, E. 2012. Pemanfaatan PXE Untuk Mengatasi Gagal Booting PC-Client dan Akses Image Sistem Operasi/File di Infrastruktur Jaringan. Yokyakarta: Sekolah tinggi Manajemen Informatika dan Komputer AMIKOM.
[4] Husni. Implementasi Jaringan Komputer Dengan Linux Redhat 9. ANDI. Yokyakarta. 2004.
[5] Pangera, A.A. & Arius. 2005. Sistem Operasi. Yogyakarta: ANDI OFFSET.
[6] Utami, E., Raharjo, S., & Sukrisno. 2007. Struktur Data Konsep & Implementasinya dalam Bahasa C & Free Pascal di GNU/LINUX. Yogyakarta: Graha Ilmu.

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 17-23
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Simulasi Pemodelan Procedure-Consumer Problem pada Sistem Operasi

Maradu Sihombing[1], Jaidup Banjarnahor[2]

[1,2]*AMIK Medan Business Polytechnic*
Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia - 20155

[1]*maradu@amikmbp.ac.id*, [2]*jaidup@amikmbp.ac.id*

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

Abstrak

Sistem operasi yang menggunakan lebih dari satu proses untuk dapat bekerja bersama mencapai tujuan yang diinginkan. Agar tujuan tercapai secara benar, proses-proses tersebut harus mensinkronkan kegiatan-kegiatannya sehingga terkendali dengan baik untuk menghindari kondisi deadlock, salah satu metode untuk menyelesaikan masalah Deadlock adalah Procedure-Consumer. Procedure-Consumer menggunakan Perangkat lunak metode sleep and wake-up untuk mencegah masalah yang terjadi ketika buffer penuh, sementara producer ingin meletakkan item ke buffer dan consumer ingin mengambil item sementara buffer telah kosong. Perangkat lunak menggunakan semaphore untuk untuk mem-blocked producer atau consumer lain ketika salah satu producer atau consumer sedang berada dalam buffer. Perangkat lunak simulasi Producer-Consumer ini merupakan illustrasi dari proses sinkronisasi, yaitu bagaimana cara mengatur beberapa proses yang mengakses beberapa variabel secara bersamaan.

Kata Kunci: Sistem Operasi, Deadlock, Procesure-Consumer, Buffer.

1. Pendahuluan

Dalam sistem operasi, lebih dari satu proses dapat bekerja bersama untuk mencapai tujuan yang diinginkan. Agar tujuan tercapai secara benar, proses-proses tersebut harus mensinkronkan kegiatan-kegiatannya sehingga terkendali dengan baik untuk menghindari kondisi deadlock.

Kasus producer-consumer digunakan sebagai ilustrasi pembahasan sinkronisasi. Kasus producer-consumer berisi masalah mutual-exclusion dan sinkronisasi. Kasus ini sering juga disebut sebagai bounded-buffer problem (masalah buffer dengan jumlah terbatas). Kasus ini dapat diilustrasikan sebagai berikut, terdapat produsen (ilustrasi dari proses yang menyimpan informasi ke buffer) menghasilkan barang (ilustrasi dari informasi) dan konsumen (ilustrasi dari proses yang mengambil informasi dari buffer) yang akan menggunakan barang yang dihasilkan produsen. Keduanya mempunyai market (ilustrasi dari buffer) bersama dan berukuran tetap. Karena ukuran market terbatas, petaka (bencana) dapat terjadi untuk producer dan consumer. Masalah bagi producer terjadi ketika market telah penuh, sementara producer ingin meletakkan barang ke market yang telah penuh itu. Sedangkan masalah bagi consumer terjadi ketika consumer ingin mengambil barang sementara market telah/sedang kosong.

Berdasarkan uraian di atas, peneliti akan merancang suatu perangkat lunak yang mampu untuk mensimulasikan proses producer-consumer dengan menerapkan metode sleep and wakeup untuk mencegah deadlock dan semaphore dan menampilkannya dalam bentuk animasi agar mudah dipahami pengguna terutama pada perkuliahan dengan mata kuliah sistem operasi.

2. Landasan Teori

Untuk dapat memahami penelitian ini lebih dalam, peneliti memaparkan teori-teori yang digunakan.

2.1. Sistem Operasi

Sistem operasi merupakan sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer. Sebelum ada sistem operasi, orang hanya dapat mengoperasikan komputer dengan menggunakan sinyal analog dan sinyal digital. Seiring dengan berkembangnya pengetahuan dan teknologi, muncullah sistem operasi yang menyediakan lingkungan untuk mengoperasikan komputer secara lebih nyaman. Untuk lebih memahami sistem operasi maka sebaiknya perlu diketahui terlebih dahulu

beberapa konsep dasar mengenai sistem operasi itu sendiri.

Pengertian sistem operasi secara umum ialah pengelola seluruh sumber daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (system calls) ke pemakai sehingga memudahkan dan menyamankan penggunaan serta pemanfaatan sumber-daya sistem komputer.

Sistem operasi mempunyai tiga sasaran utama yaitu:

1. Kenyamanan, membuat penggunaan komputer menjadi lebih nyaman.
2. Efisiensi, penggunaan sumber daya sistem komputer secara efisien.
3. Evolusi, sistem operasi harus dibangun sehingga memungkinkan dan memudahkan pengembangan, pengujian serta pengajuan sistem-sistem yang baru.

2.2. Komponen Sistem Operasi

Menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne, pada umumnya sebuah sistem operasi modern mempunyai komponen sebagai berikut:

1. Manajemen Proses.
   Proses adalah program yang sedang di eksekusi. Suatu proses membutuhkan satu atau beberapa sumber daya untuk menyelesaikan tugasnya. Sumber daya tersebut dapat berupa CPU time, memori, berkas-berkas, atau perangkat-perangkat I/O. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti:
   a. Pembuatan dan penghapusan proses pengguna dan sistem proses.
   b. Menunda atau melanjutkan proses.
   c. Menyediakan mekanisme untuk proses sinkronisasi.
   d. Menyediakan mekanisme untuk proses komunikasi.
   e. Menyediakan mekanisme untuk penanganan deadlock.
2. Manajemen Memori Utama.
   Memori utama atau lebih dikenal sebagai memori adalah suatu array besar yang terdiri dari word atau byte, yang ukurannya dapat mencapai ratusan, ribuan, atau bahkan jutaan. Setiap word atau byte mempunyai alamat tersendiri. Memori utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama merupakan tempat penyimpanan data yang sementara (volatile), artinya data dapat hilang begitu sistem dimatikan. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen memori seperti:
   a. Menjaga track dari memori yang sedang digunakan dan siapa yang menggunakannya.
   b. Memilih program yang akan di-load ke memori.
   c. Mengalokasikan dan meng-dealokasikan ruang memori sesuai kebutuhan.
3. Manajemen Secondary Storage.
   Data yang disimpan dalam memori utama bersifat sementara dan ukurannya sangat kecil jika dibandingkan dengan keseluruhan data yang terdapat dalam komputer. Oleh karena itu, untuk meyimpan keseluruhan data dan program komputer dibutuhkan secondary storage yang bersifat non volatil dan mampu menampung banyak data. Contoh dari secondary storage adalah harddisk, disket, dan lain-lain. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan disk-management seperti free-space management, alokasi penyimpanan, penjadualan disk.
4. Manajemen Sistem I/O.
   Fungsi ini sering disebut device manager, dimana sistem operasi menyediakan "device driver" yang umum sehingga operasi I/O dapat seragam (membaca atau menuliskan data tanpa mempedulikan mekanisme kerja yang berbeda dari perangkat-perangkat I/O yang ada). Contoh: pengguna menggunakan operasi yang sama untuk membaca berkas pada hard-disk, CD-ROM dan floppy disk. Komponen untuk sistem I/O, yaitu:
   a. Buffer: penampungan sementara data dari/ ke perangkat I/O.
   b. Spooling: melakukan penjadualan pemakaian I/O sistem supaya pemakaian I/O lebih efisien (antrian dan sebagainya).
   c. Driver: penerjemah instruksi antara sistem operasi dan I/O untuk dapat melakukan operasi tertentu. Setiap perangkat keras I/O memiliki driver yang berbeda-beda.
5. Manajemen File.
   File adalah kumpulan informasi yang dibuat dengan tujuan tertentu. File disimpan dalam struktur yang bersifat hirarkis, seperti direktori. Dalam manajemen file, sistem operasi bertanggung jawab dalam melakukan:
   a. Pembuatan dan penghapusan berkas.
   b. Pembuatan dan penghapusan direktori.
   c. Pemanipulasian berkas dan direktori.
   d. Pemetaan berkas ke secondary storage.
   e. Backup berkas ke media penyimpanan yang permanen (non-volatile).
6. Sistem Proteksi.
   Proteksi mengacu pada mekanisme untuk mengontrol akses yang dilakukan oleh program,

e-ISSN: 2798-9593
p-ISSN: 2798-9836

prosesor, atau pengguna ke sistem sumber daya. Mekanisme proteksi harus:
   a. membedakan antara penggunaan yang sudah diberi izin dan yang belum.
   b. menetapkan pembatasan atau pengaturan yang telah ditentukan (specify the controls to be imposed).
   c. menyediakan tata cara pelaksanaan (provide a means of enforcement).
7. Sistem Jaringan.
   Sistem ini untuk mendukung penggunaan jaringan. Sistem ini umumnya kini telah terpadu dalam sistem operasi karena kebutuhan kinerjanya serta kebutuhan komputasi telah menghendaki kemampuan ini terdapat di dalam sistem komputer. Sistem ini menyediakan akses pengguna ke bermacam sumber-daya sistem. Sistem ini membawa keuntungan dalam hal:
   a. Kecepatan komputer yang meningkat (computation speed-up).
   b. Ketersediaan data meningkat (increased data availability).
   c. Realibilitas dapat ditingkatkan (enhanced reliability).
8. User Interface (Shell).
   Sistem Operasi menunggu instruksi dari pengguna (command driven). Program yang membaca instruksi dan mengartikan control statements disebut control-card interpreter atau command-line interpreter. User Interface sangat bervariasi dari satu sistem operasi ke sistem operasi yang lain dan disesuaikan dengan tujuan dan teknologi I/O devices yang ada. Contohnya: Command Line Interface (CLI), Graphical User Interface (GUI), dan lain-lain.

2.3. Producer-Consumer

Kasus producer-consumer digunakan sebagai ilustrasi pembahasan sinkronisasi. Masalah producer-consumer disebut juga bounded-buffer problem (masalah buffer dengan jumlah terbatas).

Asumsi dalam producer-consumer problem adalah sebagai berikut:
1. Dua proses menggunakan suatu buffer yang dipakai bersama dan berukuran tetap.
2. Satu proses adalah producer yang meletakkan informasi ke buffer.
3. Proses lain adalah consumer yang mengambil informasi dari buffer.

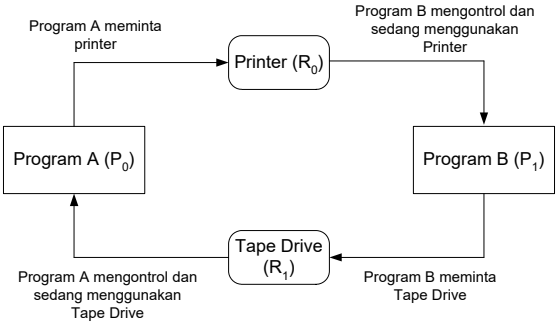
Masalah producer-consumer dapat dikembangkan menjadi masalah yang memiliki m buah produsen dan n buah konsumen. Karena buffer terbatas, masalah berikut dapat terjadi, yaitu:

1. Masalah untuk producer.
   Masalah terjadi ketika buffer telah penuh, sementara producer ingin meletakkan informasi ke buffer yang telah penuh itu.
2. Masalah untuk consumer.
   Masalah terjadi ketika consumer ingin mengambil informasi sementara buffer telah/sedang kosong.

Kedua proses perlu sinkronisasi agar keduanya dapat menghindari masalah.

2.4. Deadlock

Definisi deadlock dapat diperhatikan pada illustrasi berikut. Misalkan pada suatu komputer terdapat dua buah program, sebuah tape drive dan sebuah printer. Program A mengontrol tape drive, sementara program B mengontrol printer. Setelah beberapa saat, program A meminta printer, tapi printer masih digunakan. Berikutnya, B meminta tape drive, sedangkan A masih mengontrol tape drive. Dua program tersebut memegang kontrol terhadap sumber daya yang dibutuhkan oleh program yang lain. Tidak ada yang dapat melanjutkan proses masing-masing sampai program yang lain memberikan sumber dayanya, tetapi tidak ada yang mengalah. Kondisi inilah yang disebut Deadlock. Gambar graph deadlock dapat dilihat pada Gbr. 1. berikut.



Gbr. 1. Graph Deadlock

Deadlock mungkin dapat terjadi pada suatu proses disebabkan proses itu menunggu suatu kejadian tertentu yang tidak akan pernah terjadi. Dua atau lebih proses dikatakan berada dalam kondisi deadlock, bila setiap proses yang ada menunggu suatu kejadian yang hanya dapat dilakukan oleh proses lain dalam himpunan tersebut.

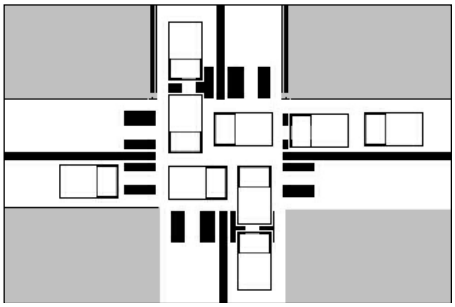
Terdapat kaitan antara overhead dari mekanisme koreksi dan manfaat dari koreksi deadlock itu sendiri. Pada beberapa kasus, overhead atau ongkos yang harus dibayar untuk membuat sistem bebas deadlock menjadi hal yang terlalu mahal dibandingkan jika mengabaikannya. Sementara pada kasus lain, seperti pada real-time process control, mengizinkan deadlock

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 17-23
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

akan membuat sistem menjadi kacau dan membuat sistem tersebut tidak berguna.

Contoh berikut ini terjadi pada sebuah persimpangan jalan. Beberapa hal yang dapat membuat deadlock pada suatu persimpangan, yaitu:

1. Terdapat satu jalur pada jalan.
2. Mobil digambarkan sebagai proses yang sedang menuju sumber daya.
3. Untuk mengatasinya beberapa mobil harus preempt (mundur).
4. Sangat memungkinkan untuk terjadinya starvation (kondisi proses tak akan mendapatkan sumber daya).



Gbr. 2. Ilustrasi *Deadlock*

3. Metodologi Penelitian

Adapun lokasi penelitian ini berada pada sistem operasi komputer dan dilakukan pada laboratorium komputer.

3.1. Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi lapangan
   Degan metode ini peneliti mengamati serta mempelajari bagaimana cara kerja dari sistem operasi khususnya pada hal-hal ayang mengakibatkan deadlock terjadi.
2. Studi Kepustakaan
   Dengan melakukan studi pustaka, peneliti mendapatkan data-data yang bersifat teori ilmiah yang dipergunakan sebagai dasar dalam melakukan penulisan dan analisa terhadap kendala-kendala yang ada sehingga kendala tersebut dapat diselesaikan dengan baik.

3.2. Langkah dalam pembuatan perangkat lunak

Langkah-langkah pembuatan perangkat lunak ini antara lain:

1. Membaca dan mempelajari buku-buku yang berhubungan dengan Sistem Operasi.
2. Mempelajari Producer-Consumer Problem.
3. Mempelajari teknik-teknik dasar pemrograman.
4. Merancang interface untuk perangkat lunak simulasi.
5. Merancang suatu perangkat lunak simulasi Producer-Consumer Problem.
6. Menguji perangkat lunak dan memperbaiki kesalahan (error) yang muncul.

4. Hasil dan Pembahasan

Adapun hasil dan pembahasan dalam penelitian ini akan dibahas disini.

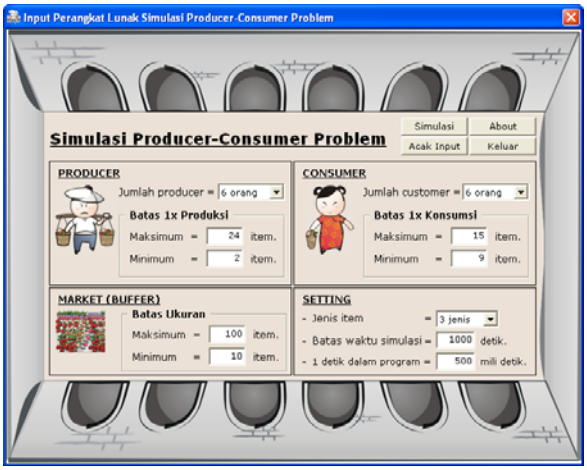
4.1. Hasil

Adapun perangkat lunak (software) yang digunakan untuk menjalankan aplikasi ini adalah lingkungan sistem operasi Microsoft Windows 98, 98 Second Edition atau Microsoft Windows NT / 2000 / XP.

Untuk menguji hasil proses eksekusi perangkat lunak, dimasukkan input sebagai berikut:

1. Jumlah producer = 6 orang
2. Batas satu kali produksi, maksimum= 24 item dan minimum = 2 item.
3. Jumlah consumer = 6 orang.
4. Batas maksimum satu kali konsumsi = 15 item dan minimum = 9 item.
5. Batas ukuran maksimum market = 100 item.
6. Batas ukuran minimum market = 10 item.
7. Banyak jenis item = 3 jenis.
8. Batas waktu simulasi = 1000 detik.

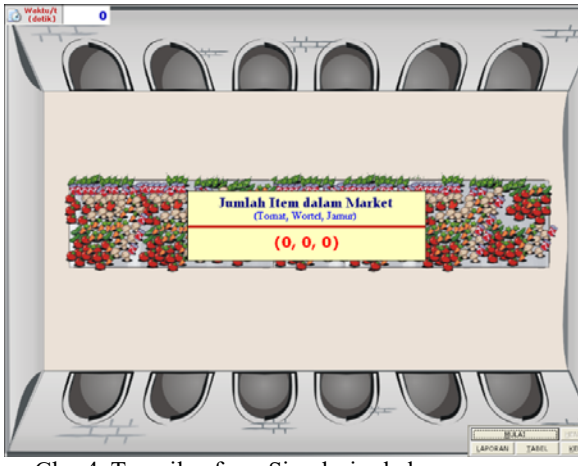
Pada perangkat lunak, form Input terlihat seperti pada Gbr. 3. berikut.



Gbr. 3. Tampilan form Input (Pengujian Program)

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 17-23
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

Tampilan awal form Simulasi dapat dilihat pada Gbr. 4. berikut.



Gbr. 4. Tampilan form Simulasi sebelum proses simulasi dimulai

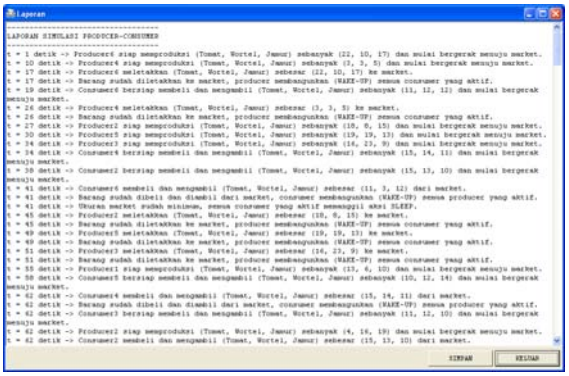
Pada saat t = 202 detik, producer berada dalam keadaan tertidur (sleep). Tampilan prosesnya dapat dilihat pada Gbr. 5. berikut.

Gbr. 5. Tampilan simulasi ketika producer berada dalam keadaan tertidur (sleep)

Pada saat t = 452 detik, consumer berada dalam keadaan tertidur (sleep). Tampilan prosesnya dapat dilihat pada Gbr. 6. berikut.

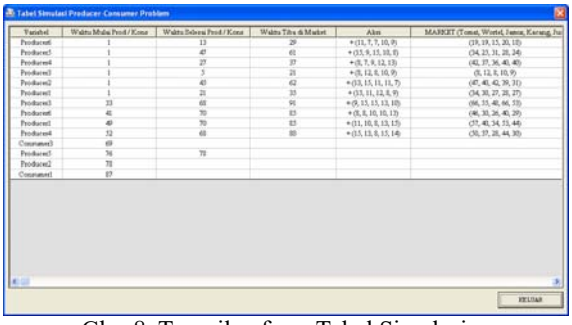
Gbr. 6. Tampilan simulasi ketika consumer berada dalam keadaan tertidur (sleep)

Tampilan form History dapat dilihat pada Gbr. 7. berikut.



Gbr. 7. Tampilan form History

Tampilan form Tabel Simulasi dapat dilihat pada Gbr. 8. berikut.



Gbr. 8. Tampilan form Tabel Simulasi

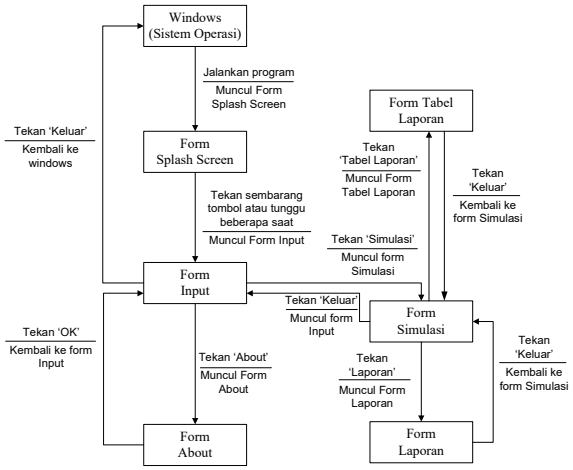
LOFIAN: Jurnal Teknologi Informasi dan Komunikasi e-ISSN: 2798-9593
Vol. 1, No. 1, September 2021, Halaman 17-23 p-ISSN: 2798-9836
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

4.2. Pembahasan

Dalam sub bab ini, akan dibahas mengenai alur kerja perangkat lunak simulasi, penggambaran objek simulasi dan proses simulasi Producer-Consumer Problem, sebagai berikut ini.

1. Alur Kerja Perangkat Lunak Simulasi
   Perangkat lunak simulasi Producer-Consumer Problem dimulai dengan tampilan splash screen. Form splash screen berisi nama / judul perangkat lunak dan nama pembuat perangkat lunak. Beberapa saat kemudian, form input akan tampil. Form input berfungsi untuk mengatur kondisi simulasi. Komponen yang dapat diatur adalah sebagai berikut:
   a. Pengaturan pada Producer, yaitu jumlah producer, batas maksimum dan minimum bagi producer dalam satu kali produksi.
   b. Pengaturan pada Consumer, yaitu jumlah consumer, batas maksimum dan minimum bagi consumer dalam satu kali konsumsi.
   c. Pengaturan pada Market, yaitu batas ukuran maksimum dan minimum market.
   d. Pengaturan lain, yaitu jenis item dan kecepatan proses simulasi.

   Setelah pengaturan pada form input, proses simulasi dapat dimulai dengan menekan tombol ’Simulasi’. Selanjutnya, form simulasi akan muncul. Ketika proses simulasi sedang berjalan, user dapat menghentikan untuk sementara proses simulasi dengan menekan tombol ’Hentikan’. User juga dapat melihat laporan proses yang terjadi dalam simulasi dengan menekan tombol ’Laporan’. Laporan juga disediakan dalam bentuk tabel.
   Alur kerja perangkat lunak simulasi ini dapat digambarkan dalam bentuk State Transition Diagram (STD), seperti terlihat pada gambar 9 berikut.



Gbr. 9. State Transition Diagram (STD)

2. Penggambaran Objek Simulasi
   Objek simulasi atau variabel yang terdapat dalam simulasi Producer-Consumer Problem diillustrasikan dalam bentuk objek gambar. Variabel dalam simulasi ini adalah producer, consumer dan market. Penggambaran variabel dilakukan dengan menggunakan aplikasi photo editor Adobe Photoshop CS.

3. Proses Simulasi Producer-Consumer Problem
   Proses simulasi Producer-Consumer Problem adalah sebagai berikut:
   a. Producer aktif memproduksi dan meletakkan item ke market (buffer). Aksi ini akan menambah jumlah item di dalam market.
   b. Consumer aktif mengambil item dari market (buffer) dan mengonsumsi item. Aksi ini akan mengurangi jumlah item di dalam market.
   c. Apabila market telah penuh atau jumlah item di dalam market telah mencapai batas maksimum, maka producer akan tidur (memanggil aksi sleep).
   d. Apabila consumer mengambil item dari market dan producer dalam keadaan sleep, maka consumer akan membangunkan (wake-up) producer.
   e. Apabila market kosong atau jumlah item di dalam market telah mencapai batas minimum, maka consumer akan tidur (memanggil aksi sleep).
   f. Apabila producer meletakkan item ke market dan consumer dalam keadaan sleep, maka producer akan membangunkan (wake-up) consumer.

   Keenam poin di atas merupakan inti dari simulasi Producer-Consumer Problem. Pencegahan kondisi deadlock (producer ingin meletakkan item ke market sedangkan market telah penuh atau consumer ingin mengambil item dari market

sedangkan market telah kosong) dihindari dengan metode sleep dan wake up. Masing-masing variabel akan memanggil aksi sleep untuk menghindari kondisi deadlock dan akan dibangunkan variabel lainnya, ketika keadaan sudah tidak menyebabkan deadlock.

Critical section dalam perangkat lunak simulasi ini adalah ketika producer ingin meletakkan item ke dalam market dan consumer ingin mengambil item dari market. Dalam hal ini, semaphore digunakan untuk mengatur producer dan consumer supaya tidak meletakkan atau mengambil item dari market atau mengakses critical section secara bersamaan. Ketika salah satu producer atau consumer meletakkan atau mengambil item, maka variabel lain di-blocked agar tidak memasuki critical section, sehingga jumlah item di dalam market tidak diakses secara bersamaan, nilainya tetap konsisten dan terjaga kebenarannya.

Proses simulasi yang dirancang dalam perangkat lunak menggunakan komponen timer yang terdapat dalam bahasa pemograman Microsoft Visual Basic 6.0. Fungsi dari komponen timer adalah untuk memeriksa keadaan suatu objek setiap satu interval waktu yang diatur kepadanya. Timer akan memeriksa dan memajukan keadaan objek ke keadaan berikutnya, sehingga terbentuk proses animasi. Misalkan, keadaan sekarang adalah producer berjalan ke bawah dengan kaki kiri, maka pada keadaan berikutnya, timer akan mengubah gambar tersebut dengan gambar producer berjalan ke bawah dengan kaki kanan. Demikian seterusnya.

5. Kesimpulan dan Saran

Setelah menyelesaikan perangkat lunak simulasi Producer-Consumer Problem, penulis menarik kesimpulan sebagai berikut:

1. Perangkat lunak menggunakan metode sleep and wake-up untuk mencegah masalah yang terjadi ketika buffer penuh, sementara producer ingin meletakkan item ke buffer dan consumer ingin mengambil item sementara buffer telah kosong.
2. Perangkat lunak menggunakan semaphore untuk untuk mem-blocked producer atau consumer lain ketika salah satu producer atau consumer sedang berada dalam buffer.
3. Perangkat lunak simulasi Producer-Consumer ini merupakan illustrasi dari proses sinkronisasi, yaitu bagaimana cara mengatur beberapa proses yang mengakses beberapa variabel secara bersamaan.

Referensi

[1] Silberschatz, A., Peterson, L.J, Operating System Concepts, Alternate Edition, Addition-Wesley Publishing Company, Inc., Juni 1978.
[2] Hariyanto, B., MT, Sistem Operasi, Edisi2, Informatika, Bandung, 1999.
[3] Hariyanto, B., MT, Sistem Operasi Lanjut, Informatika, Bandung, 2003.
[4] Kusumadewi, S., Sistem Operasi, Edisi2, Graha Ilmu, Jakarta, 2000.
[5] Hadi, R., Pemrograman Microsoft Visual Basic dengan menggunakan Windows API, PT. Elex Media Komputindo, Jakarta, 2001.
[6] Suryokusumo, A., Microsoft Visual Basic 6.0, PT. Elex Media Komputindo, 2001.
[7] Supardi, Y., Microsoft Visual Basic 6.0 Untuk Segala Tingkat, PT. Elex Media Komputindo, Jakarta, 2006.
[8] www.cs.umd.edu/~shankar/412-Notes/12-BoundedBufferProdCons.html (tanggal akses: 22 Agustus 2006).
[9] cis.poly.edu/muller/CS623/consumer_producer.htm (tanggal akses: 22 Agustus 2006).

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 24-28
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Perancangan Perangkat Lunak Pembelajaran Algoritma Hamming Code dalam Mencari Bit Error pada Komunikasi Data

Misdem Sembiring[1], Fauzi Haris Simbolon[2]

[1,2]AMIK Medan Business Polytechnic
Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia - 20155

[1]misdem@amikmbp.ac.id, [2]farisboys@amikmbp.ac.id

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

Abstrak

Dalam ilmu komputer terdapat bermacam-macam algoritma dalam mendeteksi serta mengoreksi error ketika transmisi data digital berlangsung. Salah satu algoritma untuk mendeteksi error adalah dengan menggunanakan Hamming Code dengan single error correntionn. Metode ini sangat cocok digunakan pada situasi dimana terdapat beberapa error yang teracak (randomly occurring errors). Metode Hamming Code menyisipkan (n + 1) check bit ke dalam 2n data bit. Metode ini menggunakan operasi XOR (Exclusive - OR) dalam proses pendeteksian error, dengan nilai Input dan output data berupa bilangan biner. Peneliti bermaksud untuk merancang perangkat lunak pembelajaran yang mampu menjelaskan teknik pendeteksian error dari algoritma Hamming Code. Panjang data input dan output pada algortima Hamming Code minimal harus sama dengan 4 bit. Atau dengan perkataan lain, algortima Hamming Code tidak mendukung data dengan panjang 2 bit. Algortima Hamming Code tidak mampu melakukan pengecekan terhadap posisi data error (bad bit) yang lebih dari satu buah.

Kata Kunci: Hamming Code, Perancangan Perangkat Lunak, Komunikasi Data, Bit Error.

1. Pendahuluan

Ketika transmisi data digital berlangsung, terkadang data yang ditransfer dapat mengalami kegagalan (Error). Error yang terjadi dapat mengakibatkan perubahan isi dari data yang ditransfer. Dalam ilmu komputer terdapat bermacam-macam algoritma dalam mendeteksi serta mengoreksi error tersebut. Salah satu algoritma untuk mendeteksi error adalah dengan menggunanakan Hamming Code dengan single error correntionn.

Hamming Code merupakan algoritma pendeteksi error yang mampu mendeteksi beberapa error, namun hanya mampu mengoreksi satu error (single error correntionn). Algoritma ini sangat cocok digunakan pada situasi dimana terdapat beberapa error yang teracak (randomly occurring errors). Algoritma Hamming Code menyisipkan (n + 1) check bit ke dalam 2n data bit. Algoritma ini menggunakan operasi XOR (Exclusive - OR) dalam proses pendeteksian error, dengan nilai input dan output data berupa bilangan biner.

Pendeteksian error mulai dari awal hingga akhir dalam algoritma Hamming Code berjalan dengan sangat cepat sehingga sulit untuk dijelaskan. Agar dapat mempelajari proses pendeteksian error algoritma Hamming Code secara tahap demi tahap peneliti bermaksud untuk merancang perangkat lunak pembelajaran yang mampu menjelaskan teknik pendeteksian error dari algoritma Hamming Code.

2. Tinjauan Literatur

Peneliti tidak luput meninjau penelitian-penelitian yang relevan agar dapat menyelesaikan penelitian ini dengan baik.

Dalam peninjaun akan penelitian yang dilakukan oleh Maulana et al (2014) yang berjudul Perancangan Simulasi Pengkodean Hamming (7, 4) untuk Menghitung Bit Error Rate (BER) pada Binary Symentic Channel. Tujuan peneliti untuk merancang sebuah aplikasi simulasi menggunakan bahasa pemrograman Java untuk mengambarkan proses pengkoreksian error pada pengiriman data berupa angka dalam bentuk integer dan akan diubah menjadi biner untuk memudahkan dalam proses pengecekan error. Algoritma yang digunakan proses pengecekan error pada aplikasi simulasi adalah Hammiing Code. Dari hasil pengamatan aplikasi ini, data dikirimkan akan di deteksi jika terjadi kesalahan maka aplikasi akan mengkoreksi kesalahan yang telah terdeteksi. Hasil pengamatan menunjukan bahwa error terjadi pada saat pengiriman data di karenakan kesalahan pada bit-bit yang dikirimkan, maka terjadilah error. Dan juga diharapkan dengan simulasi yang dibuat ini

e-ISSN: 2798-9593
p-ISSN: 2798-9836

bisa membantu dalam memahami tentang proses pengiriman data dan bagaimana pengkoreksian error tersebut. Kesimpulan yang didapatkan dari penelitian ini bahwa aplikasi mensimulasikan proses pengkodean dengan Kode Hamming (7,4) dan menghitung bit error rate dengan menggunakan Binary Symetric Channel (BSC) dan pengkodean Hamming (7,4) dapat mendeteksi tepat satu error, dan pengkodean Hamming dapat memperkecil bit error rate dalam sistem komunikasi digital.

Tinjauan pada penelitian Andanal (2018) dengan judul Implementasi Deteksi dan Koreksi Error Pada Komunikasi Serial Arduino Berbasis UART Dengan Metode Hamming Code, peneliti menggunakan metode Hamming Code yang diterapkan pada Arduino dan komunikasi UART. Berdasarkan hasil pengujian, algoritma Hamming Code dapat melakukan proses encode dan decode data, serta dapat melakukan deteksi dan koreksi error pada data yang mengalami error dalam proses pengujian. Rata-rata delay yang didapatkan berjumlah 102,7ms untuk data 5 bit dan 109,5ms untuk data 4 bit pada proses encode. Serta 17,5 ms untuk data 10 bit dan 100,1ms untuk data 11 bit pada prose decode. Faktor pengambilan data suhu serta jumlah bit yang dilakukan proses encode dan decode sangat mempengaruhi proses encode dan decode menggunakan algoritma Hamming Code.

Dari peninjauan penelitian-penelitian yang telah dipaparkan diatas, peneliti akan memaparkan beberapa landasan teori yang digunakan dalam penelitian ini, antara lain:

2.1. Error Detection

Pada saat data berada dalam transmission system terdapat kemungkinan data terkorupsi (data error). Data error tersebut akan diperbaiki oleh receiver melalui proses error detection dan error correction. Proses error detection dilakukan oleh transmitter dengan cara menambahkan beberapa bit tambahan ke dalam data yang akan ditransmisikan. Proses error detection dan correction ini sering digunakan pada CD Players, High speed modem, dan telepon selular (cellular phones).

Secara garis besar, metoda pendeteksi error (error detection) dan pengontrol error (error controller) dapat dibagi menjadi 2 bagian besar dengan perincian sebagai berikut:

1. Error Detection and Correction.
2. Error Controller, yaitu: Automatic Repeat Request (ARQ).

2.2. Hamming Code

Algoritma Hamming Code ditemukan oleh Richard W. Hamming pada tahun 1940-an. Algoritma Hamming Code merupakan salah satu metoda pendeteksi error (error detection) yang mampu untuk mendeteksi beberapa error, namun hanya mampu mengoreksi satu error (single error correction). Algoritma pendeteksi error ini sangat cocok digunakan pada situasi dimana terdapat beberapa error yang teracak (randomly occuring errors).

Algoritma Hamming Code merupakan salah satu algoritma pendeteksi error (error detection) dan pengoreksi error (error correction) yang paling sederhana. Algoritma ini menggunakan operasi logika XOR (Exclusive – OR) dalam proses pendeteksian error (error detection) maupun proses pengoreksian error (error correction), sedangkan input dan output data dari algoritma Hamming Code berupa bilangan biner.

1. Operasi Logika
   Operasi – operasi logika dasar terdiri dari operasi NOT, operasi AND, dan operasi OR. Operasi logika lain merupakan kombinasi dari operasi - operasi logika dasar ini. Salah satu operasi logika hasil kombinasi operasi logika dasar adalah operasi XOR (Exclusive - OR).
2. Cara Kerja Algoritma Hamming Code
   Algoritma Hamming Code menyisipkan beberapa buah check bit ke dalam data. Jumlah check bit yang disisipkan tergantung pada panjang data. Rumus perhitungan untuk menghitung jumlah check bit yang harus disisipkan ke dalam data adalah sebagai berikut.

Untuk data 2n bit, jumlah check bit yang disisipkan ada sebanyak $c = (n + 1)$ bit.

3. Metodologi Penelitian

Lokasi dari penelitian ini berada di transmisi komunikasi data.

3.1. Pengumpulan Data

Untuk menyelesaikan penelitian ini, peneliti mengumpulkan data-data yang dibutuhkan dari berbagai sumber dengan beberapa metode, antara lain:

1. Studi lapangan
   Degan metode ini peneliti mengamati bagaimana data dapat berpindah dengan dari satu tempat ke tempat lain di perangkat komputer yang terhunung dengan internet. Peneliti juga mengamati data-data

LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 24-28
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

yang berpindah kenapa mengalami corrupt dan harus dilakukan ulang penpindahan data.

2. Studi Kepustakaan
   Untuk memperdalam pengetahuan peneliti akan permasalah yang sedang diteliti. Peneliti melakukan studi pustaka yang berhubungan dengan komunikasi data. Pengetahuan yang didapatkan berupa teknik pendeteksian error dari algortima-algorima dalam mentransmisi data khusus Hamming Code serta hal-hal lain yang berhubungan dengan jaringan internet.

3.2. Metode Analisis Data

Dalam menyelesaikan penelitian ini yang bersifat deduktif dimana menganalisa data dengan cara mengambil kesimpulan berdasarkan teori algoritma Hamming Code yang telah diterima sebagai suatu kebenaran umum mengenai fakta yang diamati.

4. Hasil dan Pembahasan

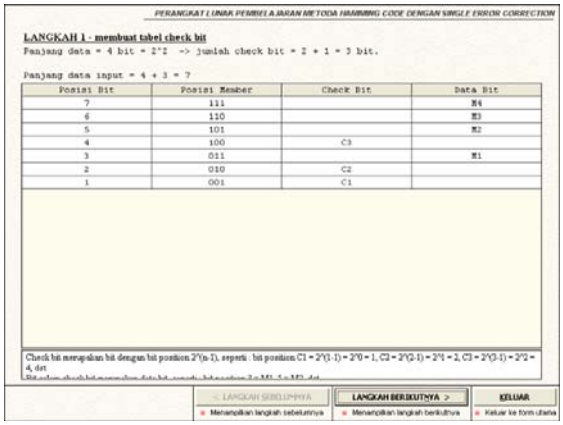
Hasil dari penelitian akan dipaparkan disini dan juga tentang pembahasannya:

4.1. Hasil

Hasil dari penelitian ini berupa perangkat lunak yang dapat menjelaskan teknik pendeteksian error dari algoritma Hamming Code.

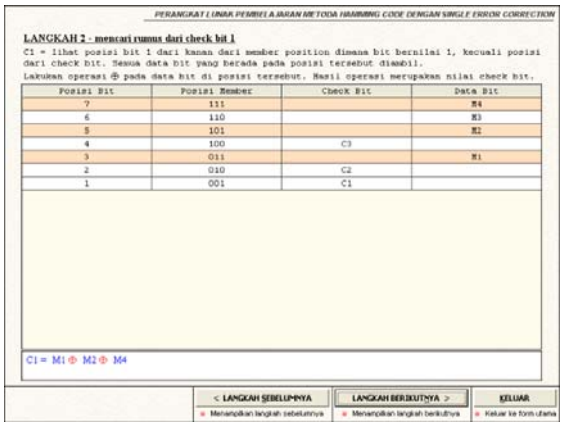
Misalkan panjang data input dan output = 4 bit, data Input = 1001, maka , langkah-langkah pendeteksian single error dengan Hamming Code adalah sebagai berikut:

1. Membuat tabel check bit



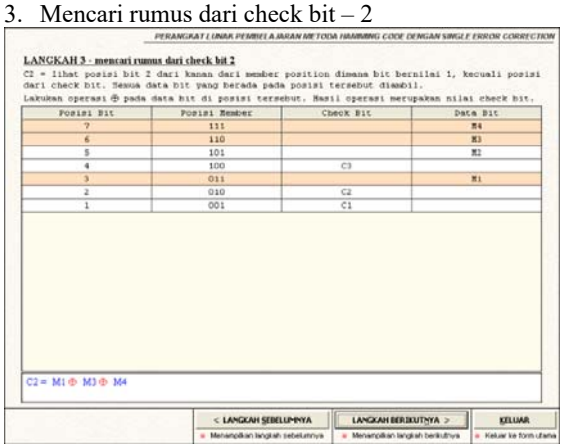
Gbr. 1. Membuat tabel check bit untuk data Input = 1001, data Output = 1000 (panjang data = 4 bit)

2. Mencari rumus dari check bit – 1



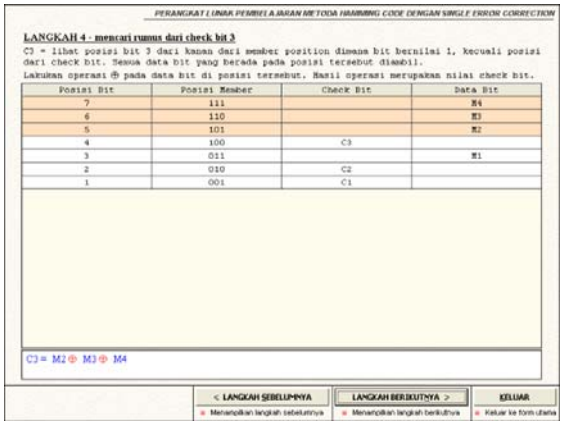
Gbr. 2. Mencari rumus dari check bit - 1 untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

3. Mencari rumus dari check bit – 2



Gbr. 3. Mencari rumus dari check bit - 2 untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

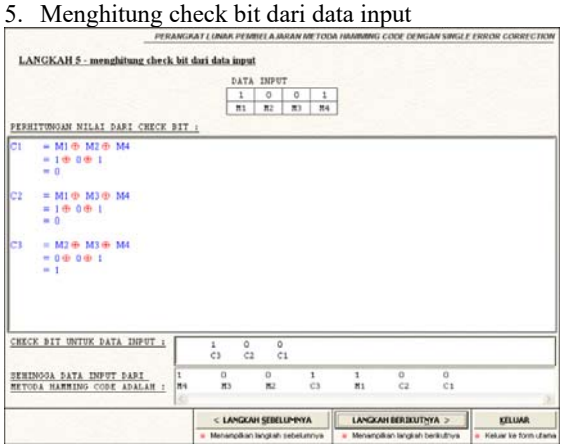
4. Mencari rumus dari check bit-3



Gbr. 4. Mencari rumus dari check bit - 3 untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

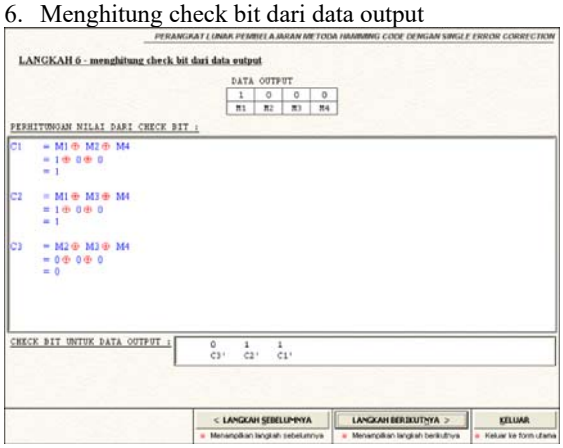
LOFIAN: Jurnal Teknologi Informasi dan Komunikasi
Vol. 1, No. 1, September 2021, Halaman 24-28
Submited: 02 Agustus 2021; Accepted: 18 September 2021; Published: 20 September 2021
e-ISSN: 2798-9593
p-ISSN: 2798-9836

5. Menghitung check bit dari data input



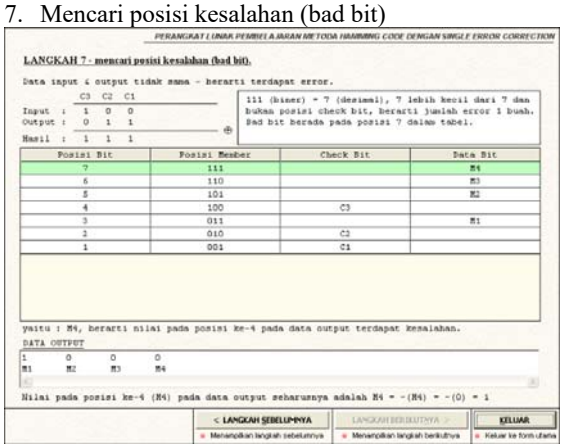
Gbr. 5. Menghitung check bit dari data input untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

6. Menghitung check bit dari data output



Gbr. 6. Menghitung check bit dari data output untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

7. Mencari posisi kesalahan (bad bit)



Gbr. 7. Mencari posisi kesalahan (bad bit) untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

4.2. Pembahasan

Algoritma Hamming Code melakukan proses pengecekan error dengan cara menyisipkan n – 1 check bit untuk 2n bit data (input dan output). Lalu dilakukan perhitungan nilai dari check bit tersebut untuk data input dan data output. Setelah itu, nilai check bit tersebut di-XOR-kan dan dilakukan perbandingan antara nilai check bit input dan nilai check bit output. Apabila nilainya tidak sama, maka terdapat error (kesalahan). Sebaliknya, apabila nilainya sama maka tidak terdapat error dalam data output.

Error tersebut dapat terjadi karena adanya gangguan pada saat transmisi data. Error yang muncul dapat berupa single error (error tunggal) yaitu error yang terjadi hanya pada 1bit data saja, ataupun double error (error ganda) yaitu error yang terjadi pada beberapa bit data. Metoda Hamming Code hanya dapat digunakan untuk mendeteksi dan mengatasi single error. Sedangkan untuk double error, metoda Hamming Code hanya dapat mendeteksinya, namun tidak mampu mengatasinya, karena tidak dapat mengetahui posisi bit error (bad bit). Keunggulan dari metoda Hamming Code hanya terletak pada cara kerjanya cukup sederhana (simple).

5. Kesimpulan dan Saran

Dari hasil penelitian dan pembahasannya, maka dapat disimpulkan dan diberikan saran sebagai berikut:

5.1. Kesimpulan

Setelah menyelesaikan perancangan perangkat lunak pembelajaran Hamming Code dengan Single Error Correction, maka peneliti dapat menarik kesimpulan sebagai berikut:

1. Algortima Hamming Code hanya mampu melakukan pengoreksian terhadap satu buah error (single error correction) saja.
2. Data input dan output pada algortima Hamming Code harus berupa hasil dari perpangkatan 2n dengan n harus lebih besar dari 1.
3. Panjang data input dan output pada algortima Hamming Code minimal harus sama dengan 4 bit. Atau dengan perkataan lain, algortima Hamming Code tidak mendukung data dengan panjang 2 bit.

e-ISSN: 2798-9593
p-ISSN: 2798-9836

5.2. Saran

Peneliti ingin memberikan beberapa saran untuk pengembangan lebih lanjut pada perancangan perangkat lunak pembelajaran metoda pendeteksi dan pengoreksi kesalahan (error detection and correction). Saran-saran tersebut antara lain:

1. Dapat dikembangkan suatu perangkat lunak dengan menggunakan metoda pendeteksi dan pengoreksi error (error detection and correction method) lainnya yang lebih canggih daripada metoda Hamming Code seperti metoda CRC (Cyclic Redundancy Check), gabungan metoda LRC (Longitudinal Redundancy Check) dan VRC (Vertical Redundancy Check) yang mampu melakukan pengecekan terhadap lebih dari satu buah error dan sebagainya.
2. Perangkat lunak dapat ditambahkan beberapa animasi lainnya agar lebih menarik.

Perangkat lunak juga dapat dikembangkan menjadi sebuah perangkat lunak multimedia dengan menambahkan efek suara, efek grafik dan efek-efek lainnya

Referensi

[1] Maulana, J., Arini, & Fahrianto, F. (2014). Perancangan Simulasi Pengkodean Hamming (7, 4) untuk Menghitung Bit Error Rate (BER) pada Binary Symentic Channel. Jurnal Teknik Informatika, 7(2), 24-34.
[2] Andana1, A. F., Akbar, S. R., & Maulana, R. (2018). Implementasi Deteksi Dan Koreksi Error Pada Komunikasi Serial Arduino Berbasis UART Dengan Metode Hamming Code. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(11), 5089-5095.
[3] Green, D. C. (2000). Komunikasi Data. Yokyakarta: Andi.
[4] Ralph J. S. (1990). Rangkaian, Piranti dan Sistem, Edisi Keempat, Jilid 1. Jakarta: Penerbit Erlangga.
[5] Stallings, W. (2001). Dasar-dasar Komunikasi Data. Jakarta: Penerbit Salemba Teknika.
[6] Malvino, Albert P., & Tjia M, O. (1994). Elektronika Komputer Digital, Edisi Kedua. Jakarta: Penerbit Erlangga.
[7] Pramono, D. (2002). Mudah menguasai Visual Basic 6. Jakarta: PT. Elex Media Komputindo.
[8] Rahadian, H. (2001). Pemrograman Microsoft Visual Basic 6.0. Jakarta: PT. Elex Media Komputindo.

ISSN 2798-9836