ISSN : 2085-4218

# Penyembunyian Citra Pada Sinyal Audio Berdasarkan Pendekatan Modifikasi OFDM

**Timotius Florean[1,*], Gelar Budiman[1], Ledya Novamizanti[1]**
1 Universitas Telkom, Fakultas Teknik Elektro
* E-mail : timytimothy@gmail.com

**Abstrak.** Dewasa ini, dimana perkembangan internet sudah semakin cepat, informasi sudah sangat mudah untuk didapatkan. Seiring dengan mudahnya penyebaran informasi saat ini, audio *watermarking* mulai banyak dikembangkan untuk menyisipkan informasi, yang umumnya berupa hak cipta atau identitas kepemilikan, ke dalam suatu file audio. Dianatara berbagai informasi yang dapat disisipkan, sebagian besar menggunakan teks sebagai informasi sisipan karena keterbatasan kapasitas penyisipan dari algoritma yang digunakan. Dalam penelitian ini, akan dirancang simulai audio *watermarking* menggunakan metode modifikasi OFDM dan modulator MPSK. Dengan parameter jump dan arch yang tepat, tidak hanya kapasitas dari sistem *watermarking* akan menjadi lebih besar tetapi juga tetap mempertahankan kualitas dari citra ekstraksi tersebut. Dengan menggunakan metode ini, kapasitas penyisipan mencapai 1475 data atau sekitar 11800 bit dalam satu detiknya dan juga PSNR dari citra hasil ekstraksi yang bernilai 20 dB hingga 35 dB.

**Kata Kunci:** Audio *Watermarking*, OFDM, *High Payload*, Perlindungan Hak Cipta

## 1. Pendahuluan

Dewasa ini, dimana teknologi sudah berkembang secara pesat, berbagai informasi sangat mudah untuk didapatkan melalui internet. Penyebaran informasi yang tanpa hambatan dan batasan itu menjadi suatu masalah tersendiri terhadap suatu karya yang memiliki hak cipta yang dapat tersebar luas, salah satunya melalui media sosial. Dengan adanya masalah seperti itu, diperlukan suatu cara agar suatu hak cipta pada konten digital bisa selalu ada dan menempel secara permanen. *Watermarking* adalah salah satu teknik yang dapat digunakan untuk mengatasi masalah tersebut. Dengan *watermarking* kita dapat menyisipkan suatu informasi, identitas ataupun hak cipta, ke dalam konten digital tanpa mengganggu kualitas dari konten digital tersebut. Oleh karena itu, algoritma *watermarking* haruslah dibuat sedemikian sehingga perbedaan dari sinyal asli dan sinyal yang telah disisipi (*watermarked*) tidak bisa dibedakan oleh indra manusia serta tahan terhadap gangguan yang ada.

Audio *watermarking* merupakan suatu teknik untuk menyisipkan suatu informasi ke dalam file audio tanpa mengganggu kualitas dari audio tersebut. Dianatara berbagai informasi yang dapat disembunyikan, sebagian besar menggunakan teks sebagai informasi sisipan karena keterbatasan kapasitas penyisipan dari algoritma yang digunakan. Disisi lain, jika algoritma tersebut tidak tahan terhadap gangguan yang mengakibatkan satu atau lebih karakter sisipan terkena distorsi maka informasi tersebut akan susah untuk dikenali. Lain halnya jika informasi sisipan berupa gambar yang lebih mudah untuk dikenali ketika satu atau lebih piksel pada gambar tersebut terkena distorsi.

Pada penelitian [1] telah didapatkan sebuah skema untuk mengatasi kapasitas pada penyembunyian informasi menggunakan pendekatan OFDM. Dengan menggunakan teknik tersebut maka kapasitas penyisipan akan menjadi lebih besar dibandingkan dengan skema yang sudah ada. Pada penelitian ini dilakukan perancangan simulasi audio *watermarking* menggunakan pendekatan OFDM sehingga informasi yang disisipkan pun tidak terbatas oleh teks saja tetapi dapat disisipkan sebuah citra abu (*grayscale*) yang merupakan logo dari si pemilik konten digital.

ISSN : 2085-4218

## 2. Teori dan Tahap Perancangan

### *2.1 OFDM dan Modulasi MPSK*

*Orthogonal Frequency Division Multiplexing* (OFDM) merupakan teknologi modulasi dan *multiplexing* yang sering digunakan saat ini. Dalam literatur, OFDM disebut sebagai *Multi-carrier*, *Multi-tone and Fourier Transform*. Konsep OFDM adalah menyebarkan data yang akan ditransmisikan ke dalam sejumlah besar *carrier* [2]. Seriap *carrier* dibuat orthogonal terhadap yang lain dengan menggunakan Transformasi Fourier. Pada transmitter, serial data stream di-*demultiplexing* ke dalam N parallel data stream lalu dipetakan ke dalam simbol menggunakan konstelasi. Untuk sinyal audio, pendengaran manusia kurang sensitif terhadap perubahan pada komponen fasa dibandingkan dengan komponen *magnitude*nya. Karena itu pada penelitian ini digunakan MPSK sebagai modulatornya.

*M-order Phase Shift Keying* (MPSK) adalah sebuah skema modulasi digital yang akan mengubah atau memodulasi fasa dari sinyal referensi. Sebuah cara untuk merepresentasikan modulasi MPSK adalah dengan menggunakan diagram konstelasi. Pada MPSK, titik-titik konstelasi dipisahkan dengan jarak yang sama.

### *2.2 Parameter Jump dan Arch*

Parameter jump [1] mewakili pemisahan dari setiap komponen fasa yang akan disisipkan, sebagai contoh, jika jump bernilai 3 dan komponen fasa yang akan dimodifikasi dimulai dari komponen ke seratus, dilambagkan sebagai Fasa(100), maka sampel berikutnya yang akan dimodifikasi adalah Fasa(103), Fasa(106), Fasa(109), dll, sampai seluruh spektrum frekuensi ditutupi.

Sedangkan parameter arch [1] merepresentasikan ruang yang tersedia. Sebagai contoh, jika arch = $5^o$ dan komponen fasa yang akan disisipi memiliki fasa $10^o$ berarti bahwa ruang konstelasi yang tersedia adalah antara $-5^o$ dan $5^o$ dari $10^o$ yaitu berkisar $5^o$ sampai $15^o$.

### *2.3 Segmentasi Audio dan Penyisipan Informasi*

Pada tahap ini *host* audio, yang berekstensi WAV dengan frekuensi sampling 44100 Hz, akan disegmentasi ke dalam beberapa segmen dengan jumlah 4096 data setiap segmennya [1] dan setiap segmen akan dilakukan transformasi ke dalam domain frekuensi dengan menggunakan algoritma FFT. Pada citra abu akan dilakukan proses *preprocessing* dimana citra yang asalnya berupa matriks akan ditransformasi ke dalam vektor. Setelah itu vektor tersebut akan ditambahkan *header* yang berupa pola khusus, sebagai penanda mulainya penyisipan, dan juga ukuran dari citra itu sendiri.

Setelah itu akan dilakukan penyisipan ke dalam setiap segmen. Proses penyisipan hanya dilakukan pada komponen fasa dari setiap segmen. Penyisipan pada setiap titik pada komponen fasa dilakukan dengan cara sebagai berikut:

1. Titik pertama yang akan disisipkan pada setiap segmen adalah titik ke 120 yang berupakan korespondensi dengan frekuensi 1.3 KHz [1]
2. Ambil fasa pada titik tersebut (Pj) lalu tambahkan dengan parameter arch untuk menentukan titik pertama dari konstelasi MPSK. Setelah itu tentukan fasa baru (Pj') yang merupakan hasil *mapping* dari data informasi ke dalam fasa pada titik tersebut dengan persamaan sebagai berikut:

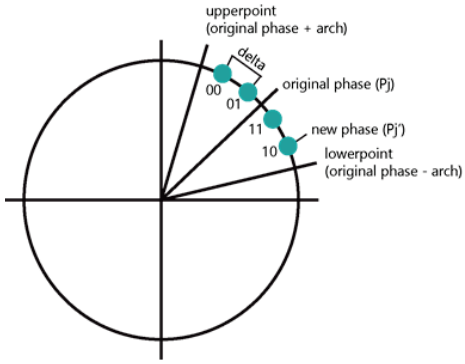
$$P_j' = upperpoint - (delta * mappingpoint) \quad (1)$$

$$upperpoint = P_j + arch \quad (2)$$

$$delta = 2 * \frac{arch}{M} \quad (3)$$

mappingpoint merupakan titik ke-n dari konstelasi MPSK yang mewakili data yang akan disisipkan. Sebagai contoh jika data yang akan disisipi adalah 2 (dalam desimal) dan menggunakan MPSK dengan M = 4, maka data akan di*mapping* pada titik konstelasi ke 4 (mappingpoint).

ISSN : 2085-4218




Gambar 12. Penentuan fasa baru

3. Titik FFT baru didapatkan dengan mengalikan *magnitude* pada titik tersebut dengan fasa baru yang telah diperoleh pada persamaan di bawah ini

$$newFFTpoint = originalMagnitude * \exp(i * P_j') \quad (4)$$

Pada penelitian ini, sistem *watermarking* yang diusulkan menggunakan skema *blind*, yaitu audio asli tidak diperlukan dalam proses ekstraksi data, sehingga jumlah kapasitas berkurang menjadi setengahnya. Proses penyisipan dengan skema *blind* dilakukan sebagai berikut [1]:

1. Proses penyisipan dilakukan pada komponen fasa ($P_j$') untuk setengah kanal yang ada
2. Hitung *interpolation-error* $e_j$ menggunakan fasa asli ($P_j$) dan fasa yang telah diubah ($P_j$') dengan rumus $e_j$ sebagai berikut: $e_j = P_j - P_j'$ (5)
3. Tambahkan informasi *interpolation-error* yang telah didapatkan pada kanal yang bersebelahan sebagai berikut: $P_{j+jump}$ = round($P_{j+jump}$) + $\frac{e_j}{500}$ (6)
   dimana *round()* merupakan fungsi pembulatan ke bilangan bulat terdekat
4. Penyisipan data akan dilakukan kembali pada titik setelah informasi $e_j$ disisipkan, j = j + 2*jump
5. Proses penyisipan data ke dalam fasa (Pj) dilakukan sesuai dengan yang telah dijelaskan sebelumnya

Ketika semua data sudah disisipkan, transformasikan kembali setiap segmen ke dalam domain waktu dengan menggunakan *inverse* FFT.

*2.4 Pengambilan Sisipan Informasi*

Proses ekstraksi dilakukan dengan cara mensegmentasi file audio dengan ukuran 4096 sampel. Setiap segmen ditransformasikan ke dalam domain frekuensi dengan menggunakan FFT. Setelah mendapatkan komponen frekuensinya lakukan tahap berikut [1]:

1. Titik pertama yang akan dideteksi dari setiap segmen adalah titik ke 120
2. Tentukan fasa asli ($P_j$)dengan menghitung *interpolation-error* ($e_j$) sebagai berikut:
   $$e_j = 500 * (P_{j+jump} - round(P_{j+jump})) \quad (7)$$
3. Setelah mendapatkan nilai $P_j$, tentukan titik pertama konstelasi MPSK dengan:
   $$upperpoint = P_j + arch \quad (8)$$
4. Tentukan nilai n dari Pj'
   $$n = \frac{(upperpoint - P_j')}{delta} \quad (9)$$
5. Nilai informasi dari titik tersebut adalah simbol dari titik ke-n pada konstelasi MPSK

Setelah didapatkan semua simbol informasi, selanjutnya akan dicari pola khusus dari seluruh simbol yang didapatkan. Pola khusus tersebut merupakan tanda yang merupakan sisipan pertama. Setelah mendapatkan letak dimana pola khusus tersebut berada, bentuk simbol tersebut ke dalam matriks. Data hasil ekstraksi yang akan diambil merupakan data dengan nilai kemunculan terbanyak. Setelah didapatkan data ekstraksi, bentuk kembali citra abu dengan ukuran yang terdapat pada data pertama dan data kedua.

ISSN : 2085-4218

## 3. Hasil dan pengujian

*3.1 Minimum Pengulangan Sisipan*

Pada tahap ini yang pertama kali dilakukan adalah menghitung jumlah minimum pengulangan sisipan ke dalam file audio. Proses rekonstruksi citra pada proses ekstraksi dilakukan dengan pengambilan keputusan berdasarkan banyaknya kemunculan dari suatu nilai. Sehingga untuk mendapatkan citra hasil ekstraksi dengan kualitas yang baik, diperlukan proses pengulangan penyisipan citra ke dalam audio karena jika semakin banyak pengulangan yang terjadi maka *error* bisa diminimalisasi pada saat proses rekonstruksi. Panjang dari suatu audio pun akan mempengaruhi banyaknya pengulangan yang bisa dilakukan maka dari itu pada tahap ini akan dilakukan análisis pengaruh banyaknya pengulangan terhadap nilai MSE dan PSNR yang didapatkan. Berikut ini merupakan hasil dari análisis tersebut:

**Tabel 7.** Nilai MSE dan PSNR terhadap banyaknya pengulangan

| Pengulangan | MSE | PSNR | Pengulangan | MSE | PSNR |
|---|---|---|---|---|---|
| 1 | 1587.583134 | 16.12343884 | 13 | 141.7329851 | 26.61609427 |
| 2 | 1564.138358 | 16.18805194 | 14 | 115.309403 | 27.51215637 |
| 3 | 1468.024328 | 16.46347108 | 15 | 103.1247761 | 27.99717342 |
| 4 | 1339.79597 | 16.86041694 | 16 | 92.92910448 | 28.44928609 |
| 5 | 1180.213284 | 17.41119862 | 17 | 79.20253731 | 29.14341266 |
| 6 | 985.9279104 | 18.192352 | 18 | 70.69970149 | 29.63662781 |
| 7 | 761.0195522 | 19.31684546 | 19 | 60.50522388 | 30.31287489 |
| 8 | 536.601194 | 20.83428726 | 20 | 59.13089552 | 30.41265905 |
| 9 | 411.3373134 | 21.98882254 | 21 | 59.29686567 | 30.40048623 |
| 10 | 289.2310448 | 23.51835455 | 22 | 57.40641791 | 30.54119913 |
| 11 | 218.1923881 | 24.74240765 | 23 | 57.25462687 | 30.55269772 |
| 12 | 178.8043284 | 25.60702333 | 24 | 60.10283582 | 30.34185397 |

Dari hasil tersebut dapat dilihat bahwa mulai pengulangan ke 19 sampai 24 nilai PSNR stabil pada posisi 30 dB. Sedangkan citra hasil ekstraksi pada pengulangan ke 20 pun sudah sangat baik dan dapat dikenali. Maka dari itu 20 merupakan nilai pengulangan minimal pada proses penyisipan sehingga hasil ekstraksi citra yang nantinya akan diperoleh tidak terlalu buruk.

*3.2 Ukuran Maksimal Citra Terhadap Panjang Audio*

Pada tahap ini akan dilakukan perhitungan ukuran maksimum dari suatu citra yang akan disisipkan terhadap panjang dari *host* audio yang ada. Pada tahap sebelumnya telah didapatkan bahwa pengulangan penyisipan citra ke dalam host audio dilakukan minimal 20 kali sehingga pada tahap ini perhitungan akan melibatkan nilai pengulangan dari hasil yang telah didapatkan sebelumnya. Banyaknya data yang bisa disisipi untuk setiap segmen adalah:

$$data\_satu\_segmen = \frac{\left(\frac{4096}{2}\right) - 120}{jump * 2} \quad (10)$$

Ketika sudah diketahui banyaknya data sisipan pada satu segmen, maka banyaknya sisipan dalam audio dengan panjang y detik adalah banyaknya jumlah segmen yang dihasilkan dikali dengan banyak data yang dapat disisipi dalam satu segmen. Tetapi jumlah tersebut belum termasuk pengulangan sehingga totalnya harus dibagi dengan 20.

$$jumlah\_data\_sisipan = \frac{data\_satu\_segmen * \frac{44100}{4096}}{20} \quad (11)$$

Dari hasil tersebut maka ukuran maksimal citra (dalam n*n piksel) yang dapat disisipi adalah:

$$n = \sqrt{jumlah\_data\_sisipan} \quad (12)$$

Berikut ini adalah ukuran maksimal dari citra ketika parameter jump bernilai 7.

ISSN : 2085-4218

Tabel 8. Ukuran maksimal citra dengan panjang audio tertentu

| Panjang audio (y) dalam detik | jumlah data sisipan | Ukuran maksimal citra |
|---|---|---|
| 20 | 1475 | 38 |
| 40 | 2950 | 54 |
| 60 | 4425 | 66 |
| 80 | 5900 | 76 |
| 100 | 7375 | 85 |
| 120 | 8850 | 94 |
| 140 | 10325 | 101 |
| 160 | 11800 | 108 |

| Panjang audio (y) dalam detik | jumlah data sisipan | Ukuran maksimal citra |
|---|---|---|
| 180 | 13275 | 115 |
| 200 | 14750 | 121 |
| 220 | 16225 | 127 |
| 240 | 17700 | 133 |
| 260 | 19175 | 138 |
| 280 | 20650 | 143 |
| 300 | 22125 | 148 |

3.3 Nilai MSE dan PSNR

Analisis dilakukan dengan menyisipkan empat gambar ke dalam empat audio berbeda (satu audio dengan satu gambar sisipan) dengan masing-masing audio diberi parameter jump dengan nilai 2, 7, dan 12 serta parameter arch dengan nilai 25, 30, dan 35. Setelah masing-masing audio disisipi oleh masing-masing gambar dan dengan parameter yang telah ditentukan, maka akan terbentuk 36 file audio yang telah tersisipi citra abu. Kemudian informasi dalam 36 file audio tersebut akan diekstraksi dan dibandingkan dengan dengan citra aslinya. Dari sini dapat dihitung nilai MSE dan PSNR dari hasil proses ekstraksi.

Tabel 9. Nilai MSE dan PSNR terhadap perubahan parameter jump dan arch

| Gambar 1 – Audio 1 | | | |
|---|---|---|---|
| Jump | Arch | MSE | PSNR |
| 2 | 25 | 57.40328 | 30.54144 |
| 2 | 30 | 51.76045 | 30.99082 |
| 2 | 35 | 25.4909 | 34.06695 |
| 7 | 25 | 115.6606 | 27.49895 |
| 7 | 30 | 90.59433 | 28.55979 |
| 7 | 35 | 67.65821 | 29.8276 |
| 12 | 25 | 236.7122 | 24.3886 |
| 12 | 30 | 155.7133 | 26.20755 |
| 12 | 35 | 120.3637 | 27.32585 |

| Gambar 2 – Audio 2 | | | |
|---|---|---|---|
| Jump | Arch | MSE | PSNR |
| 2 | 25 | 125.862 | 27.13186 |
| 2 | 30 | 91.73126 | 28.50563 |
| 2 | 35 | 59.97207 | 30.35131 |
| 7 | 25 | 203.2167 | 25.05121 |
| 7 | 30 | 137.7952 | 26.73846 |
| 7 | 35 | 115.3687 | 27.50992 |
| 12 | 25 | 373.4503 | 22.40847 |
| 12 | 30 | 252.5818 | 24.10678 |
| 12 | 35 | 190.6575 | 25.32827 |

| Gambar 3 – Audio 3 | | | |
|---|---|---|---|
| Jump | Arch | MSE | PSNR |
| 2 | 25 | 76.5713 | 29.29014 |
| 2 | 30 | 56.2577 | 30.62898 |
| 2 | 35 | 53.3711 | 30.85774 |
| 7 | 25 | 163.601 | 25.99294 |
| 7 | 30 | 122.7573 | 27.24033 |
| 7 | 35 | 92.8796 | 28.4516 |
| 12 | 25 | 438.2676 | 21.71341 |
| 12 | 30 | 258.5898 | 24.00469 |
| 12 | 35 | 176.6951 | 25.65856 |

| Gambar 4 – Audio 4 | | | |
|---|---|---|---|
| Jump | Arch | MSE | PSNR |
| 2 | 25 | 109.2444 | 27.74681 |
| 2 | 30 | 98.62415 | 28.19097 |
| 2 | 35 | 50.04756 | 31.13697 |
| 7 | 25 | 245.8848 | 24.22349 |
| 7 | 30 | 164.1921 | 25.97728 |
| 7 | 35 | 111.0905 | 27.67403 |
| 12 | 25 | 314.5635 | 23.15372 |
| 12 | 30 | 252.3402 | 24.11094 |
| 12 | 35 | 171.1667 | 25.79661 |

Dari hasil tersebut dapat terlihat bahwa semakin kecil parameter jump maka akan semakin besar PSNR yang dihasilkan. Hal tersebut terjadi karena semakin kecil jump, maka akan semakin banyak pengulangan data sisipan yang akan berpengaruh besar pada saat penghitungan kemunculan data sisipan. Parameter arch juga mempengaruhi nilai PSNR karena jika semakin besar arch, maka semakin besar ruang konstelasi dan jarak antara titik konstelasi akan semakin besar yang akan meminimalisasi *error* pada saat proses ekstraksi.

ISSN : 2085-4218

Tabel 10. Rentang Nilai PSNR [3]

| PSNR (dB) | *Picture Quality* |
|---|---|
| 60 | *Excellent, no noise apparent* |
| 50 | *Good, a small amount of noise but picture quality good* |
| 40 | *Reasonable, fine grain or snow in the picture, some fine detail lost* |
| 30 | *Poor picture with a great deal of noise* |
| 20 | *Unusable* |

Dari hasil yang didapatkan pun terlihat bahwa nilai PSNR berada pada rentang 20 hingga 35. Menurut tabel di atas, nilai PSNR pada rentang tersebut merupakan gambar yang rusak yang diakibatkan oleh banyaknya *noise*. Hal tersebut terjadi karena kesalahan pada proses ekstraksi data. Jika parameter arch bernilai 25, maka rentang dari setiap titik konstelasi bernilai arch/MPSK = 25/256 = 0.0977, nilai tersebut sangatlah kecil sehingga ketika mengalami sedikit pergeseran nilai maka hasil ekstraksi akan menjadi salah sehingga nilai MSE menjadi besar. Walaupun nilai PSNR dalam skema ini sangatlah kecil tetapi citra hasil ekstraksi masih sangat bisa dikenali karena perubahan yang terjadi pada setiap piksel sangatlah kecil, sebagai contoh nilai piksel yang awalnya 92 berubah menjadi 90.
Berikut ini merupakan citra hasil ekstraksi pada setiap gambar dengan parameter jump = 7 dan arch = 30

Tabel 11. Citra hasil ekstraksi

| Gambar 1 | Gambar 2 | Gambar 3 | Gambar 4 |
|---|---|---|---|
|  |  |  |  |

## 4. Kesimpulan

*Watermarking* dengan menggunakan modifikasi OFDM ini membuat kapasitas penyisipan menjadi lebih besar. Dengan jump yang bernilai 7, skema ini dapat menyisipkan sekitar 1475 data atau sekitar 11800 bit dalam satu detiknya. Dengan demikian algoritma *watermark* ini dapat menggunakan citra abu sebagai informasi sisipannya. Walaupun dengan nilai PSNR yang kecil, yaitu antara 20 dB hingga 35 dB, citra hasil ekstraksi tidak rusak dan masih sangat bisa dikenali.

## 5. Daftar Referensi

[1] Jose Juan Garcia-Hernandeza, Ramon Parra-Michelb, Claudia, Feregrino-Uribec, Rene Cumplido. High payload data-hiding in audio signals based on a modified OFDM approach, 2012.
[2] Manushree Bhardwaj, Arun Gangwar, Devendra Soni, "A Review on OFDM: Concept, Scope & its Applications" IOSR Journal of Mechanical and Civil Engineering (IOSRJMCE), 2012, pp 07-11.
[3] Constant, Mike. 2015. Signal to Noise Ratio. [Online] Available at: http://www.cctv-information.co.uk/i/Signal_to_Noise_Ratio. [Accessed 2 January 2016].