

## ABSTRAK

Pada era digital ini, pertukaran informasi berupa data sering terjadi di internet. Hal ini menimbulkan permasalahan pencurian hak cipta pada data di internet. Solusinya adalah *watermarking*, yaitu upaya penyisipan informasi secara rahasia pada suatu data guna memberikan tanda kepemilikan.

Tugas akhir ini fokus pada audio sebagai *host* daripada *watermark* serta analisa kualitas skema sistem audio *watermarking* yang dirancang menggunakan metode *Quantization Index Modulation* (QIM) berbasis *Discrete Wavelet Transform* (DWT), *Discrete Cosine Transform* (DCT), *Singular Value Decomposition* (SVD), dan *Cartesian to Polar Transformation* (CPT), yang diimplementasikan pada Raspberry Pi. Pada tugas akhir ini, penulis menggunakan metode DWT untuk mendekomposisi sinyal *host* audio berdasarkan frekuensinya. Kemudian metode DCT akan mentransformasikan sinyal *host* ke domain frekuensi. Setelah itu, sinyal *host* akan didekomposisi menjadi 3 submatriks yaitu submatriks  $U$ ,  $V$ , dan  $S$  menggunakan metode SVD. Selanjutnya, submatriks  $S$  dipilih penulis untuk ditransformasi kedalam nilai koordinat *polar* menggunakan metode CPT. Terakhir, sinyal *host* disisipkan *watermark* dengan metode QIM, sehingga sinyal audio memiliki perlindungan *copyright*. Skema sistem audio *watermarking* diimplementasikan pada Raspberry Pi agar lebih *portable* dan praktis.

Sistem, menggunakan *watermark* berupa citra biner berukuran  $16 \times 16$  *pixel* pada 5 buah tipe fail audio dengan mengaktifkan serangan, menghasilkan sinyal audio ter-*watermark* yang rata – rata nilai BER sebesar 0,1687,  $SNR \geq 20$  dB, Data Payload sebesar 1336,36 bps, serta  $MOS \geq 4$ . Sistem audio *watermarking* tahan terhadap serangan LPF, BPF, *Resampling*, *Time Scale Modification* (TSM), *Linear Speed Change*, *Equalizer*, Kompresi MP3 dan MP4, serta *Delay*, sehingga sistem audio *watermarking* bersifat *robust*, *high capacity*, dan *imperceptible*.

**Kata kunci:** *Audio Watermarking, DWT, DCT, SVD, CPT, QIM, Raspberry Pi.*