EECCIS 06

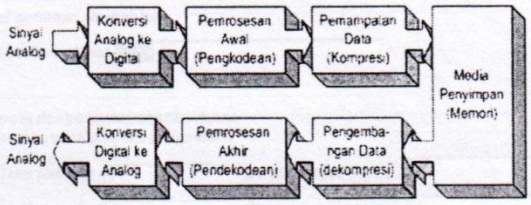
Implementasi Transformasi Wavelet Diskrit Sebagai Pemroses Awal Kompresi dan Dekompresi Data Sinyal Satu Dimensi Menggunakan FPGA

Ali Husein Alasiry, Hary Oktavianto, Bambang Sumantri, Gilang Unthari, Zulvatur Rofiah
Jurusan Elektronika, Politeknik Elektronika Negeri Surabaya - ITS
Kampus ITS, Keputih-Sukolilo, Surabaya-60111
ali@eepis-its.edu, hary@eepis-its.edu, bambang@eepis-its.edu

Abstract — **Makalah ini merupakan bagian dari penelitian tentang penerapan transformasi wavelet diskrit (DWT) dalam proses kompresi dan dekompresi data secara hardware menggunakan FPGA. Proses dekompresi yang akan dijelaskan disini merupakan kelanjutan dari penelitian yang telah dipublikasikan sebelumnya yaitu bagian kompresi [1]. Data yang tersimpan dalam bentuk koefisisen wavelet terkompresi selanjutnya diproses balik menggunakan Inverse DWT (IDWT) untuk mendapatkan kembali sinyal asli. Data hasil dekompresidibandingkan secara korelasi silang untuk mengetahui tingkat kemiripan kedua sinyal. Selanjutnya dilakukan pengaturan koefisien pada proses kuantisasi dibagian kompresi untuk mendapatkan perbandingan tingkat kemiripan hasil rekonstruksi daengan angka kompresi yang terbaik.**

I. PENDAHULUAN

Kompresi dan dekompresi data seringkali dibutuhkan dalam penyimpanan ataupun pengiriman data untuk mengatasi keterbatasan kapasitas memori maupun bandwidth komunikasi dan juga untuk mempercepat prosesnya. Permasalahan selanjutnya adalah bagaimana mendapatkan nilai kompresi yang semaksimal mungkin dengan seminimal mungkin informasi hilang pada proses kompresi dan dekompresi. Untuk meningkatkan efisiensi dan kinerja dari proses kompresi diantaranya adalah dengan melakukan pemrosesan awal berupa pengkodean ataupun transformasi sebelum proses kompresi dilakukan.



Gambar 1. Proses kompresi – dekompresi data.

Teknik pengkodean yang telah dikenal antara lain dengan pengkodean simbol, pengkodean *fixed length*, pengkodean Huffman. Sedangkan cara transformasi diantaranya DCT dan DWT.

Untuk merealisasikan DWT dapat dilakukan secara perangkat keras ataupun lunak. Untuk realisasi menggunakan perangkat lunak atau program perlu dipertimbangkan kecepatan CPU memproses data.

Bila CPU tidak cukup cepat dalam proses kompresinya akan menyebabkan turunnya performa system secara waktu nyata. Hal terburuk yang mungkin terjadi ialah hilangnya sebagian data akibat tidak terproses. Realisasi dengan perangkat keras dipilih dengan alasan ini.

II. DASAR TEORI

2.1. Kompresi Data Sinyal 1 Dimensi

Sinyal 1 dimensi merupakan sinyal dengan 1 variabel yang berubah dalam skala waktu. Sinyal ini dapat bersifat periodik, misalnya keluaran dari suatu osilator, atau aperiodik, misalnya sinyal random atau musik. Sinyal juga dapat mengandung satu atau lebih dari satu komponen frekuensi. Kompresi data berarti pengecilan jumlah data dalam merepresentasikan hal yang sama.

Secara umum ada dua metode kompresi data yaitu *lossless compression* dann *lossy compression*. *Lossless compression* merupakan metoda kompresi yang mempertahankan keutuhan informasi yang dikandung oleh data, sehingga proses rekonstruksi yang eksak dapat dilakukan sehingga diperoleh data yang sama persis dengan data aslinya. Biasanya dipakai pada kompresi file dan komunikasi data. Sedangkan pada *lossy compression*, ada informasi yang hilang pada proses kompresi data sehingga rekonstruksi eksak tidak dapat dilakukan. Biasanya diterapkan pada file gambar yang tidak membutuhkan kesamaan bit [5].

2.2. Transformasi Wavelet Diskrit sebagai Pemroses Awal

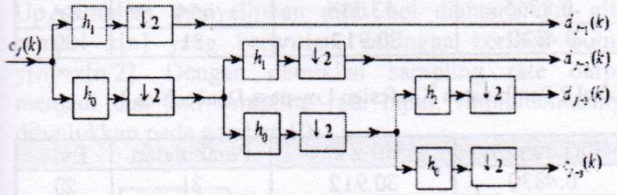
Secara umum transformasi wavelet kontinyu untuk sinyal $f(x)$ berdimensi satu didefinisikan pada persamaan (1) [6]:

$$W_a f(b) = \int f(x)\psi_{a,b}(x)dx \qquad (1)$$

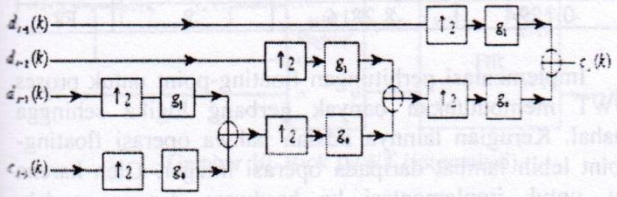
Transformasi wavelet mendekomposisi sinyal $f(x)$ kedalam bentuk varian sinyal induk wavelet Ψ yang ter-dilasi dan ter-translasi. Dengan kata lain sinyal $f(x)$ direpresentasikan sebagai jumlah dari kumpulan *dilated-version* dan *translated-version* fungsi induk wavelet. Fungsi induk ter-dilasi dengan faktor a dan ter-translasi sebesar b. Persamaan (1) dapat dibentuk kedalam bentuk diskrit dengan memberikan a dan b nilai diskrit ($a=2n, b \in Z$). Sinyal input mengalami dekom-posisi kedalam bentuk aproksimasi dari sejumlah koefisien yang merupakan hasil pergeseran dan penskalaan dari suatu bentuk fungsi induk wavelet seperti pada persamaan (2).

$$\psi_{a,b}(x)=\frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \qquad (2)$$

DWT dapat dihitung baik secara maju (FDWT) maupun terbalik (IDWT) dengan memenggunakan piramida Mallat [8] pada gambar 2 dan 3. Proses dekomposisi sinyal menjadi koefisien-koefisien wavelet dapat dilakukan dengan menggunakan transformasi maju (forward DWT, FDWT) seperti pada gambar 2 dimana h0 dan h1 koefisien dilasi berkaitan dengan penskalaan dan fungsi wavelet. Sedangkan rekonstruksi ulang sinyal dapat dilakukan dengan menggunakan transformasi terbaliknya (inverse DWT, IDWT) seperti pada gambar 2, dimana g0 dan g1 adalah koefisien high-pass dan low-pass filter berhubungan dengan fungsi induk wavelet.



Gambar 2. Pohon Mallat proses analisis.



Gambar 3. Pohon Mallat proses sintesis.

Persamaan untuk DWT maju (FDWT) dan DWT terbalik (IDWT) dapat dinyatakan seperti pada kumpulan persamaan 3. Untuk menghitung koefisien-koefisien wavelet diatas digunakan kumpulan persamaan berikut

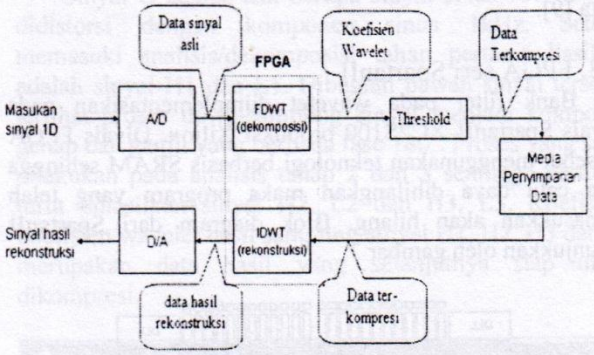
$$\textbf{Forward DWT:}\quad \begin{cases} c_J(k)=x(k) \\ c_{j-1}(k)=\sum_m h_0(m-2k)c_j(m) \\ d_{j-1}(k)=\sum_m h_1(m-2k)c_j(m) \\ \text{for } j=J,J-1,\cdots,j_0+1 \\ \{d_{j-1}(k),d_{j-2}(k),\cdots,d_{j_0+1}(k),d_{j_0}(k),c_{j_0}(k)\} \end{cases} \qquad (3)$$

$$\textbf{Inverse DWT:}\quad \begin{cases} c_{j+1}(k)=\sum_m c_j(m)g_0(k-2m)+\sum_m d_j(k)g_1(k-2m) \\ \text{for } j=j_0,j_0+1,\cdots,J-1 \\ x(k)=c_J(k) \end{cases}$$

III. DISAIN IMPLEMENTASI

3.1. Desain sistem

Gambar 4 menunjukkan blok-blok proses yang akan dilakukan. Disini tampak untuk implementasi DWT dilakukan pada chip FPGA. Pengubah analog ke digital dan sebaliknya memanfaatkan chip yang sudah ada pada modul extensi. Sedangkan proses threshold dan penyimpanan data sampai saat ini masih dalam tahapan simulasi.



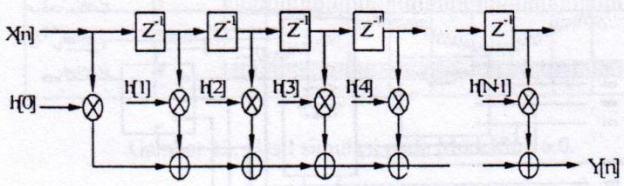
Gambar 4. Blok diagram sistem.

3.2. Implementasi filter FIR

Persamaan DWT dapat dihitung secara efisien menggunakan *quadratic mirror filter* (QMF) seperti ditunjukkan pada gambar diatas. Kesemua struktur piramid FDWT dan IDWT dapat disusun menggunakan filter FIR. Secara umum filter FIR dengan panjang M bisa dinyatakan dalam persamaan (4).

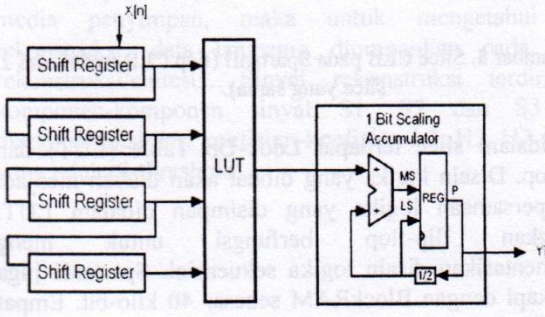
$$H(z)=\sum_{k=0}^{M-1}h[k]z^{-k} \qquad (4)$$

Pada kebanyakan aplikasi, bentuk standar filter FIR adalah seperti ditunjukkan pada gambar 5. Setiap tap filter terdiri dari sebuah elemen tunda (delay), sebuah penjumlah (adder) dan sebuah pengali (multiplier) [7].



Gambar 5. Struktur filter FIR langsung.

Implementasinya pada hardware secara blok diagram seperti ditunjukkan pada gambar 6. FIR terdiri dari empat shift register yang dikaskade, sebuah tabel cari lihat (*look up table*, LUT) dan sebuah akumulator.

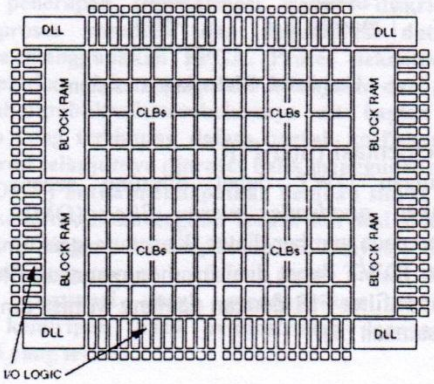


Gambar 6. Komponen blok FIR pada FPGA.

LUT berisi semua kemungkinan yang terdapat pada table koefisien wavelet Daubechies. Wavelet Daubechies dipilih karena diketahui memiiiki keunggulan dalam hal kompresi data [9].

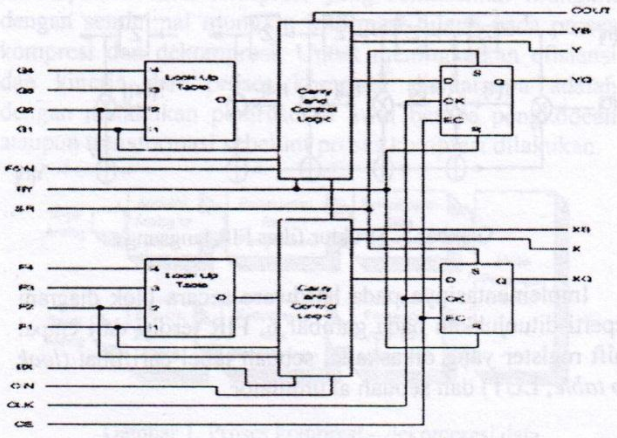
3.3. FPGA seri SpartanII

Bank filter pada wavelet diimplementasikan pada divais SpartanII XC2S100 produksi Xilinx. Divais FPGA tersebut menggunakan teknologi berbasis SRAM sehingga bila catu daya dihilangkan maka program yang telah dimasukkan akan hilang. Blok diagram dari SpartanII ditunjukkan oleh gambar 7.



Gambar 7. Blok diagram divais SpartanII.

CLB berfungsi untuk mengimplemen-tasikan disain yang dibuat. Pada divais XC2S100 terdapat 600 CLB Tiap CLB terdiri atas dua buah slice seperti yang ditunjukkan oleh gambar 8.



Gambar 8. Slice CLB pada SpartanII (tiap CLB terdiri atas 2 slice yang sama).

Didalam slice terdapat Look-Up Table (LUT) dan Flip-flop. Disain logika yang dibuat akan diubah menjadi tabel persamaan logika yang disimpan didalam LUT. Sedangkan flip-flop berfungsi untuk meng implementasikan disain logika sekuensial. SpartanII juga dilengkapi dengan BlockRAM sebesar 40 kilo-bit. Empat buah Digital Locked Loop (DLL) disediakan untuk menghasilkan clock dengan delay rendah yang mempunyai jalur khusus ke semua bagian. Input/Output Block (IOB) berada pada tiap tepi divais yang terhubung langsung dengan pin IC FPGA.

3.4. Bank filter Daubechies orde-4

Penelitian ini menegimplementasikan bank filter orthogonal Daubechies orde-4 (db4). Filter ini telah digunakan secara luas karena mempunyai sifat orthogonal dan mudah dibuat karena merupakan filter yang pendek. Juga mempunyai kondisi hasil rekontruksi yang sempurna dan memuaskan. Koefesien-koefisien filter ditunjukkan oleh tabel 1 dan 2.

Tabel I Pembulatan Koefesien Highpass Daubechies 4

G1(Highpass)	G1 x 2^6	Pembulatan	Byte
0.1294	8.2816	8	08
0.2241	14.3424	14	0E
-0.8365	-53.536	-54	CA
0.4830	30.912	31	20

Tabel 2 Pembulatan Koefesien Lowpass Daubechies 4

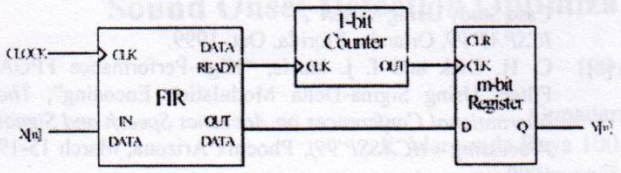
G0(Lowpass)	G0 x 2^6	Pembulatan	Byte
0.4830	30.912	31	20
0.8365	53.536	54	36
0.2241	14.3424	14	0E
-0.1294	-8.2816	-8	F8

Implementasi perhitungan floating-point untuk proses DWT membutuhkan banyak gerbang logika sehingga mahal. Kerugian lainnya adalah bahwa operasi floating-point lebih lambat daripada operasi integer. Oleh karena itu, untuk implementasi ke hardware dengan mudah, koefesien pecahan tersebut digeser kekiri sebanyak 2^6 dan mengambil nilai bulatnya saja. Cara ini cukup bagus untuk mendapatkan ketelitian sampai 3 angka desimal. Menggunakan operasi integer membuat proses perhitungan menjadi cepat dan membutuhkan sedikit gerbang logika yang pada akhirnya akan menurunkan konsumsi daya. Pada akhir perhitungan, nilai output dikembalikan lagi dengan cara menggeser kekanan sebanyak 2^6 untuk mendapatkan hasil yang benar.

Realisasi perangkat keras yang paling mungkin dan ekonomis adalah menggunakan FPGA. Disini perlu dipertimbangkan kapasitas keping FPGA yang digunakan, disamping kapasitas memori RAM. IC seri XC4000 seperti XC4010XL mempunyai kapasitas sebesar 10.000 *logic gates*. Sedangkan IC seri SpartanII seperti XC2S100 mempunyai kapasitas 100.000 *logic gates* dan mempunyai blok-RAM internal sebanyak 40Kb. Karena SpartanII mempunyai RAM internal, diharapkan disain hardware dapat menjadi lebih sederhana.

3.5. Implementasi FDWT dan IDWT

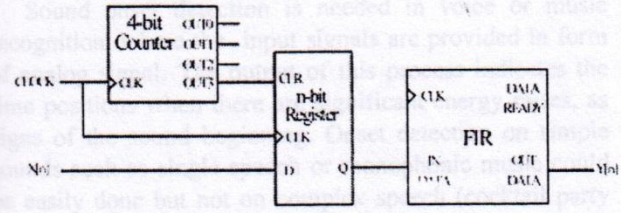
Blok dasar FDWT filter bank terdiri dari desimator berupa FIR yang diikuti dengan operator down sampling [10]. Down sampling mengurutkan input x[n] dengan n merupakan kelipatan 2, sehingga berlaku output y[n]=x[2n]. Implementasinya seperti ditunjukkan pada gambar 9.



Gambar 9. Blok FDWT (decimator).

Diasumsikan terdapat sinyal input 8-bit. Jika pin DATA-READY diaktifkan, siyal masuk menuju filter FIR. Setiap selesai operasi, filter FIR memberikan sinyal trigger ke counter untuk memproses data berikutnya. Proses ini berulang dimana bila output counter bernilai 1 data dari FIR disimpan, sedangkan bila 0, data diabaikan.

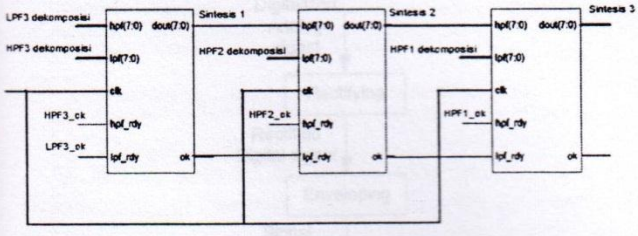
Blok dasar FDWT filter bank terdiri dari interpolator berupa FIR yang diikuti dengan operator up sampling [10]. Up samplers menyelipkan nilai nol diantara dua nilai sampel x[n] yang berurutan, sehingga berlaku output y[n]=x[n/2]. Dengan demikian sampling rate output menjadi dua kali sampling rate input. Implementasinya ditunjukkan pada gambar 10.



Gambar 10. Blok IDWT (interpolasi).

Diasumsikan sumber sinyal mengirimkan data secara berurutan dalam 16 clock kedalam pembeban paralel (parallel load register) kemudian data diteruskan ke filter FIR. Selama nilai MSB counter adalah 0 yaitu pada 8 data pertama, data ditransfer ke output, sedangakan ketika MSB menjadi 1, nilai 0 ditransfer ke output. Proses berlaku scara berulang.

Hasil implementasi menggunakan VHDL yang telah dikonversi dalam bentuk skematik blok rekontruksi/sintesis tiga tingkat ditunjukkan pada gambar 11.

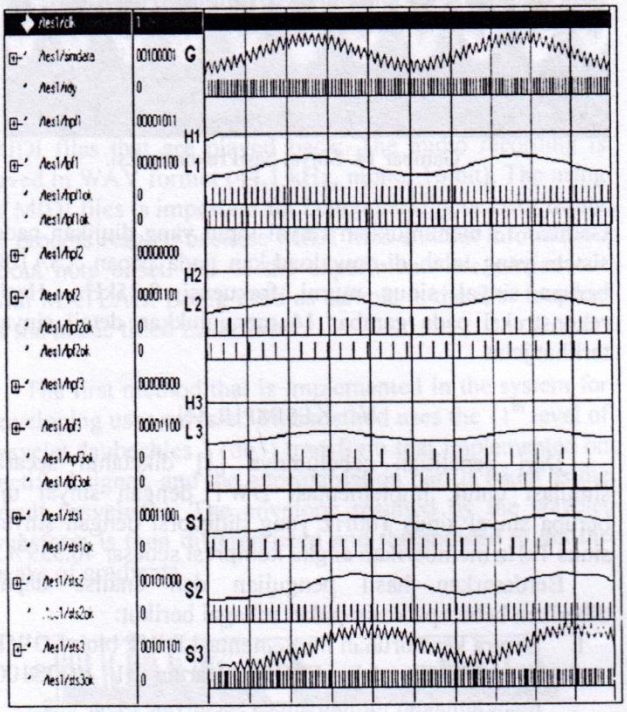


Gambar 11. Blok komponen rekonstruksi level skematik.

IV. Simulasi Dan Pengujian

Setelah dilakukan penulisan kode program dalam VHDL, kemudian dilakukan simulasi pada software ModelSim© 6.0, untuk mengamati proses pada masing-masing tahap DWT beserta besarnya sumber daya FPGA yang digunakan. Hasil simulasi pada ModelSim® 6.0 ditunjukkan pada gambar 12.

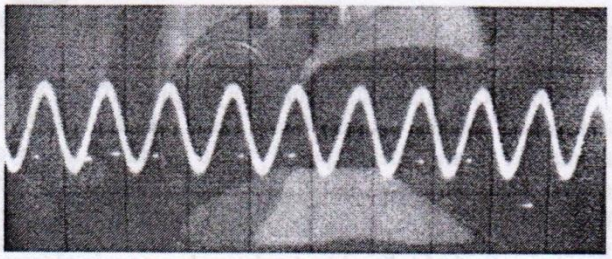
Sinyal G adalah asli berupa sinyal sinus 100Hz yang didistorsi dengan komponen sinus 1kHz. Setelah memasuki analisis/dekomposisi tahap pertama hasilnya adalah sinyal H1 dan L1. Dibagian bawah sinyal tersebut terlihat proses down-sampling sinyal menjadi komponen genap dan ganjil yang berbeda fase 180°. Proses yang sama dilakukan pada analisis tahap 2 dan 3 sehingga berturut-turut dihasilkan sinyal H2, L2 dan H3, L3. Koefisien-koefisien wavelet hasil sampling sinyal H1, H2, H3 dan L3 merupakan data hasil yang selanjutnya siap untuk dikompresi.



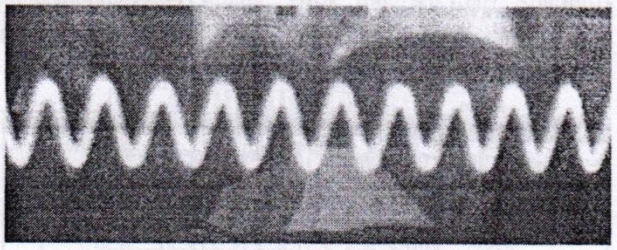
Gambar 12. Hasil simulasi pada ModelSim 6.0.

Pada tahap kompresi, koefisien-koefisien hasil pemfilteran FIR Highpass diberi nilai threshold sehingga koefisien-koefisien dengan nilai dibawah threshold akan menjadi 0. Dalam penelitian sebelumnya [1] digunakan nilai threshold 2 dan -2 dimana diperoleh total kompresi 65,32%. Data ini dapat langsung disimpan dalam media penyimpanan dengan teknik zero-surpression.

Pada penelitian ini data tidak sampai disimpan pada media penyimpan, maka untuk mengetahui hasil rekonstruksi, data langsung diumpankan pada proses rekonstruksi/sintesis. Sinyal rekonstruksi terdiri dari komponen-komponen sinyal S1, S2 dan S3 yang direkonstruksi dari koefisien-koefisien L3, H3, H2 dan H1 yang telah di-threshold.



Gambar 13. Sinyal input (asli).



Gambar 14. Sinyal hasil rekonstruksi.

Gambar 13 menunjukkan sinyal input yang diujikan pada sistem yang telah di-download-kan pada papan XSA100 berupa sinyal sinus murni frekuensi 9.25kHz. Hasil rekonstruksi pada gambar 14 menunjukkan detail sinyal terlihat jelas.

V. KESIMPULAN

Dari penelitian sebelumnya [1] diketahui secara simulasi untuk implementasi DWT dengan sinyal uji berupa sinyal sinus 100Hz yang didistorsi dengan sinyal sinus 1kHz memberikan angka kompresi sebesar 46.339%.

Berdasarkan hasil pengujian dan analisa dapat diberikan beberapa kesimpulan sebagai berikut:

1. Secara keseluruhan implementasi DWT blok FDWT dan IDWT pada FPGA Spartan II XC2S100 menggunakan jumlah slices sebanyak 74%.
2. Sinyal rekonstruksi yang dihasilkan memiliki nilai cross korelasi dengan sinyal input sebesar 0.9 atau tingkat kemiripan 90%.

PUSTAKA DAN REFERENSI

[1] A. H. Alasiry, H. Oktavianto, B. Sumantri, G. Unthari, Z. Rofiah, "Implementasi Transformasi Wavelet Diskrit Sebagai Pemroses Awal Kompresi Data Sinyal 1 Dimensi Menggunakan FPGA" , SITIA 2006.

[2] Mallat, S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Recognition and Machine Intelligence*.Vol. 11, No. 7, pp. 674-693, July 1989.

[3] Daubechies, I., "The Wavelet Transform, Time-Frequency Localization and Signal Analysis", *IEEE Transactions on Information Theory*. Vol. 36, No. 5, pp. 961-1005, September 1990.

[4] S.A. White, "Applications of Distributed Arithmetic to Digital Signal Processing", *IEEE ASSP Magazine*, Vol. 6(3), pp. 4-19, July 1989.

[5] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[6] Xilinx Inc., *The Programmable Logic Data Book*, 1999.

[7] C. H. Dick and f. j. harris, "FPGA Multirate Filters: A Case Study Using Virtex", *ICSPAT'99*, Orlando, Florida, Oct. 1999.

[8] C. H. Dick and f. j. harris, "High-Performance FPGA Filters Using Sigma-Delta Modulation Encoding", *The International Conferences on Acoustics Speech and Signal Processing - (ICASSP'99)*, Phoenix Arizona, March 15-19 1999.

[9] E. B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation", *IEEE. Trans. Acoust., Speech Signal Processing*, Vol. 29, No. 2, pp. 155-162, April 1981.

[10] Brigham, E. O., *The Fast Fourier Transform*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

[11] Allen, J. B. and Rabiner, L. R., "A Unified Approach to Short-Time Fourier Analysis and Synthesis", *Proceedings of IEEE*. Vol. 65, No. 11, pp. 1558-1564, 1977

[12] Burrus, C. S.; Gopinath, R. A.; and Guo, H., *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall Inc., Upper Saddle River, New Jersey, 1998.

[13] Strang, G. and Nguyen, T., Wavelets and Filter Banks. Wellesley-Cambridge Press, Wellesley, Massachusetts, 1996.

[14] T. I. Laakso and V. Valimaki, "Splitting the Unit Delay", *IEEE Signal Processing Magazine*, pp. 30-60, Jan. 1996.