

BAB III

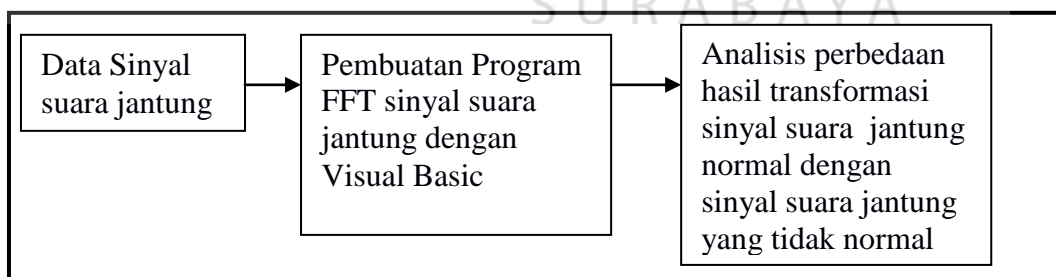
METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam pengerjaan tugas akhir ini adalah studi kepustakaan, pembuatan program dan analisis. Dengan ini penulis berusaha untuk mengumpulkan data dan informasi-informasi serta materi yang bersifat teoritis yang sesuai dengan permasalahan. Hal tersebut diperoleh dari buku-buku, materi perkuliahan serta literatur dari internet, jurnal dan dengan bantuan *Matlab* serta *Visual Basic 6.0*.

Penelitian diawali dengan pencarian data sinyal suara jantung yang didapat dari internet [<http://www.med.umich.edu/lrc/psb/heartsounds/>], setelah itu data yang didapat dicoba dengan menggunakan software *Matlab* untuk mendapatkan hasil yang akan dicapai dalam pembuatan program konversi sinyal.

Analisis sinyal jantung dengan menggunakan analisis spektrum ini dapat dijelaskan dengan blok diagram dibawah ini :



Gambar 3.1. Blok Diagram Analisis Sinyal Suara Jantung Dengan Analisis Spektrum

Pada Gambar 3.1 kegiatan penelitian dapat di kelompokkan menjadi 3 yaitu proses pencarian data input sinyal suara jantung, proses pembuatan program, dan analisa hasil transformasi sinyal.

1. Proses Pembuatan Program

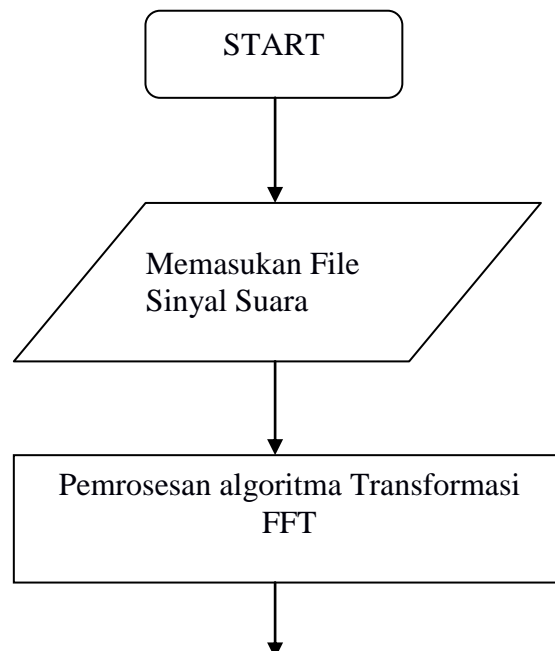
Proses pembuatan program dimulai dengan membuat flowchart dari program dan menuliskan kode program ke program yang dibuat. Flowchart dari program akan dijelaskan pada sub bab 3.2.

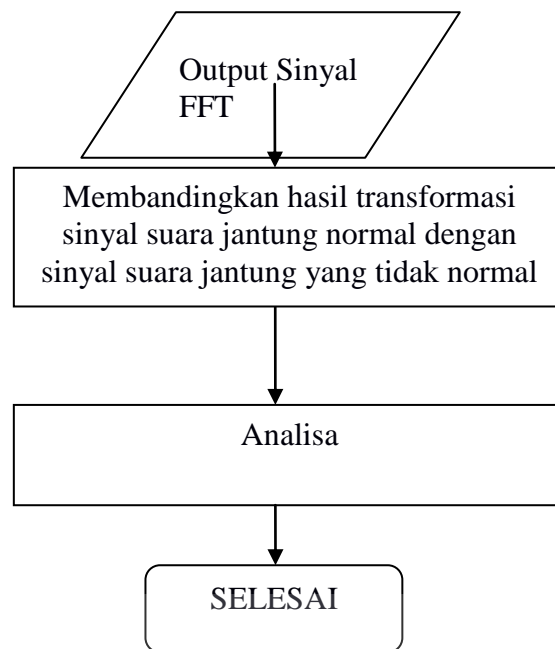
2. Analisa transformasi sinyal

Program yang telah dibuat digunakan untuk mentransformasikan sinyal suara jantung yang telah didownload sebelumnya. Sinyal jantung yang normal dan yang tidak normal diinputkan ke program untuk mendapatkan spektrum sinyal hasil transformasi, kemudian dibandingkan untuk melihat perbedaan hasil transformasi sinyal normal dan tidak normal.

3.2 Prosedur Penelitian

Prosedur ini menjelaskan tentang langkah-langkah yang dilakukan dalam penelitian seperti pada Gambar 3.2.



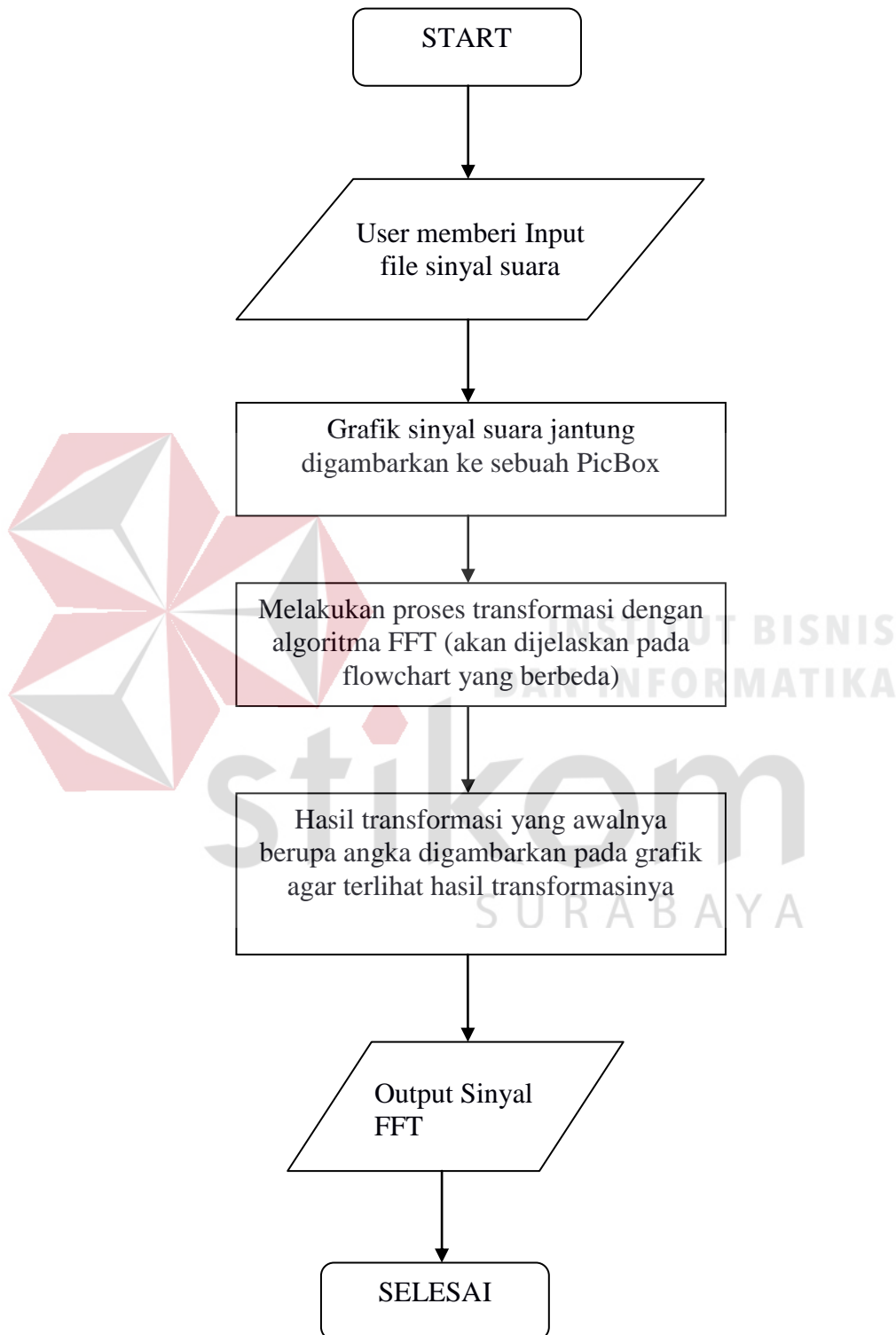


Gambar 3.2. *Flowchart* Proses Penelitian Transformasi Sinyal Suara Jantung

1. Proses dimulai dengan meloading atau menginputkan data sinyal suara jantung pada program yang telah dibuat sebelumnya. Sinyal suara jantung di inputkan untuk selanjutnya diproses dengan Transformasi Fourier untuk memperoleh hasil transformasinya.
2. Setelah file yang di inputkan diproses maka langkah selanjutnya adalah menganalisa hasil transformasi yang berupa grafik
3. Hasil transformasi sinyal suara jantung yang normal dan tidak normal dibandingkan untuk memperoleh hasil penelitian

3.2.1. Desain Flowchart Program

Flowchart dari program yang dibuat akan dijelaskan pada gambar dibawah ini :



Gambar 3.3. Flowchart Program Transformasi Sinyal Suara Jantung

Penjelasan flowchart :

1. Pada awalnya user memberikan input sinyal suara jantung ke dalam program untuk dibuatkan grafik.
2. File sinyal suara jantung yang di inputkan dibuatkan grafiknya agar user dapat melihat grafik dari sinyal suara jantung yang dibuat.
3. Kemudian user menekan tombol “Transform” untuk memulai proses transformasi sinyal suara jantung
4. Sinyal suara jantung yang telah diproses dengan Fast Fourier Transform dapat dilihat hasilnya di grafik.

3.2.2. Fast Fourier Transform

Fast Fourier Transform adalah algoritma untuk menghitung DFT dengan memanfaatkan sifat simetris dari DFT agar perhitungan dapat lebih efisien dan cepat dengan mengabaikan perhitungan yang sama (Smith, 2011).

Rumus dasar dari DFT ditunjukkan pada persamaan 3.1.

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-jk2\pi n} \quad (3.1)$$

Keterangan :

X_k : Representasi domain frekuensi untuk x_n .

N : Periode pada sinyal diskrit pada contoh atau soal

n : bilangan deret dengan batas N .

Rumus diatas dapat juga ditulis kembali seperti ditunjukkan pada persamaan 3.2.

$$X_k = \sum_{n=0}^{N-1} x_n w^{k n} \text{ menjadi } w = e^{-\frac{j2\pi}{N}} \quad (3.2)$$

Keuntungan dari pemakaian FFT adalah efisiensi perhitungan yang besar.

Pada awalnya formula ini dibuat karena jumlah perhitungan yang sangat besar

jika menggunakan metode DFT tradisional. Karena pada pada perhitungan DFT setiap garis spektral X_n harus dilakukan evaluasi sebanyak N . dan untuk memperoleh seluruh spektrum yang ada maka diperlukan N^2 perkalian dan N^2 evaluasi fungsi. Jika difokuskan pada jumlah perkaliannya saja maka algoritma untuk memperoleh Transformasi Fourier Diskrit dikatakan mempunyai orde N^2 .

Penghematan yang didapatkan dari FFT dapat dihitung dengan persamaan

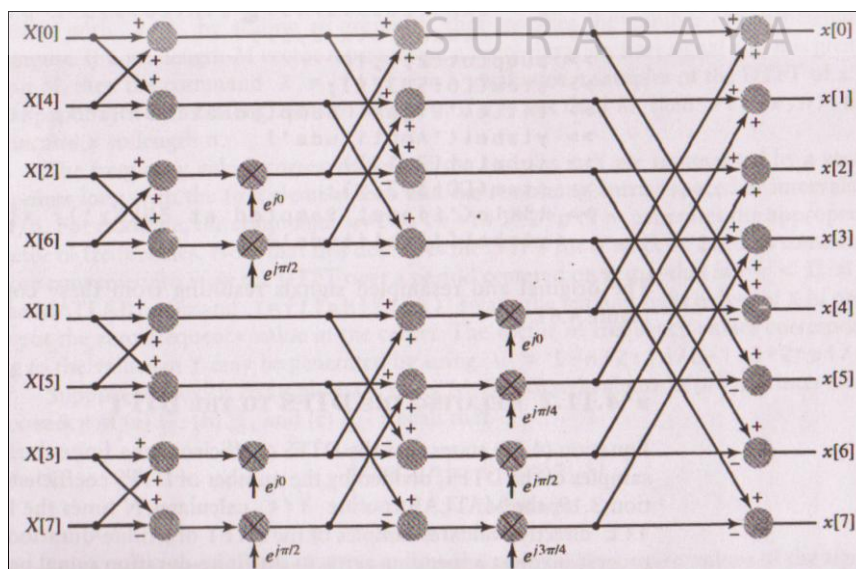
3.3.

$$\frac{N^2}{N \log_2 N} = \frac{N}{M} \quad (3.3)$$

Rumus ini didapatkan dari :

$N = 2^M$ (M disebut radiks) sehingga $M = \log_2 N$. Pada setiap tingkat harus dilakukan N perkalian sehingga algoritma FFT memerlukan $N \log_2 N$ perkalian. Contoh jika $N = 1024$ maka M adalah 10 sehingga $N/M = 1024/10 = 102.4$, Sehingga keuntungan menggunakan FFT adalah 102.4 kali lebih efisien dari metode DFT tradisional.

Berikut adalah proses perhitungan FFT jika digambarkan dengan model.



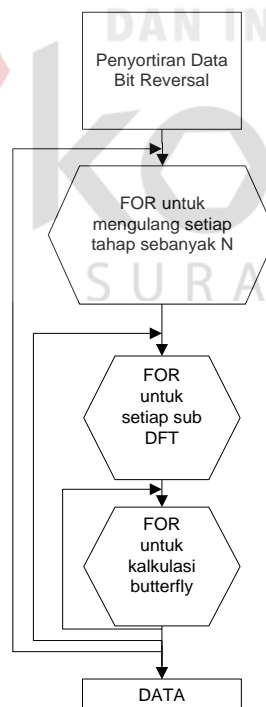
Gambar 3.4. Diagram Butterfly (Haykin, 2003)

Gambar menunjukkan perhitungan dengan $N=8$. Proses pemisahan antara bit genap dan ganjil diulang untuk menukar urutan dari koefisien DTFS pada input. Proses penukaran ini disebut *bit reversal*, karena metode ini menggunakan pembalikan bit untuk menentukan lokasi dari $X[k]$. Contoh :

- $X[6]$ mempunyai index 6 atau $k = 6$. Bila dijadikan biner maka akan menjadi $6 = 110_2$. sekarang membalik bit yang ada maka akan merubah 110_2 menjadi 011_2 atau $k' = 3$ sehingga $X[6]$ berada pada baris ke empat.

Kemudian dilakukan pertukaran bit dengan diagram diatas yang disebut diagram butterfly atau kupu – kupu karena kemiripannya.

Sehingga secara garis besar Flowchart dari FFT dapat digambarkan sebagai berikut:



Gambar 3.5. Flowchart algoritma FFT.

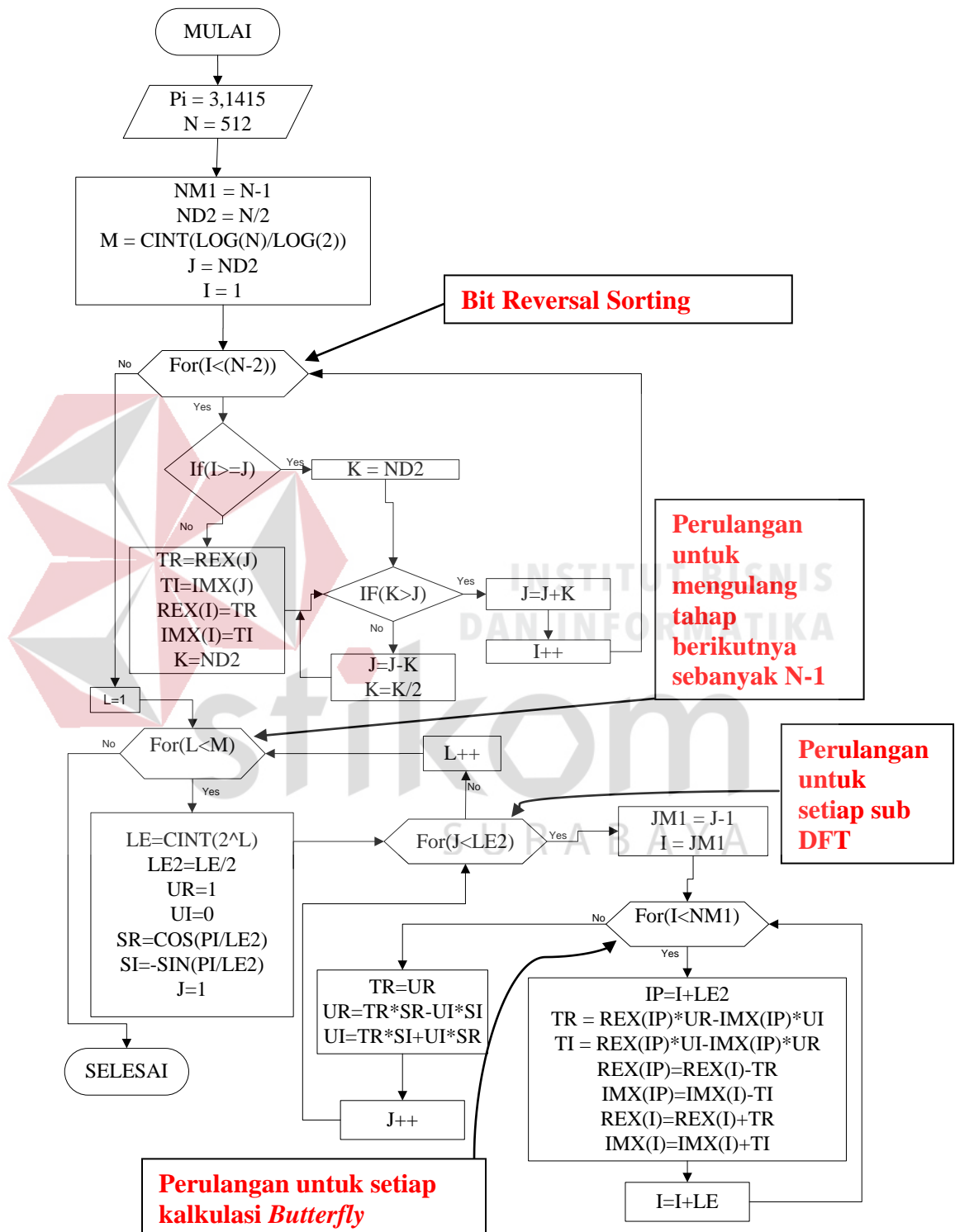
Penjelasan dari Flowchart :

1. Proses transformasi fourier dimulai dengan penyortiran bit secara terbalik atau *reversal*.
2. Setelah itu proses dilanjutkan dengan perulangan FOR untuk melakukan perulangan di setiap tahap sampai dengan batas perulangan yang dinyatakan dengan N yang pada program ini adalah 512.
3. Tahap berikutnya adalah perulangan FOR untuk mengulang perhitungan sub-DFT.
4. Tahap berikutnya adalah perulangan FOR untuk kalkulasi Butterfly.



3.2.3. Flowchart dari Logika FFT

Berikut adalah flowchart lengkap dari logika FFT yang digunakan pada program.



Gambar 3.6. Flowchart algoritma FFT yang digunakan pada program transformasi sinyal suara jantung

Penjelasan Dari flowchart algoritma Fast Fourier Transform(FFT).

Algoritma *Fast Fourier Transform* (FFT) ini dibagi menjadi 6 bagian yaitu :

1. Deklarasi konstanta yang diperlukan.

Pada Tahap ini dilakukan deklarasi konstanta yang diperlukan antara lain :

N dan Pi. Nilai N ditetapkan sebesar 512 dan Pi sebesar 3.14.

2. Proses untuk menetapkan nilai beberapa variabel yang digunakan dalam FFT.

Pada tahap ini dilakukan penetapan nilai dari beberapa variabel yang digunakan. Beberapa variabel itu adalah :

1. NM1 yang di tetapkan nilainya dari $N-1$ atau $512 - 1 = 511$, jadi nilai variabel NM1 sebagai 511.

2. ND2 yang ditetapkan nilainya dari $N/2$ atau $512/2 = 256$, jadi nilai variabel ND2 adalah 256.

3. M yang ditetapkan nilainya dari $\log_2(N)$ atau $\log_2(512) = 9$, jadi nilai variabel M adalah 9.

4. J yang ditetapkan nilainya sama dengan ND2 atau 256.

3. Bit Reversal Sorting

Pada Tahap ini setiap array di REX[] dan IMX[] nilainya akan ditukar dengan nilai array yang lain. Array REX[] berfungsi untuk menyimpan nilai perhitungan yang REAL dan array IMX[] untuk menyimpan nilai Imajiner. Fungsi perulangan FOR bertugas untuk mengulang proses penukaran nilai variabel array REX[] dan IMX[]. Contoh :

Misal nilai $REX[256] = 120$ dan berjalan pada proses selanjutnya maka nilai REX tersebut akan diinputkan pada REX[1] sehingga nilai $REX[1] = 120$

.Nilai 1 diambil dari variabel I yang berfungsi untuk menyimpan nilai urutan looping. Untuk melakukan pembalikan bit yang diperlukan pada program.

4. Looping untuk semua proses berikutnya.

Looping ini adalah untuk mengubah nilai gelombang yang dalam domain waktu menjadi domain frekuensi. Tahap ini akan berlangsung sebanyak 8 kali karena nilai M didapatkan dari $\log_2(N)$ yang dimana nilai N ditetapkan sebesar 512. Pada looping ini proses yang terjadi adalah menyiapkan variabel- variabel yang diperlukan di proses berikutnya yaitu untuk proses pada perhitungan sub-DFT dan kalkulasi butterfly.

5. Proses di dalam sub DFT yang diulang pada setiap sub DFT nya.

Perulangan sebanyak $N/2$ yang nilainya disimpan pada variabel J. Perulangan ini bertujuan untuk menyiapkan variabel JM1 yang ditambahkan 1 setiap perulangan terjadi dan berisi kalkulasi butterfly.

6. Looping/Perulangan untuk proses kalkulasi Butterfly

Pada Perulangan ini berisi proses kalkulasi butterfly yang berlangsung jika nilai dari variabel I tidak lebih dari NM1. Proses Butterfly adalah proses transformasi 2 point kompleks menjadi 2 point yang kompleks pula. Proses butterfly di ulang untuk menghitung nilai setiap spektrum frekuensi. Dalam Tahap ini dilakukan juga perhitungan sinusoida yang diperlukan di dalam kalkulasi butterfly. Nilai sinusoida didapatkan dari variabel SR dan SI.