



**APLIKASI PENCITRAAN DATA DIGITAL PADA
OBJEK SAYAP CAPUNG MENGGUNAKAN
PEMOGRAMAN MATLAB**

skripsi

disajikan sebagai salah satu syarat

untuk memperoleh gelar Sarjana Sains Program Studi Fisika

oleh

Muhammad Denny Steve Bob Hudson

NIM 4250404034

JURUSAN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SEMARANG

2011

PERSETUJUAN PEMBIMBING

Skripsi ini yang berjudul “*Aplikasi Pencitraan Data Digital Pada Objek Sayap Capung Menggunakan Pemograman Matlab*”, telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi.

Semarang, September 2011

Pembimbing I

Pembimbing II

Dr. Agus Yulianto, M.Si
NIP. 19660705 199003 1002

Sunarno, S.Si, M.Si
NIP.19720112 199903 1003

PERPUSTAKAAN
UNNES

PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, 8 September 2011

Muhammad Denny Steve Bob Hudson
NIM. 4250404034

PERPUSTAKAAN
UNNES

PENGESAHAN

Skripsi yang berjudul

Aplikasi Pencitraan Data Digital Pada Objek Sayap Capung Menggunakan

Pemrograman Matlab

disusun oleh

Muhammad Denny Steve Bob Hudson

4250404034

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES
pada tanggal 8 September 2011

Panitia:

Ketua

Sekretaris

Drs. Kasmadi Imam S, M.S.
NIP. 19511115 1979 1 001

Dr. Putut Marwoto, M.S.
NIP. 19630821 198803 1 004

Ketua Penguji

Dr. Khumaedi, M.Si.
NIP. 19630610 198901 1 002

Anggota Penguji/
Pembimbing Utama

Anggota Penguji/
Pembimbing Pendamping

Dr. Agus Yulianto, M.Si.
NIP. 19660705 199003 1 002

Sunarno, S.Si, M.Si.
NIP. 19720112 199903 1 003

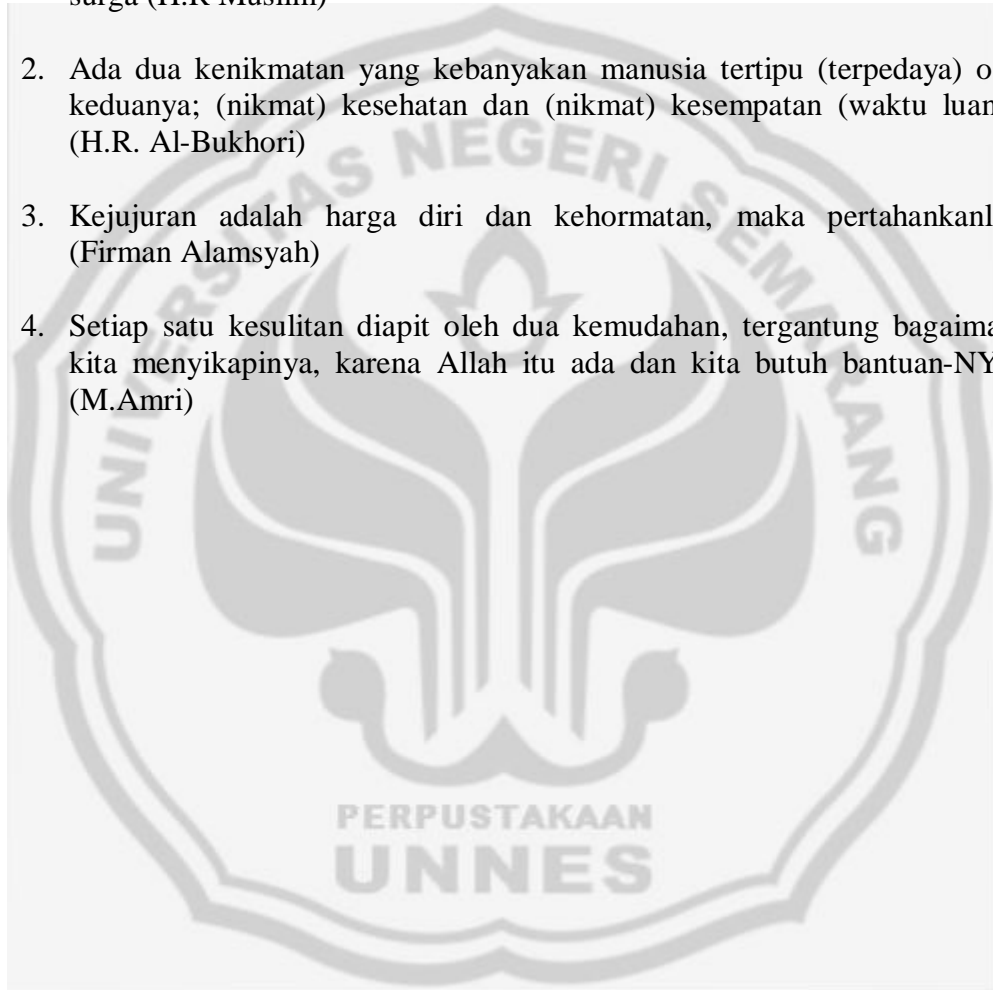
PERSEMBAHAN

1. Untuk Ayah, Ibu, dan Kakak-Kakak
2. Untuk Dr. Agus Yulianto, M.Si
3. Untuk Sunarno, S.Si, M.Si
4. Untuk Dr. Khumaedi, M.Si
5. Untuk Teman-teman Fisika
6. Untuk Teman-teman Kost



MOTTO

1. Barang siapa menempuh jalan untuk menuntut ilmu maka Allah SWT akan memudahkan bagi orang itu karena ilmu tersebut jalan menuju ke surga (H.R Muslim)
2. Ada dua kenikmatan yang kebanyakan manusia tertipu (terpedaya) oleh keduanya; (nikmat) kesehatan dan (nikmat) kesempatan (waktu luang). (H.R. Al-Bukhori)
3. Kejujuran adalah harga diri dan kehormatan, maka pertahankanlah. (Firman Alamsyah)
4. Setiap satu kesulitan diapit oleh dua kemudahan, tergantung bagaimana kita menyikapinya, karena Allah itu ada dan kita butuh bantuan-NYA. (M.Amri)



PRAKATA

Alhamdulillahirroobil'alamin, puji syukur penulis panjatkan kehadirat Allah Subhanahu Wa ta'ala, Satu - satunya Dzat yang berhak menerimanya, dan yang mempunyai kekuatan memberikan segenap karunia pada semua hambanya, sehingga penulis dapat menyelesaikan skripsi dengan judul "Aplikasi Pencitraan Data Digital Pada Objek Sayap Capung Menggunakan Pemograman Matlab". Bantuan berbagai pihak yang dibutuhkan dalam penulisan skripsi, penulis mengucapkan banyak terima kasih kepada :

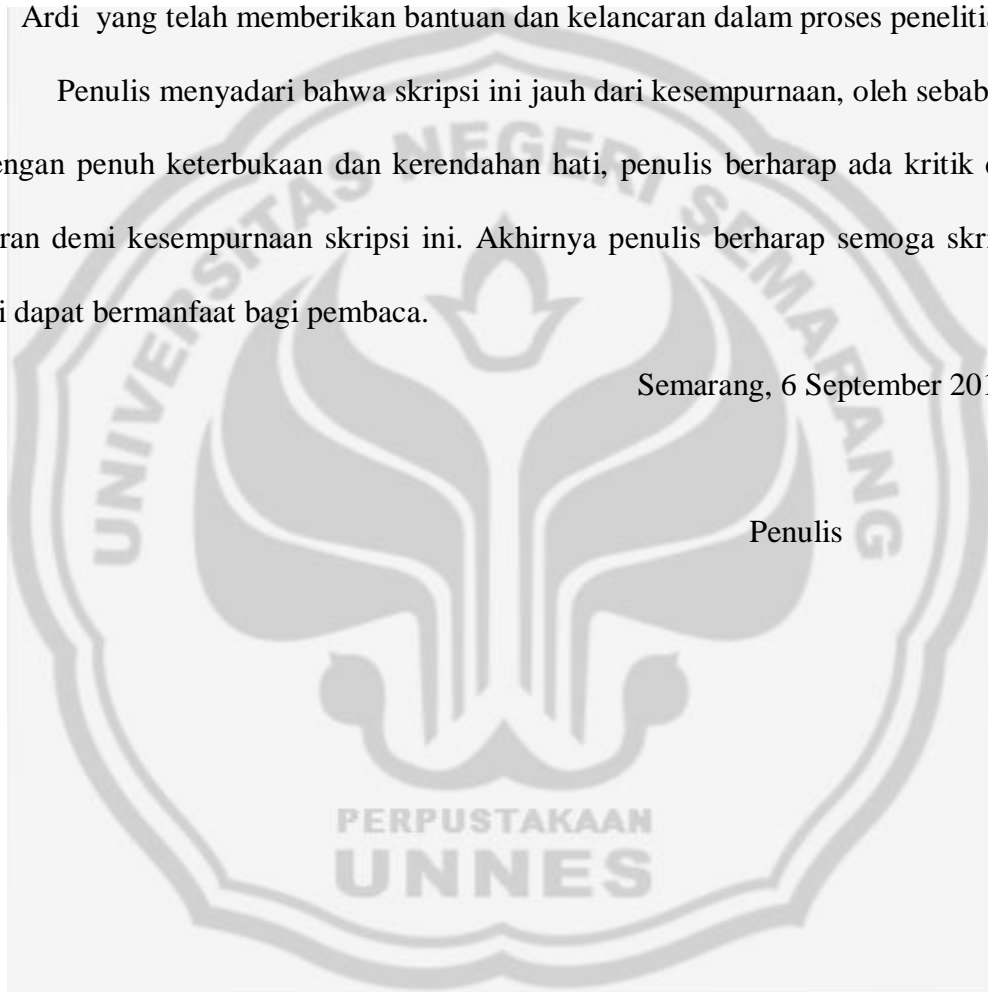
1. Dr. Agus Yulianto, M.Si Selaku Pembimbing I juga sebagai Kaprodi Jurusan Fisika yang penuh kesabaran dan pengertian memberikan bimbingan selama pembuatan skripsi.
2. Sunarno, S.Si, M.Si, Selaku Pembimbing II yang penuh kesabaran dan pengertian memberikan bimbingan selama pembuatan skripsi.
3. Prof Dr. Sudijono Sastroatmojo, M.Si, selaku Rektor Universitas Negeri Semarang yang telah memberikan kesempatan kepada penulis untuk menyelesaikan studi saya di Universitas Negeri Semarang.
4. Drs. Kasmadi Imam S,M.S, Selaku Dekan FMIPA Universitas Negeri Semarang yang telah membantu kelancaran administrasi dalam menyelesaikan penulisan skripsi saya.
5. Dr. Putut Marwoto, M.S, Selaku Ketua Jurusan Fisika FMIPA Universitas Negeri Semarang yang telah membantu kelancaran dalam penyelesaian skripsi saya.

6. Isa Akhlis, M.Si, Selaku Dosen Wali yang telah memberi petunjuk selama saya menjadi mahasiswa.
7. Edy, Rudi, Joko, Nunung, keluarga besar Pak Abdul Kholik, Ana, Ucup dan Ardi yang telah memberikan bantuan dan kelancaran dalam proses penelitian.

Penulis menyadari bahwa skripsi ini jauh dari kesempurnaan, oleh sebab itu dengan penuh keterbukaan dan kerendahan hati, penulis berharap ada kritik dan saran demi kesempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat bagi pembaca.

Semarang, 6 September 2011

Penulis



ABSTRAK

Hudson, Muhammad Denny Steve Bob. 2011. *Aplikasi Pencitraan Data Digital Pada Objek Sayap Capung Menggunakan Pemograman Matlab Telah Dilakukan.* Skripsi, Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Utama : Dr. Agus Yulianto, M.S, Pembimbing Pendamping : Sunarno, S.Si, M.Si.

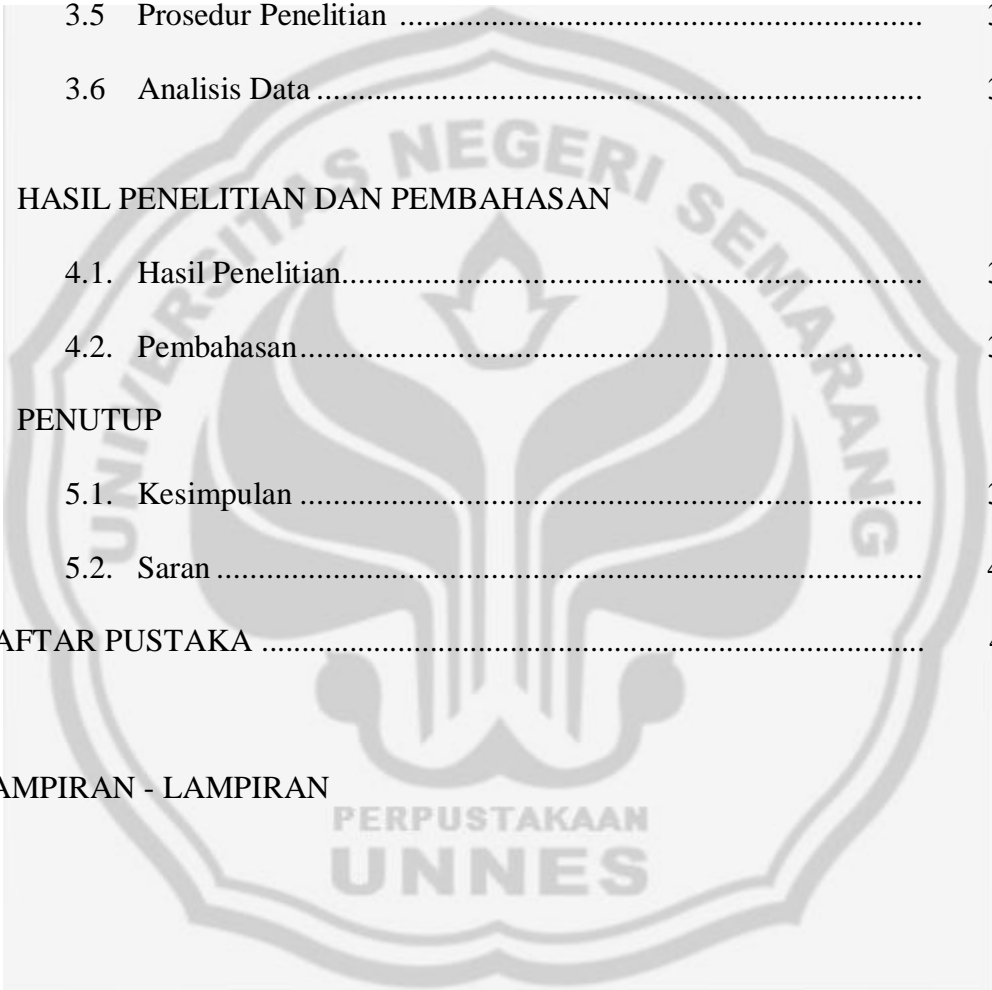
Kata kunci: Sayap Capung, Pemograman Matlab, Fungsi Transformasi Fourier.

Aplikasi pencitraan data digital pada objek sayap capung menggunakan pemograman matlab telah dilakukan. Studi ini dilakukan menggunakan tiga jenis sayap capung untuk menentukan fungsi transformasi fouriernya. Ketiga sampel sayap capung memiliki pola periodik *square*. Fungsi transformasi fourier dari suatu pola *square* sayap capung tergantung pada lebar pola periodik sayap capung tersebut. Semakin panjang jarak $a-b$ pada pola *square* sayap, maka grafik transformasi fourier semakin rapat. Sebaliknya semakin pendek jarak $a-b$ pada *square* pola sayap, maka grafik transformasi fourier semakin renggang.

DAFTAR ISI

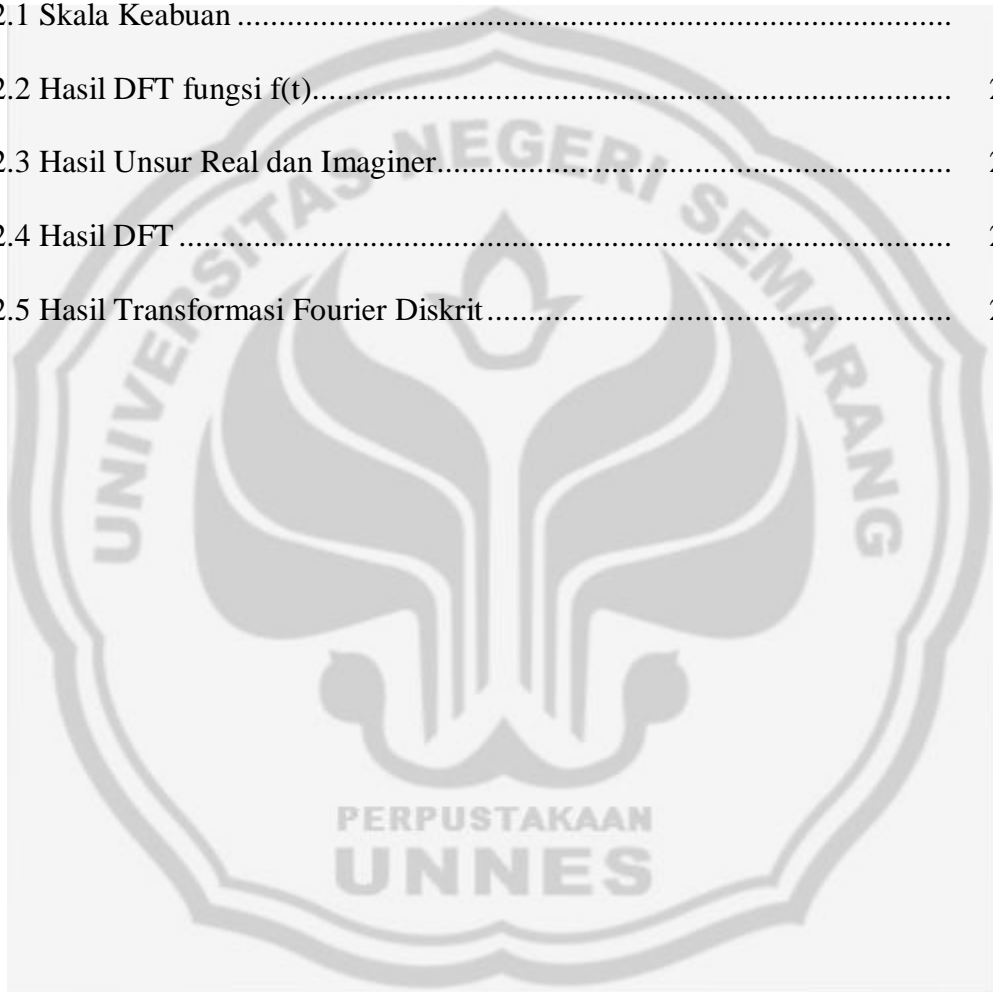
	Halaman
PRAKATA	vii
ABSTRAK	ix
DARTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xv
BAB	
1. PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Sistematika Skripsi	2
2. TINJAUAN PUSTAKA	
2.1 Spesifikasi Capung	5
2.2 Pengolahan Citra	8
2.3 Skala Keabuan.....	9
2.4 Histogram.....	10
2.5 MATLAB.....	12
3. METODE PENELITIAN	
3.1 Metode Pengumpulan Data.....	30

3.2	Waktu dan Tempat Penelitian	30
3.3	Alat dan Bahan Penelitian.....	30
3.4	Diagram Alir Penelitian	31
3.5	Prosedur Penelitian	32
3.6	Analisis Data	32
4.	HASIL PENELITIAN DAN PEMBAHASAN	
4.1.	Hasil Penelitian.....	33
4.2.	Pembahasan.....	37
5.	PENUTUP	
5.1.	Kesimpulan	39
5.2.	Saran	40
	DAFTAR PUSTAKA	41
	LAMPIRAN - LAMPIRAN	



DAFTAR TABEL

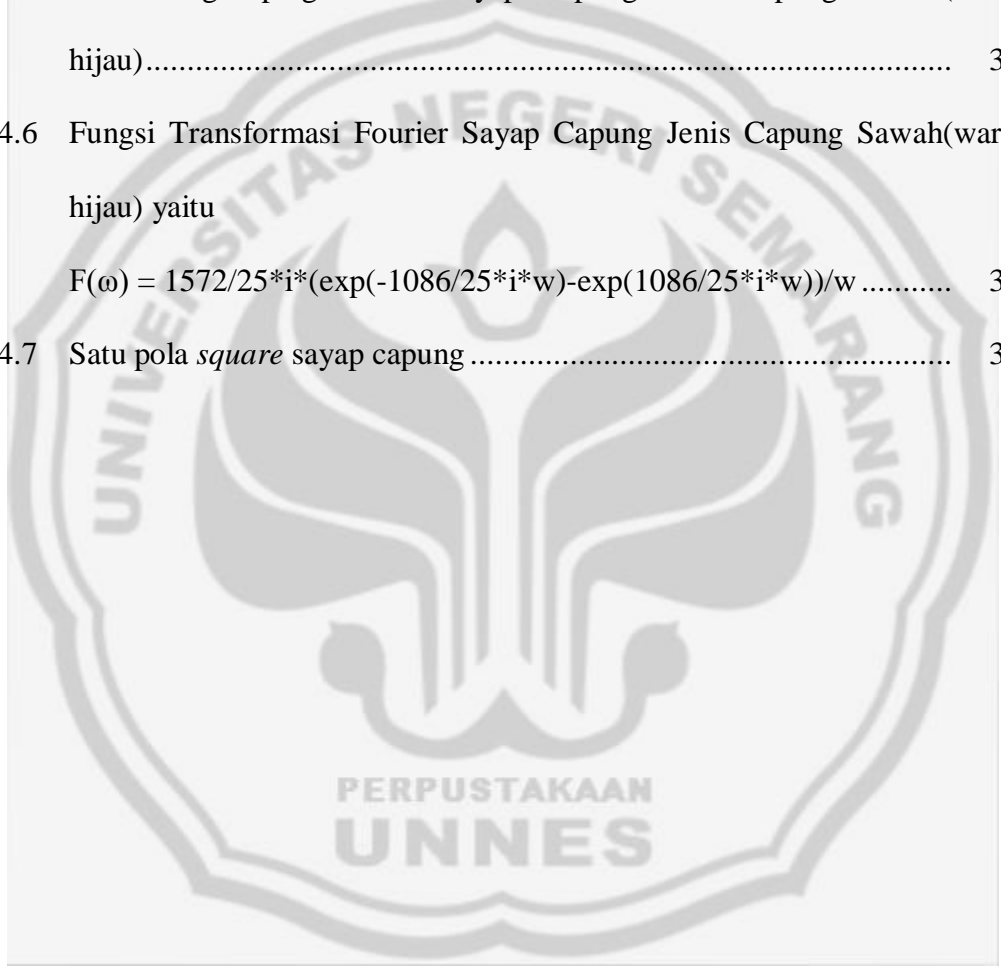
Tabel	Halaman
2.1 Skala Keabuan	9
2.2 Hasil DFT fungsi $f(t)$	22
2.3 Hasil Unsur Real dan Imaginer.....	23
2.4 Hasil DFT	25
2.5 Hasil Transformasi Fourier Diskrit.....	27



DAFTAR GAMBAR

Gambar	Halaman
2.1 Capung.....	7
2.2 Grafik Histogram	10
2.3 Transfromasi Fourier	16
2.4 Contoh Hasil Transformasi Fourier.....	18
2.5 Contoh Hasil Transformasi Fourier 2D.....	19
2.6 Hasil Transformasi Fourier Dalam Surface	20
2.7 DFT Bagian Real.....	23
2.8 DFT Bagian Imaginer.....	24
2.9 DFT Magnitude.....	24
2.10 DFT Phase	23
2.11 Contoh Citra Dalam $f(x,y)$	27
2.12 Bagian Real Hasil DFT 2D.....	28
2.13 Bagian Imaginer Hasil DFT 2D.....	28
2.14 Hasil DFT 2D Magnitude	29
2.15 Hasil DFT 2D Phase.....	29
3.1 Diagram blok perancangan sistem pengolahan citra sayap capung	31
4.1 Hasil Pengkropingan Pola Sayap Capung Jenis Capung Jarum	33
4.2 Fungsi Transformasi Fourier Sayap Capung Jenis Capung Jarum yaitu $F(\omega) = 213/10 * i * (\exp(-4943/100 * i * w) - \exp(4943/100 * i * w)) / w$	34
4.3 Hasil Pengkropingan Pola Sayap Capung Jenis Capung Sungai	34

4.4	Fungsi Transformasi Fourier Sayap Capung Jenis Capung Sungai(warna orange) yaitu	
	$F(\omega) = 562/5*i*(\exp(-12643/200*i*w)-\exp(12643/200*i*w))/w$	35
4.5	Hasil Pengkropingan Pola Sayap Capung Jenis Capung Sawah(warna hijau).....	35
4.6	Fungsi Transformasi Fourier Sayap Capung Jenis Capung Sawah(warna hijau) yaitu	
	$F(\omega) = 1572/25*i*(\exp(-1086/25*i*w)-\exp(1086/25*i*w))/w$	36
4.7	Satu pola <i>square</i> sayap capung	37



DAFTAR LAMPIRAN

Lampiran	Halaman
1. Pemograman MATLAB	41
2. Pengkropingan Citra Digital Pola Sayap Capung	47
3. Hasil Transformasi Fourier Sayap Capung	49



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Capung merupakan salah satu jenis serangga yang banyak dijumpai. Capung serangga yang tidak menggigit ataupun bersengat. Seperti serangga pada umumnya, tubuh capung terdiri dari tiga bagian. Kepala dengan mata faset (mata majemuk), dada atau thorax dengan empat sayap panjang yang tidak bisa dilipat dan dilengkapi tiga pasang kaki, dan abdomen dengan sepuluh segmen.

Masing-masing bagian tubuh capung mempunyai fungsi yang berbeda-beda. Pada penelitian ini hanya dikhususkan pada sayap capung yang akan diolah ke dalam pencitraan data digital.

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Pemanfaatan pengolahan citra dalam era yang hampir serta digital ini sangat dibutuhkan. Analisa komposisi suatu citra dapat berdampak terhadap perubahan kehidupan. Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diterjemahkan oleh manusia atau mesin (dalam hal ini komputer). Pada sayap capung jika dilihat dengan mata telanjang akan terlihat biasa, akan tetapi jika menggunakan pencitraan data digital akan terlihat lebih spesifik, sehingga dapat diolah menjadi diagram fungsi. Sayap capung memiliki bentuk yang lebih sederhana dibandingkan bagian tubuh yang lain seperti mata, kepala dan badan

capung sehingga harapannya memiliki fungsi-fungsi *Fourier Transform* yang juga lebih sederhana (studi awal).

Dengan latar belakang tersebut, peneliti tertarik untuk menyusun skripsi dengan judul “**APLIKASI PENCITRAAN DATA DIGITAL PADA OBJEK SAYAP CAPUNG MENGGUNAKAN PEMOGRAMAN MATLAB**”

1.2 Perumusan Masalah

Permasalahan yang dapat diambil dalam penelitian ini adalah Bagaimanakah fungsi *Fourier Transform* dari sayap capung?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menentukan fungsi *Fourier Transform* sayap capung dari tiga *species* capung.

1.4 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini dapat menemukan fungsi-fungsi *Fourier Transform* sayap capung dari 3 *species* capung.

1.5 Sistematika Penulisan Skripsi

Sistematika dalam skripsi ini disusun dengan tujuan agar pokok-pokok masalah yang dibahas dapat urut, terarah dan jelas. Sistematika skripsi ini terdiri dari tiga bagian yaitu : bagian awal, bagian isi dan bagian akhir.

1.5.1 Bagian awal skripsi

Bagian ini berisi halaman judul, sari, persetujuan pembimbing, halaman pengesahan, halaman pernyataan, halaman motto, halaman persembahan, halaman abstrak, kata pengantar, daftar isi, daftar tabel, daftar gambar dan daftar lampiran.

1.5.2 Bagian isi skripsi

Bagian ini terdiri dari 5 (lima) bab yang meliputi :

1.5.2.1 Bab 1 Pendahuluan

Bab ini memuat latar belakang, penegasan istilah, permasalahan, tujuan penelitian, manfaat penelitian, dan sistematika penulisan skripsi.

1.5.2.2 Bab 2 Landasan Teori

Bab ini terdiri dari kajian mengenai landasan teori yang mendasari permasalahan skripsi ini serta penjelasan yang merupakan landasan teori yang diterapkan dalam skripsi dan pokok-pokok bahasan yang terkait dalam pelaksanaan penelitian.

1.5.2.3 Bab 3 Metode Penelitian

Bab ini menguraikan metode penelitian yang digunakan dalam penyusunan skripsi. Metode penelitian ini meliputi: penentuan objek penelitian, variabel penelitian, alat dan bahan penelitian, prosedur penelitian, waktu dan tempat penelitian, dan metode analisis data.

1.5.2.4 Bab 4 Hasil Penelitian dan Pembahasan

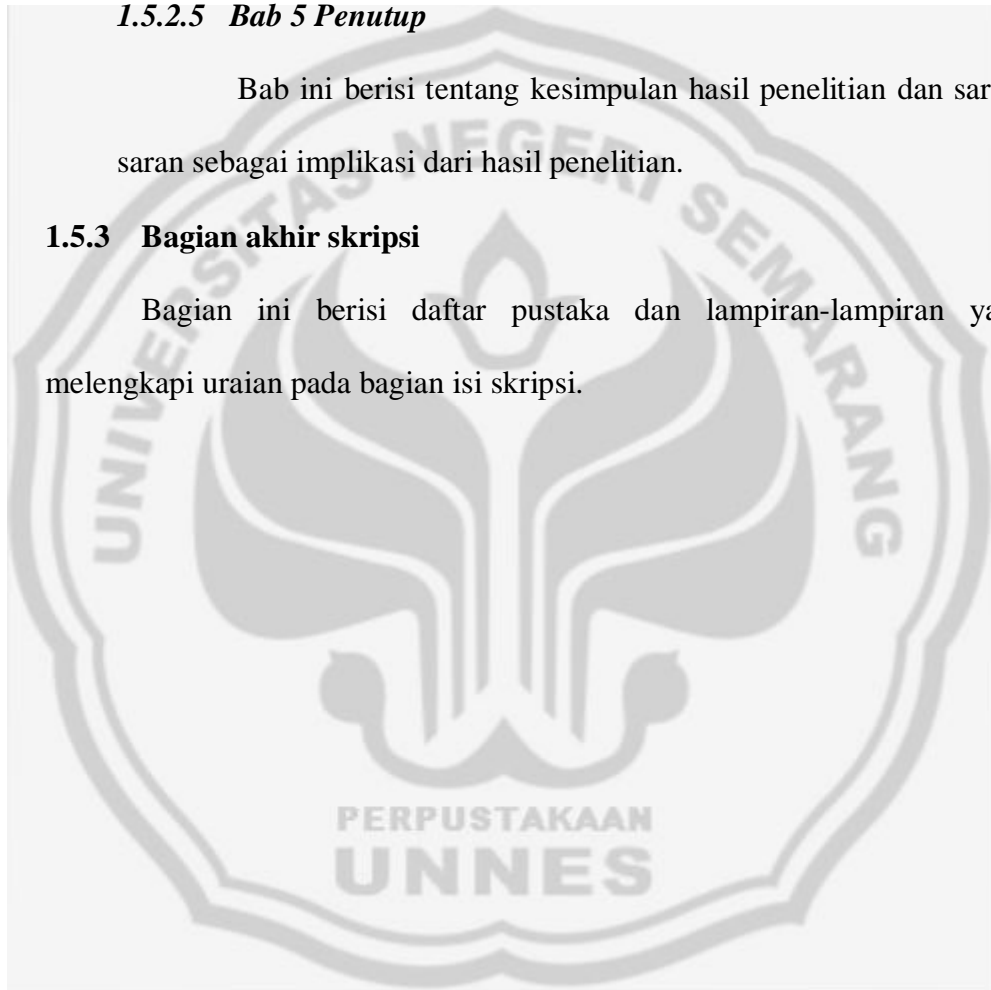
Bab ini berisi tentang pelaksanaan penelitian, semua hasil penelitian yang dilakukan dan pembahasan terhadap hasil penelitian.

1.5.2.5 Bab 5 Penutup

Bab ini berisi tentang kesimpulan hasil penelitian dan saran-saran sebagai implikasi dari hasil penelitian.

1.5.3 Bagian akhir skripsi

Bagian ini berisi daftar pustaka dan lampiran-lampiran yang melengkapi uraian pada bagian isi skripsi.



BAB 2

TINJAUAN PUSTAKA

2.1 Spesifikasi Capung

Sayap capung tidak dapat dilipat pada tubuhnya. Selain itu, cara otot terbang digunakan ketika sayap bergerak, berbeda dengan kebanyakan serangga lainnya. Karena sifat ini, para evolusionis menyatakan bahwa capung adalah “serangga terbelakang.”

Padahal sebaliknya, sistem terbang makhluk yang disebut “serangga terbelakang” ini tidak lain adalah keajaiban perancangan. Pembuat helikopter terbaik dunia, Sikorsky, menuntaskan perancangan satu dari helikopter mereka dengan menjadikan capung sebagai model.⁶ IBM, mitra Sikorsky dalam proyek ini memulai dengan menempatkan suatu model capung ke dalam komputer (IBM 3081). Dua ribu jenis penggambaran khusus dilakukan di komputer dalam hal manuver (gerakan jungkir balik) capung di udara. Jadi, model helikopter Sikorsky yang ditujukan untuk pengangkutan tentara dan persenjataan telah dibuat berdasarkan contoh yang berasal dari capung.

Gilles Martin, seorang fotografer alam, telah melakukan pengamatan 2 tahun untuk meneliti capung, dan dia juga menyimpulkan bahwa makhluk ini memiliki cara terbang yang sangat rumit.

Tubuh capung menyerupai bentuk pilin yang terbungkus logam. Dua sayapnya saling silang pada badannya yang menampakkan bias warna dari biru

muda hingga merah marun. Karena bentuk begini, capung dilengkapi dengan kemampuan manuver yang luar biasa. Tak peduli pada kecepatan atau arah bagaimana pun ia telah bergerak, capung dapat mendadak berhenti dan mulai terbang kembali dengan arah berlawanan. Atau, capung dapat tetap diam di udara untuk berburu. Pada kedudukan seperti itu, ia dapat bergerak dengan sangat cepat menuju mangsanya. Ia dapat mempercepat gerakannya hingga kecepatan yang sangat mengejutkan untuk seekor serangga: 25 mil per jam (40 kilometer/jam), yang dapat disejajarkan dengan seorang atlet lari 100 meter di Olimpiade dengan kecepatan 24,4 mil per jam (39 kilometer/jam).

Pada kecepatan ini, capung bertabrakan dengan mangsanya. Guncangan tabrakan ini sangat kuat. Namun, ketahanan capung sangat lentur sekaligus tahan terhadap benturan. Bentuk yang lentur dari tubuhnya meredam guncangan benturan. Sebaliknya, hal yang sama tidak akan terjadi pada mangsanya. Mangsa capung akan kehilangan kesadaran atau bahkan mati karena benturan itu.

Menyusul benturan ini, kaki belakang capung berperan sebagai senjatanya yang paling mematikan. Kaki menjulur ke depan dan menangkap mangsa yang kaget, kemudian dengan tangkas dicabik-cabik dan dimakan dengan rahangnya yang kuat.

Penglihatan capung sama mengesankannya dengan kemampuannya menunjukkan manuver mendadak pada kecepatan tinggi. Mata capung diakui sebagai contoh terbaik di antara semua serangga. Capung memiliki sepasang mata, tiap matanya memiliki sekitar 30 ribu lensa berbeda. Dua mata nyaris bulat, masing-masing hampir separuh ukuran kepalanya, memberi serangga ini wilayah

pandang yang sangat luas. Karena mata-mata ini, capung hampir selalu dapat mengetahui keadaan di belakangnya.

Karena itu, capung merupakan gabungan sistem-sistem, yang masing-masingnya memiliki bentuk tersendiri dan sempurna. Tidak berjalannya salah satu saja dari sistem-sistem ini akan merusak sistem yang lainnya juga. Walaupun begitu, seluruh sistem ini diciptakan tanpa cacat, sehingga makhluk ini tetap bertahan.



Gambar 2.1 Capung

Bagian tubuh yang paling penting dari capung adalah sayapnya. Akan tetapi, tidaklah mungkin menggunakan model evolusi perkembangan untuk menjelaskan cara terbang yang memungkinkan penggunaan sayap ini. Pertama, teori evolusi tidak punya penjelasan tentang masalah asal mula sayap, karena sayap hanya dapat bekerja jika berkembang bersama sekaligus agar dapat bekerja dengan benar. (Yahya, Harun: 2002)

2.2 Pengolahan Citra

Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang.

Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang yang menyatakan hal ini adalah pengolahan citra (*image processing*). Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik (Munir 2004: 3).

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik – teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi masukannya adalah citra dan keluarannya adalah citra, namun citra keluaran mempunyai kualitas lebih baik dari pada citra masukan. Termasuk ke dalam bidang ini juga adalah pemampatan citra (Munir 2004: 5).

2.3 Skala Keabuan

Proses kuantisasi skala keabuan (0,L) menjadi G sebuah skala yang dinyatakan dengan suatu harga bilangan bulat (integer), biasanya G diambil perpangkatan dari 2,

$$G = 2^m \quad (2.1)$$

dalam hal ini,

G = skala keabuan

m = bilangan bulat positif

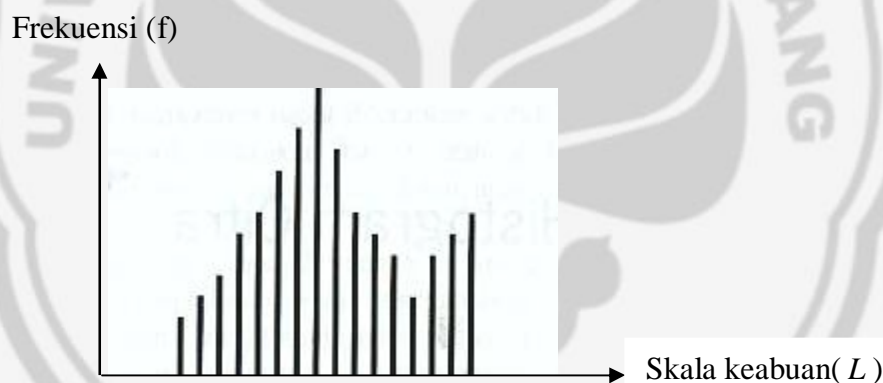
Skala Keabuan	Rentang Nilai Keabuan	<i>Pixel Depth</i>
2^1 (2 nilai)	0 sampai 1	1 bit
2^2 (4 nilai)	0 sampai 3	2 bit
2^3 (8 nilai)	0 sampai 7	3 bit
2^8 (256 nilai)	0 sampai 255	8 bit

Tabel 2.1 Skala Keabuan

Hitam dinyatakan dengan nilai skala keabuan terendah, yaitu 0, sedangkan putih dinyatakan dengan dengan nilai skala keabuan tertinggi, misalnya 15 untuk 16 level. Jumlah bit yang di butuhkan untuk merepresentasikan nilai keabuan *pixel* disebut kedalaman *pixel* (*pixel dept*). Jadi citra dalam kedalaman 8 bit disebut juga citra 8 bit (atau citra 256 warna). Pada kebanyakan aplikasi, citra hitam putih dikuantisasi pada skala 256 dan membutuhkan 1 byte (8 bit) untuk representasi setiap pixelnya ($G = 256 = 2^8$).

Besarnya skala keabuan yang digunakan menentukan resolusi kecerahan dari gambar yang diperoleh. Sebagai contoh, jika digunakan 3 bit untuk menyimpan harga bilangan bulat, maka jumlah skala keabuan yang diperoleh hanya 8 level, jika digunakan 4 bit, maka skala keabuan yang diperoleh adalah 16 level. Semakin banyak jumlah skala keabuan (berarti jumlah bit kuantisasinya makin banyak), semakin bagus citra yang diperoleh karena kemenerusan skala keabuan akan semakin tinggi sehingga mendekati citra aslinya (Munir 2004: 23).

2.4 Histogram



Gambar 2.2 Grafik Histogram

Histogram skala keabuan adalah fungsi yang menunjukkan informasi suatu citra. Absis (sumbu-x)-nya adalah skala keabuan, dan ordinat (sumbu-y)-nya adalah frekuensi kemunculan atau banyaknya titik dengan nilai keabuan tertentu. Citra digital yang memiliki L skala keabuan, yaitu mempunyai rentang nilai skala keabuan dari nilai 0 sampai $(L - 1)$. Misalnya pada citra dengan kuantisasi skala keabuan 8 bit, maka rentang nilai skala keabuannya dari 0 sampai 255.

Informasi penting mengenai isi citra digital dapat diketahui dengan membuat histogram citra. Histogram citra adalah grafik yang menggambarkan

penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relatif*) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras dari sebuah citra. Karena itu, histogram adalah alat bantu yang berharga dalam pekerjaan pengolahan citra baik secara kualitatif maupun kuantitatif (Munir 2004: 95). Histogram juga dapat digunakan untuk mengetahui kehomogenitasan dari suatu citra digital, yaitu dengan cara jika grafik histogram mempunyai rentang yang sempit berarti intensitas pixel - pixel penyusun step pada stepwedge (obyek) semakin mendekati nilai yang sebenarnya (nilai homogen). Sedangkan jika suatu citra tidak homogen maka histogram tersebut mempunyai rentang yang lebar, ini berarti semakin banyak intensitas pixel - pixel penyusun yang berbeda, karena semakin lebar histogram maka akan semakin menjauhi nilai yang sebenarnya (nilai homogen) (Hidayah 2009 : 84). Tingkat ketajaman/resolusi warna pada citra digital tergantung pada jumlah "bit" yang digunakan oleh komputer untuk merepresentasikan setiap pixel tersebut. Tipe yang sering digunakan untuk merepresentasikan citra adalah "8-bit citra" yaitu mempunyai jumlah skala keabuan 256 dengan rentang nilai skala keabuan 255 (0 untuk hitam - 255 untuk putih), tetapi dengan kemajuan teknologi perangkat keras grafik, kemampuan tampilan citra di komputer hingga 32 bit.

Kegunaan dari histogram citra adalah untuk melihat apakah distribusi informasi yang ada dalam suatu citra sudah baik atau belum. Contoh distribusi informasi dalam citra kurang baik adalah pada kasus dimana detail tekstur citra

kurang terlihat. Histogram juga banyak digunakan dalam *texture analysis*. *Texture analysis* adalah analisa untuk melihat apakah kedua tekstur sama atau berbeda. Misalkan seberapa mirip tekstur step pada stepwedge tingkat 1 dengan tekstur step pada stepwedge tingkat 2. Contoh lainnya adalah untuk melihat apakah pencahayaan dan *contrast* suatu citra sudah cukup atau belum (terlalu terang atau terlalu gelap). Histogram dari suatu citra terlalu terang biasanya cenderung mengumpul di nilai skala keabuan yang tinggi (ke arah nilai 255), sebaliknya *histogram* dari suatu citra yang terlalu gelap biasanya cenderung mengumpul di nilai skala keabuan yang rendah (ke arah nilai 0).

Pembagian citra menjadi ukuran tertentu menentukan resolusi (yaitu rincian skala keabuan yang dapat dilihat). Semakin tinggi resolusinya, yang berarti semakin kecil ukuran *pixel* (atau semakin banyak jumlah *pixel*-nya), semakin halus gambar yang diperoleh karena informasi yang hilang akibat pengelompokkan skala keabuan pada penyekenan semakin kecil (Munir 2004: 22).

2.5 MATLAB

MATLAB adalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang – bidang :

- 1) Matematika dan Komputasi

- 2) Pembentukan Algoritma
- 3) Akusisi Data
- 4) Pemodelan, Simulasi dan Pembuatan Prototipe
- 5) Analisa Data, Explorasi dan Visualisasi
- 6) Grafik Keilmuan dan Rekayasa

MATLAB merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu array sehingga tidak lagi kita dipusingkan dengan masalah dimensi. Hal ini memungkinkan kita untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, khususnya yang berhubungan dengan matrix dan formulasi vektor, yang mana masalah tersebut merupakan momok apabila kita harus menyelesaikannya dengan menggunakan bahasa level rendah seperti Pascall, C dan Basic.

Nama MATLAB merupakan singkatan dari *matrix laboratory*. MATLAB pada awalnya ditulis untuk memudahkan akses perangkat lunak matrik yang telah dibentuk oleh LINPACK dan EISPACK. Saat ini perangkat MATLAB telah menggabung dengan LAPACK dan BLAS library, yang merupakan satu kesatuan dari sebuah seni tersendiri dalam perangkat lunak untuk komputasi matrix.

Dalam lingkungan perguruan tinggi, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Di industri, MATLAB merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tinggi, pengembangan dan analisisnya.

Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih kita kenal dengan nama *toolbox*. Sangat penting bagi seorang pengguna Matlab, toolbox mana yang mendukung untuk *learn* dan *apply* teknologi yang sedang dipelajarinya. Toolbox toolbox ini merupakan kumpulan dari fungsi-fungsi MATLAB (Mfiles) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas particular. Area-area yang sudah bisa dipecahkan dengan toolbox saat ini meliputi pengolahan sinyal, system kontrol, neural networks, fuzzy logic, wavelets, dan lain-lain.

Sebagai sebuah sistem, MATLAB tersusun dari 5 bagian utama :

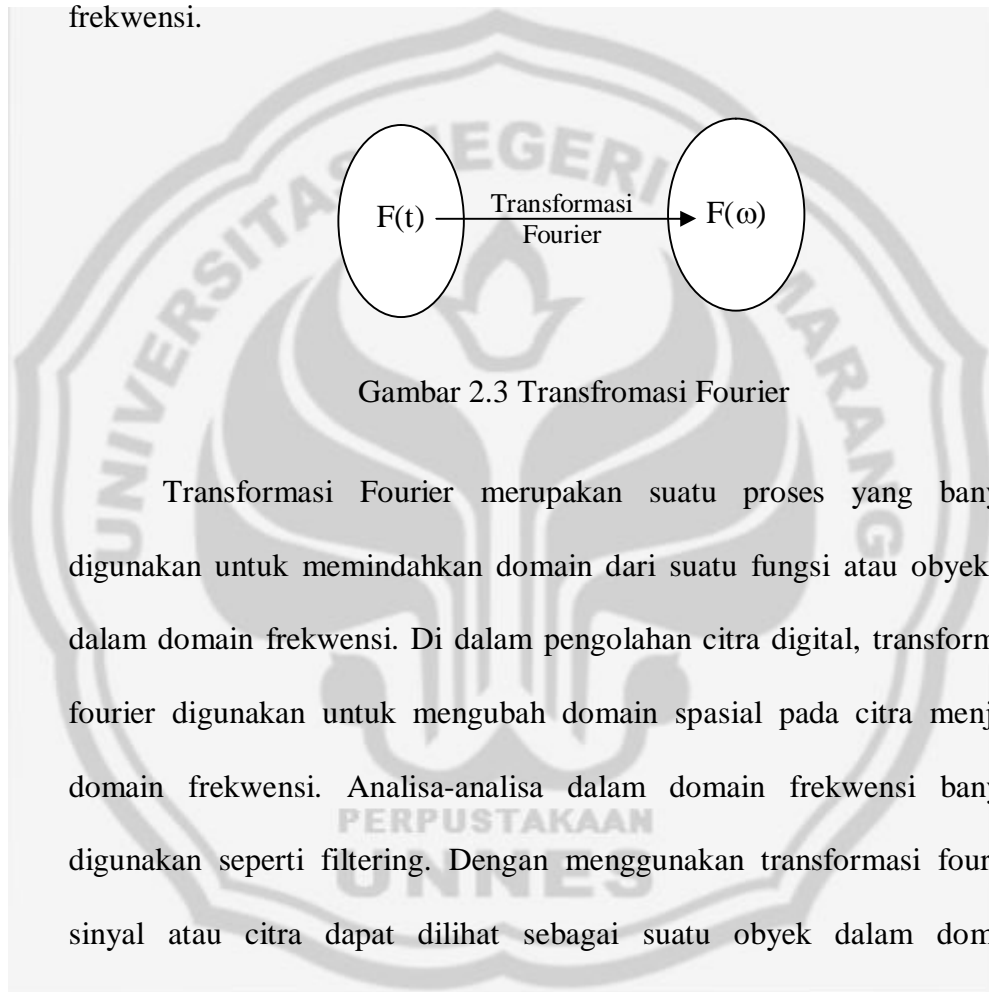
- a. **Development Environment.** Merupakan sekumpulan perangkat dan fasilitas yang membantu untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah graphical user interfaces (GUI). Termasuk didalamnya adalah MATLAB desktop dan Command Window, command history, sebuah editor dan debugger, dan browsers untuk melihat help, workspace, files, dan search path.
- b. **MATLAB Mathematical Function Library.** Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: sum, sin, cos, dan complex arithmetic, sampai dengan fungsi – fungsi yang lebih komplek seperti matrix inverse, matrix eigenvalues, Bessel functions, dan fast Fourier transforms.
- c. **MATLAB Language.** Merupakan suatu high-level matrix/array language dengan control flow statements, functions, data structures,

input/output, dan fitur-fitur object-oriented programming. Ini memungkinkan untuk melakukan "pemrograman dalam lingkup sederhana " untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

- d. **Graphics.** MATLAB memiliki fasilitas untuk menampilkan *vector* dan *matrices* sebagai suatu grafik. Didalamnya melibatkan high-level functions (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, image processing, animation, dan presentation graphics. Ini juga melibatkan fungsi level rendah yang memungkinkan untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan graphical user interfaces pada aplikasi MATLAB.
- e. **MATLAB Application Program Interface (API).** Merupakan suatu library yang memungkinkan program yang telah ditulis dalam bahasa C dan Fortran mampu berinteraksi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan routines dari MATLAB (dynamic linking), pemanggilan MATLAB sebagai sebuah computational engine, dan untuk membaca dan menuliskan MAT-files. (Erick & Nataliani: 2007)

1. Transformasi Fourier

Transformasi Fourier adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekwensi.



Transformasi Fourier merupakan suatu proses yang banyak digunakan untuk memindahkan domain dari suatu fungsi atau obyek ke dalam domain frekwensi. Di dalam pengolahan citra digital, transformasi fourier digunakan untuk mengubah domain spasial pada citra menjadi domain frekwensi. Analisa-analisa dalam domain frekwensi banyak digunakan seperti filtering. Dengan menggunakan transformasi fourier, sinyal atau citra dapat dilihat sebagai suatu obyek dalam domain frekwensi.

a) Transformasi Fourier 1 Dimensi

Transformasi Fourier kontinu 1 Dimensi (1D) dari suatu fungsi waktu $f(t)$ didefinisikan dengan:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt \quad (2.2)$$

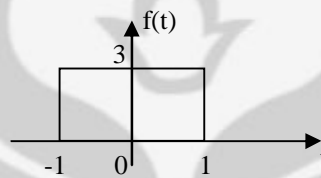
dimana $F(\omega)$ adalah fungsi dalam domain frekwensi ω adalah frekwensi radial $0 - 2\pi f$,

atau dapat dituliskan bahwa

$$\omega = 2\pi f \quad (2.3)$$

Contoh 1.1 :

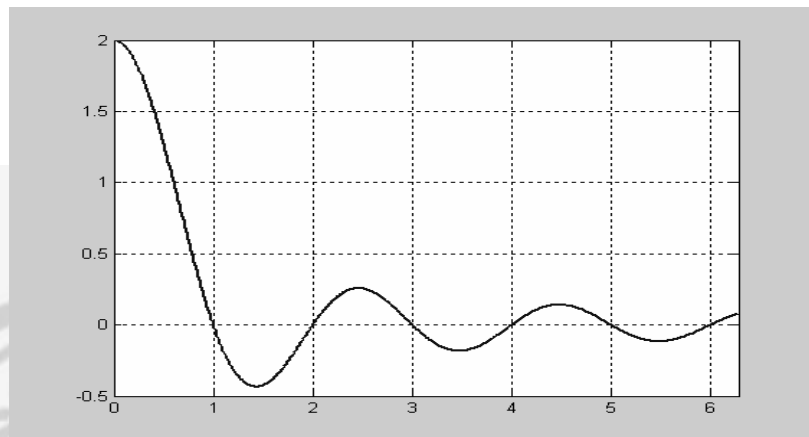
Diketahui fungsi $f(t)$ sebagai berikut:



Transformasi Fourier dari $f(t)$ di atas adalah:

$$\begin{aligned} F(\omega) &= \int_{-1}^1 (3)e^{-j\omega t} dt = 3 \int_{-1}^1 e^{-j\omega t} dt \\ &= -\frac{3}{j\omega} e^{-j\omega t} \Big|_{-1}^1 \\ &= -\frac{3}{j\omega} [e^{-j\omega} - e^{j\omega}] = \frac{6 \sin(\omega)}{\omega} \end{aligned}$$

Hasil dari transformasi Fourier untuk $\omega = 0$ s/d 2π adalah :



Gambar 2.4 Contoh hasil transformasi fourier

b) Transformasi Fourier 2 Dimensi

Transformasi Fourier kontinu 2 Dimensi (2D) dari suatu fungsi spasial $f(x,y)$ didefinisikan dengan:

$$F(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot e^{-j(\omega_1 x + \omega_2 y)} dx dy \quad (2.4)$$

dimana $F(\omega_1, \omega_2)$ adalah fungsi dalam domain frekwensi

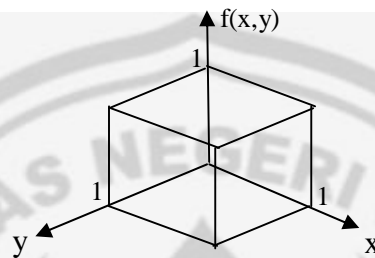
$f(x,y)$ adalah fungsi spasial atau citra

ω_1 dan ω_2 adalah frekwensi radial $0 - 2\pi$.

Transformasi fourier yang digunakan dalam pengolahan citra digital adalah transformasi fourier 2D.

Contoh 2.1 :

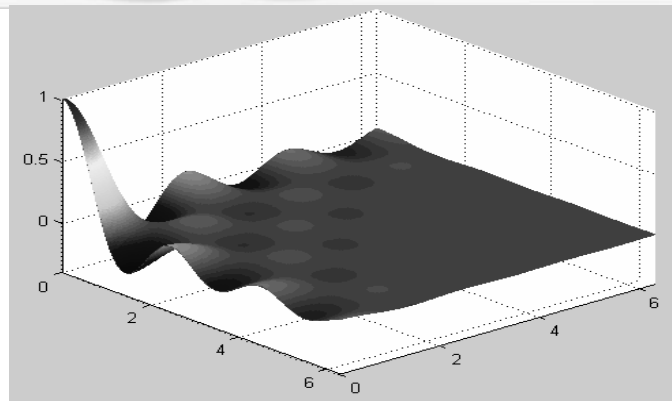
Diketahui fungsi spasial $f(x,y)$ berikut:



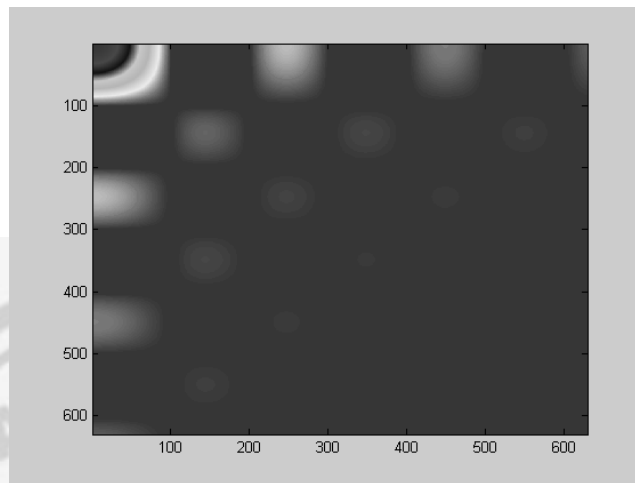
Transformasi fourier dari $f(x,y)$ di atas adalah:

$$\begin{aligned}
 F(\omega_1, \omega_2) &= \int_{-1}^1 \int_{-1}^1 (1) \cdot e^{-j(\omega_1 x + \omega_2 y)} dy dx \\
 &= \int_{-1}^1 \left[-\frac{e^{-j\omega_1 x}}{j\omega_2} e^{-j\omega_2 y} \right]_{-1}^1 dx = \int_{-1}^1 \frac{\sin(\omega_2)}{\omega_2} e^{-j\omega_1 x} dx \\
 &= \frac{\sin(\omega_2)}{\omega_2} \left[-\frac{e^{-j\omega_1 x}}{j\omega_1} \right]_{-1}^1 = \frac{\sin(\omega_2)}{\omega_2} \cdot \frac{\sin(\omega_1)}{\omega_1} \\
 &= \frac{\sin(\omega_2) \sin(\omega_1)}{\omega_2 \omega_1}
 \end{aligned}$$

Hasil dari transformasi fourier untuk $0 < \omega_1, \omega_2 < 2\pi$, adalah sebagai berikut :



Gambar 2.5 Contoh hasil transformasi fourier 2D



Gambar 2.6 Hasil transformasi fourier dalam surface
Transformasi Fourier semacam ini disebut dengan continuous fourier transform, dan sulit dikomputasi karena ada operasi integral dan sifat kontinunya itu sendiri.

2. Transformasi Fourier Diskrit

Transformasi fourier diskrit atau disebut dengan *Discrete Fourier Transform* (DFT) adalah model transformasi fourier yang dikenakan pada fungsi diskrit, dan hasilnya juga diskrit. DFT didefinisikan dengan :

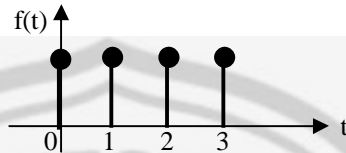
$$F(k) = \sum_{n=1}^N f(n).e^{-j2\pi knT/N} \quad (2.5)$$

a) Transformasi Fourier Diskrit 1 Dimensi

DFT seperti rumus di atas dinamakan dengan DFT 1 dimensi, DFT semacam ini banyak digunakan dalam pengolahan sinyal digital.

Contoh 1.1 :

Diketahui $f(t)$ dalam bentuk diskrit $f(n)$ sebagai berikut :



DFT dengan $T=1$ dari fungsi $f(n)$ di atas adalah :

$$k=0 \quad \rightarrow \quad F(0) = \sum_{n=0}^3 f(n) \cdot e^{-jn0} = \sum_{n=0}^3 f(n) \\ = 1 + 1 + 1 + 1 = 4$$

$$k=1 \quad \rightarrow \quad F(1) = \sum_{n=0}^3 f(n) \cdot e^{-j2\pi n/4} = \\ \sum_{n=0}^3 f(n) \cdot e^{-0.5j\pi n} = 0$$

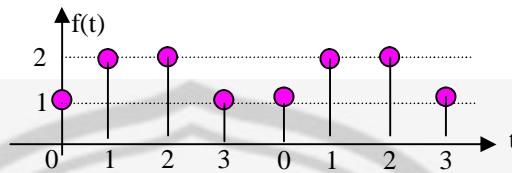
$$k=2 \quad \rightarrow \quad F(2) = \sum_{n=0}^3 f(n) \cdot e^{-j4\pi n/4} = \sum_{n=0}^3 f(n) \cdot e^{-j\pi n} = 0$$

$$k=3 \quad \rightarrow \quad F(3) = \sum_{n=0}^3 f(n) \cdot e^{-j6\pi n/4} = \sum_{n=0}^3 f(n) \cdot e^{-j1.5\pi n} = 0$$

Hasil dari DFT untuk T (periode sampling) yang berbeda akan juga berbeda. Sehingga dalam proses perhitungan DFT, penentuan nilai T juga merupakan perhatian penting. Sebagai acuan dapat digunakan aturan frekwensi Niquist bahwa frekwensi sampling minimal dua kali frekwensi informasi (data), atau dengan kata lain periode sampling maksimal setengah kali periode dari nilai fungsinya.

Contoh 1.2 :

Diketahui $f(t)$ dalam bentuk diskrit $f(n)$ sebagai berikut :



DFT dengan $T=1$ dari fungsi $f(n)$ di atas adalah :

$$F(k) = \sum_{n=0}^7 f(n) \cdot e^{-j2\pi nk/8} = \sum_{n=0}^7 f(n) \cdot e^{-j\pi nk/4}$$

Hasil DFT fungsi $f(t)$ di atas adalah :

k	F(k)
0	12
1	0
2	$-2 - 2j$
3	0
4	0
5	0
6	$-2 + 2j$
7	0

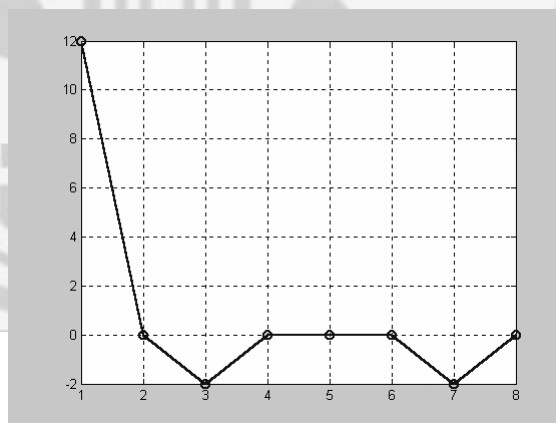
Tabel 2.2 Hasil DFT fungsi $f(t)$

Terlihat bahwa hasil dari DFT adalah bilangan kompleks, yang terdiri dari unsur real dan imajiner. Sehingga dapat dipisahkan dalam unsur real dan imajiner sebagai berikut :

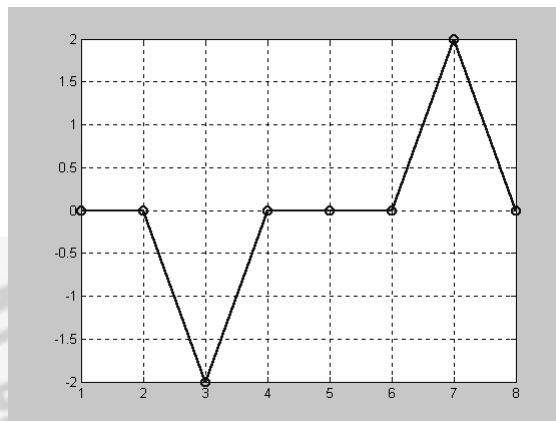
k	Real{F(k)}	Im{F(k)}
0	12	0
1	0	0
2	-2	-2
3	0	0
4	0	0
5	0	0
6	-2	2
7	0	0

Tabel 2.3 Hasil unsur real dan imajiner

Dan dapat digambarkan sebagai berikut :



Gambar 2.7 DFT Bagian Real

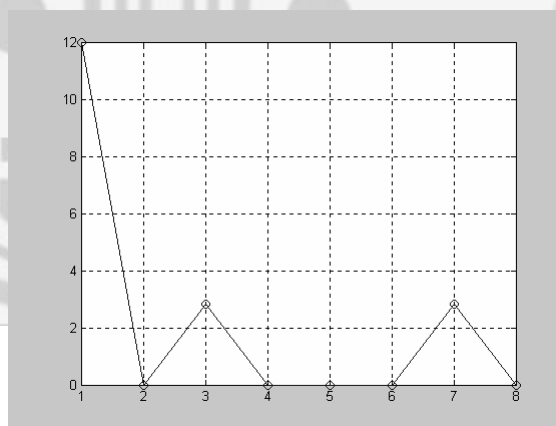


Gambar 2.8 DFT Bagian Imaginer

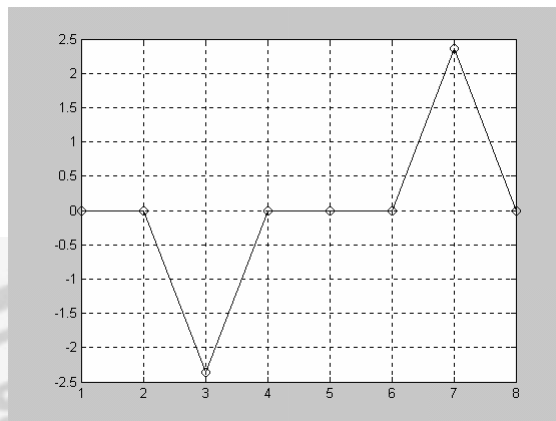
Atau dapat dinyatakan dalam *magnitude* dan *phase* dengan definisi sebagai berikut :

$$\text{Magnitude : } |F(k)| = \sqrt{(\text{Re}\{f(k)\})^2 + (\text{Im}\{f(k)\})^2}$$

$$\text{Phase : } \text{Arg}\{F(k)\} = \frac{\text{Im}\{F(k)\}}{\text{Re}\{F(k)\}}$$



Gambar 2.9 DFT Magnitude



Gambar 2.10 DFT Phase

Bila DFT dihitung untuk $k=0$ s/d 15 maka hasilnya adalah:

k	F(k)	K	F(k)
0	12	8	12
1	0	9	0
2	$-2 - 2j$	10	$-2 - 2j$
3	0	11	0
4	0	12	0
5	0	13	0
6	$-2 + 2j$	14	$-2 + 2j$
7	0	15	0

Tabel 2.4 Hasil DFT

Terlihat terjadi pengulangan hasil, hal ini disebabkan proses DFT memang mengakibatkan terjadinya periodik. Ini sebagai akibat dari adanya unsur radial 2π dalam bentuk transformasi

fourier. Sehingga dalam proses perhitungan DFT, perhitungan cukup dilakukan sampai 1/2 periodik saja. Dan perhitungan inilah yang dinamakan dengan FFT (Fast Fourier Transform).

b) Transformasi Fourier Diskrit 2 Dimensi

Transformasi Fourier Diskrit (DFT) 2 Dimensi adalah tranformasi fourier diskrit yang dikenakan pada fungsi 2D (fungsi dengan dua variabel bebas), yang didefinisikan sebagai berikut :

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) \cdot e^{-j2\pi T(k_1 n_1 / N_1 + k_2 n_2 / N_2)} \quad (2.6)$$

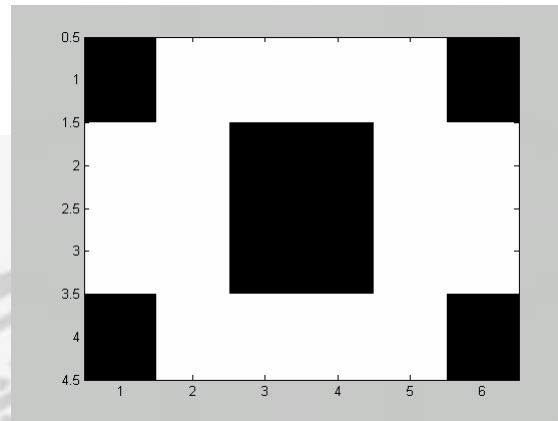
DFT 2D ini banyak digunakan dalam pengolahan citra digital, karena data citra dinyatakan sebagai fungsi 2D.

Contoh 2.1

Diketahui $f(x,y)$ adalah sebagai berikut :

0	1	1	1	1	0
1	1	0	0	1	1
1	1	0	0	1	1
0	1	1	1	1	0

Bila digambarkan, hasilnya sebagai berikut :



Gambar 2.11 Contoh citra dalam $f(x,y)$

Transformasi Fourier Diskrit dari fungsi diatas adalah :

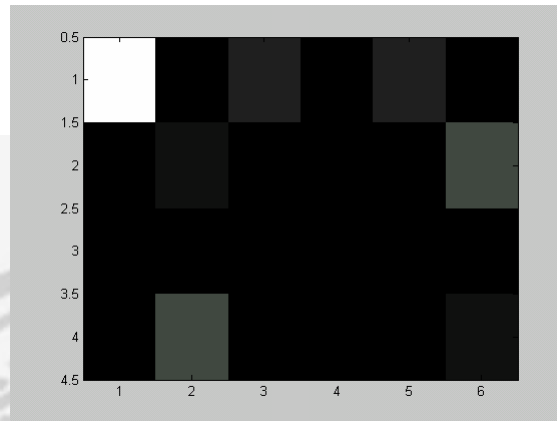
$$F(k_1, k_2) = \sum_{n_1=0}^4 \sum_{n_2=0}^6 f(n_1, n_2) \cdot e^{-j2\pi T(k_1 n_1 / 4 + k_2 n_2 / 6)} \quad (2.7)$$

Hasil dari Transformasi Fourier Diskrit adalah sebagai berikut :

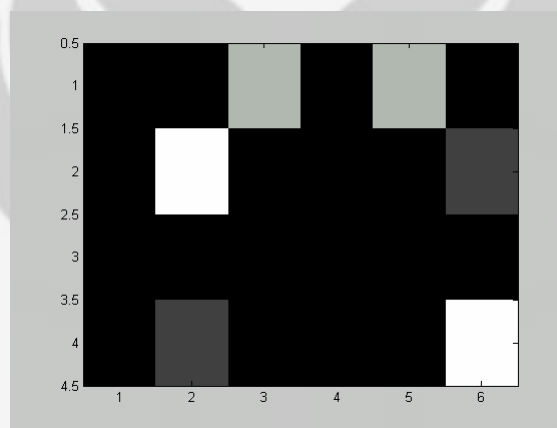
16	0	-2 - 3.46i	0	-2 + 3.46i	0
0	-1.27 - 4.73i	0	0	0	4.73 - 1.27i
0	0	0	0	0	0
0	-4.73 + 1.27i	0	0	0	1.27 + 4.73i

Tabel 2.5 Hasil Transformasi Fourier Diskrit

Secara grafis dapat ditunjukkan bahwa :



Gambar 2.12 Bagian Real hasil DFT 2D



Gambar 2.13 Bagian Imaginer hasil DFT 2D

Hasil DFT dalam bentuk magnitude dan phase adalah sebagai

berikut :

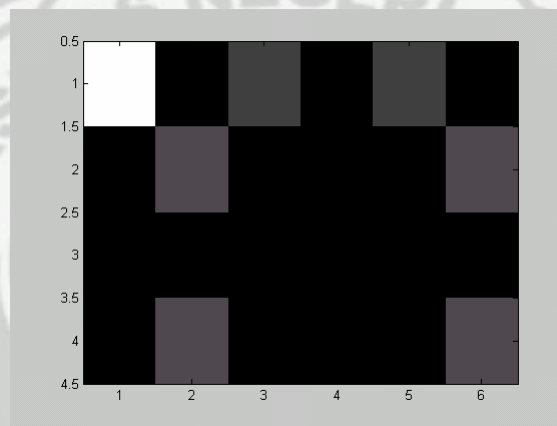
Magnitude =

16.0000	0	4.0000	0	4.0000	0
0	4.8990	0	0	0	4.8990
0	0	0	0	0	0
0	4.8990	0	0	0	4.8990

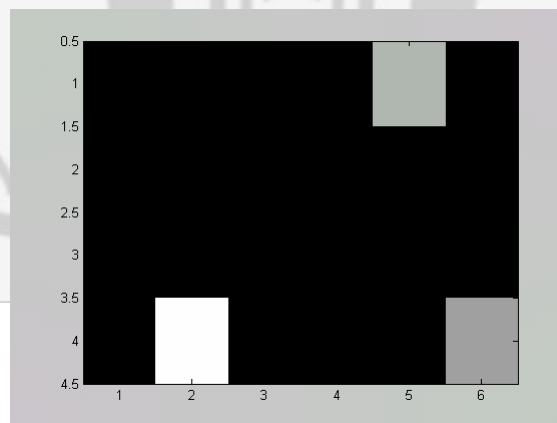
Phase =

0	0	-2.0944	0	2.0944	0
0	-1.8326	0	0	0	-2.8798
0	0	0	0	0	0
0	2.8798	0	0	0	1.8326

Secara grafis dapat digambarkan sebagai berikut :



Gambar 2.14 Hasil DFT 2D Magnitude



Gambar 2.15 Hasil DFT 2D Phase

(Zulkaryanto, Eko. 2002)

BAB 3

METODE PENELITIAN

3.1 Metode Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan pemotretan dan penelitian secara langsung di ruangan Lab.Fisika Universitas Negeri Semarang (UNNES).

3.2 Waktu dan Tempat Penelitian

Waktu : Juni 2011- selesai

Tempat : Lab.Fisika Universitas Negeri Semarang (UNNES).

3.3 Alat dan Bahan Penelitian

Penelitian ini melibatkan alat pemindai citra berupa kamera mikroskop dan seperangkat komputer. Objek sayap capung dipasang melintang dan kemudian difoto. Selanjutnya pengidentifikasian fungsi dari sayap capung menggunakan Transformasi Fourier dengan menggunakan perangkat lunak MATLAB.

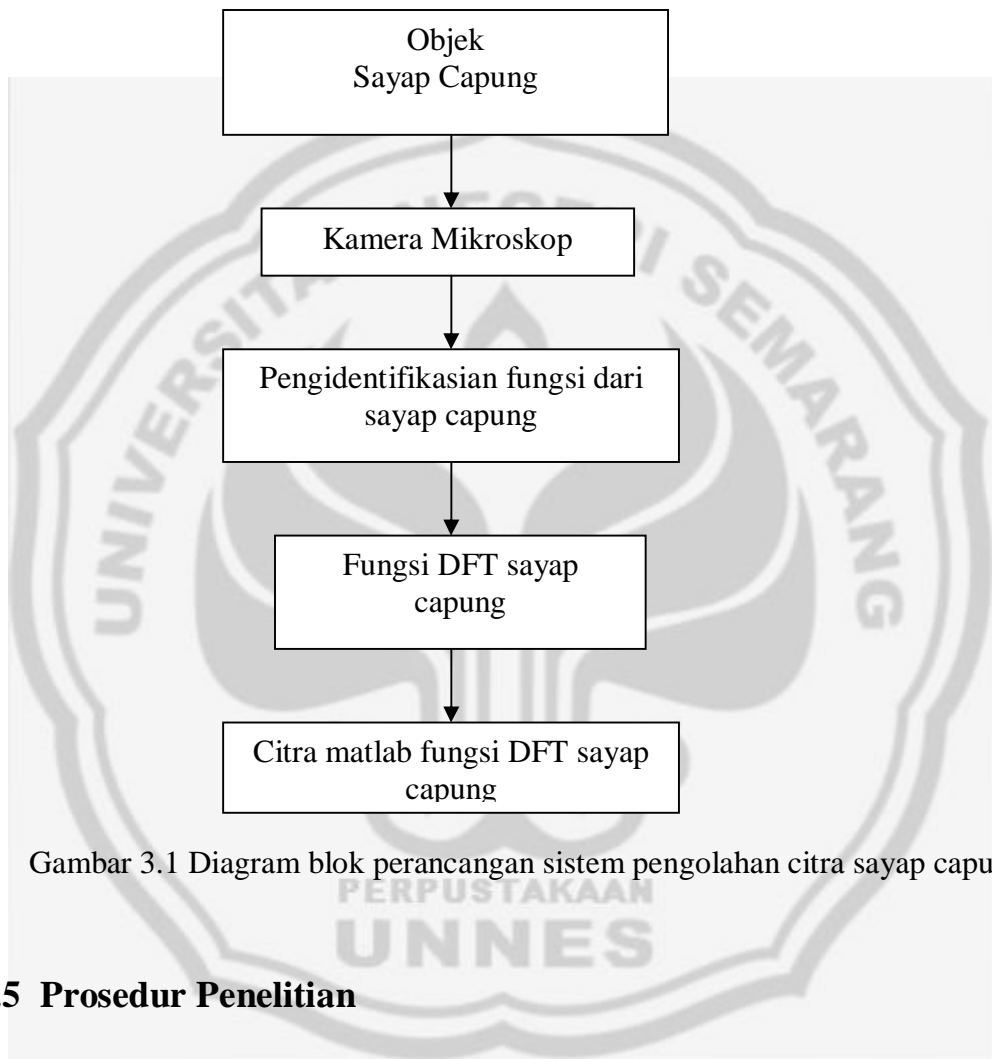
3.3.1 Alat yang dipergunakan berupa:

1. Seperangkat komputer
2. Kamera mikroskop(motic)

3.3.2 Bahan yang dipergunakan berupa:

1. Tiga jenis sayap capung

3.4 Diagram Alir Penelitian



Gambar 3.1 Diagram blok perancangan sistem pengolahan citra sayap capung

3.5 Prosedur Penelitian

- Mempersiapkan sayap capung
- Memotret sayap capung dengan menggunakan kamera mikroskop.
- Menentukan pola sayap capung
- Menentukan fungsi sayap capung
- Mensimulasikan sayap capung dengan menggunakan MATLAB

3.6 Analisis Data

Data dianalisis dengan cara membandingkan fungsi dari tiga jenis sayap capung, sehingga bisa ditentukan bentuk dan ukuran pola sayap capung dengan transformasi fouriernya.

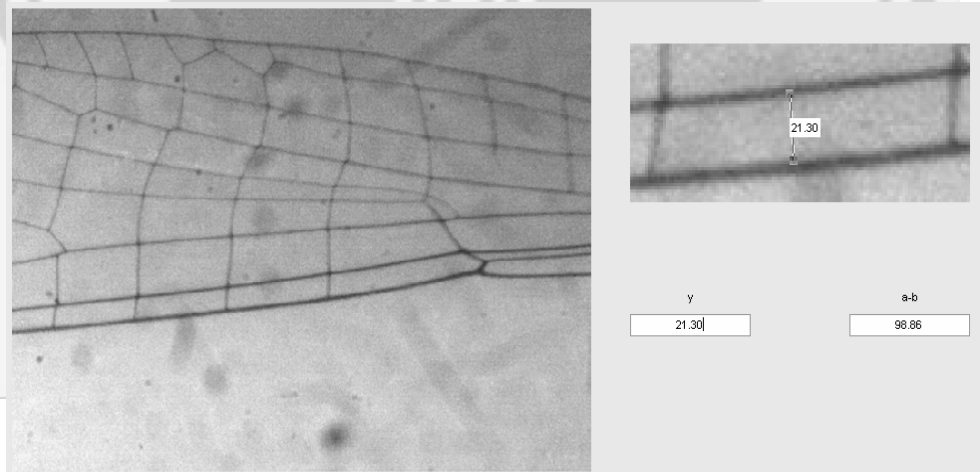


BAB 4

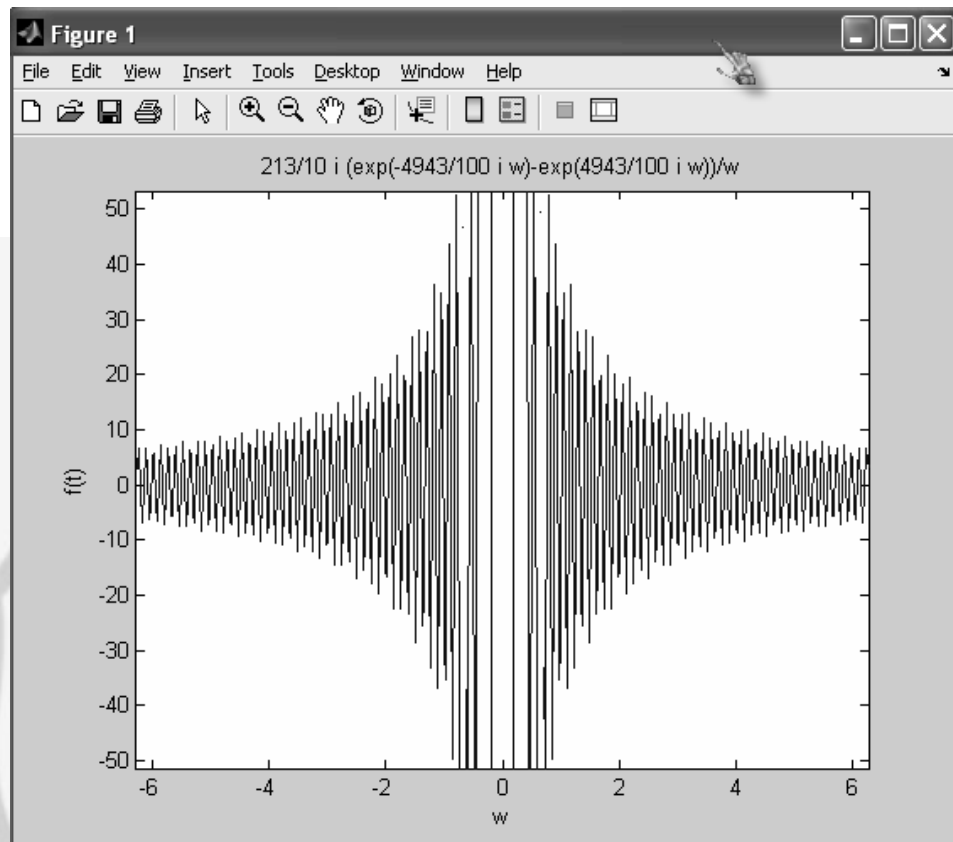
HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Dalam skripsi ini yang berjudul “Aplikasi Pencitraan Data Digital pada Objek Sayap Capung Menggunakan Pemograman Matlab” yang akan dicari fungsi dari ketiga jenis sayap capung. Dari penelitian ini terlebih dahulu menentukan salah satu jenis pola sayap capung dengan mengkroping salah satu pola *square* dengan menggunakan matlab. Sehingga diperoleh jarak dan lebar pola sayap capung. Hasil pengkropingan dan fungsi dari citra digital pola sayap capung dapat dilihat pada Gambar 4.1 sampai dengan 4.6

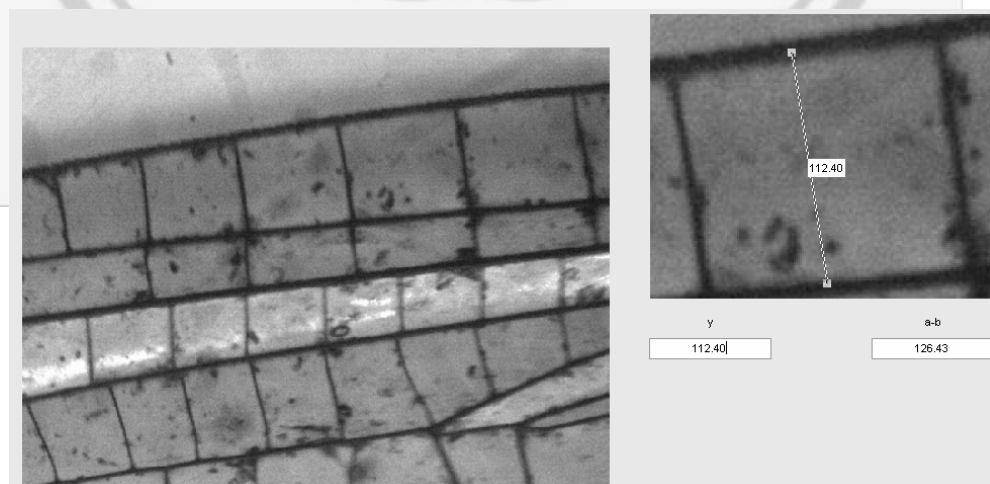


Gambar 4.1 Hasil Pengkropingan Pola Sayap Capung Jenis Capung Jarum.

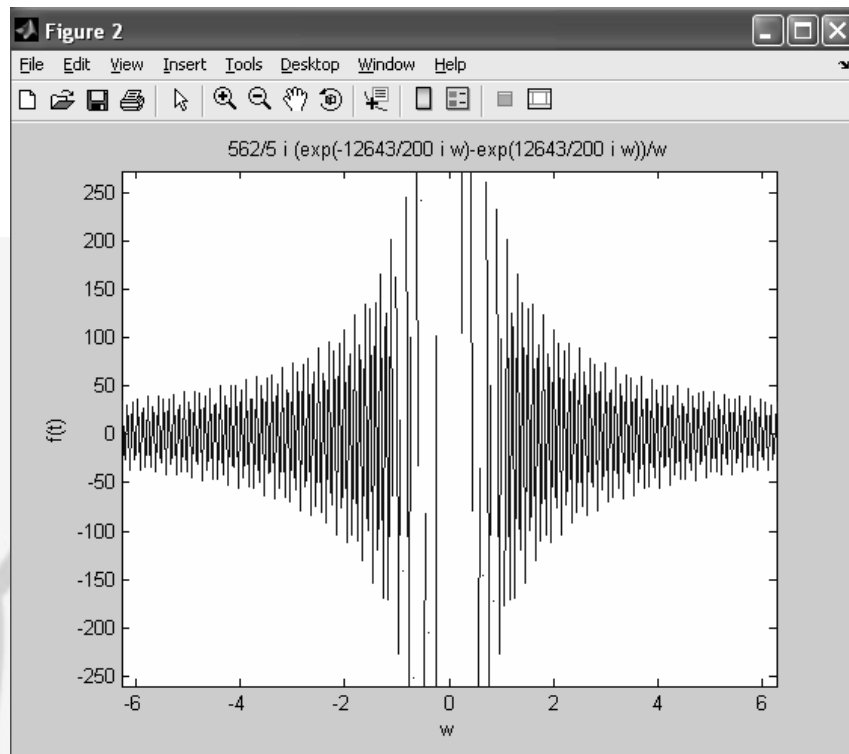


Gambar 4.2 Fungsi Transformasi Fourier Sayap Capung Jenis Capung yaitu

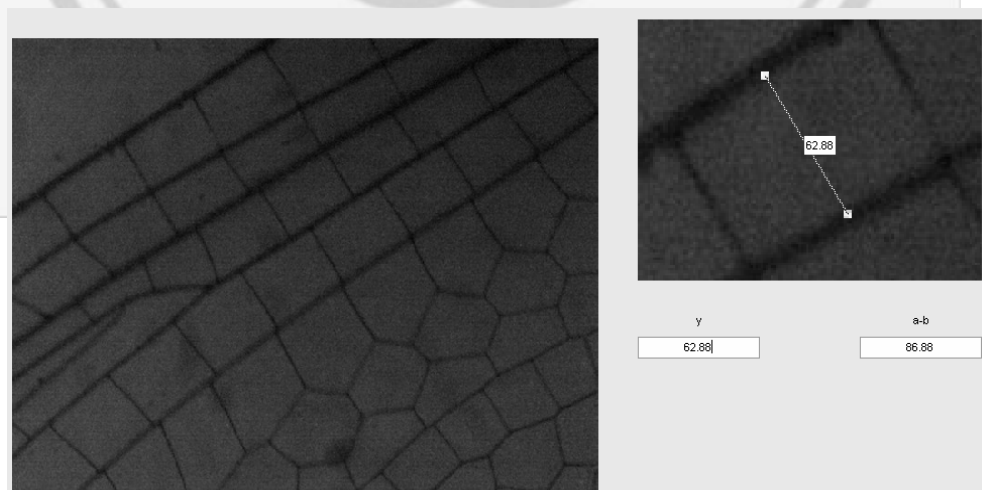
$$F(\omega) = 213/10 * i * (\exp(-4943/100 * i * \omega) - \exp(4943/100 * i * \omega)) / \omega$$



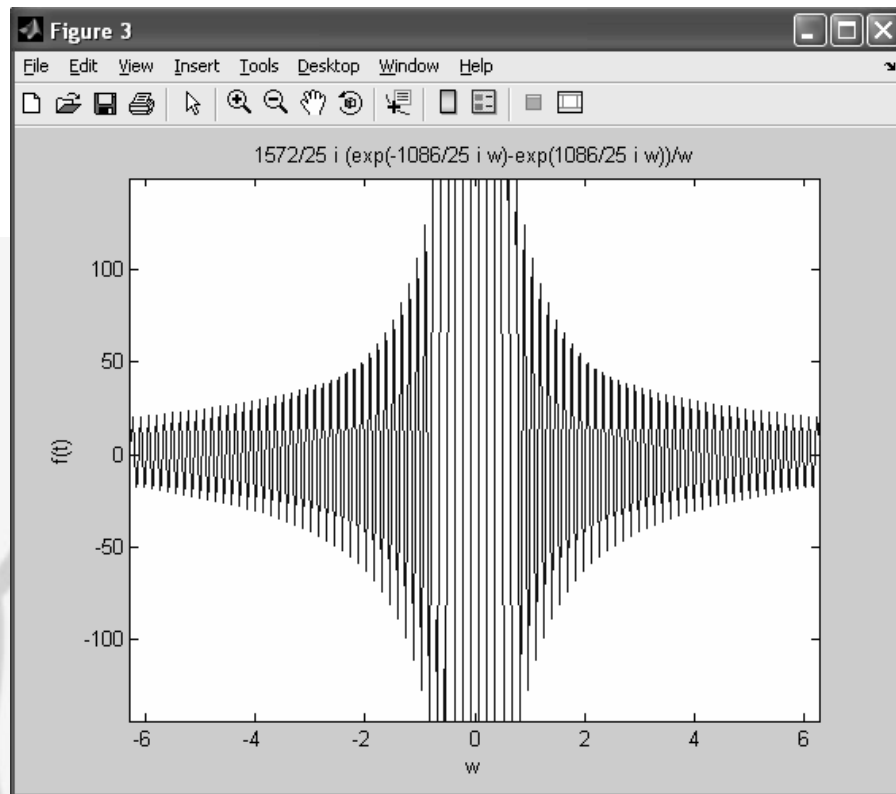
Gambar 4.3 Hasil Pengkropingan Pola Sayap Capung Jenis Capung Sungai(warna orange)



Gambar 4.4 Fungsi Transformasi Fourier Sayap Capung Jenis Capung Sungai(warna orange) yaitu $F(\omega) = \frac{562}{5}i \cdot (\exp(-12643/200 \cdot i \cdot \omega) - \exp(12643/200 \cdot i \cdot \omega)) / \omega$



Gambar 4.5 Hasil Pengkropingan Pola Sayap Capung Jenis Capung Sawah(warna hijau)



Gambar 4.6 Fungsi Transformasi Fourier Sayap Capung Jenis Capung

Sawah(warna hijau) yaitu $F(\omega) = 1572/25*i*(\exp(-1086/25*i*w)-\exp(1086/25*i*w))/w$

4.2 Pembahasan

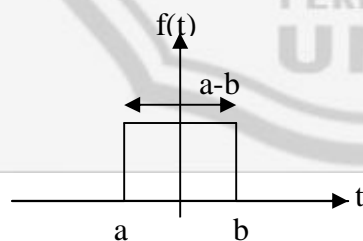
Dari hasil penelitian tentang pencarian fungsi transformasi fourier sayap capung dapat dilihat dari hasil pengkropingan pola dari ketiga jenis sayap capung dan diperoleh hasil dan fungsi yang berbeda-beda.

Pada penelitian ini pola yang diambil adalah pola *square* dengan anggapan semua pola periodik. Dengan mengambil salah satu pola *square* dengan cara pengkropingan pada gambar dengan menggunakan MATLAB dapat ditentukan fungsi transformasi fourier.

Persamaan yang digunakan pada pemrograman MATLAB:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} dt \quad (1)$$

Untuk $f(t) = y = \text{konstan}$, diperoleh persamaan untuk satu pola *square* sayap capung yang ditunjukkan pada gambar (6) dapat dinyatakan dengan menggunakan persamaan (2).



Gambar 4.7. Satu pola *square* sayap capung

$$F(\omega) = y \int_a^b e^{-j\omega t} dt \quad (2)$$

Dengan menggunakan pemrograman MATLAB fungsi-fungsi didapatkan pada masing- masing pola sayap capung setelah dilakukannya pengkropingan sebagai berikut:

$$F(\omega) = \frac{213}{10} \cdot i \left(\frac{e^{-i \frac{4943}{100} \omega} - e^{i \frac{4943}{100} \omega}}{\omega} \right)$$

$$F(\omega) = \frac{562}{5} \cdot i \left(\frac{e^{-i \frac{12643}{200} \omega} - e^{i \frac{12643}{200} \omega}}{\omega} \right)$$

$$F(\omega) = \frac{1572}{25} \cdot i \left(\frac{e^{-i \frac{1086}{25} \omega} - e^{i \frac{1086}{25} \omega}}{\omega} \right)$$

Dari fungsi-fungsi tersebut didapatkan juga grafik yang berbeda-beda dan hasil analisis grafik menyatakan semakin panjang jarak a-b pada pola *square* sayap, maka grafik transformasi fourier semakin rapat. Sebaliknya semakin pendek jarak a-b pada pola *square* sayap, maka grafik transformasi fourier semakin renggang.

BAB 5

PENUTUP

5.1 Kesimpulan

Dari Hasil penelitian dan pembahasan dapat ditarik kesimpulan sebagai berikut :

Bahwa dari pembahasan dapat diketahui fungsi transformasi fourier dari ketiga sayap capung sebagai berikut:

Sayap capung jarum memiliki fungsi:
$$F(\omega) = \frac{213}{10} \cdot i \left(\frac{e^{-i\frac{4943}{100}\omega} - e^{i\frac{4943}{100}\omega}}{\omega} \right)$$

Sayap capung sungai memiliki fungsi:
$$F(\omega) = \frac{562}{5} \cdot i \left(\frac{e^{-i\frac{12643}{200}\omega} - e^{i\frac{12643}{200}\omega}}{\omega} \right)$$

Sayap capung sawah memiliki fungsi:
$$F(\omega) = \frac{1572}{25} \cdot i \left(\frac{e^{-i\frac{1086}{25}\omega} - e^{i\frac{1086}{25}\omega}}{\omega} \right)$$

Semakin panjang jarak a-b pada pola *square* sayap, maka grafik transformasi fourier semakin rapat. Sebaliknya semakin pendek jarak a-b pada pola *square* sayap, maka grafik transformasi fourier semakin renggang.

5.2 Saran

Untuk penelitian lebih lanjut, terdapat saran sebagai berikut :

1. Penulis menghimbau pada pembaca untuk mengembangkan penelitian ini menggunakan pola periodik sayap yang lebih rumit.
2. Penulis menghimbau pada pembaca untuk mengembangkan penelitian ini menggunakan bahasa pemrograman yang lain.

DAFTAR PUSTAKA

Erick, P. & Nataliani, Y. 2007. *GUI MATLAB*. Yogyakarta: Andi Offset.

http://id.wikipedia.org/wiki/transformasi_fourier

Munir, Rinaldi. 2004. *Pengolahan Citra Digital*. Bandung: Informatika Bandung.

Sugiharto, Aris. 2006. *Pemrograman GUI dengan MATLAB*. Yogyakarta: CV. Andi Offset.

www.snh.org.uk/publications/on-line/naturallscottish/dragonfly

Yahya, Harun. 2002. *Keajaiban Desain di Alam*. Jakarta: Flobal Cipta Publishing.

Zulkaryanto, Eko. 2002. *Transformasi Fourier 1 Dimensi*. Bogor: Computer Science Bogor Agricultural University



LAMPIRAN 1

Pemograman MATLAB

```

function varargout = FFT_Prog(varargin)
% FFT_PROG M-file for FFT_Prog.fig
%   FFT_PROG, by itself, creates a new FFT_PROG or raises the existing
%   singleton*.
%
%   H = FFT_PROG returns the handle to a new FFT_PROG or the handle to
%   the existing singleton*.
%
%   FFT_PROG('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in FFT_PROG.M with the given input
arguments.
%
%   FFT_PROG('Property','Value',...) creates a new FFT_PROG or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before FFT_Prog_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to FFT_Prog_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help FFT_Prog

% Last Modified by GUIDE v2.5 25-Aug-2011 12:53:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @FFT_Prog_OpeningFcn, ...
                  'gui_OutputFcn',  @FFT_Prog_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});

```



```

end
% End initialization code - DO NOT EDIT

% --- Executes just before FFT_Prog is made visible.
function FFT_Prog_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to FFT_Prog (see VARARGIN)

% Choose default command line output for FFT_Prog
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FFT_Prog wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = FFT_Prog_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% -----
function open_Callback(hObject, eventdata, handles)
% hObject handle to open (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
proyek=guidata(gcbo);
[nama_file1,nama_path1]=uigetfile({'*.jpg';*.bmp';...
    '*.png';*.tif'},'Buka Gambar');
if ~isequal(nama_file1,0)
    h=waitbar(0,'Please Wait...');
    for i=1:200;
        waitbar(i/200)
    end
end

```

```

close(h);
I= imread(fullfile(nama_path1,nama_file1));
axes(handles.axes1)
set(proyek.figure1,'CurrentAxes',proyek.axes1);
imshow(I)
axis tight
axis off
set(proyek.axes1,'Userdata',I);
set(proyek.figure1,'Userdata',I);
I=get(proyek.axes1,'Userdata');
h=proyek;
hp = uipanel('Title','Pixel Info','FontSize',10,...
    'Position',[0.78 0.05 .20 .07]);
axes(handles.axes1)
set(proyek.figure1,'CurrentAxes',proyek.axes1);
h = imshow(I);
hText = impixelinfoval(hp,h);
set(hText,'FontWeight','Bold');
set(hText,'FontSize',12);
set(hText,'FontName','Tahoma');
set(proyek.axes1,'Userdata',I);
else
    return;
end

% -----
function file_Callback(hObject, eventdata, handles)
% hObject handle to file (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----
function exit_Callback(hObject, eventdata, handles)
% hObject handle to exit (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close;

% -----
function capture_Callback(hObject, eventdata, handles)
% hObject handle to capture (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% -----
function crop_Callback(hObject, eventdata, handles)
% hObject handle to crop (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
axes(handles.axes1)
proyek=guidata(gcbo);
I=get(proyek.axes1,'Userdata');
J = imcrop;
set(proyek.figure1,'CurrentAxes',proyek.axes2);
imshow(J,[]);
set(proyek.axes2,'Userdata',J);
axes(handles.axes2)
proyek=guidata(gcbo);
I=get(proyek.axes2,'Userdata');
set(proyek.figure1,'CurrentAxes',proyek.axes2);
imshow(J,[]);
set(proyek.axes2,'Userdata',I);

% -----
function tool_Callback(hObject, eventdata, handles)
% hObject handle to tool (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----
function distance_Callback(hObject, eventdata, handles)
% hObject handle to distance (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
imdistline

% -----
function calcu_Callback(hObject, eventdata, handles)
% hObject handle to calcu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----

```

```

function FFT_Callback(hObject, eventdata, handles)
% hObject handle to FFT (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
proyek=guidata(gcbo);
h=str2num (get(handles.edit1,'string'));
u=str2num (get(handles.edit2,'string'));
e=u/2;
a=-e;
b=e;
r=h;
syms t w
d=r*exp(-i*t*w)
f=int(d, t, a, b)
figure
ezplot(f)
xlabel('w'),ylabel('f(t)')
hold on

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

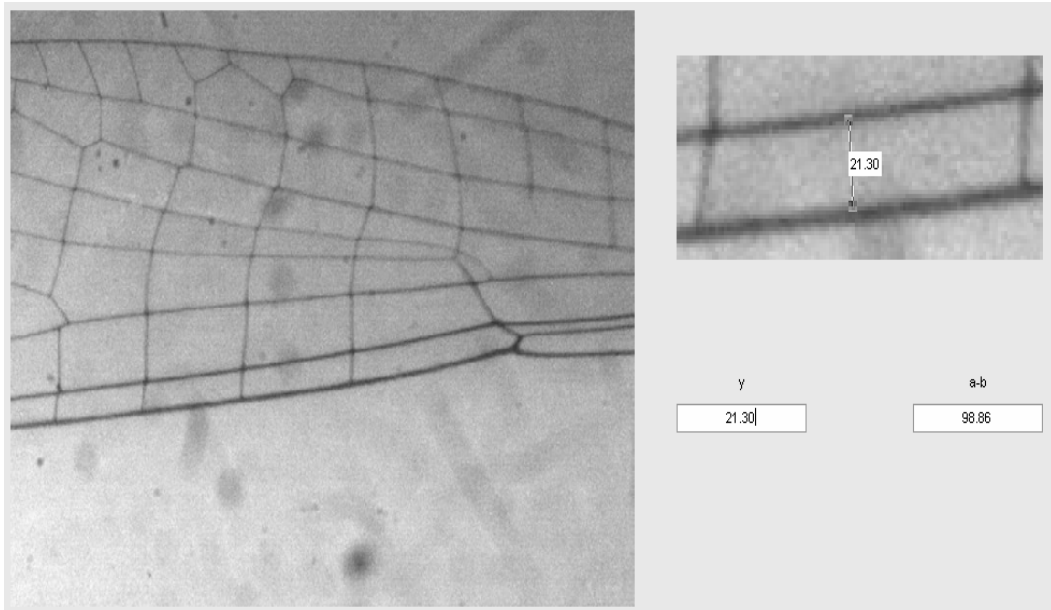
% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

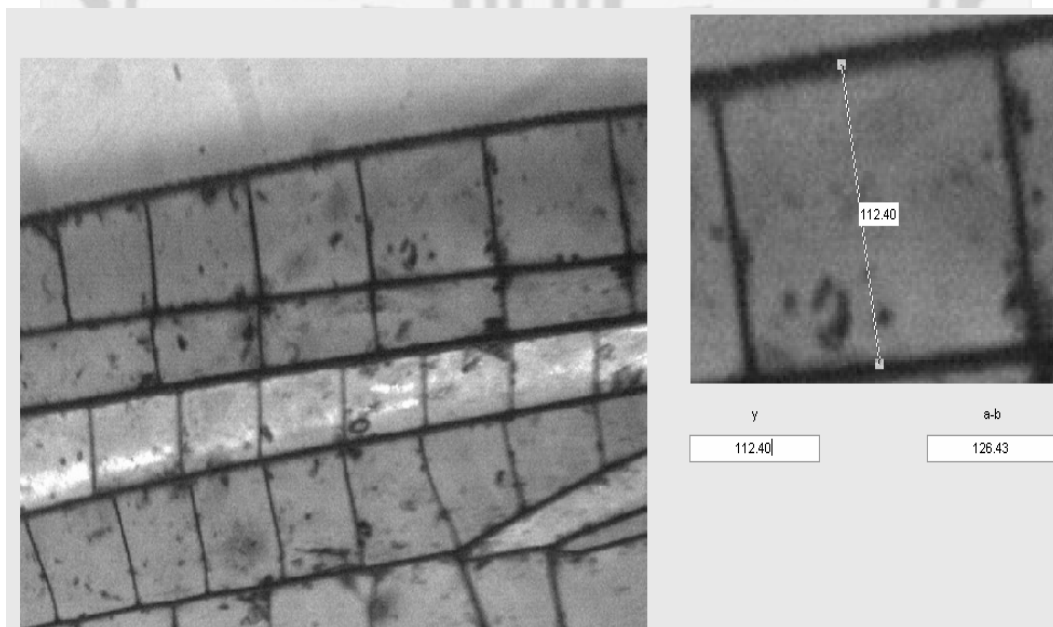
LAMPIRAN 2

PENGKROPINGAN CITRA DIGITAL POLA SAYAP CAPUNG

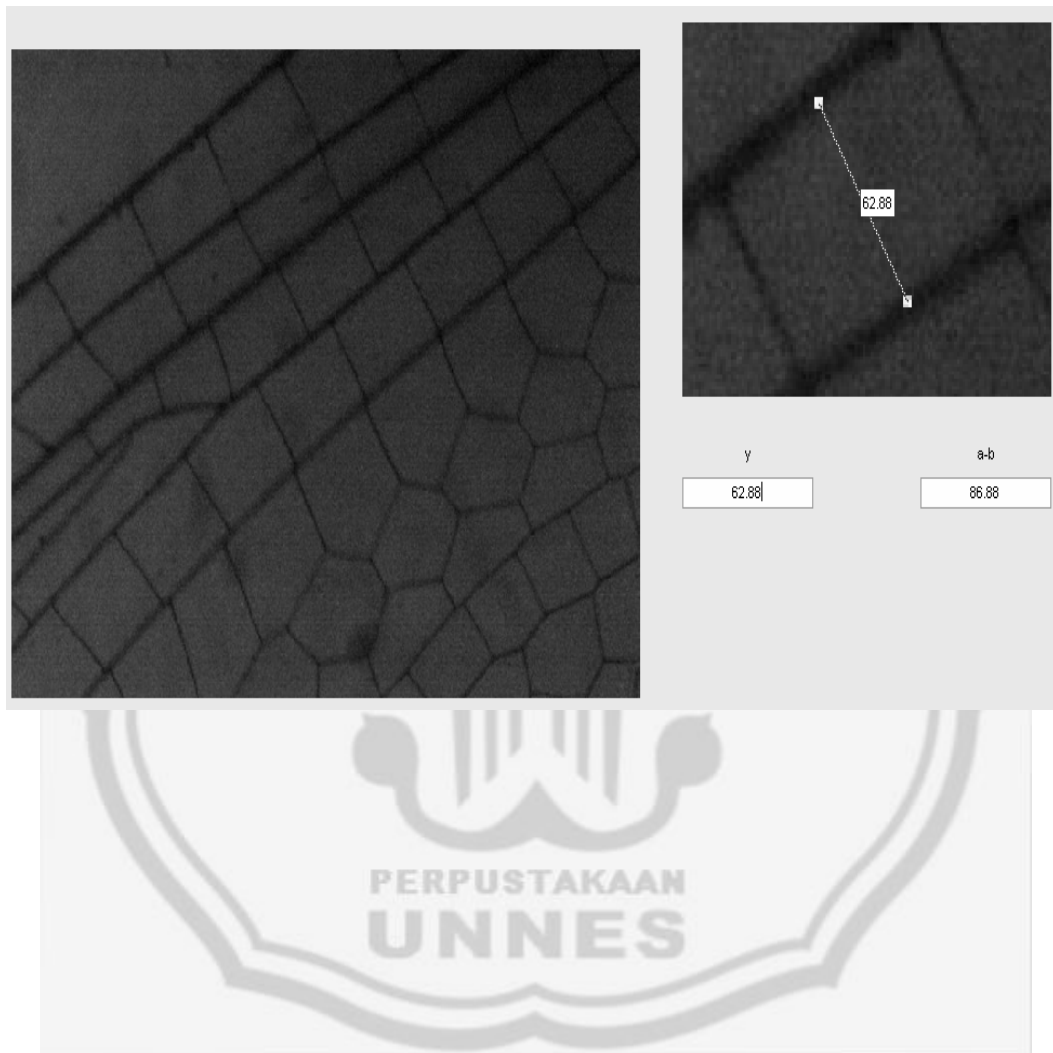
Pengkropingan citra digital pola sayap capung jenis capung jarum



Pengkropingan citra digital pola sayap capung jenis capung sungai(warna orange)



Pengkropingan citra digital pola sayap capung jenis capung sawah(warna hijau)

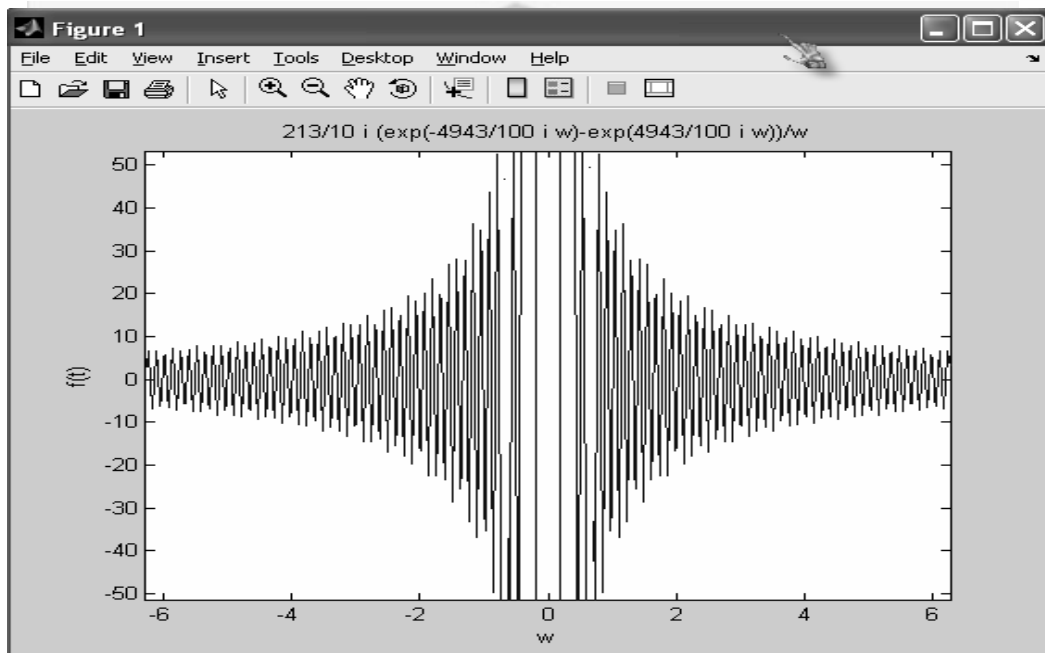


LAMPIRAN 3

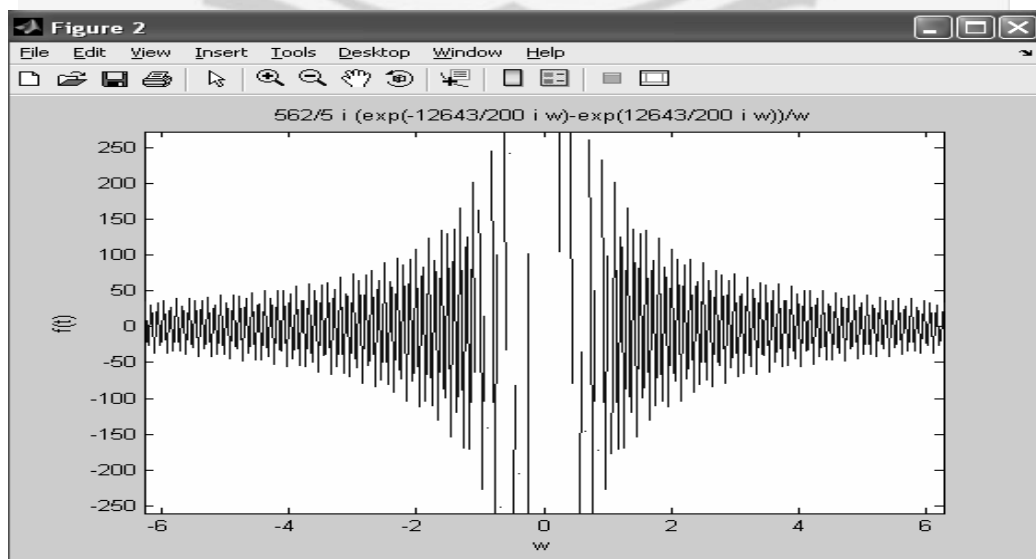
HASIL TRANSFORMASI FOURIER SAYAP CAPUNG

Fungsi transformasi fourier sayap capung jenis capung jarum yaitu

$$F(\omega) = 213/10 * i * (\exp(-4943/100 * i * \omega) - \exp(4943/100 * i * \omega)) / \omega$$



Fungsi transformasi fourier sayap capung jenis capung sungai(warna orange) yaitu $F(\omega) = 562/5 * i * (\exp(-12643/200 * i * \omega) - \exp(12643/200 * i * \omega)) / \omega$



Fungsi Transformasi Fourier Sayap Capung Jenis Capung Sawah(warna hijau)
yaitu $F(\omega) = 1572/25 * i * (\exp(-1086/25 * i * \omega) - \exp(1086/25 * i * \omega)) / \omega$

