

SKRIPSI

RANCANG BANGUN APLIKASI KOMPRESI DATA PADA *HANDPHONE* BERBASIS *ANDROID* MENGGUNAKAN ALGORITMA *HUFFMAN*



Disusun Oleh

DINASTI CAHYO PRASETYO

07. 12. 648

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI KOMPRESI DATA PADA
HANDPHONE BERBASIS ANDROID MENGGUNAKAN
ALGORITMA HUFFMAN

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik*

Disusun oleh :

DINASTI CAHYO PRASETYO

07. 12. 648

Mengetahui,

Ketua Jurusan Teknik Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Michael Ardita, ST, MT
NIP.P. 1031000434

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2012

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : DINASTI CAHYO PRASETYO
NIM : 07.12.648
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, Maret 2012

Yang membuat Pernyataan,



Dinasti Cahyo Prasetyo

NIM. 07.12.648

ABSTRAK

RANCANG BANGUN APLIKASI KOMPRESI DATA PADA *HANDPHONE* BERBASIS *ANDROID* MENGGUNAKAN ALGORITMA *HUFFMAN*

Dinasti Cahyo Prasetyo, NIM 07.12.648

Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT dan Michael Ardita, ST, MT

Kompresi data (pemampatan data) merupakan suatu teknik untuk memperkecil jumlah ukuran data (hasil kompresi) dari data aslinya. Pemampatan data digunakan untuk mengurangi jumlah bit-bit yang dihasilkan dari setiap simbol yang muncul dengan tujuan untuk memperkecil ukuran data dalam media penyimpanan (storage device). Pengkodean dengan metode Huffman dibangun dari panjang variabel kode-kode yang disusun dari bit-bit. Simbol dengan probabilitas yang tinggi akan memperoleh kode-kode paling pendek sedangkan simbol dengan probabilitas paling rendah akan memperoleh kode terpanjang. Kode Huffman mempunyai atribut unik yang sempurna, dalam arti kode-kode tersebut dapat mengembalikan (decoder) kode yang panjang dan decirable dalam arti kode Huffman tersebut tidak menjadi prefik kode Huffman yang lain.

Algoritma Huffman ini akan diimplementasikan ke dalam Handphone yang berbasis Android. Aplikasi ini digunakan untuk kompresi dan dekompresi berbagai macam data tanpa merusak data aslinya. Sistem ini dibangun dengan menggunakan eclipse sebagai ruang lingkup kerjanya.

Dari beberapa sample yang diujikan, Aplikasi kompresi data menggunakan algoritma Huffman ini sangat baik untuk mengkompresi file teks.

Kata Kunci : Kompresi, Huffman, Android

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, yang telah memberikan berkat-Nya, sehingga penulis dapat menyelesaikan laporan Skripsi ini dengan baik dan lancar.

Laporan Skripsi ini merupakan salah satu persyaratan akademik dalam menyelesaikan program Strata 1 Jurusan Teknik Elektro, Konsentrasi Komputer & Informatika, Institut Teknologi Nasional Malang. Adapun judul laporan Skripsi ini adalah:

RANCANG BANGUN APLIKASI KOMPRESI DATA PADA HANDPHONE BERBASIS ANDROID MENGGUNAKAN ALGORITMA HUFFMAN

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
2. Bapak Dr. Aryuanto Soetedjo, ST, MT selaku Sekertaris Jurusan Teknik Elektro S-1 ITN Malang dan pengusul serta penyedia ruang Skripsi.
3. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Dosen Pembimbing I
4. Bapak Michael Ardita, ST, MT selaku Dosen Pembimbing II
5. Bapak Ir. Yusuf Ismail Nahkoda, MT, selaku Dosen Wali.
6. ALLAH SWT, Nabi Muhammad SAW, Ibu, Ayah, dan adikku yang telah memberikan dukungan untuk selalu berdoa, berusaha dan naschat yang telah diberikan sampai saat ini.
7. Seluruh dosen dan pegawai ITN Kampus 2 Malang.

8. Mahasiswa Elektro ITN Malang, Malang Cyber Crew, Indonesian Codcr, teman-teman Laboratorium Jaringan Komputer dan CISCO, Lab. Jaringan Telekomunikasi, Klix ITN Malang, Elite FM, Moslem Hax0r.
9. Sahabat dan teman-temanku : Fiko, Shierly, UburUbur (luqvia), Aris, Mba' Nihan, Thoriq, Whendi, Ahmad Purnama, Reni, Tamie, Becca, Agung, Shaadow, Topan, Daphi, Pandu, Rendayu Anindita, RR Shinta, Anas, Kipli, Nto, Fitrianda, Sari, Saigon, Takari, Anton Black, Tahol, Mas Bima, Lia, Reva (Repong), Arif kdb, Joko, Margaretha, Deblo, Atoz, Christian, teman-teman Amburadulz FC.
10. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis berharap agar buku laporan Skripsi ini dapat memberikan banyak manfaat bagi semua pihak yang membutuhkan, khususnya bagi rekan-rekan mahasiswa. Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, oleh karena itu mohon maaf apabila dalam buku ini terdapat hal-hal yang kurang berkenan dihati para pembaca.

Penulis juga mengharap koreksi, kritik serta saran-saran yang bermanfaat demi kesempurnaan buku Laporan Skripsi ini.

Malang, Februari 2012

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan.....	2
1.5. Metodologi Penelitian.....	3
1.6. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1. Data	5
2.2. File Gif (.gif).....	5
2.3. File JPEG(.jpg)	6
2.4. File PNG (.png).....	6
2.5. File BMP (.bmp).....	6
2.6. Kompresi Data (<i>Data Compression</i>)	7
2.7. Rasio Kompresi.....	8

2.8. Algoritma Huffman	8
2.9. Android	12
2.9.1. Jenis – Jenis Android.....	12
2.10. Android SDK (<i>Software Development Kit</i>)	15
2.11. Eclipse	15
2.12. ADT (<i>Android Development Tools</i>)	16
2.13. AVD (<i>Android Virtual Device</i>)	16
2.14. <i>Java Development Kit</i>	16
BAB III ANALISA DAN PERANCANGAN SISTEM.....	18
3.1. Deskripsi Sistem	18
3.2. Analisis Sistem.....	18
3.3. Perancangan Sistem	19
3.4. Desain Sistem.....	19
3.5. Gambaran Umum Perangkat Lunak.....	19
3.6. Proses Kompresi dan Dekompresi Algoritma <i>Huffman</i>	21
3.6.1. Metode Kompresi Huffman.....	21
3.6.2. Metode Kompresi Huffman.....	27
3.7. Rancangan Sistem.....	28
3.7.1. Perancangan <i>Form</i> Utama	28
3.7.2. Daftar Fungsi Tombol Aplikasi.....	39
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....	41
4.1. Implementasi	41

4.1.1. Perangkat Lunak	41
4.1.2. Perangkat Keras	41
4.2. Implementasi Sistem.....	42
4.2.1. Spesifikasi <i>Handphone</i> dan <i>Emulator</i>	42
4.3. Pengujian Aplikasi	45
4.3.1. Pengujian dengan Menggunakan <i>Emulator</i>	45
4.3.1.1. Hasil Pengujian Aplikasi dengan Menggunakan <i>Emulator</i>	51
4.3.2. Pengujian Dengan Menggunakan <i>Handphone</i> Android Galaxy Mini....	53
4.3.2.1. Hasil pengujian Aplikasi Dengan Menggunakan <i>Handphone</i> Galaxy Mini	60
4.4. Pengujian Sistem Terhadap Berbagai Macam <i>File</i>	63
4.4.1. Pengujian Kompresi	63
4.4.1.1. Hasil Pengujian Kompresi pada <i>Emulator</i> dan <i>Handphone</i>	78
4.4.2. Pengujian Dekompresi	80
4.4.2.1. Hasil Pengujian Dekompresi pada <i>Emulator</i> dan <i>Handphone</i>	95
4.4.3. Hasil Pengujian Kompresi dan Dekompresi pada <i>Emulator</i> dan <i>Handphone</i>	97
BAB V KESIMPULAN DAN SARAN.....	99
5.1. Kesimpulan.....	99
5.2. Saran.....	100
DAFTAR PUSTAKA.....	101
LAMPIRAN – LAMPIRAN	

DAFTAR TABEL

BAB II TINJAUAN PUSTAKA

Tabel 2.1. Frekuensi Kemunculan String	10
--	----

BAB III ANALISA DAN PERANCANGAN SISTEM

Tabel 3.1. Tabel Frekuensi.....	22
---------------------------------	----

Tabel 3.2. Tabel <i>Huffman</i>	26
---------------------------------------	----

Tabel 3.3. Tabel Daftar <i>Form</i>	29
---	----

Tabel 3.4. Fungsi Aplikasi.....	39
---------------------------------	----

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Tabel 4.1. Hasil Pengujian Halaman <i>Splash Screen</i> pada <i>Emulator</i>	51
--	----

Tabel 4.2. Hasil Pengujian Halaman Menu Utama pada <i>Emulator</i>	51
--	----

Tabel 4.3. Hasil Pengujian Halaman Menu <i>Compress File</i> pada <i>Emulator</i> ...	52
---	----

Tabel 4.4. Hasil Pengujian Halaman Menu <i>Decompress File</i> pada <i>Emulator</i>	53
--	----

Tabel 4.5. Hasil Pengujian Halaman <i>Splash Screen</i> pada <i>Handphone</i>	61
---	----

Tabel 4.6. Hasil Pengujian Halaman Menu Utama pada <i>Handphone</i>	61
---	----

Tabel 4.7. Hasil Pengujian Halaman Menu <i>Compress File</i> pada <i>Handphone</i>	62
--	----

Tabel 4.8. Hasil Pengujian Halaman Menu <i>Decompress File</i> pada <i>Handphone</i>	62
---	----

Tabel 4.9. Kompresi Data pada <i>File</i> teks, <i>image</i> , <i>audio</i> pada <i>Emulator</i>	79
--	----

Tabel 4.10. Kompresi Data pada <i>File</i> teks, <i>image</i> , <i>audio</i> pada <i>Handphone</i>	79
--	----

Tabel 4.11. Dekompresi Data pada <i>File</i> teks, <i>image</i> , <i>audio</i> pada <i>Emulator</i>	96
--	----

Tabel 4.12. Dekompresi Data pada <i>file</i> teks, <i>image</i> , <i>audio</i> pada <i>Handphone</i> .	96
Tabel 4.13. Hasil Pengujian Kompresi dan Dekompresi pada <i>Emulator</i> dan <i>Handphone</i>	97

DAFTAR GAMBAR

BAB II TINJAUAN PUSTAKA

Gambar 2.1. Perbandingan Metode Kompresi <i>Lossy</i> dan <i>Lossless</i>	7
Gambar 2.2. Proses Pembentukan <i>Huffman Tree</i>	9

BAB III ANALISA DAN PERANCANGAN SISTEM

Gambar 3.1. Proses <i>Transfer file</i> ke <i>Handphone</i>	18
Gambar 3.2. Desain Sistem Aplikasi Kompresi Data.....	19
Gambar 3.3. Diagram Alir Proses Kompresi Secara Umum.....	20
Gambar 3.4. Diagram Alir Proses Dekompresi Secara Umum	21
Gambar 3.5. Langkah I Pembuatan Pohon <i>Huffman</i>	23
Gambar 3.6. Langkah II Pembuatan Pohon <i>Huffman</i>	23
Gambar 3.7. Langkah III Pembuatan Pohon <i>Huffman</i>	24
Gambar 3.8. Langkah IV Pembuatan Pohon <i>Huffman</i>	24
Gambar 3.9. Langkah V Pembuatan Pohon <i>Huffman</i>	25
Gambar 3.10. Langkah VI Pembuatan Pohon <i>Huffman</i>	25
Gambar 3.11. Langkah VII Pembuatan Pohon <i>Huffman</i>	26
Gambar 3.12. <i>Form main.xml</i> pada <i>Workspace Eclipse</i>	29
Gambar 3.13. Tampilan <i>Form main.xml</i> pada <i>Emulator</i>	30
Gambar 3.14. Tampilan <i>Form main.xml</i> pada <i>Handphone</i>	30
Gambar 3.15. <i>Form Main Menu (menu.xml)</i> pada <i>Workspace Eclipse</i>	31
Gambar 3.16. Tampilan <i>Form menu.xml</i> pada <i>Emulator</i>	31
Gambar 3.17. Tampilan <i>Form menu.xml</i> pada <i>Handphone</i>	32

Gambar 3.18. <i>Form Compress (fcompress.xml)</i> pada <i>Workspace Eclipse</i>	33
Gambar 3.19. Tampilan <i>Form fcompress.xml</i> pada <i>Emulator</i>	33
Gambar 3.20. Tampilan <i>Form fcompress.xml</i> pada <i>Handphone</i>	34
Gambar 3.21. <i>Form Decompress (fdecompress.xml)</i> pada <i>Workspace Eclipse</i>	35
Gambar 3.22. Tampilan <i>Form fdecompress.xml</i> pada <i>Emulator</i>	35
Gambar 3.23. Tampilan <i>Form fdecompress.xml</i> pada <i>Handphone</i>	36
Gambar 3.24. <i>Form file_dialog_main.xml</i> pada <i>Workspace Eclipse</i>	37
Gambar 3.25. Tampilan <i>Form file_dialog_main.xml</i> pada <i>Emulator</i>	37
Gambar 3.26. Tampilan <i>Form file_dialog_main.xml</i> pada <i>Handphone</i>	37
Gambar 3.27. <i>Form file_dialog_row.xml</i> pada <i>Workspace Eclipse</i>	38
Gambar 3.28. Tampilan <i>Form file_dialog_row.xml</i> pada <i>Emulator</i>	38
Gambar 3.29. Tampilan <i>Form file_dialog_row.xml</i> pada <i>Handphone</i>	39

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Gambar 4.1. <i>Emulator</i> yang menggunakan <i>Android</i> versi 2.1 (Eclair)	43
Gambar 4.2. <i>Handphone</i> Samsung Galaxy Mini	44
Gambar 4.3. Halaman <i>Splash Screen</i> pada <i>Emulator</i>	45
Gambar 4.4. Halaman Menu Utama pada <i>Emulator</i>	46
Gambar 4.5. Menu Exit pada <i>Emulator</i>	46
Gambar 4.6. Menu About pada <i>Emulator</i>	47
Gambar 4.7. Halaman <i>Compress File</i> pada <i>Emulator</i>	48
Gambar 4.8. Halaman Pilih <i>File</i> pada <i>Emulator</i>	48
Gambar 4.9. Proses Kompresi pada <i>Emulator</i>	49
Gambar 4.10. Halaman Dekompresi <i>File</i> pada <i>Emulator</i>	50

Gambar 4.11. Halaman Pilih <i>File</i> pada <i>Emulator</i>	50
Gambar 4.12. Proses Dekompresi pada <i>Emulator</i>	51
Gambar 4.13. Halaman Utama pada <i>Handphone</i>	54
Gambar 4.14. Halaman Menu Utama pada <i>Handphone</i>	55
Gambar 4.15. Halaman Exit pada <i>Handphone</i>	55
Gambar 4.16. Menu About pada <i>Handphone</i>	56
Gambar 4.17. Halaman <i>Compress File</i> pada <i>Handphone</i>	57
Gambar 4.18. Pilih <i>File</i> pada <i>Handphone</i>	57
Gambar 4.19. Proses Kompresi pada <i>Handphone</i>	58
Gambar 4.20. Halaman <i>Decompress File</i> pada <i>Handphone</i>	59
Gambar 4.21. Pilih <i>File</i> pada <i>Handphone</i>	60
Gambar 4.22. Proses Dekompresi pada <i>Handphone</i>	60
Gambar 4.23. Kompresi <i>File .txt</i> pada <i>Emulator</i>	64
Gambar 4.24. Kompresi <i>File .txt</i> pada <i>Handphone</i>	64
Gambar 4.25. Kompresi <i>File .doc</i> pada <i>Emulator</i>	65
Gambar 4.26. Kompresi <i>File .doc</i> pada <i>Handphone</i>	66
Gambar 4.27. Kompresi <i>File .pdf</i> pada <i>Emulator</i>	66
Gambar 4.28. Kompresi <i>File .pdf</i> pada <i>Handphone</i>	67
Gambar 4.29. Kompresi <i>File .rtf</i> pada <i>Emulator</i>	68
Gambar 4.30. Kompresi <i>File .rtf</i> pada <i>Handphone</i>	68
Gambar 4.31. Kompresi <i>File .mp3</i> pada <i>Emulator</i>	69
Gambar 4.32. Kompresi <i>File .mp3</i> pada <i>Handphone</i>	70
Gambar 4.33. Kompresi <i>File .mp4</i> pada <i>Emulator</i>	70

Gambar 4.34. Kompresi <i>File</i> .mp4 pada <i>Handphone</i>	71
Gambar 4.35. Kompresi <i>File</i> .flv pada <i>Emulator</i>	72
Gambar 4.36. Kompresi <i>File</i> .flv pada <i>Handphone</i>	72
Gambar 4.37. Kompresi <i>File</i> .bmp pada <i>Emulator</i>	73
Gambar 4.38. Kompresi <i>File</i> .bmp pada <i>Handphone</i>	74
Gambar 4.39. Kompresi <i>File</i> .gif pada <i>Emulator</i>	74
Gambar 4.40. Kompresi <i>File</i> .gif pada <i>Handphone</i>	75
Gambar 4.41. Kompresi <i>File</i> .jpg pada <i>Emulator</i>	76
Gambar 4.42. Kompresi <i>File</i> .jpg pada <i>Handphone</i>	76
Gambar 4.43. Kompresi <i>File</i> .png pada <i>Emulator</i>	77
Gambar 4.44. Kompresi <i>File</i> .jpg pada <i>Handphone</i>	78
Gambar 4.45. Dekompresi <i>File</i> .txt pada <i>Emulator</i>	81
Gambar 4.46. Dekompresi <i>File</i> .txt pada <i>Handphone</i>	81
Gambar 4.47. Dekompresi <i>File</i> .doc pada <i>Emulator</i>	82
Gambar 4.48. Dekompresi <i>File</i> .doc pada <i>Handphone</i>	83
Gambar 4.49. Dekompresi <i>File</i> .pdf pada <i>Emulator</i>	83
Gambar 4.50. Dekompresi <i>File</i> .pdf pada <i>Handphone</i>	84
Gambar 4.51. Dekompresi <i>File</i> .rtf pada <i>Emulator</i>	85
Gambar 4.52. Dekompresi <i>File</i> .rtf pada <i>Handphone</i>	85
Gambar 4.53. Dekompresi <i>File</i> .mp3 pada <i>Emulator</i>	86
Gambar 4.54. Dekompresi <i>File</i> .mp3 pada <i>Handphone</i>	87
Gambar 4.55. Dekompresi <i>File</i> .mp4 pada <i>Emulator</i>	87
Gambar 4.56. Dekompresi <i>File</i> .mp4 pada <i>Handphone</i>	88

Gambar 4.57.Dekompresi <i>File</i> .flv pada <i>Emulator</i>	89
Gambar 4.58.Dekompresi <i>File</i> .flv pada <i>Handphone</i>	89
Gambar 4.59.Dekompresi <i>File</i> .bmp pada <i>Emulator</i>	90
Gambar 4.60.Dekompresi <i>File</i> .bmp pada <i>Handphone</i>	91
Gambar 4.61.Dekompresi <i>File</i> .gif pada <i>Emulator</i>	91
Gambar 4.62.Dekompresi <i>File</i> .gif pada <i>Handphone</i>	92
Gambar 4.63.Dekompresi <i>File</i> .jpg pada <i>Emulator</i>	93
Gambar 4.64.Dekompresi <i>File</i> .jpg pada <i>Handphone</i>	93
Gambar 4.65.Dekompresi <i>File</i> .png pada <i>Emulator</i>	94
Gambar 4.66.Dekompresi <i>File</i> .png pada <i>Handphone</i>	95

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Semakin berkembangnya teknologi pada *handphone* maka semakin banyak pula data yang ingin kita simpan, tetapi kapasitas media penyimpanan (*storage device*) yang kita miliki terkadang tidak sebanding dengan banyaknya data yang akan kita simpan. Jika data yang akan disimpan pada media penyimpanan semakin bertambah dan berukuran besar, maka media penyimpanan tidak dapat menyimpan data tersebut karena melebihi kapasitas. Oleh karena itu, untuk mengatasi masalah ini digunakanlah pemampatan data (*data compression*). Pemampatan data (*data compression*) merupakan salah satu kajian di dalam ilmu komputer yang bertujuan untuk mengurangi ukuran file sebelum menyimpan atau memindahkan data tersebut ke dalam media penyimpanan (*storage device*). Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan bandwidth.

Secara umum cara kerja kompresi data adalah dengan merubah suatu simbol-simbol menjadi suatu kode-kode. Pemampatan dikatakan efektif jika ukuran perolehan kode-kode tersebut sangat kecil dibandingkan dengan ukuran kode simbol aslinya. Dalam kompresi data, terdapat 4 (empat) faktor penting yang perlu diperhatikan, yaitu: *Time Process* (waktu yang dibutuhkan dalam menjalankan proses), *Completeness* (kelengkapan data setelah file-file tersebut dikompres), *Ratio Compress* (ukuran data setelah dilakukan kompresi), *Optimaly* (perbandingan apakah ukuran file sebelum dikompres sama atau tidak sama dengan file yang telah dikompres).

Pada skripsi ini penulis akan merancang dan membangun aplikasi kompresi data pada *handphone* berbasis *android* menggunakan algoritma *huffman*, dimana fungsi dari aplikasi ini adalah untuk memperkecil ukuran data yang tersimpan sehingga media penyimpanan pada *handphone* bisa menyimpan data dengan jumlah yang lebih banyak.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka dalam pembuatan aplikasi kompresi data pada *handphone* berbasis *android* menggunakan algoritma *huffman* didapat beberapa permasalahan yaitu :

1. Bagaimana merancang dan membangun aplikasi kompresi data pada *handphone* berbasis *android* menggunakan algoritma *huffman* ?
2. Bagaimana memperkecil ukuran data yang tersimpan pada media penyimpanan (*storage device*) *handphone* berbasis *android*?

1.3 Tujuan

Tujuan yang ingin dicapai dari pembuatan aplikasi kompresi data pada *handphone* berbasis *android* menggunakan algoritma *huffman* ini adalah :

1. Membangun aplikasi kompresi data pada *handphone* berbasis *android* menggunakan algoritma *huffman*.
2. Memperkecil ukuran data yang tersimpan pada media penyimpanan (*storage device*) *handphone* berbasis *android*, sehingga penggunaan media penyimpanan (*storage device*) menjadi lebih efisien.

1.4 Batasan Masalah

Permasalahan yang akan dibahas dalam penulisan ini dibatasi pada:

1. Membahas perancangan dan pembuatan aplikasi kompresi data pada *handphone* berbasis *android* menggunakan algoritma *Huffman*.
 2. Aplikasi ini dapat dijalankan minimal pada *Android* versi 2.1/ *Éclair* (API minimal level 7).
 3. Algoritma yang digunakan untuk kompresi dan dekompresi data adalah algoritma *Huffman*.
 4. Data-data yang akan dikompres berbentuk file teks (.txt, .doc, .pdf, .rtf), audio video (.mp3, .mp4, .flv), *image/gambar* (.jpg, .gif, .bmp, .png).
-

1.5 Metodologi Penelitian

Pembuatan tugas akhir ini terbagi menjadi beberapa tahap pengerjaan yang digunakan sebagai acuan dalam penyelesaiannya. Berikut adalah tahap-tahap pengerjaan tugas akhir sejak pengajuan proposal penelitian hingga laporan tugas akhir ini disusun :

1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan keperustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan kedalam sistem.

4. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

Bab III : Perancangan dan Analisa Sistem

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik *software* maupun *hardware* yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

Bab IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai¹²¹. Data yang terformat adalah data dengan suatu format tertentu. Misalnya data yang menyatakan tanggal atau jam, atau menyatakan nilai mata uang.

Teks adalah sederetan huruf, angka, dan symbol-simbol khusus (misalnya “+” dan “\$”) yang kombinasinya tidak tergantung pada masing-masing *item* secara individual. Contoh teks adalah koran.

Citra (*image*) adalah data dalam bentuk gambar. Citra dapat berupa grafik, foto, hasil *rontgen*, dan tanda tangan, ataupun gambar yang lain.

Audio adalah data dalam bentuk suara. Instrumen musik, suara orang atau suara binatang, gemericik air, detak jantung merupakan beberapa contoh dari *audio*.

Video menyatakan data dalam bentuk sejumlah gambar yang bergerak dan bias saja dilengkapi dengan suara. *Video* dapat digunakan untuk mengabadikan suatu kejadian atau aktivitas.

2.2 File Gif (.gif)

Graphic Interface Format (GIF) merupakan standar untuk menampilkan gambar pada *World Wide Web* dan pelayanan online lainnya. *File* ini mempunyai kompresi data yang baik tanpa menghilangkan detail. Metode kompresi ini disebut *lossless*. Kompresi demikian dapat digunakan dengan baik pada warna-warna yang terbatas. Dengan demikian *file* GIF ini tidak dapat menampilkan lebih dari 256 warna. Dari alasan inilah *file* format ini tidak efektif untuk menampilkan gambar fotografi secara online dan tidak direkomendasikan untuk cetak komersial^{16]}.

2.3 File JPEG (.jpg)

Format Joint Photographic Expert Group (JPEG) merupakan format yang umum digunakan untuk menampilkan gambar foto dan gambar separasi di dalam file HTML di web dan media online lain. File JPEG ini mendukung mode warna CMYK, RGB dan grayscale. JPEG tetap menyimpan semua informasi dalam gambar mode RGB. JPEG menggunakan skema kompresi *lossy* dalam mengurangi besar ukuran file, dimana data ekstra dan tidak begitu penting dalam tampilan akan dihapuskan. Kompresi yang sangat tinggi akan menghasilkan kualitas gambar yang rendah, sedangkan kompresi gambar dengan menggunakan pilihan kualitas yang maksimal, akan menghasilkan gambar yang tidak berbeda dengan aslinya. Saat membuka file gambar JPEG maka secara otomatis akan dilakukan pemekaran data kembali [6].

2.4 File PNG (.png)

Format Portable Network Graphics (PNG) menggunakan metode kompresi *lossless* untuk menampilkan gambar foto 24-bit atau warna-warna solid pada media online. PNG dikembangkan secara terbuka sebagai alternative selain format GIF. Format ini mendukung transparansi di dalam *alpha channel*. Format PNG sangat baik digunakan pada dokumen online, dan mempunyai dukungan warna yang lebih baik saat dicetak daripada format GIF. Akan tetapi warna pada gambar PNG akan *di-place* pada dokumen *InDesign* sebagai gambar bitmap RGB, sehingga hanya dapat dicetak sebagai gambar komposit bukan pada gambar separasi [6].

2.5 File BMP (.bmp)

BMP merupakan format gambar bitmap standar *Window* pada computer yang mempunyai sistem operasi DOS maupun Windows. Akan tetapi BMP tidak mendukung CMYK, dan hanya mendukung warna yang sangat terbatas, dari 1, 4, 8, atau 24 bits. Mode gambar ini tidak disarankan untuk digunakan pada cetak komersial maupun dokumen online, dan tidak didukung oleh *web browser*. BMP dapat diterima pada saat dicetak dengan kualitas cetak rendah atau pada printer *non-PostScript* [6].

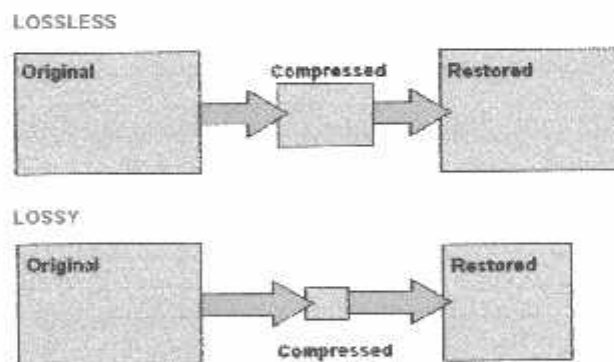
2.6 Kompresi Data (*Data Compression*)

Kompresi data (pemampatan data) merupakan suatu teknik untuk memperkecil jumlah ukuran data (hasil kompresi) dari data aslinya. Pemampatan data umumnya diterapkan pada mesin komputer, hal ini dilakukan karena setiap simbol yang dimunculkan pada komputer memiliki nilai bit-bit yang berbeda. Misal pada ASCII setiap simbol yang dimunculkan memiliki panjang bit 8 bit, misal kode A pada ASCII mempunyai nilai desimal = 65, jika dirubah dalam bilangan biner menjadi 01000001. Pemampatan data digunakan untuk mengurangi jumlah bit-bit yang dihasilkan dari setiap simbol yang muncul. Dengan pemampatan ini diharapkan dapat mengurangi (memperkecil ukuran data) dalam ruang penyimpanan.

Berdasarkan kompresi yang dihasilkan, metode kompresi dapat dibagi ke dalam dua kategori, yaitu :

a. Metode kompresi *lossy*

Metode kompresi *lossy* adalah dimana ketika mengkompres data dan kemudian didekompresi data yang dihasilkan tidak sama dengan data aslinya, tetapi cukup dan masih bisa digunakan sebatas keperluan.



Gambar 2.1

Perbandingan Metode Kompresi *Lossy* dan *Lossless*

b. Metode kompresi *lossless*

Metode kompresi *lossless* adalah metode kompresi data yang menghasilkan data yang sama persis dengan data aslinya dengan merekonstruksi dari data yang dikompresi. Karakteristik dari metode

kompresi ini adalah tekniknya tidak berdasarkan media dengan spesifikasi dan karakteristik tertentu namun berdasarkan urutan data. Metode *lossless* menghasilkan data yang identik dengan data aslinya, hal ini dibutuhkan untuk banyak tipe data, contohnya: *executable code*, *word processing files*, *tabulated numbers*, dan sebagainya. Kadang kala ada data-data yang setelah dikompresi dengan menggunakan metode ini ukurannya menjadi lebih besar atau sama dengan ukuran data aslinya (sebelum dilakukan kompresi).

Kriteria algoritma dan aplikasi kompresi data :

1. Kualitas data hasil *encoding* : ukuran lebih kecil, data tidak rusak (untuk kompresi *lossy*).
2. Kecepatan, rasio, dan efisiensi proses kompresi dan dekompresi.
3. Ketepatan proses dekompresi data : data hasil dekompresi tetap sama dengan data sebelum dikompres (untuk kompresi *lossless*).

2.7 Rasio Kompresi

Rasio Kompresi adalah ukuran *file* hasil kompresi dibagi dengan ukuran *file* asli. Dan rasio kompresi yang mempunyai nilai lebih dari 1, berarti ukuran *file* atau data yang dikompresi bukan menjadi lebih kecil tetapi malah menjadi besar [3].

$$\text{Rasio Kompresi} = \frac{\text{ukuran file hasil kompresi}}{\text{ukuran file asli}} \quad (2.1)$$

2.8 Algoritma Huffman

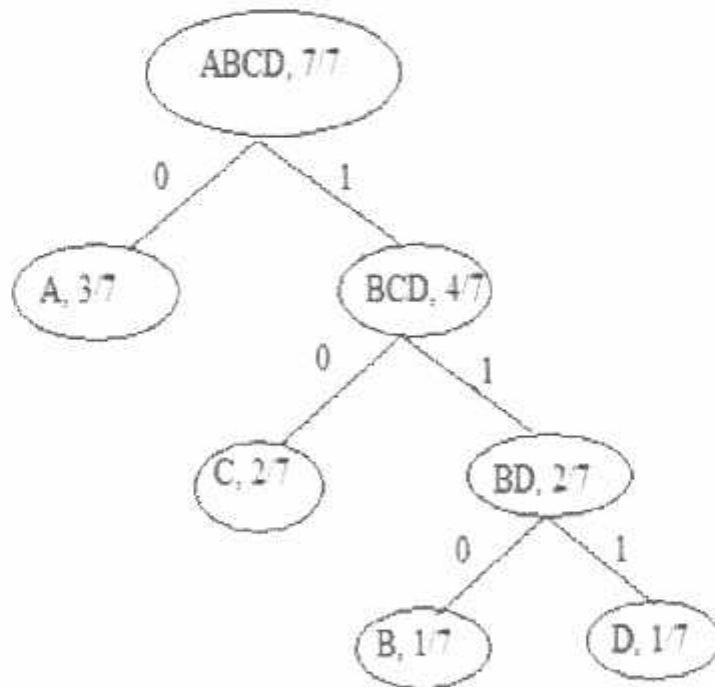
Metode *Huffman* ditemukan oleh seorang mahasiswa MIT bernama David *Huffman* pada tahun 1952. Algoritma ini bekerja dengan ide pengkodean karakter dengan kemunculan tinggi dengan bit yang sedikit dan karakter dengan kemunculan rendah dikodekan dengan bit yang lebih besar. Dengan demikian, ukuran memori untuk menyimpan informasi dapat diminimalisasikan tanpa ada informasi yang terbuang (*lossless*). Pengkodean dengan metode *Huffman* dibangun dari panjang variabel kode-kode yang disusun dari bit-bit. Simbol

dengan probabilitas yang tinggi akan memperoleh kode-kode paling pendek sedangkan simbol dengan probabilitas paling rendah akan memperoleh kode terpanjang. Contoh untuk string 'ABACCCA' mempunyai panjang bit sebanyak 56 bit karena 1 karakter dikodekan dengan 8 bit (ASCII) akan diperoleh jumlah bit untuk tiap simbolnya dengan jumlah yang lebih sedikit atau bitnya lebih pendek yaitu 13 bit, sehingga secara otomatis ukuran *filenya* berkurang^[1].

Sebagai contoh, dalam kode ASCII string 7 huruf "ABACCCA" membutuhkan representasi $7 \times 8 \text{ bit} = 56 \text{ bit}$ (7 byte), dengan rincian sebagai berikut :

$\underbrace{01000001}_A$ $\underbrace{01000010}_B$ $\underbrace{01000001}_A$ $\underbrace{01000011}_C$ $\underbrace{01000011}_C$ $\underbrace{01000100}_D$ $\underbrace{01000001}_A$

Untuk mengurangi jumlah bit yang dibutuhkan, panjang kode untuk tiap karakter dapat dipersingkat, terutama untuk karakter yang frekuensi kemunculannya besar. Pada string di atas, frekuensi kemunculan $A = 3$, $B = 1$, $C = 2$, dan $D = 1$, sehingga dengan menggunakan algoritma di atas diperoleh kode *Huffman* sebagai berikut:



Gambar 2.2 Proses Pembentukan *Huffman Tree*

Tabel 2.1 Frekuensi kemunculan string

Simbol	Frekuensi	Peluang	Kode Huffman
A	3	$3/7$	0
B	1	$1/7$	110
C	2	$2/7$	10
D	1	$1/7$	111

Dengan menggunakan kode *Huffman* ini, string "ABACCCA" direpresentasikan menjadi rangkaian bit : 0 110 0 10 10 111 0. Jadi, jumlah bit yang dibutuhkan hanya 13 bit. Tampak bahwa kode untuk sebuah simbol atau karakter tidak boleh menjadi awalan dari kode simbol yang lain guna menghindari keraguan (*ambiguitas*) dalam proses dekompresi atau *decoding*. Karena tiap kode *Huffman* yang dihasilkan unik, maka proses dekompresi dapat dilakukan dengan mudah. Contoh : saat membaca kode bit pertama dalam rangkaian bit "[0]11001010110", yaitu bit "0", dapat langsung disimpulkan bahwa kode bit "0" merupakan pemetaan dari simbol "A". Kemudian baca kode bit selanjutnya, yaitu bit "1". Tidak ada kode *Huffman* "1", lalu baca kode bit selanjutnya, sehingga menjadi "11". Tidak ada juga kode *Huffman* "11", lalu baca lagi kode bit berikutnya, sehingga menjadi "110". Rangkaian kode bit "110" adalah pemetaan dari simbol "B". Metode *Huffman* yang diterapkan dalam penelitian ini adalah tipe statik, dimana dilakukan dua kali pembacaan (*two-pass*) terhadap *file* yang akan dikompresi, pertama untuk menghitung frekuensi kemunculan karakter dalam pembentukan pohon *Huffman*, dan kedua untuk mengkodekan simbol.

Selanjutnya adalah proses penguraian (*decompress*). Penguraian kode (*decompress*) adalah sebuah proses untuk menyusun kembali data yang telah dikodekan (dikompresi) sebelumnya sehingga informasi yang diterima dapat dibaca dan diolah. Penguraian kode (*decompress*) ini adalah lawan dari pemampatan (*compress*). Ada 2 cara penguraian kode *Huffman*. Cara yang pertama adalah menggunakan pohon *Huffman* sedangkan cara yang kedua adalah menggunakan tabel kode *Huffman*. Berikut adalah penjelasan untuk cara pertama, yaitu menggunakan pohon *Huffman*. Seperti yang dijelaskan sebelumnya, pohon *Huffman* adalah pohon biner dengan menggunakan kode awalan (*prefix code*). Hal ini memudahkan proses penguraian kode. Langkah – langkah yang dilakukan

dalam penguraian kode (*decompress*) menggunakan pohon *Huffman* adalah sebagai berikut (lihat gambar 2.2) :

1. Baca bit pertama dari string biner masukan
2. Lakukan pembacaan ulang pada pohon *Huffman* mulai dari akar sesuai dengan bit yang dibaca. Jika bit yang dibaca adalah 0 maka baca anak kiri, tetapi jika bit yang dibaca adalah 1 maka baca anak kanan.
3. Jika anak dari pohon bukan merupakan daun (masih berupa cabang) maka baca bit berikutnya dari string biner masukan.
4. Hal ini diulang hingga ditemukan daun.
5. Pada daun tersebut simbol ditemukan dan proses penguraian kode selesai.
6. Proses penguraian kode ini dilakukan hingga keseluruhan string biner masukan diproses.

Contoh cara dekompresi dengan menggunakan pohon *Huffman* : Dari hasil pengkodean string “ABACCD A” ke dalam string biner dengan menggunakan pohon *Huffman* (*Huffman Tree*) adalah 0110010101110. Bit pertama dari string biner tersebut adalah 0. Dengan menggunakan pohon *Huffman* yang ditunjukkan dalam gambar 2.2, ditemukan bahwa anak kiri nya adalah daun yang menyimpan simbol A. Pada bit kedua (angka 1), karena bit kedua adalah angka 1 maka harus diambil anak kanan dan ditemukan bahwa anak kanan tersebut bukanlah sebuah daun. Oleh karena itu harus diambil bit berikutnya yaitu bit ketiga (angka 1). Karena bit ketiga adalah 1 maka harus diambil anak kanan. Pada anak kanan tersebut belum ditemukan sebuah daun, maka dilanjutkan dengan mengambil bit selanjutnya, yaitu bit keempat (angka 0). Karena bit keempat adalah 0 maka harus diambil anak kiri. Pada anak kiri tersebut ditemukan sebuah daun yang menyimpan simbol B. Dengan melakukan hal ini secara berulang hingga bit terakhir, maka akan diperoleh bahwa string biner 0110010101110 adalah hasil kompresi dari string “ABACCD A” .

Cara kedua untuk menguraikan kode *Huffman* adalah dengan menggunakan tabel kode *Huffman*. Oleh karena kode *Huffman* disusun menggunakan kode awalan (*prefix code*) maka dapat dipastikan bahwa sebuah kode untuk sebuah simbol atau karakter yang satu tidak boleh menjadi awalan dari kode atau simbol

yang lain. Oleh karena itu dapat dipastikan string biner yang berisi hasil enkripsi dapat dipisahkan dengan mudah berdasarkan setiap rangkaian bit-nya untuk diuraikan menjadi informasi semula. Yang perlu dilakukan hanyalah melihat setiap rangkaian bit yang ditemukan dalam string biner hasil enkripsi di dalam tabel kode *Huffman*.

2.9 *Android*

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia^[7].

2.9.1 *Jenis-jenis Android*

1. *Android* versi 1.1

Pada 9 Maret 2009, Google merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. *Android* versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan *Android* dan SDK (*Software Development Kit*) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. *Android* versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. *Android* versi 2.0/2.1 (Fclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset *Android*, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam sistem operasi *Android* adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi *Android* dalam situs Internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh MySpace dan Facebook.

5. *Android* versi 2.2 (Froyo : Frozen Yoghurt)

Pada 20 Mei 2010, *Android* versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan

Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi *Android Market*.

6. *Android* versi 2.3 (Gingerbread)

Pada 6 Desember 2010, *Android* versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari *Android* versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. *Android* versi 3.0 (Honeybread)

Android Honeycomb dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom. Perangkat tablet dengan platform *Android* 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8. *Android* versi 4.0 (ICS : Ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2.10 *Android SDK (Software Development Kit)*

Android SDK adalah sebuah *kit* pengembangan perangkat lunak yang memungkinkan pengembang untuk membuat aplikasi untuk *platform Android*. *Android SDK* mencakup proyek sampel dengan *sourcecode*, *tools* pembangunan, *emulator*, dan *library* yang dibutuhkan untuk membangun aplikasi *Android*. Aplikasi yang ditulis menggunakan bahasa pemrograman Java dan berjalan di Dalvik, mesin virtual yang dirancang khusus untuk penggunaan *embedded* yang berjalan di atas kernel Linux.

2.11 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*)^[2]. Berikut ini adalah sifat dari Eclipse:

- **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- **Multilanguage:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE (*Integrated Development Environment*) favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.12 ADT (*Android Development Tools*)

Android menyediakan plugin untuk IDE yang disebut *Android Development Tools* (ADT). *Tools* ini sangat membantu sekali dalam *development eclipse*. *Eclipse* dan ADT merupakan salah satu cara yang disarankan untuk membangun aplikasi *Android* karena memberikan kemudahan dan meningkatkan produktifitas *developer*. Kemampuan yang diberikan adalah pembuatan *project* yang mudah, membuat UI, melakukan *debugging*, sampai kepada *export APK (Android Packages)* untuk di- distribusikan ^[13].

2.13 AVD (*Android Virtual Device*)

AVD adalah sebuah *emulator*, yang mengemulasikan perangkat handphone *Android* dengan CPU, sehingga kita dapat menjalankan perangkat *Android* sesuai model yang dipilih ^[14]. Setiap AVD terdiri dari :

1. Sebuah profil perangkat keras. Dapat mengatur opsi untuk menentukan fitur *hardware* emulator. Misalnya : Untuk menentukan apakah menggunakan perangkat kamera, apakah menggunakan *keyboard* QWERTY fisik atau tidak, berapa banyak memori internal, dan lain-lain.
2. Sebuah pemetaan versi *Android*. Dapat menentukan versi dari platform *Android* akan berjalan pada emulator.
3. Pilihan lainnya. Dapat menentukan skin yang ingin digunakan pada emulator, yang memungkinkan untuk menentukan dimensi layar, tampilan, dan sebagainya. Selain itu, juga dapat menentukan SD Card virtual untuk digunakan dengan di emulator.

2.14 *Java Development Kit*

JDK (*Java Development Kit*) adalah semacam kotak peralatan (*kit*) yang digunakan untuk pengembangan aplikasi yang ditulis dengan bahasa pemrograman java, JDK ini berguna saat menulis kode program. Komponen yang membantu kerja JDK, antara lain:

1. JVM (*Java Virtual Machine*)
-

JVM inilah yang merupakan jantung dari Java Platform. JVM ini adalah pihak yang bertanggung jawab untuk mengeksekusi program Java menjadi bahasa mesin untuk diproses oleh prosesor. JVM mampu menerjemahkan code-code Java ke hampir semua *platform*. JVM ini-lah yang membuat Java "*write once, run everywhere*" alias *multi-platform*.

2. JRE (*Java Runtime Environment*)

JRE inilah yang memungkinkan sebuah program Java dapat berjalan di mesin Anda. JRE ini mengeksekusi bilangan-bilangan biner dari kelas-kelas dan mengirimnya ke JVM untuk diproses lagi ke prosesor.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Deskripsi Sistem

Sistem yang akan dibahas dalam tugas akhir ini adalah aplikasi kompresi data yang digunakan pada ponsel berbasis sistem operasi *Android* dengan menggunakan algoritma *Huffman*. Pada sistem ini diujikan berbagai macam *file* atau data yang umum digunakan. Hasil dari pengujian sistem yang menggunakan algoritma *Huffman* yang telah terkompresi ini dianalisa dan selanjutnya dilakukan perbandingan dari kinerja metode kompresi tersebut. Kinerja dari sistem tersebut akan terlihat setelah dilakukan kompresi dan dekompresi. Data dari hasil penelitian sistem (data setelah dilakukan kompresi) dibandingkan dengan data aslinya. Dari perbandingan ini akan dihasilkan rasio kompresi, yaitu perbandingan ukuran data asli dibandingkan dengan ukuran data setelah dilakukan kompresi. Pada proses dekompresi akan terlihat data yang telah dikompresi ukurannya akan kembali (tepat) sama seperti sebelum dilakukan kompresi. Sistem ini dibangun dengan menggunakan *eclipse* sebagai ruang lingkup kerja.

3.2 Analisa Sistem

Sistem yang telah dibuat akan menghasilkan sebuah *file* dengan ekstensi *.apk* (*dot apk*). Pada *handphone* yang berbasis sistem operasi *Android*, *file* yang berekstensi *.apk* inilah yang akan digunakan untuk menginstal sistem yang telah dibuat ke dalam *handphone*. Sebelum sistem (aplikasi) dapat digunakan, terlebih dahulu aplikasi kompresi ini harus diinstall ke dalam *handphone* atau ponsel yang sistem operasinya berbasis *Android*, sistem ini juga dapat dijalankan pada PC (komputer), yaitu dengan menggunakan AVD (*Android Virtual Device*).



Gambar 3.1 Proses tranfer *file* ke Handphone

3.3 Perancangan Sistem

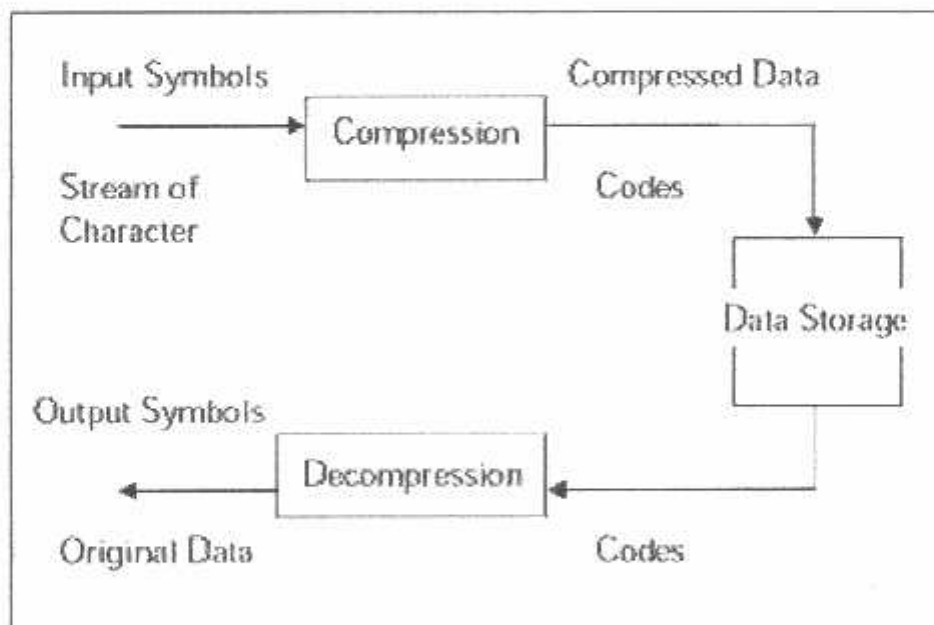
Rancangan dan langkah-langkah yang dilakukan dalam membangun aplikasi kompresi data yang digunakan pada ponsel berbasis sistem operasi *Android* dengan menggunakan algoritma *Huffman*.

Penelitian dilakukan dengan langkah-langkah sebagai berikut :

1. Mempelajari metode yang digunakan, yaitu algoritma *Huffman*.
2. Menganalisa dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
3. Membuat sistem berdasarkan analisis dan perancangan yang dilakukan.
4. Uji coba sistem.
5. Evaluasi hasil uji coba.

3.4 Desain Sistem

Sebelum membangun sebuah sistem, perlu merancang desain sistem terlebih dahulu, untuk dapat mempermudah pembuatan sistem, dan memberikan gambaran umum tentang sistem tersebut. Dijelaskan pada gambar 3.2



Gambar 3.2 Desain Sistem Aplikasi Kompresi Data

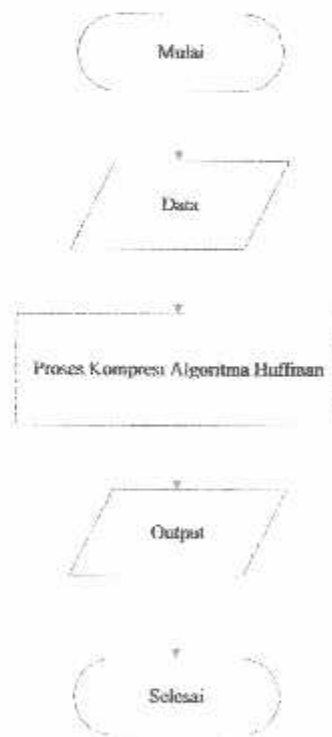
3.5 Gambaran Umum Perangkat Lunak

Sistem yang dikembangkan adalah implementasi dari algoritma *Huffman*

dalam bentuk aplikasi untuk kompresi dan dekompresi data. Sistem akan mengkompresi berbagai jenis data yang diinputkan oleh *user* dan akan memberikan keluaran berupa *file* atau data yang telah dikompresi yang dalam tugas akhir ini penulis memberikan ekstensi *.huf* untuk *file* yang telah dikompresi. Setelah dilakukan kompresi, untuk mengembalikan *file* ke bentuk aslinya maka dilakukan proses dekompresi. Pada proses dekompresi ini akan terlihat apakah ukuran data sebelum dilakukan kompresi sama dengan ukuran data setelah dilakukan dekompresi.

Langkah-langkah kompresi data menggunakan algoritma *Huffman* diilustrasikan dalam gambar 3.3, dengan tahapan sebagai berikut :

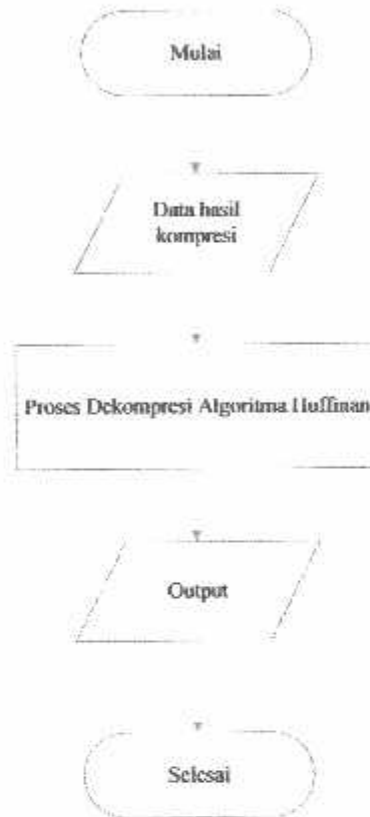
1. Data masukan (*input*) adalah berbagai macam tipe *file* atau data.
2. Proses kompresi algoritma *Huffman* terdiri dari pembuatan pohon *Huffman* (*Huffman Tree*) dan Tabel *Huffman*.
3. Keluaran (*output*), merupakan hasil kompresi (*file* hasil kompresi memiliki ekstensi *.huf*).



Gambar 3.3 Diagram Alir Proses Kompresi Secara Umum

Langkah-langkah dekompresi data menggunakan algoritma *Huffman* diilustrasikan dalam gambar 3.4, dengan tahapan sebagai berikut :

1. Data masukan (*input*) adalah *file* dengan ekstensi *.huf* (*file* yang telah dikompresi sebelumnya).
2. Proses dekomposisi algoritma *Huffman* yaitu pembacaan ulang terhadap pohon *Huffman* dan Tabel *Huffman*.
3. Keluaran (*output*), merupakan hasil dekomposisi (*file* hasil dekomposisi memiliki tambahan nama Huf didepannya, misalnya : Huf-dokumen.doc).



Gambar 3.4 Diagram Alir Proses Dekompresi Secara Umum

3.6 Proses Kompresi dan Dekompresi Algoritma *Huffman*

3.6.1 Metode Kompresi *Huffman*

Metode *Huffman* merupakan salah satu teknik kompresi dengan cara melakukan pengkodean dalam bentuk bit untuk mewakili data karakter. Cara kerja atau algoritma metode ini adalah sebagai berikut :

- a. Menghitung banyaknya jenis karakter dan jumlah dari masing-masing karakter yang terdapat dalam sebuah *file*.

- b. Menyusun setiap jenis karakter dengan urutan jenis karakter yang jumlahnya paling sedikit ke yang jumlahnya paling banyak.
- c. Membuat pohon biner berdasarkan urutan karakter dari yang jumlahnya terkecil ke yang terbesar, dan memberi kode untuk tiap karakter.
- d. Mengganti data yang ada dengan kode bit berdasarkan pohon biner.
- e. Menyimpan jumlah bit untuk kode bit yang terbesar, jenis karakter yang diurutkan dari frekuensi keluarnya terbesar ke terkecil beserta data yang sudah berubah menjadi kode bit sebagai data hasil kompresi.

Contoh teknik kompresi dengan menggunakan metode *Huffman* pada *file* teks. Misalkan sebuah *file* teks yang isinya "WINDA WINANTI". *File* ini memiliki ukuran 13 byte (104 bit) atau satu karakter sama dengan 1 byte. Berdasarkan pada cara kerja di atas, maka perlu dibuat pohon *Huffman* sebagai berikut :

- Misalkan String yang kita masukkan untuk kompresi adalah WINDA WINANTI, langkah selanjutnya adalah buat tabel frekuensi dari string tersebut.

Tabel 3.1 Tabel Frekuensi

Simbol	Kekerapan	Peluang
A	2	2 / 13
D	1	1 / 13
I	3	3 / 13
N	3	3 / 13
T	1	1 / 13
W	2	2 / 13
<spasi>	1	1 / 13

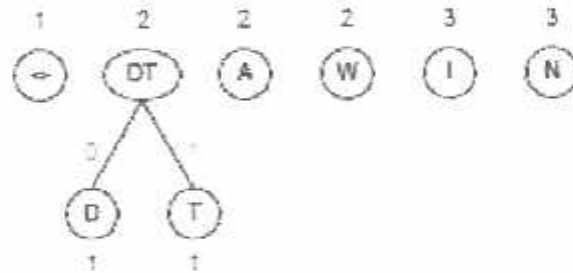
- Buat semua karakter menjadi simpul bebas dan sertakan probabilitas kemunculan karakter tersebut dalam string. Dalam contoh ini string yang digunakan adalah 'WINDA<spasi>WINANTI'.

Maka simpulnya setelah diurut dari terkecil hingga terbesar menurut frekuensinya adalah seperti gambar 3.5 di bawah ini :



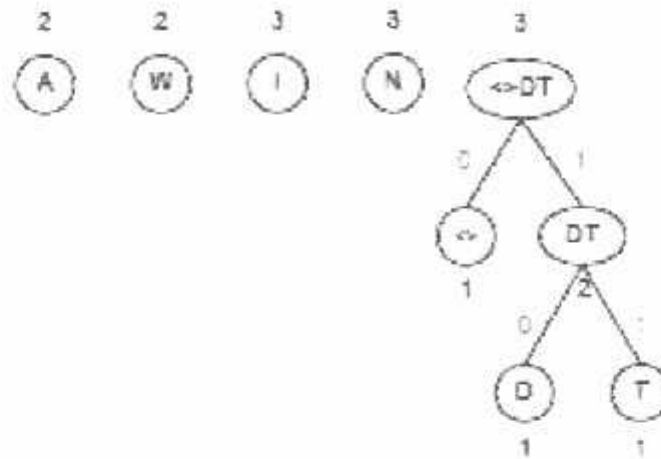
Gambar 3.5 Langkah I Pembuatan Pohon *Huffman*

- Pilih dua simpul dengan frekuensi paling kecil lalu gabungkan kedua simpulnya menjadi simpul baru. Dalam contoh ini dua simpul terkecil adalah D dan T, lalu gabungkan menjadi simpul baru dengan menerapkan $1 + 1 = 2$. Setelah simpul DT terbentuk maka urutkan lagi simpul DT dengan simpul sebelumnya sehingga menjadi seperti :



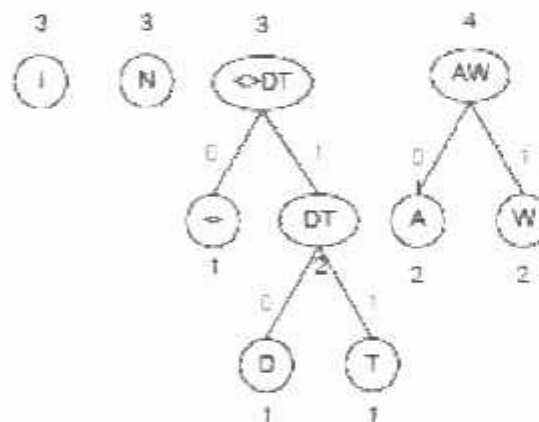
Gambar 3.6 Langkah II Pembuatan Pohon *Huffman*

- Lakukan kembali langkah 2 untuk semua simpul hingga terbentuk pohon *Huffman*. Dengan demikian maka langkah berikutnya adalah menggabungkan dua simpul dengan frekuensi terkecil yaitu simpul "<spasi>" dan "DT" sehingga menghasilkan simpul baru yaitu "<spasi>DT" dengan kecerapan $1 + 2 = 3$. Setelah simpul "<spasi>DT" terbentuk maka urutkan lagi simpul "<spasi>DT" dengan simpul sebelumnya sehingga menjadi seperti :



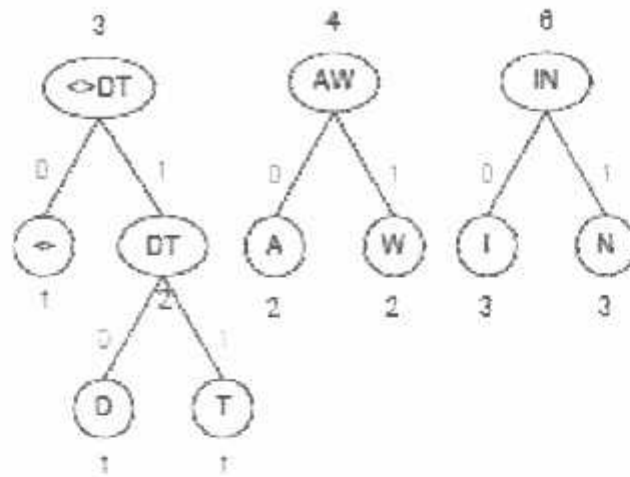
Gambar 3.7 Langkah III Pembuatan Pohon *Huffman*

- Gabungkan dua simpul dengan kekerapan terkecil yaitu simpul A dan W sehingga menghasilkan simpul baru yaitu AW dengan kekerapan $2 + 2 = 4$. Setelah simpul AW terbentuk maka urutkan lagi simpul AW dengan simpul sebelumnya sehingga menjadi seperti :



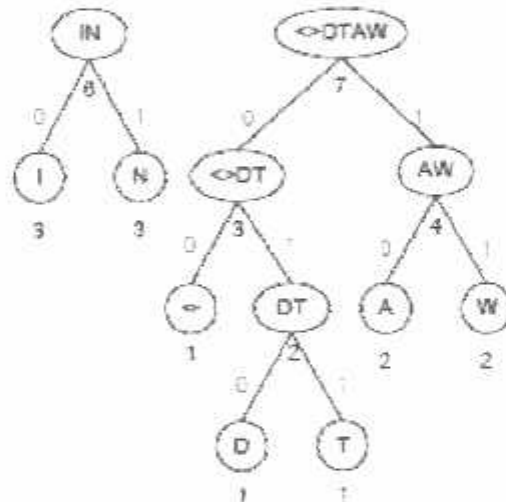
Gambar 3.8 Langkah IV Pembuatan Pohon *Huffman*

- Gabungkan dua simpul dengan kekerapan terkecil yaitu simpul I dan N sehingga menghasilkan simpul baru yaitu IN dengan kekerapan $3 + 3 = 6$. Setelah simpul IN terbentuk maka urutkan lagi simpul IN dengan simpul sebelumnya sehingga menjadi seperti :



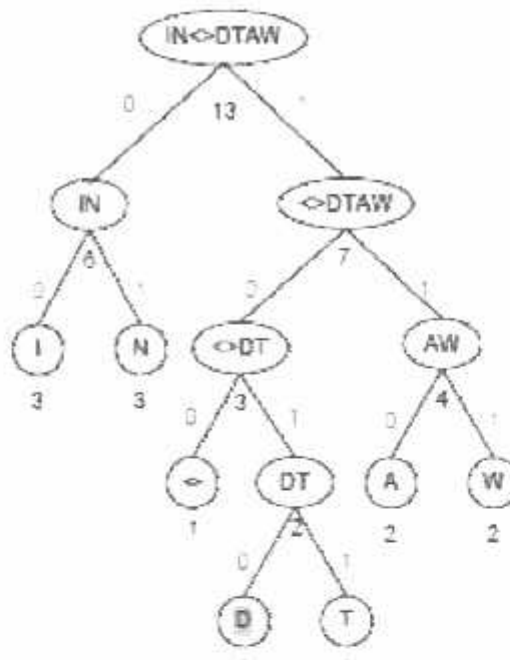
Gambar 3.9 Langkah V Pembuatan Pohon *Huffman*

- Gabungkan dua simpul dengan kekerapan terkecil yaitu simpul <spasi>DT dan AW sehingga menghasilkan simpul baru yaitu <spasi>DTAW dengan kekerapan $3 + 4 = 7$. Setelah simpul <spasi>DTAW terbentuk maka urutkan lagi simpul <spasi>DTAW dengan simpul sebelumnya sehingga menjadi seperti :



Gambar 3.10 Langkah VI Pembuatan Pohon *Huffman*

- Gabungkan dua simpul dengan kekerapan terkecil yaitu simpul IN dan <spasi>DTAW sehingga menghasilkan simpul baru yaitu IN<spasi>DTAW dengan kekerapan $6 + 7 = 13$. Simpul ini sudah menjadi akar dari pohon *Huffman* dan pembentukan pohon telah selesai.

Gambar 3.11 Langkah VII Pembuatan Pohon *Huffman*

- Langkah selanjutnya adalah membuat table *Huffman*, dengan cara mengganti data yang ada dengan kode bit berdasarkan pohon biner yang dibuat. Penggantian karakter menjadi kode biner, dilihat dari node yang paling atas atau disebut node akar :

Tabel 3.2 Tabel *Huffman*

Simbol	Keke- rapan	Peluang	Kode Huffman
A	2	2 / 13	110
D	1	1 / 13	1010
I	3	3 / 13	00
N	3	3 / 13	01
T	1	1 / 13	1011
W	2	2 / 13	111
<spasi>	1	1 / 13	100

- Selanjutnya berdasarkan pada kode biner masing-masing karakter ini, semua karakter dalam *file* dapat diganti menjadi :

111 00 01 1010 110 100 111 00 01 110 01 1011 00

Karena angka 0 dan angka 1 mewakili 1 bit, sehingga data bit di atas

terdiri dari 35 bit. Hasil ini lebih kecil daripada ukuran *file* sebelum dikompresi 104 bit.

3.6.2 Metode Dekompresi *Huffman*

Setiap data yang telah mengalami kompresi, tentu harus dapat merekonstruksi kembali data tersebut sesuai dengan aslinya. Merekonstruksi data lebih dikenal sebagai metode dekompresi data. Metode dekompresi *Huffman* dapat digunakan untuk mengembalikan data kode biner menjadi *file*. Ada 2 cara penguraian kode *Huffman*. Cara yang pertama adalah menggunakan pohon *Huffman* sedangkan cara yang kedua adalah menggunakan tabel kode *Huffman*. Berikut adalah penjelasan untuk cara pertama, yaitu menggunakan pohon *Huffman*. Seperti yang dijelaskan sebelumnya, pohon *Huffman* adalah pohon biner dengan menggunakan kode awalan (*prefix code*). Hal ini memudahkan proses penguraian kode. Langkah – langkah yang dilakukan dalam penguraian kode (*decompress*) menggunakan pohon *Huffman* adalah sebagai berikut (lihat gambar 3.11) :

1. Baca bit pertama dari string biner masukan
2. Lakukan pembacaan ulang pada pohon *Huffman* mulai dari akar sesuai dengan bit yang dibaca. Jika bit yang dibaca adalah 0 maka baca anak kiri, tetapi jika bit yang dibaca adalah 1 maka baca anak kanan.
3. Jika anak dari pohon bukan merupakan daun (masih berupa cabang) maka baca bit berikutnya dari string biner masukan.
4. Hal ini diulang hingga ditemukan daun.
5. Pada daun tersebut simbol ditemukan dan proses penguraian kode selesai.
6. Proses penguraian kode ini dilakukan hingga keseluruhan string biner masukan diproses.

Contoh cara dekompresi dengan menggunakan pohon Huffman : Dari hasil pengkodean string “WINDA WINANTI” ke dalam string biner dengan menggunakan pohon *Huffman* yang ditunjukkan dalam gambar 3.11 adalah 11100011010110100111000111001101100. Bit pertama dari string biner tersebut

adalah 1 (point 1). Sesuai dengan langkah-langkah yang disebutkan diatas jika bit yang dibaca adalah "1", maka yang harus dibaca adalah anak kanan (point 2). Ditemukan bahwa anak kanan bukan merupakan sebuah daun, baca bit berikutnya (point 3). Bit berikutnya adalah "1", sesuai langkah 2 maka yang dibaca adalah anak kanan. Anak kanan selanjutnya bukan merupakan sebuah daun, maka baca bit berikutnya (point 3). Bit selanjutnya adalah "1", baca anak kanan (point 2). Ditemukan bahwa anak kanan ini merupakan sebuah daun yang menyimpan symbol "W". Dengan melakukan hal ini secara berulang hingga bit terakhir, maka akan diperoleh bahwa string biner 11100011010110100111000111001101100 adalah hasil kompresi dari string "WINDA WINANTI" .

Cara kedua untuk menguraikan kode *Huffman* adalah dengan menggunakan tabel kode *Huffman*. Oleh karena kode *Huffman* disusun menggunakan kode awalan (*prefix code*) maka dapat dipastikan bahwa sebuah kode untuk sebuah simbol atau karakter yang satu tidak boleh menjadi awalan dari kode atau simbol yang lain. Oleh karena itu dapat dipastikan string biner yang berisi hasil enkripsi dapat dipisahkan dengan mudah berdasarkan setiap rangkaian bit-nya untuk diuraikan menjadi informasi semula. Yang perlu dilakukan hanyalah melihat setiap rangkaian bit yang ditemukan dalam string biner hasil enkripsi di dalam tabel kode *Huffman*.

3.7 Rancangan Sistem

3.7.1 Perancangan *Form* Utama

Sebelum mengimplementasikan sistem maka perlu dibuat suatu rancangan *form* agar sistem yang telah dibuat mudah untuk diterapkan dan digunakan oleh user. Pembuatan *form* untuk aplikasi ini dikerjakan dengan menggunakan perangkat lunak *Eclipse*. Dalam rancangan *form* tersebut berisi semua proses-proses yang terdapat pada program aplikasi kompresi ini. Rancangan *form* digunakan untuk mempermudah pemakaian sistem. Dalam pembuatan rancangan digunakan konsep interaksi manusia dengan *handphone* sehingga user dengan hanya melihat *form* akan mudah mengenali apa yang akan dilakukan selanjutnya. Berikut ini adalah rancangan *form* dari aplikasi kompresi yang diimplementasikan dalam bentuk bahasa pemrograman :

Tabel 3.3 Tabel Daftar Form

No	Nama File	Deskripsi
1.	main.xml	Form ini berisi splash screen.
2.	menu.xml	Form menu utama.
3.	fcompress.xml	Form untuk melakukan proses kompresi.
4.	fdecompress.xml	Form untuk melakukan proses dekompresi.
5.	file_dialog_main.xml	Form untuk memilih file untuk dikompresi maupun didekompres. Form ini berisi baris (<i>listing</i>) direktori atau file yang ada pada <i>Android storage device</i> .
6.	file_dialog_row.xml	Form ini berfungsi untuk menampilkan gambar (<i>icon</i>) untuk folder (direktori) maupun file.

a. Form main.xml

Form ini adalah form pertama yang terbuka pada saat aplikasi dijalankan.

Form akan menampilkan *splash screen*.



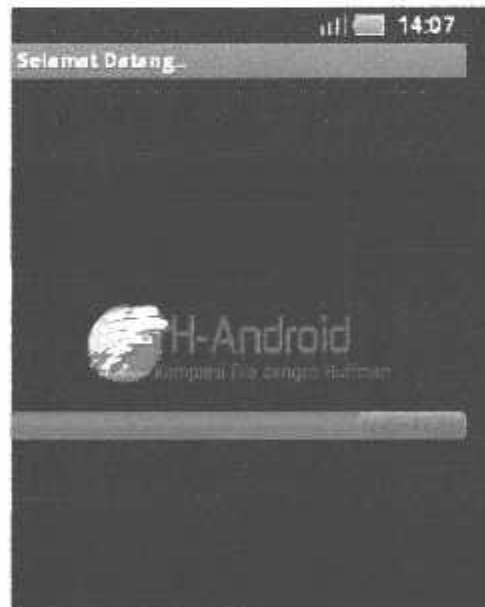
Gambar 3.12 Gambar Form main.xml pada *Workspace* Eclipse

Pada *emulator* tampilan halaman *splash screen* akan ditunjukkan pada gambar 3.13



Gambar 3.13 Tampilan Form *main.xml* pada *Emulator*

Tampilan halaman *splash screen* pada *handphone* Samsung Galaxy mini ditunjukkan pada gambar 3.14



Gambar 3.14 Tampilan Form *main.xml* pada *Handphone*

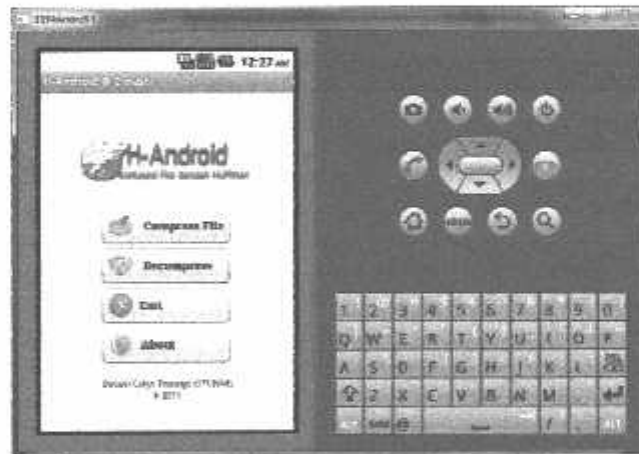
b. Form Main Menu (menu.xml)

Form ini adalah form utama pada aplikasi kompresi ini. Pada form ini terdapat empat buah menu, yaitu : *Compress File*, *Decompress*, *Exit*, *About*.



Gambar 3.15 Form Main Menu (menu.xml) pada *Workspace Eclipse*

Tampilan Form menu.xml pada *emulator* ditunjukkan pada gambar 3.16



Gambar 3.16 Tampilan Form menu.xml pada *Emulator*

Tampilan menu.xml pada *handphone* Samsung Galaxy mini ditunjukkan pada gambar 3.17



Gambar 3.17 Tampilan Form menu.xml pada *Handphone*

c. *Form Compress (fcompress.xml)*

Form ini digunakan untuk melakukan proses kompresi terhadap *file* atau data yang ada pada *storage device* (media penyimpanan) *Android*. Pada *form* ini terdapat *button* (tombol) *Open File* untuk mencari *file* yang ingin dikompresi, pada saat kita menekan tombol *Open File* maka aplikasi akan menuju ke *form file dialog*. Terdapat juga tombol proses untuk melakukan kompresi pada *file* yang telah dipilih, atau tombol batal untuk kembali ke menu utama (menu.xml). *File* yang telah dikompresi akan ditambahkan ekstensi baru yaitu *.huf*. Contoh : *file* coba.doc setelah dikompresi akan berubah menjadi coba.doc.huf



Gambar 3.18 Form Compress (fcompress.xml) pada *Workspace* Eclipse

Tampilan form fcompress.xml pada *emulator* ditunjukkan pada gambar 3.19



Gambar 3.19 Tampilan Form fcompress.xml pada *Emulator*

Tampilan fcompress.xml pada handphone Samsung Galaxy mini ditunjukkan pada gambar 3.20



Gambar 3.20 Tampilan Form fcompress.xml pada *Handphone*

d. Form Decompress (*fdecompress.xml*)

Form ini digunakan untuk melakukan proses dekompresi terhadap *file* atau data yang ada pada *storage device* (media penyimpanan) *Android* yang telah dikompresi sebelumnya. Untuk diingat, *file* yang telah dikompresi akan mendapat tambahan ekstensi yaitu *.huf*. *Form* dekompresi ini hanya akan membaca *file* dengan ekstensi *.huf*. Jika *user* memasukkan sembarang *file*, maka akan muncul pemberitahuan bahwa *file* belum dikompresi. Pada *form* ini terdapat *button* (tombol) *Open File* untuk mencari *file* yang ingin didekompresi, pada saat kita menekan tombol *Open File* maka aplikasi akan menuju ke *form file_dialog*. Terdapat juga tombol proses untuk melakukan dekompresi pada *file* yang telah dipilih, atau tombol batal untuk kembali ke menu utama (*menu.xml*)



Gambar 3.21 Form Decompress (fdecompress.xml) pada *Workspace* Eclipse

Tampilan form fdecompress.xml pada *emulator* ditunjukkan pada gambar 3.22



Gambar 3.22 Tampilan Form fdecompress.xml pada *Emulator*

Tampilan `fdecompress.xml` pada handphone Samsung Galaxy mini ditunjukkan pada gambar 3.23



Gambar 3.23 Tampilan Form `fdcompress.xml` pada *Emulator*

e. Form Pilih File (`file_dialog_main.xml`)

Form ini akan muncul pada saat *user* memilih *file*, baik pada *form* kompres (`fcompress.xml`) maupun *form* dekompres (`fdecompress.xml`). *Form* ini berisi baris (*listing*) direktori atau *file* yang ada pada *Android storage device*. Pada *form* ini juga akan ditunjukkan *path* (lokasi) direktori. Default *path* (lokasi) direktori pada aplikasi ini adalah “/sdcard”.



Gambar 3.24 Form `file_dialog_main.xml` pada *Workspace Eclipse*
 Tampilan form `file_dialog_main.xml` pada *emulator* ditunjukkan pada gambar 3.25



Gambar 3.25 Tampilan Form `file_dialog_main.xml` pada *Emulator*

Tampilan `file_dialog_main.xml` pada handphone Samsung Galaxy mini ditunjukkan pada gambar 3.26



Gambar 3.26 Tampilan Form `file_dialog_main.xml` pada *Handphone*

f. Form *file_dialog_row.xml*

Form ini berfungsi menampilkan gambar (*icon*) untuk memperjelas *listing* yang dikeluarkan oleh form *file_dialog_main.xml*, apakah berupa folder (direktori) atau *file*.



Gambar 3.27 Form *file_dialog_row.xml* pada *Workspace* Eclipse

Tampilan form *file_dialog_row.xml* pada *emulator* ditunjukkan pada gambar 3.28



Gambar 3.28 Tampilan Form *file_dialog_row.xml* pada *Emulator*

Tampilan file `dialog_row.xml` pada handphone Samsung Galaxy mini ditunjukkan pada gambar 3.29



Gambar 3.29 Tampilan Form file `dialog_row.xml` pada *Handphone*

3.7.2 Daftar Fungsi Tombol Aplikasi

Pada tabel dibawah ini akan ditunjukkan fungsi dari keseluruhan tombol yang ada pada aplikasi kompresi data pada *handphone* berbasis *Android* menggunakan algoritma *Huffman*.

Tabel 3.4
Fungsi Aplikasi

No	DaftarTombol	Fungsi
1.	Compress File	Membuka <i>form</i> kompresi
2.	Decompress	Membuka <i>form</i> dekompresi
3.	Exit	Keluar dari program
4.	About	Membuka jendela about

5.	Cari File	Untuk mencari <i>file</i> yang akan dikompres maupun didekompres
6.	Pilih	Tombol untuk memilih <i>file</i> yang akan dikompres atau didekompres
7.	Proses	Tombol untuk melakukan proses kompresi atau dekompresi
8.	Batal	Tombol untuk kembali ke Menu Utama

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat lunak dan lingkungan perangkat keras yang digunakan untuk membangun aplikasi kompresi data pada *handphone* berbasis *Android* menggunakan algoritma *Huffman*.

4.1.1 Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam membangun aplikasi ini antara lain, yaitu :

1. Sistem operasi Windows 7 Ultimate
2. Java Development Kit (JDK) Version 6.0
3. Android Sdk Updater Revision 16
4. *Android Development Tools (ADT) plugin 16.0*
5. Android Virtual Device (AVD) manager
6. Eclipse for Java developer

4.1.2 Perangkat Keras

Adapun perangkat keras yang digunakan dalam membangun aplikasi ini antara lain, yaitu :

1. Processor Pentium Dual-Core CPU T4400 @ 2.20 GHz
2. Memory 2048MB RAM
3. Hard Disk 300 GB
4. *Monitor SVGA 14"*

5. Keyboard dan Mouse

6. *Handphone* Samsung Galaxy Mini

4.2 Implementasi Sistem

Pada sub bab ini akan dibahas mengenai implementasi dari sistem kompresi data pada *handphone* berbasis *Android* menggunakan algoritma *Huffman*.

4.2.1 Spesifikasi *Handphone* dan *Emulator*

Sebelum aplikasi dijalankan pada *handphone*, disini penulis menjalankan aplikasi dengan menggunakan *Emulator* terlebih dahulu guna mengurangi resiko error pada *handphone*. Alasan lain penulis menggunakan *Emulator* adalah karena dengan menggunakan *Emulator*, penulis bisa memilih versi android yang diinginkan. Hal ini tentu berbeda apabila langsung menggunakan *handphone* yang memiliki sistem operasi *Android*, karena jika ingin menggunakan atau merubah versi *Android* maka perlu dilakukan *upgrade*. Untuk dapat menggunakan *Emulator Android* pada PC (komputer), penulis menggunakan AVD (*Android Virtual Device*) yang terdapat pada *Eclipse* yang telah diintegrasikan dengan *plugin* ADT (*Android Development Tools*). Aplikasi ini dapat dijalankan dengan minimum API (*Application Programming Interface*) level 7 yang artinya minimum penggunaan jenis (versi) *Android* untuk aplikasi ini adalah *Android* versi 2.1 (Eclair).

Pengujian ini di lakukan untuk mengetahui tingkat keberhasilan dari tugas akhir yang telah di rancang serta di rencanakan sebelumnya pada bab perancangan. Adapun spesifikasi *Emulator* dan *smartphone* yang digunakan pada saat pengujian adalah sebagai berikut :

a. *Emulator : Android versi 2.1 (Éclair) HVGA*

- General 2G Network GSM 850 / 900 / 1800 / 1900
 - 16/32/64 GB storage
 - Prosesor 600MHz, 256 RAM
 - 3G HSDPA, 7.2 Mbps
 - WLAN Wi-Fi 802.11 b/g/n
-

- Bluetooth Yes, v3.0 with A2DP, HS
- USB Yes, microUSB v2.0
- Features OS Android2.1 QVGA)
- CPU 832 MHz processor
- Browser WAP 2.0/xHTML, HTML
- GPS Yes, with A-GPS support
- Java Yes, MIDP 2.1
- Google Search, Maps, Gmail



Gambar 4.1

Emulator yang menggunakan Android versi 2.1 (Éclair)

b. Handphone : Samsung Galaxy Mini

- General 2G Network GSM 850 / 900 / 1800 / 1900
- 3G Network HSDPA 900 / 2100
- Size Dimensions 104 x 58 x 11.5 mm
- Display Type TFT capacitive touchscreen, 256K colors
- Size 240 x 320 pixels, 3.0 inches (~133 ppi pixel density)
- TouchWiz UI
- Touch sensitive controls
- Multi-touch input method

- RAM 384MB (yang terbaca di sistem 278MB)
- CPU 600 MHz processor
- Memory Internal 160 MB
- Card slot microSD, up to 32GB
- Data GPRS Yes
- 3G HSDPA, 7.2 Mbps
- WLAN Wi-Fi 802.11 b/g/n
- Bluetooth Yes, v3.0 with A2DP, HS
- Infrared port No
- USB Yes, microUSB v2.0
- Camera Primary 2 MP, 1600x1200 pixels
- Features OS Android OS, v2.2 (Froyo)
- CPU 832 MHz processor
- Browser WAP 2.0/xHTML, HTML
- GPS Yes, with A-GPS support
- Java Yes, MIDP 2.1
- Google Search, Maps, Gmail.



Gambar 4.2

Handphone Samsung Galaxy Mini

4.3 Pengujian Aplikasi

Pengujian aplikasi dilakukan dengan menggunakan AVD (Android Virtual Device) dan menggunakan *handphone* Samsung Galaxy Mini. Pada pengujian ini dilakukan pengujian untuk seluruh form yang telah dibuat, masing-masing form diuji untuk mengetahui apakah form-form setelah dijalankan menghasilkan output atau tampilan sesuai dengan rancangan dan untuk mengetahui fungsi kerja masing-masing tombol pada suatu halaman apakah telah berjalan sesuai dengan rancangan awal.

4.3.1 Pengujian Dengan Menggunakan *Emulator*

Berdasarkan perancangan tampilan pada sub bab 3, maka dihasilkan *Form* yang terdiri dari :

1. *Halaman Splash Screen*

Halaman ini merupakan tampilan awal ketika aplikasi dijalankan. Dalam halaman ini terdapat *splash screen* yang berupa *progress bar*. Ketika *progress bar* telah selesai berjalan, *user* akan dibawa ke halaman menu utama.

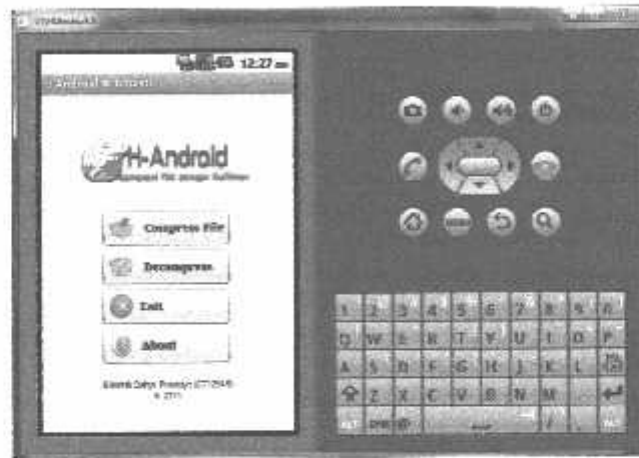


Gambar 4.3
Halaman *Splash Screen* pada *Emulator*

2. *Halaman Menu Utama*

Halaman ini adalah tampilan utama dari aplikasi kompresi menggunakan algoritma *Huffman*. Pada menu utama ini terdapat empat buah tombol (*button*) : **Compress File**, **Decompress**, **Exit**, dan **About**. Menu ini akan

membawa *user* menuju halaman yang diinginkan. Sebagai contoh, jika ingin mengkompres *File* maka *user* dapat memilih menu **Compress File**. Menu **Decompress** untuk melakukan dekompresi, Menu **Exit** untuk keluar dari program dan menu **About** untuk menampilkan profil penulis. Halaman ini dijelaskan pada gambar 4.4, gambar 4.5 dan gambar 4.6.



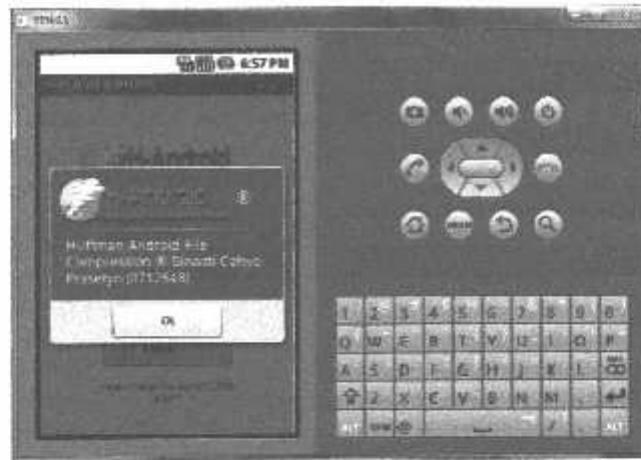
Gambar 4.4

Halaman Menu Utama pada *Emulator*



Gambar 4.5

Menu Exit pada *Emulator*



Gambar 4.6

Menu About pada *Emulator*

3. Halaman Menu *Compress File*

Setelah memilih menu *Compress File* pada menu utama, pengguna akan dibawa ke halaman ini. Pada halaman ini terdapat 3 (tiga) buah tombol, yaitu : **Proses**, **Batal**, dan **Cari File**. Untuk melakukan kompresi *File*, langkah-langkahnya adalah sebagai berikut (dijelaskan pada gambar 4.7, gambar 4.8, dan gambar 4.9):

- Pilih (klik) tombol (button) **Cari File**
- Aplikasi akan menampilkan baris folder (direktori) dan *File* pada direktori `"/sdcard"` tampilan baris dan *File* ini ditangani oleh form ***File_dialog*** (*File_dialog_main.xml* dan *File_dialog_row.xml*)
- Setelah *File* dipilih, tekan tombol (button) "**Pilih**"
- Aplikasi akan kembali ke halaman menu **Compress File**, lokasi (*path*) *File* dan nama *File* akan ditampilkan pada *text view*
- Untuk memulai kompresi klik tombol (button) **Proses**
- Tunggu beberapa saat dan ketika *File* selesai dikompresi akan muncul pemberitahuan pada **Log Proses**
- **Log Proses** berisi informasi tentang *File* yang telah dikompresi
- *File* yang telah dikompresi akan tersimpan pada direktori (folder) yang sama dengan *File* aslinya. *File* yang telah dikompresi

ditambahkan ekstensi .huf

- Untuk kembali ke Menu Utama klik tombol (*button*) **Batal**



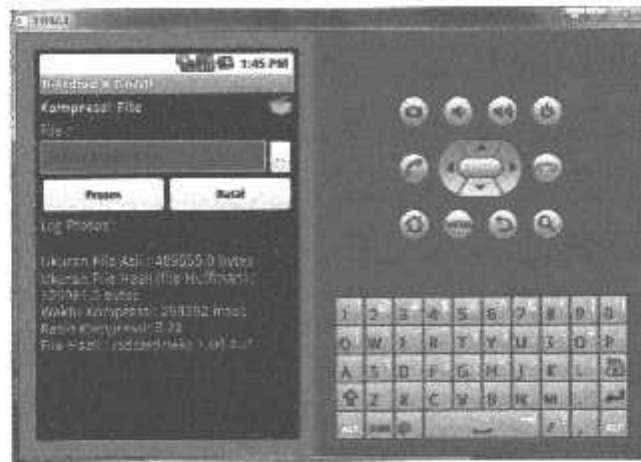
Gambar 4.7

Halaman *Compress File* pada *Emulator*



Gambar 4.8

Halaman *Pilih File* pada *Emulator*



Gambar 4.9

Proses Kompresi pada *Emulator*

4. Halaman Menu *Decompress*

Halaman ini hampir sama dengan halaman menu *Compress File*, yang membedakan adalah fungsi atau kegunaan dari halaman ini yaitu untuk melakukan dekompresi untuk *File* yang telah dikompresi sebelumnya. Padahal halaman ini juga terdapat 3 (tiga) buah tombol (*button*), yaitu : **Proses**, **Batal**, dan **Cari File**. Untuk melakukan dekompresi *File*, langkah-langkahnya adalah sebagai berikut (dijelaskan pada gambar 4.10, gambar 4.11, dan gambar 4.12):

- Pilih (klik) tombol (*button*) **Cari File**
- Aplikasi akan menampilkan baris folder (direktori) dan *File* pada direktori “/sdcard” tampilan baris dan *File* ini ditangani oleh form **File_dialog** (*File_dialog_main.xml* dan *File_dialog_row.xml*)
- *File* yang akan didekompresi harus berupa *File* dengan ekstensi .huf yang merupakan output dari proses kompresi. Jika pengguna (*user*) memasukkan *File* yang bukan berekstensi .huf, maka aplikasi akan memunculkan pesan pemberitahuan bahwa *File* belum dikompresi
- Setelah *File* dipilih, tekan tombol (*button*) “**Pilih**”
- Aplikasi akan kembali ke halaman menu **Decompress File**, lokasi (*path*) *File* dan nama *File* akan ditampilkan pada *text view*

- Untuk memulai dekompresi klik tombol (*button*) **Proses**
- Tunggu beberapa saat dan ketika *File* selesai didekompresi akan muncul pemberitahuan pada **Log Proses**
- **Log Proses** berisi informasi tentang *File* yang telah didekompresi
- *File* yang telah didekompresi akan tersimpan pada direktori (folder) yang sama dengan *File* aslinya. *File* yang telah didekompresi ditambahkan nama Huf-“nama *File*”. Misalnya : *File* dokumen.doc.huf setelah didekompresi akan berubah menjadi Huf-dokumen.doc
- Untuk kembali ke Menu Utama klik tombol (*button*) **Batal**



Gambar 4.10

Halaman Dekompresi *File* pada *Emulator*



Gambar 4.11

Pilih *File* pada *Emulator*



Gambar 4.12

Proses Dekompresi pada *Emulator*

4.3.1.1 Hasil Pengujian Aplikasi dengan Menggunakan *Emulator*

Setelah melakukan pengujian dengan menggunakan *Emulator* didapatkan hasil sebagai berikut :

a. Pengujian Halaman Splash Screen

Pada pengujian ini dilakukan pengujian untuk halaman *Splash Screen*, yaitu halaman pertama pada saat aplikasi dibuka. Dan hasilnya ditunjukkan pada table 4.1 .

Tabel 4.1

Hasil Pengujian Halaman Splash Screen pada *Emulator*

Jenis Pengujian	Keterangan	Hasil
Membuka Aplikasi	Membuka Halaman	OK

b. Pengujian Halaman Menu Utama

Pada pengujian ini dilakukan Pengujian pada Halaman Menu Utama, dan juga tombol menu yang ada didalamnya. Hasil pengujian ditunjukkan pada table 4.2.

Tabel 4.2

Hasil Pengujian Halaman Menu Utama pada *Emulator*

Jenis Pengujian	Keterangan	Hasil
Tombol <i>Compress File</i>	Membuka Halaman <i>Compress File</i>	OK
Tombol Decompress	Membuka Halaman <i>Decompress File</i>	OK
Tombol Exit	Keluar dari Aplikasi	OK
Tombol About	Menampilkan profil penulis	OK

c. Pengujian Halaman Menu *Compress File*

Pada pengujian ini dilakukan pengujian pada Halaman *Compress File* dan semua tombol yang ada pada halaman ini. Hasil pengujian ditunjukkan pada table 4.3.

Tabel 4.3

Hasil Pengujian Halaman Menu *Compress File* pada *Emulator*

Jenis Pengujian	Keterangan	Hasil
Tombol Pilih <i>File</i>	Untuk mencari dan memilih <i>File</i> yang akan dikompresi	OK
Tombol Proses	Untuk melakukan proses kompresi <i>File</i>	OK
Tombol Batal	Untuk kembali ke Menu Utama	OK
Log Proses	Memberikan informasi tentang <i>File</i> yang telah dikompresi	OK

d. Pengujian Halaman Menu *Decompress File*

Pada pengujian ini dilakukan pengujian pada Halaman *Decompress File* dan semua tombol yang ada pada halaman ini. Hasil pengujian ditunjukkan pada table 4.4.

Tabel 4.4

Hasil Pengujian Halaman Menu *Decompress File* pada *Emulator*

Jenis Pengujian	Keterangan	Hasil
Tombol Pilih <i>File</i>	Untuk mencari dan memilih <i>File</i> yang akan didekompresi	OK
Tombol Proses	Untuk melakukan proses dekomposisi <i>File</i>	OK
Tombol Batal	Untuk kembali ke Menu Utama	OK
Log Proses	Memberikan informasi tentang <i>File</i> yang telah didekompresi	OK

4.3.2 Pengujian Dengan Menggunakan *Handphone* Android Galaxy Mini

Berdasarkan perancangan tampilan pada sub bab 3, maka dihasilkan *Form* yang terdiri dari :

1 Halaman *Splash Screen*

Halaman ini merupakan tampilan awal ketika aplikasi dijalankan. Dalam halaman ini terdapat *splash screen* yang berupa *progress bar*. Ketika *progress bar* telah selesai berjalan, *user* akan dibawa ke halaman menu utama.



Gambar 4.13

Halaman Utama pada *Handphone*

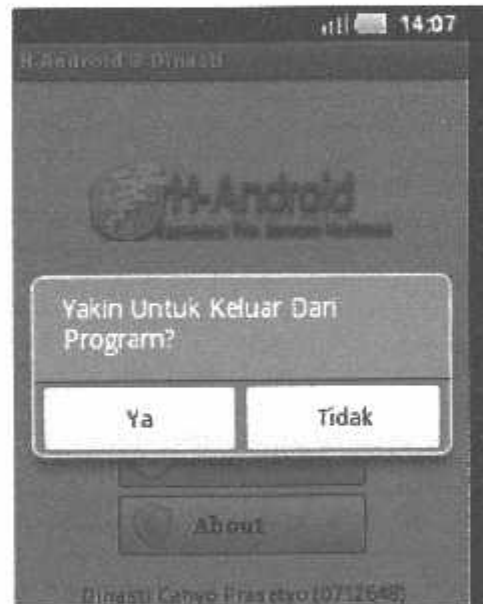
2 Halaman Menu Utama

Halaman ini adalah tampilan utama dari aplikasi kompresi menggunakan algoritma *Huffman*. Pada menu utama ini terdapat empat buah tombol (*button*) : **Compress File**, **Decompress**, **Exit**, dan **About**. Menu ini akan membawa *user* menuju halaman yang diinginkan. Sebagai contoh, jika ingin mengkompres *File* maka *user* dapat memilih menu **Compress File**. Menu **Decompress** untuk melakukan dekompresi, Menu **Exit** untuk keluar dari program dan menu **About** untuk menampilkan profil penulis. Ditunjukkan pada gambar 4.14, gambar 4.15, dan gambar 4.16.



Gambar 4.14

Halaman Menu Utama pada *Handphone*



Gambar 4.15

Menu Exit pada *Handphone*



Gambar 4.16

Gambar Menu About pada *Handphone*

3 Halaman Menu *Compress File*

Setelah memilih menu *Compress File* pada menu utama, pengguna akan dibawa ke halaman ini. Pada halaman ini terdapat 3 (tiga) buah tombol, yaitu : **Proses**, **Batal**, dan **Cari File**. Untuk melakukan kompresi *File*, langkah-langkahnya adalah sebagai berikut (ditunjukkan pada gambar 4.17, gambar 4.18, dan gambar 4.19) :

- Pilih (klik) tombol (button) **Cari File**
- Aplikasi akan menampilkan baris folder (direktori) dan *File* pada direktori “/sdcard” tampilan baris dan *File* ini ditangani oleh form **File_dialog** (*file_dialog_main.xml* dan *file_dialog_row.xml*)
- Setelah *File* dipilih, tekan tombol (button) “**Pilih**”
- Aplikasi akan kembali ke halaman menu **Compress File**, lokasi (*path*) *File* dan nama *File* akan ditampilkan pada *text view*
- Untuk memulai kompresi klik tombol (button) **Proses**
- Tunggu beberapa saat dan ketika *File* selesai dikompresi akan muncul pemberitahuan pada **Log Proses**
- **Log Proses** berisi informasi tentang *File* yang telah dikompresi
- *File* yang telah dikompresi akan tersimpan pada direktori (folder)

yang sama dengan *File* aslinya. *File* yang telah dikompresi ditambahkan ekstensi *.huf*

- Untuk kembali ke Menu Utama klik tombol (*button*) **Batal**



Gambar 4.17

Halaman *Compress File* pada *Handphone*



Gambar 4.18

Pilih *File* pada *Handphone*



Gambar 4.19

Proses Kompresi pada *Handphone*

4 Halaman Menu *Decompress*

Halaman ini hampir sama dengan halaman menu *Compress File*, yang membedakan adalah fungsi atau kegunaan dari halaman ini yaitu untuk melakukan dekompresi untuk *File* yang telah dikompresi sebelumnya. Padahal halaman ini juga terdapat 3 (tiga) buah tombol (*button*), yaitu : **Proses**, **Batal**, dan **Cari File**. Untuk melakukan dekompresi *File*, langkah-langkahnya adalah sebagai berikut (ditunjukkan pada gambar 4.20, gambar 4.21, dan gambar 4.22) :

- Pilih (klik) tombol (*button*) **Cari File**
- Aplikasi akan menampilkan baris folder (*direktori*) dan *File* pada direktori “/sdcard” tampilan baris dan *File* ini ditangani oleh form ***File_dialog*** (*File_dialog_main.xml* dan *File_dialog_row.xml*)
- *File* yang akan didekompresi harus berupa *File* dengan ekstensi .huf yang merupakan output dari proses kompresi. Jika pengguna (*user*) memasukkan *File* yang bukan berekstensi .huf, maka aplikasi akan memunculkan pesan pemberitahuan bahwa *File* belum dikompresi
- Setelah *File* dipilih, tekan tombol (*button*) “**Pilih**”

- Aplikasi akan kembali ke halaman menu **Decompress File**, lokasi (*path*) *File* dan nama *File* akan ditampilkan pada *text view*
- Untuk memulai dekompresi klik tombol (*button*) **Proses**
- Tunggu beberapa saat dan ketika *File* selesai didekompresi akan muncul pemberitahuan pada **Log Proses**
- **Log Proses** berisi informasi tentang *File* yang telah didekompresi
- *File* yang telah didekompresi akan tersimpan pada direktori (*folder*) yang sama dengan *File* aslinya. *File* yang telah didekompresi ditambahkan nama Huf-“nama *File*”. Misalnya : *File* dokumen.doc.huf setelah didekompresi akan berubah menjadi Huf-dokumen.doc
- Untuk kembali ke Menu Utama klik tombol (*button*) **Batal**



Gambar 4.20

Halaman Decompress *File* pada *Handphone*



Gambar 4.21

Pilih *File* pada *Handphone*



Gambar 4.22

Proses Dekompresi pada *Handphone*

4.3.2.1 Hasil pengujian Aplikasi dengan Menggunakan *Handphone* Galaxy

Mini

Setelah melakukan pengujian didapatkan hasil sebagai berikut,yaitu:

a. Pengujian Halaman Splash Screen

Pengujian ini dilakukan dengan cara membuka aplikasi, apabila setelah aplikasi dibuka dan kemudian muncul halaman *splash screen* maka pengujian dikatakan berhasil. Hasil pengujian ditunjukkan pada table 4.5

Tabel 4.5

Hasil Pengujian Halaman *Splash Screen* pada *Handphone*

Jenis Pengujian	Keterangan	Hasil
Membuka Aplikasi	Membuka Halaman	OK

b. Pengujian Halaman Menu Utama

Pengujian ini dilakukan dengan cara memilih salah satu tombol (menu) pada halaman utama. Hasil pengujian ditunjukkan pada table 4.6

Tabel 4.6

Hasil Pengujian Halaman Menu Utama pada *Handphone*

Jenis Pengujian	Keterangan	Hasil
Tombol <i>Compress File</i>	Membuka Halaman <i>Compress File</i>	OK
Tombol <i>Decompress</i>	Membuka Halaman <i>Decompress File</i>	OK
Tombol <i>Exit</i>	Keluar dari Aplikasi	OK
Tombol <i>About</i>	Menampilkan profil penulis	OK

c. Pengujian Halaman Menu *Compress File*

Pada pengujian ini dilakukan pengujian untuk mengetahui apakah semua tombol (menu) pada halaman kompresi dapat berfungsi. Hasil dari pengujian ditunjukkan pada table 4.7.

Tabel 4.7

Hasil Pengujian Halaman Menu *Compress File* pada *Handphone*

Jenis Pengujian	Keterangan	Hasil
Tombol Pilih <i>File</i>	Untuk mencari dan memilih <i>File</i> yang akan dikompresi	OK
Tombol Proses	Untuk melakukan proses kompresi <i>File</i>	OK
Tombol Batal	Untuk kembali ke Menu Utama	OK
Log Proses	Memberikan informasi tentang <i>File</i> yang telah dikompresi	OK

d. Pengujian Halaman Menu *Decompress File*

Pada pengujian ini dilakukan pengujian untuk mengetahui apakah semua tombol (menu) pada halaman dekompresi dapat berfungsi. Hasil dari pengujian ditunjukkan pada table 4.8.

Tabel 4.8

Hasil Pengujian Halaman Menu *Decompress File* pada *Handphone*

Jenis Pengujian	Keterangan	Hasil
Tombol Pilih <i>File</i>	Untuk mencari dan memilih <i>File</i> yang akan didekompresi	OK

Tombol Proses	Untuk melakukan proses dekompresi <i>File</i>	OK
Tombol Batal	Untuk kembali ke Menu Utama	OK
Log Proses	Memberikan informasi tentang <i>File</i> yang telah didekompresi	OK

4.4 Pengujian Sistem Terhadap Berbagai Macam *File*

Pada pengujian ini, akan diujikan *File* atau data yang berupa teks, audio video, dan gambar (*image*). Akan diambil beberapa sample *File* atau data yang berupa teks, audio video dan gambar. Untuk *File* yang berupa teks, *File* yang akan diambil sampel adalah .txt, .doc, .pdf, .rtf. Untuk *File* yang berupa audio video *File* yang akan diambil : .mp3, .mp4, .flv, . Dan untuk *File* yang berupa gambar (*image*) *File* yang akan diambil sampel : .jpg, .gif, .bmp, .png

4.4.1 Pengujian Kompresi

Dalam pengujian ini, dilakukan pada *Emulator* dan pada *handphone* Samsung Galaxy Mini. Hasil kompresi data dengan *File* yang berjenis teks, image dan audio ditunjukkan pada tabel 4.9 dan table 4.10. Rasio Kompresi dihitung menggunakan persamaan 2.1. Apabila ratio kompresi < 1 menunjukkan bahwa *file* berhasil dikompresi dan apabila ratio kompresi > 1 berarti *file* menjadi lebih besar.

a. Pengujian Kompresi pada untuk *File* .txt

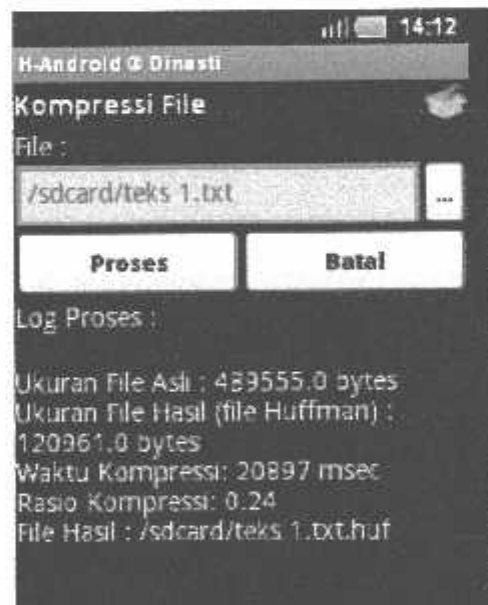
Pada pengujian ini dilakukan kompresi pada *file* .txt dengan ukuran 489 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file* .txt dengan ukuran 120 kb dan membutuhkan waktu 269 detik. Ratio kompresi untuk pengujian ini = 0,24 yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.23.



Gambar 4.23

Kompresi File .txt pada Emulator

Pada pengujian ini dilakukan kompresi pada *file* .txt dengan ukuran 489 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file* .txt dengan ukuran 120 kb dan membutuhkan waktu 20 detik. Ratio kompresi untuk pengujian ini = 0,24 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.24.



Gambar 4.24

Kompresi File .txt pada Handphone

b. Pengujian Kompresi pada untuk *File .doc*

Pada pengujian ini dilakukan kompresi pada *file .doc* dengan ukuran 105 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .doc* dengan ukuran 47 kb dan membutuhkan waktu 64 detik. Ratio kompresi untuk pengujian ini = 0,44 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.25.



Gambar 4.25

Kompresi *File .doc* pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file .doc* dengan ukuran 105 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file .doc* dengan ukuran 47 kb dan membutuhkan waktu 6 detik. Ratio kompresi untuk pengujian ini = 0,44 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.26.



Gambar 4.26

Kompresi *File .doc* pada *Handphone*

c. Pengujian Kompresi pada untuk *File .pdf*

Pada pengujian ini dilakukan kompresi pada *file .pdf* dengan ukuran 1.575 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .pdf* dengan ukuran 1570 kb dan membutuhkan waktu 1336 detik. Ratio kompresi untuk pengujian ini = 0,99 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.27.



Gambar 4.27

Kompresi *File .pdf* pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file* .pdf dengan ukuran 1.575 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file* .pdf dengan ukuran 1570 kb dan membutuhkan waktu 178 detik. Ratio kompresi untuk pengujian ini = 0,99 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.28.



Gambar 4.28

Kompresi *File* .pdf pada *Handphone*

d. Pengujian Kompresi pada untuk *File* .rtf

Pada pengujian ini dilakukan kompresi pada *file* .rtf dengan ukuran 197 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file* .rtf dengan ukuran 121 kb dan membutuhkan waktu 110 detik. Ratio kompresi untuk pengujian ini = 0,61 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.29.



Gambar 4.29

Kompresi File .rtf pada Emulator

Pada pengujian ini dilakukan kompresi pada file .rtf dengan ukuran 197 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan file .rtf dengan ukuran 121 kb dan membutuhkan waktu 11 detik. Ratio kompresi untuk pengujian ini = 0,61 , yang berarti file berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.30.



Gambar 4.30

Kompresi File .rtf pada Handphone

e. Pengujian Kompresi pada untuk *File .mp3*

Pada pengujian ini dilakukan kompresi pada *file .mp3* dengan ukuran 155 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .mp3* dengan ukuran 154 kb dan membutuhkan waktu 126 detik. Ratio kompresi untuk pengujian ini = 0,99 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.31.



Gambar 4.31

Kompresi *File .mp3* pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file .mp3* dengan ukuran 155 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file .mp3* dengan ukuran 154 kb dan membutuhkan waktu 16 detik. Ratio kompresi untuk pengujian ini = 0,99 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.32.



Gambar 4.32

Kompresi *File* .mp3 pada *Handphone*

f. Pengujian Kompresi pada untuk *File* .mp4

Pada pengujian ini dilakukan kompresi pada *file* .mp4 dengan ukuran 1.674 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file* .mp4 dengan ukuran 1.642 kb dan membutuhkan waktu 1.418 detik. Ratio kompresi untuk pengujian ini = 0,98 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.33.



Gambar 4.33

Kompresi *File* .mp4 pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file* .mp4 dengan ukuran 1.674 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file* .mp4 dengan ukuran 1.642 kb dan membutuhkan waktu 188 detik. Ratio kompresi untuk pengujian ini = 0,98 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.34.



Gambar 4.34

Kompresi *File* .mp4 pada *Handphone*

g. Pengujian Kompresi pada untuk *File* .flv

Pada pengujian ini dilakukan kompresi pada *file* .flv dengan ukuran 3.973 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file* .flv dengan ukuran 3.937 kb dan membutuhkan waktu 3.329 detik. Ratio kompresi untuk pengujian ini = 0,99 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.35.



Gambar 4.35

Kompresi File .flv pada Emulator

Pada pengujian ini dilakukan kompresi pada file .flv dengan ukuran 3.973 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan file .flv dengan ukuran 3.937 kb dan membutuhkan waktu 486 detik. Ratio kompresi untuk pengujian ini = 0,99 , yang berarti file berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.36.



Gambar 4.36

Kompresi File .flv pada Handphone

h. Pengujian Kompresi pada untuk *File .bmp*

Pada pengujian ini dilakukan kompresi pada *file .bmp* dengan ukuran 198 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .bmp* dengan ukuran 102 kb dan membutuhkan waktu 170 detik. Ratio kompresi untuk pengujian ini = 0,51 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.37.



Gambar 4.37

Kompresi *File .bmp* pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file .bmp* dengan ukuran 198 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file .bmp* dengan ukuran 102 kb dan membutuhkan waktu 24 detik. Ratio kompresi untuk pengujian ini = 0,51 , yang berarti *file* berhasil dimampatkan (dikompres). Hasil dari pengujian ditunjukkan pada gambar 4.38.



Gambar 4.38

Kompresi *File .bmp* pada *Handphone*

i. Pengujian Kompresi pada untuk *File .gif*

Pada pengujian ini dilakukan kompresi pada *file .gif* dengan ukuran 92 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .gif* dengan ukuran 96 kb dan membutuhkan waktu 85 detik. Ratio kompresi untuk pengujian ini = 1,04 , yang berarti *file* bertambah besar setelah dilakukan kompresi. Hasil dari pengujian ditunjukkan pada gambar 4.39.



Gambar 4.39

Kompresi *File .gif* pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file .gif* dengan ukuran 92 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file .gif* dengan ukuran 96 kb dan membutuhkan waktu 11 detik. Ratio kompresi untuk pengujian ini = 1,04 , yang berarti *file* bertambah besar setelah dilakukan kompresi. Hasil dari pengujian ditunjukkan pada gambar 4.40.



Gambar 4.40

Kompresi *File .gif* pada *Handphone*

j. Pengujian Kompresi pada untuk *File .jpg*

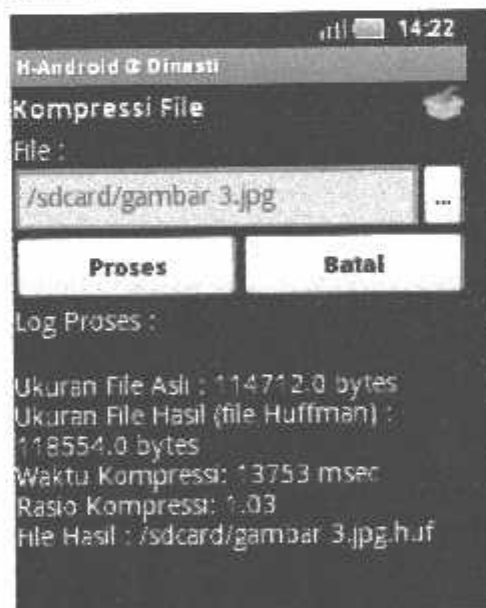
Pada pengujian ini dilakukan kompresi pada *file .jpg* dengan ukuran 114 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .jpg* dengan ukuran 118 kb dan membutuhkan waktu 100 detik. Ratio kompresi untuk pengujian ini = 1,03 , yang berarti *file* bertambah besar setelah dilakukan kompresi. Hasil dari pengujian ditunjukkan pada gambar 4.41.



Gambar 4.41

Kompresi File .jpg pada Emulator

Pada pengujian ini dilakukan kompresi pada file .jpg dengan ukuran 114 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan file .jpg dengan ukuran 118 kb dan membutuhkan waktu 100 detik. Rasio kompresi untuk pengujian ini = 1,03 , yang berarti file bertambah besar setelah dilakukan kompresi. Hasil dari pengujian ditunjukkan pada gambar 4.42.



Gambar 4.42

Kompresi File .jpg pada Handphone

k. Pengujian Kompresi pada untuk *File .png*

Pada pengujian ini dilakukan kompresi pada *file .png* dengan ukuran 128 kb pada *emulator*. Setelah dilakukan kompresi pada *emulator* dihasilkan *file .png* dengan ukuran 133 kb dan membutuhkan waktu 151 detik. Ratio kompresi untuk pengujian ini = 1,03 , yang berarti *file* bertambah besar setelah dilakukan kompresi. Hasil dari pengujian ditunjukkan pada gambar 4.43.



Gambar 4.43

Kompresi *File .png* pada *Emulator*

Pada pengujian ini dilakukan kompresi pada *file .png* dengan ukuran 128 kb pada *handphone*. Setelah dilakukan kompresi pada *handphone* dihasilkan *file .png* dengan ukuran 133 kb dan membutuhkan waktu 151 detik. Ratio kompresi untuk pengujian ini = 1,03 , yang berarti *file* bertambah besar setelah dilakukan kompresi. Hasil dari pengujian ditunjukkan pada gambar 4.44.



Gambar 4.44

Kompresi *File .png* pada *Handphone*

4.4.1.1 Hasil Pengujian Kompresi pada *Emulator* dan *Handphone*

Dari hasil pengujian kompresi menggunakan *emulator* dan *handphone* hampir semua jenis *file* dapat terkompresi (termampatkan), kecuali pada *file* dengan ekstensi *.gif*, *.jpg*, dan *.png*. Hal ini dikarenakan ketiga ekstensi *file* tersebut sudah merupakan *file* atau data yang telah mengalami proses kompresi pada proses pembentukannya (dijelaskan pada bab II). Hasil yang diperoleh dari pengujian, baik menggunakan *emulator* ataupun menggunakan *handphone* didapatkan bahwa hasil kompresi dengan menggunakan *emulator* dan *handphone* menghasilkan pemampatan (kompresi) dan ratio kompresi yang sama (menghasilkan ukuran *file* yang sama). Satu hal yang membedakan adalah waktu yang diperlukan untuk mengkompresi (memampatkan) sebuah *file*, dari hasil pengujian didapatkan bahwa waktu yang diperlukan *emulator* untuk mengkompresi (memampatkan) sebuah *file* lebih lama daripada waktu yang diperlukan *handphone* untuk mengkompresi (memampatkan) sebuah *file*. Hasil dari pengujian dengan menggunakan *emulator* ditunjukkan pada tabel 4.9. Dan hasil pengujian dengan menggunakan *handphone* ditunjukkan pada tabel 4.10.

Tabel 4.9

Kompresi Data pada *File* teks, image, audio pada *Emulator*

Nama File	Ukuran File Asli	Ukuran File Kompres	Rasio	Waktu (s)
Teks1.txt	489 kb	120 kb	0.24	269.392
Teks2.doc	105 kb	47 kb	0.4	64.156
Teks3.pdf	1.575 kb	1.570 kb	0.99	1336.527
Teks4.rtf	197 kb	121 kb	0.61	110.500
Audio1.mp3	155 kb	154 kb	0.99	126.268
Video1.mp4	1.674 kb	1.642 kb	0.98	1418.789
Video2.flv	3.973 kb	3.937 kb	0.99	3329.028
Gambar1.bmp	198 kb	102 kb	0.51	170.541
Gambar2.gif	92 kb	96 kb	1.04	85.467
Gambar3.jpg	114 kb	118 kb	1.03	100.112
Gambar4.png	128 kb	133 kb	1.03	151.105

Tabel 4.10

Kompresi Data pada *File* teks, image, audio pada *Handphone*

Nama File	Ukuran File Asli	Ukuran File Kompres	Ratio	Waktu (s)
Teks1.txt	489 kb	120 kb	0.24	20.897
Teks2.doc	105 kb	47 kb	0.44	6.761

Teks3.pdf	1.575 kb	1.570 kb	0.99	178.818
Teks4.rtf	197 kb	121 kb	0.61	11.065
Audio1.mp3	155 kb	154 kb	0.99	16.987
Video1.mp4	1.674 kb	1.642 kb	0.98	188.182
Video2.flv	3.973 kb	3.937 kb	0.99	486.852
Gambar1.bmp	198 kb	102 kb	0.51	24.953
Gambar2.gif	92 kb	96 kb	1.04	11.260
Gambar3.jpg	114 kb	118 kb	1.03	13.753
Gambar4.png	128 kb	133 kb	1.03	14.915

4.4.2 Pengujian Dekompresi

Dalam pengujian ini, dilakukan pada *Emulator* dan pada *handphone* Samsung Galaxy Mini. Hasil dekompresi data dengan *File* yang berjenis teks, image dan audio ditunjukkan pada tabel 4.11 dan tabel 4.12

a. Pengujian Dekompresi pada untuk *File .txt*

Pada pengujian ini dilakukan dekompresi pada *file .txt* dengan ukuran 120 kb pada *emulator*. Setelah dilakukan dekompresi pada *emulator* dihasilkan *file .txt* dengan ukuran 489 kb dan membutuhkan waktu 175 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.45.



Gambar 4.45

Dekomposisi File .txt pada Emulator

Pada pengujian ini dilakukan dekomposisi pada file .txt dengan ukuran 120 kb pada *handphone*. Setelah dilakukan dekomposisi pada *handphone* dihasilkan file .txt dengan ukuran 489 kb dan membutuhkan waktu 13 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.46.



Gambar 4.46

Dekomposisi File .txt pada Handphone

b. Pengujian Dekompresi pada untuk *File .doc*

Pada pengujian ini dilakukan dekomposisi pada *file .doc* dengan ukuran 47 kb pada *emulator*. Setelah dilakukan dekomposisi pada *emulator* dihasilkan *file .doc* dengan ukuran 105 kb dan membutuhkan waktu 42 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.47.



Gambar 4.47

Dekomposisi *File .doc* pada *Emulator*

Pada pengujian ini dilakukan dekomposisi pada *file .doc* dengan ukuran 47 kb pada *handphone*. Setelah dilakukan dekomposisi pada *handphone* dihasilkan *file .doc* dengan ukuran 105 kb dan membutuhkan waktu 42 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.48.



Gambar 4.48

Dekomposisi *File .doc* pada *Handphone*

c. Pengujian Dekompresi pada untuk *File .pdf*

Pada pengujian ini dilakukan dekomposisi pada *file .pdf* dengan ukuran 1.570 kb pada *emulator*. Setelah dilakukan dekomposisi pada *emulator* dihasilkan *file .pdf* dengan ukuran 1.575 kb dan membutuhkan waktu 878 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.49.



Gambar 4.49

Dekomposisi *File .pdf* pada *Emulator*

Pada pengujian ini dilakukan dekompresi pada *file* .pdf dengan ukuran 1.570 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan *file* .pdf dengan ukuran 1.575 kb dan membutuhkan waktu 74 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.50.



Gambar 4.50

Dekomposisi *File* .pdf pada *Handphone*

d. Pengujian Dekompresi pada untuk *File* .rtf

Pada pengujian ini dilakukan dekompresi pada *file* .rtf dengan ukuran 121 kb pada *emulator*. Setelah dilakukan dekompresi pada *emulator* dihasilkan *file* .rtf dengan ukuran 197 kb dan membutuhkan waktu 89 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.51.



Gambar 4.51

Dekompresi *File .rtf* pada *Emulator*

Pada pengujian ini dilakukan dekompresi pada file *.rtf* dengan ukuran 121 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan file *.rtf* dengan ukuran 197 kb dan membutuhkan waktu 7 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.52.



Gambar 4.52

Dekompresi *File .rtf* pada *Handphone*

e. Pengujian Dekompresi pada untuk *File .mp3*

Pada pengujian ini dilakukan dekomposisi pada *file .mp3* dengan ukuran 154 kb pada *emulator*. Setelah dilakukan dekomposisi pada *emulator* dihasilkan *file .mp3* dengan ukuran 155 kb dan membutuhkan waktu 91 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.53.



Gambar 4.53

Dekomposisi *File .mp3* pada *Emulator*

Pada pengujian ini dilakukan dekomposisi pada *file .mp3* dengan ukuran 154 kb pada *handphone*. Setelah dilakukan dekomposisi pada *handphone* dihasilkan *file .mp3* dengan ukuran 155 kb dan membutuhkan waktu 6 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.54.



Gambar 4.54

Dekomposisi *File .mp3* pada *Handphone*

f. Pengujian Dekompresi pada untuk *File .mp4*

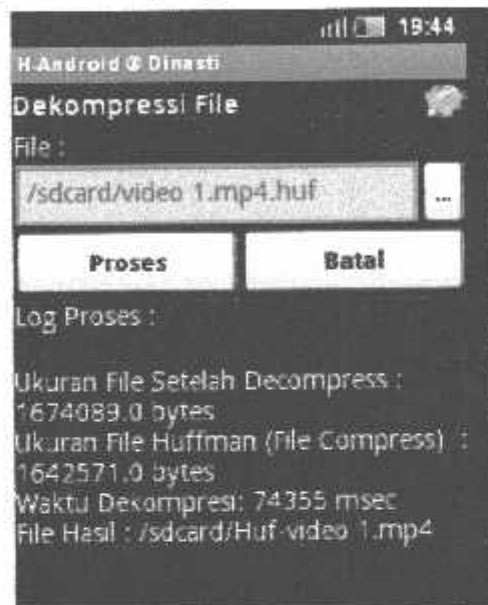
Pada pengujian ini dilakukan dekomposisi pada *file .mp4* dengan ukuran 1.642 kb pada *emulator*. Setelah dilakukan dekomposisi pada *emulator* dihasilkan *file .mp4* dengan ukuran 1.647 kb dan membutuhkan waktu 899 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.55.



Gambar 4.55

Dekomposisi *File .mp4* pada *Emulator*

Pada pengujian ini dilakukan dekompresi pada *file* .mp4 dengan ukuran 1.642 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan *file* .mp4 dengan ukuran 1.647 kb dan membutuhkan waktu 74 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.56.

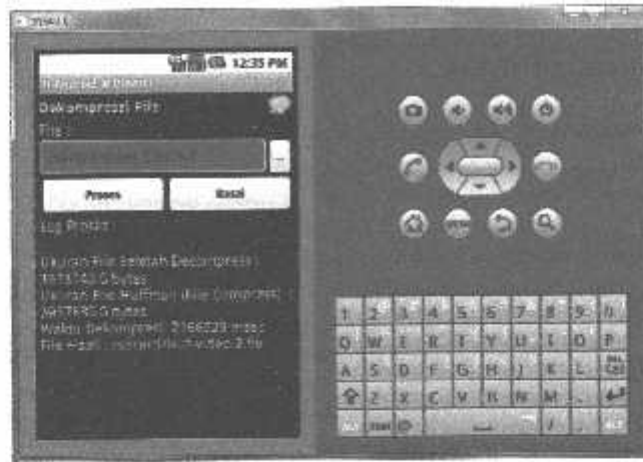


Gambar 4.56

Dekompresi *File* .mp4 pada *Handphone*

g. Pengujian Dekompresi pada untuk *File* .flv

Pada pengujian ini dilakukan dekompresi pada *file* .flv dengan ukuran 3.937 kb pada *emulator*. Setelah dilakukan dekompresi pada *emulator* dihasilkan *file* .flv dengan ukuran 3.973 kb dan membutuhkan waktu 2166 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.57.



Gambar 4.57

Dekompresi File .flv pada Emulator

Pada pengujian ini dilakukan dekompresi pada file .flv dengan ukuran 3.937 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan file .flv dengan ukuran 3.973 kb dan membutuhkan waktu 188 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.58.



Gambar 4.58

Dekompresi File .flv pada Handphone

h. Pengujian Dekompresi pada untuk *File .bmp*

Pada pengujian ini dilakukan dekomposisi pada *file .bmp* dengan ukuran 102 kb pada *emulator*. Setelah dilakukan dekomposisi pada *emulator* dihasilkan *file .bmp* dengan ukuran 198 kb dan membutuhkan waktu 84 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.59.



Gambar 4.59

Dekomposisi *File .bmp* pada *Emulator*

Pada pengujian ini dilakukan dekomposisi pada *file .bmp* dengan ukuran 102 kb pada *handphone*. Setelah dilakukan dekomposisi pada *handphone* dihasilkan *file .bmp* dengan ukuran 198 kb dan membutuhkan waktu 6 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.60.



Gambar 4.60

Dekomposisi *File* .bmp pada *Handphone*

i. Pengujian Dekompresi pada untuk *File* .gif

Pada pengujian ini dilakukan dekomposisi pada *file* .gif dengan ukuran 96 kb pada *emulator*. Setelah dilakukan dekomposisi pada *emulator* dihasilkan *file* .gif dengan ukuran 92 kb dan membutuhkan waktu 51 detik. Dari diperoleh bahwa, setelah dilakukan proses dekomposisi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.61.



Gambar 4.61

Dekomposisi *File* .gif pada *Emulator*

Pada pengujian ini dilakukan dekompresi pada *file* .gif dengan ukuran 96 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan *file* .gif dengan ukuran 92 kb dan membutuhkan waktu 4 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.62.



Gambar 4.62

Dekompresi *File* .gif pada *Handphone*

j. Pengujian Dekompresi pada untuk *File* .jpg

Pada pengujian ini dilakukan dekompresi pada *file* .jpg dengan ukuran 118 kb pada *emulator*. Setelah dilakukan dekompresi pada *emulator* dihasilkan *file* .jpg dengan ukuran 114 kb dan membutuhkan waktu 64 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.63.



Gambar 4.63

Dekompresi File .jpg pada Emulator

Pada pengujian ini dilakukan dekompresi pada file .jpg dengan ukuran 118 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan file .jpg dengan ukuran 114 kb dan membutuhkan waktu 5 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.64.



Gambar 4.64

Dekompresi File .jpg pada Handphone.

k. Pengujian Kompresi pada untuk *File .png*

Pada pengujian ini dilakukan dekompresi pada *file .png* dengan ukuran 133 kb pada *emulator*. Setelah dilakukan dekompresi pada *emulator* dihasilkan *file .png* dengan ukuran 128 kb dan membutuhkan waktu 72 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.65.



Gambar 4.65

Dekompresi *File .png* pada *Emulator*

Pada pengujian ini dilakukan dekompresi pada *file .png* dengan ukuran 133 kb pada *handphone*. Setelah dilakukan dekompresi pada *handphone* dihasilkan *file .png* dengan ukuran 128 kb dan membutuhkan waktu 5 detik. Dari diperoleh bahwa, setelah dilakukan proses dekompresi file akan kembali ke ukuran asal (sebelum dilakukan proses kompresi). Hasil dari pengujian ditunjukkan pada gambar 4.66.



Gambar 4.66

Dekomposisi *File* .png pada *Handphone*

4.4.2.1 Hasil Pengujian Dekompresi pada *Emulator* dan *Handphone*

Dari hasil pengujian dekompresi menggunakan *emulator* dan *handphone* didapatkan bahwa semua jenis *file* yang didekompresi akan kembali ke ukuran semua (sebelum dilakukan kompresi). Hasil yang diperoleh dari pengujian, baik menggunakan *emulator* ataupun menggunakan *handphone* didapatkan bahwa hasil dekompresi dengan menggunakan *emulator* dan *handphone* menghasilkan ukuran dekompresi yang sama. Satu hal yang membedakan adalah waktu yang diperlukan untuk melakukan dekompresi sebuah *file*, dari hasil pengujian didapatkan bahwa waktu yang diperlukan *emulator* untuk dekompresi sebuah *file* lebih lama daripada waktu yang diperlukan *handphone* untuk dekompresi sebuah *file*. Hasil dari pengujian dengan menggunakan *emulator* ditunjukkan pada tabel 4.11 . Dan hasil pengujian dengan menggunakan *handphone* ditunjukkan pada tabel 4.12 .

Tabel 4.11

Dekompresi Data pada *File* teks, image, audio pada *Emulator*

Nama <i>File</i>	Ukuran <i>File</i> Kompres	Ukuran <i>File</i> Dekompres	Waktu (s)
Teks1.txt	120 kb	489 kb	175.320
Teks2.doc	47 kb	105 kb	42.915
Teks3.pdf	1.570 kb	1.575 kb	878.596
Teks4.rtf	121 kb	197 kb	89.721
Audio1.mp3	154 kb	155 kb	91.256
Video1.mp4	1.642 kb	1.647 kb	899.270
Video2.flv	3.937 kb	3.973 kb	2166.029
Gambar1.bmp	102 kb	198 kb	84.145
Gambar2.gif	96 kb	92 kb	51.833
Gambar3.jpg	118 kb	114 kb	64.009
Gambar4.png	133 kb	128 kb	72.133

Tabel 4.12

Dekompresi data pada *File* teks, image, audio pada *handphone*

Nama <i>File</i>	Ukuran <i>File</i> Kompres	Ukuran <i>File</i> Dekompres	Waktu (s)
Teks1.txt	120 kb	489 kb	13.954

Teks2.doc	47 kb	105 kb	8.648
Teks3.pdf	1.570 kb	1.575 kb	74.528
Teks4.rtf	121 kb	197 kb	7.492
Audio1.mp3	154 kb	155 kb	6.873
Video1.mp4	1.642 kb	1.674 kb	74.355
Video2.flv	3.937 kb	3.973 kb	188.349
Gambar1.bmp	102 kb	198 kb	6.913
Gambar2.gif	96 kb	92 kb	4.286
Gambar3.jpg	118 kb	114 kb	5.143
Gambar4.png	133 kb	128 kb	5.811

4.4.3 Hasil Pengujian Kompresi dan Dekompresi pada *Emulator* dan *Handphone*

Tabel 4.13 merupakan tabel yang menunjukkan percobaan kompresi dan dekompresi menggunakan *emulator* dan menggunakan *handphone*. Pada *file* dengan ekstensi *.gif*, *.jpg*, dan *.png* dikatakan pengujian gagal. Hal ini dikarenakan ketiga ekstensi *file* tersebut sudah merupakan *file* atau data yang telah mengalami proses kompresi pada proses pembentukannya (dijelaskan pada bab II).

Tabel 4.13

Hasil Pengujian Kompresi dan Dekompresi pada *Emulator* dan *Handphone*

<i>File</i> yang diujikan	Kompresi (<i>Emulator</i>)	Kompresi (<i>Handphone</i>)	Dekompresi (<i>Emulator</i>)	Dekompresi (<i>Handphone</i>)
---------------------------	------------------------------	-------------------------------	--------------------------------	---------------------------------

Teks1.txt	OK	OK	OK	OK
Teks2.doc	OK	OK	OK	OK
Teks3.pdf	OK	OK	OK	OK
Teks4.rtf	OK	OK	OK	OK
Audio1.mp3	OK	OK	OK	OK
Video1.mp4	OK	OK	OK	OK
Video2.flv	OK	OK	OK	OK
Gambar1.bmp	OK	OK	OK	OK
Gambar2.gif	Gagal	Gagal	OK	OK
Gambar3.jpg	Gagal	Gagal	OK	OK
Gambar4.png	Gagal	Gagal	OK	OK

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil percobaan dan pembahasan dapat disimpulkan beberapa hal sebagai berikut:

1. Dihasilkan sebuah aplikasi yang dapat digunakan untuk melakukan kompresi dan dekompresi terhadap berbagai jenis *file*.
2. Dari sampel *file* yang diujikan *file* yang berhasil dikompresi (dimampatkan) adalah : .txt, .doc, .pdf, .rtf, .bmp, .mp3, .mp4, dan .flv. Untuk *file* .gif, .jpg, dan .png ukuran *file* semakin membesar. Aplikasi kompresi dengan menggunakan algoritma *Huffman* ini berhasil memampatkan *file* dengan format : .txt, .doc, .pdf, .rtf, .bmp, .mp3, .mp4, dan .flv. *File* .gif, .jpg, dan .png merupakan *file* yang sudah dimampatkan (dikompresi), seperti yang telah dijelaskan pada Bab II. Dari pengujian yang telah dilakukan, aplikasi ini tidak dapat digunakan untuk mengkompresi *file* yang sudah terkompresi.
3. Kecepatan proses kompresi dan dekompresi data setara dengan ukuran dan jenis *file*.
4. Jika ratio kompresi < 1 menunjukkan bahwa *file* berhasil dikompresi (ukurannya menjadi lebih kecil) dan apabila ratio kompresi > 1 berarti *file* menjadi lebih besar.
5. Percobaan dengan menggunakan *Handphone Android* dan dengan *Emulator* memiliki hasil yang sama dalam hal kompresi dan dekompresi, tetapi waktu yang dibutuhkan *Emulator* lebih lama dari pada menggunakan *Handphone Android*. Dengan spesifikasi *hardware* yang hampir sama (tetapi berbeda sistem operasi) waktu yang digunakan *Handphone Android* lebih cepat daripada *Emulator*.
6. Tingkat keamanan data setelah dikompresi cukup terjaga, dengan kata lain *file* tidak berkurang atau mengalami kerusakan setelah proses kompresi data dilakukan.

5.2 Saran

Untuk pengembangan selanjutnya diharapkan program aplikasi ini disarankan menggunakan algoritma kompresi data lainnya, agar didapatkan metode yang lebih efektif. Penulis juga menyarankan untuk pengembangan selanjutnya agar program aplikasi yang dibuat lebih variatif dan inovatif dari program aplikasi yang telah ada sebelumnya.

DAFTAR PUSTAKA

1. Jubilee Enterprise (2010). *Rahasia Manajemen File*. Jakarta : Elex Media Komputindo.
2. Kadir, Abdul (2003). *Pengenalan Sistem Informasi*. Yogyakarta : Penerbit Andi.
3. Mengyi Pu, Ida (2006). *Fundamental Data Compression*. Butterworth-Heinemann.
4. Michael Siregar, Ivan (2010). *Mengembangkan Aplikasi Enterprise Berbasis Android*. Bandung : Penerbit Gava Media.
5. Michael Siregar, Ivan (2011). : *Membongkar Source Code Berbagai Aplikasi Android*. Bandung : Penerbit Gava Media.
6. Mulyanta, Edi S (2005). Menjadi Desainer Layout Andal dengan *Adobe InDesign CS*. Yogyakarta : Penerbit Andi. Halaman 172 174.
7. Nazruddin Safaat H (2011). *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Informatika
8. Stephen, Agustinus (2007). *File Piracy*. Jakarta : Elex Media Komputindo
9. Wahyu Winarno, Wing, (2010). *Let's android the world panduan menguasai sistem android*. Yogyakarta : Penerbit Paska Media.
10. <http://android.gits.co.id/2010/08/26/arti-istilah-avd/#more-128> diakses tanggal 5 November 2011
11. http://en.wikipedia.org/wiki/Lossless_data_compression diakses tanggal 27 Desember 2011
12. http://id.wikipedia.org/wiki/Android_%28sistem_operasi%29 diakses tanggal 3 November 2011
13. http://id.wikipedia.org/wiki/Eclipse_%28perangkat_lunak%29 diakses tanggal 10 November 2011
14. http://id.wikipedia.org/wiki/Kompresi_data diakses tanggal 2 November 2011
15. <http://michael.dipperstein.com/huffman/> diakses tanggal 27 Desember 2011





**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**


NAMA : DINASTI CAHYO PRASETYO
NIM : 07.12.648
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 5 Desember 2011 s/d 5 Juni 2012
JUDUL : RANCANG BANGUN APLIKASI KOMPRESI DATA PADA
HANDPHONE MENGGUNAKAN ALGORITMA HUFFMAN
BERBASIS ANDROID

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :


Hari : Selasa
Tanggal : 21 Februari 2012
Dengan Nilai : 86,3 (A) *α*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji,



Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Sekretaris Majelis Penguji,


Dr. Eng. Arzuanto S, ST, MT
NIP.Y.1030800417

ANGGOTA PENGUJI

Dosen Penguji I


M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Penguji II


Bima Aulia F, ST



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
 FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
 PROGRAM PASCASARJANA MAGISTER TEKNIK**

T. BNI (PERSERO) MALANG
 BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa:

NAMA : DINASTI CAHYO PRASETYO
 NIM : 07.12.648
 JURUSAN : Teknik Elektro S-1
 KONSENTRASI : Teknik Komputer dan Informatika
 MASA BIMBINGAN : 5 Desember 2011 s/d 5 Juni 2012
 JUDUL : **RANCANG BANGUN APLIKASI KOMPRESI DATA PADA
 HANDPHONE MENGGUNAKAN ALGORITMA HUFFMAN
 BERBASIS ANDROID**

No	Tanggal	Uraian	Paraf
1	Penguji I 21 - 02 - 2012	Tambahkan keterangan pada tabel dan gambar	
		Tambahkan pengujian metode yang digunakan dibandingkan dengan yang lain	
2	Penguji II 21 - 02 - 2012	Tambahkan ratio kompresi pada kesimpulan	 3/5 12
		Tambahkan app delete	
		Tambahkan keterangan pada tabel dan gambar	

Disetujui,

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
 NIP.P.1030100358

Dosen Penguji II

Bima Aulia F, ST

Mengetahui,

Dosen Pembimbing I

Ir. Yustuf Ismail Nakhoda, MT
 NIP.Y.1018800189

Dosen Pembimbing II

Michael Ardita, ST, MT
 NIP.P.1031000434



Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Dinasti Cahyo Prasetyo
NIM : 07.12648
Perbaikan meliputi :

tambahan literatur pd glb dan tabel.
tambahan pengisian metode arda di
bab dua dan sebagainya.

Malang.


M. Ibrahim Achrisman



Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA
NIM
Perbaikan melalui

Dinask Cahya P
07.12.648

*) Tambahkan Rata kompresi
pada kesimpulan

*) Tambahkan App ~~dit~~ delete

*) Tambahkan ref. gambar +
tabel.

Malang, 21-02-2012


(Bima Adika P)



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : DINASTI CAHYO P.
 NIM : 07.12.648
 Semester : VIII
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : TEKNIK ELEKTRONIKA
TEKNIK ENERGI LISTRIK
TEKNIK KOMPUTER DAN INFORMATIKA
TEKNIK KOMPUTER
TEKNIK TELEKOMUNIKASI
 Alamat : Jl. TERUSAN SIGURA - GURA 1B MALANG

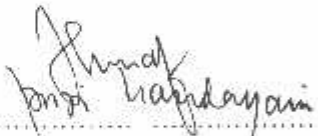
Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat *SKRIPSI Tingkat Sarjana*. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan *SKRIPSI* adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
Recording Teknik Elektro

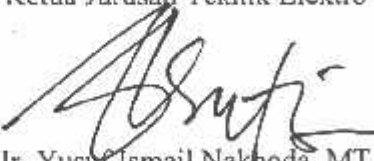

(.....)

Malang, 21 APRIL 201

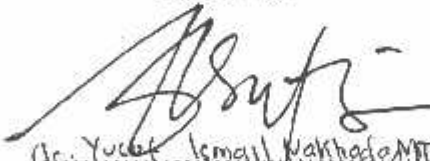
Pemohon


(.....DINASTI.....)

Disetujui
Ketua Jurusan Teknik Elektro


Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Mengetahui
Dosen Wali


(.....Yusuf Ismail Nakhoda MT.....)

Catatan :

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. 506/138 3.69
2.
3. praktikum lengkap



DAFTAR PRESTASI AKADEMIK PRAKTIKUM
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

Nama Mahasiswa	:	DENASTI SYARI P.
NIM	:	0712145
Tempat, Tanggal Lahir	:	LUMAJANG, 2 NOVEMBER 1988
Jenjang	:	Strata 1 (S1)
Fakultas	:	Teknologi Industri
Jurusan / Program Studi	:	Teknik Elektro
Konsentrasi	:	Teknik Komputer dan Informatika

Praktikum Laboratorium	Kode	Nama Praktikum	SKS	Nilai
I	EL-2215 27	Fisika	1	A
		Rangkaian Listrik		B+
		Rangkaian Logika dan Digital		B
		Dasar Komputer dan Pemrograman		B
II	EL-4216 27	Dasar Elektronika	1	B
		Dasar Sistem Telekomunikasi		A-
		Mikrokontroler		A
		Sistem Pengukuran		B
III	EL-5316 26	Dasar Sistem Kendali	1	B+
		Basis Data		B
		Administrasi Jaringan		A
IV	EL-6317 2-11	Sistem Operasi	1	B-
		Pemrograman Internet		B
		Pemrograman Objek		B+
V	EL-7318 23	Rekayasa Perangkat Lunak Sistem Informasi	1	B+
		Peripheral dan Antar Muka		B+
		Pemrosesan Sinyal Digital		B
		Multimedia		B
		Pemrograman Jaringan		A

$$\frac{506}{137} = 3.69$$

$$\frac{532.5}{144} = 3.70$$

Malang, 25-4-2011

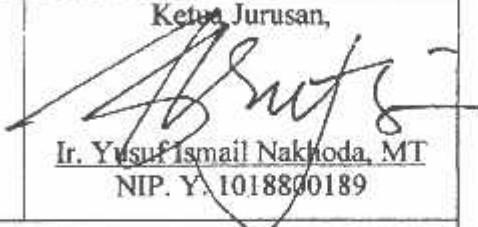
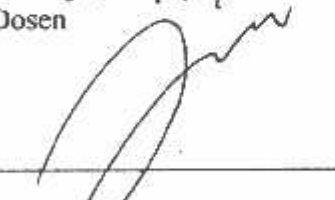
Recording
Jurusan Teknik Elektro S1

Puji Handayani
Puji Handayani



**FORMULIR PENGAJUAN JUDUL SKRIPSI
 JURUSAN TEKNIK ELEKTRO S-1**

Konsentrasi : ~~Teknik Energi Listrik / Teknik Elektronika / Teknik Komputer & Informatika / Teknik Komputer / Teknik Telekomunikasi *~~

1.	Nama Mahasiswa : <u>DINASTI CAHYO PRASETYO</u>	Nim : <u>0712648</u>
2.	Waktu Pengajuan :	Tanggal: _____ Bulan: <u>11</u> Tahun: <u>2011</u>
Spesifikasi Judul (berilah tanda silang)**)		
3.	a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri	c. Elektronika & Komponen f. Elektro Digital & Komputer g. Elektronika Komunikasi h. Lainnya
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen ^{*)} : <u>Dr. Aryanto, ST, MT</u>	Ketua Jurusan,  <u>Ir. Yusuf Ismail Nakhoda, MT</u> NIP. Y. 1018800189
5.	Judul yang diajukan Mahasiswa :	<u>PANCANG BANGUN APLIKASI KOMPRESI DATA PADA HANDPHONE KEUGUNJAYAN ALGORITMA HOFFMAN BERBASIS ANDROID</u>
6.	Perubahan Judul yang Disetujui Dosen materi bidang ilmu
7.	Catatan :
8.	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui, <u>29/11</u> / 2011 Dosen 

Perhatian :

1. Formulir Pengajuan ini harap dikembalikan kepada Jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1.
2. Keterangan : *) Coret yang tidak perlu
**) dilingkari a, b, c,atau g, sesuai bidang Keahlian.

Form.S-2

Lampiran : 1 (satu) berkas

Pembimbing Skripsi

Kepada : Yth. Bapak Ir. Yusuf Ismail Nakhoda, MT
Dosen Institut Teknologi Nasional Malang

Yang bertanda tangan di bawah ini:

Nama : Dinasti Cahyo Prasetyo
NIM : 07.12.648
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama untuk penyusunan Skripsi dengan judul (proposal terlampir):

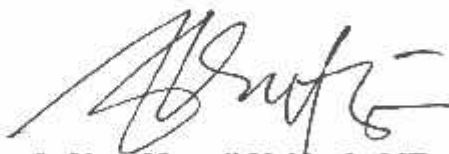
“ Rancang Bangun Aplikasi Kompresi Data Pada Handphone Menggunakan Algoritma Huffman Berbasis Android “

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Jurusan Teknik Elektro


Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 101.880.0189

Malang , 29 November 2011

Hormat kami


Dinasti Cahyo Prasetyo
0712648

Form S-3a

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Dinasti Cahyo Prasetyo
NIM : 07.12.648
Semester : IX (Sembilan)
Jurusan : Teknik Elektro S-I
Konsentrasi : Teknik Komputer dan Informatika


Dengan ini menyatakan bersedia/ ~~tidak bersedia~~ *) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

“ Rancang Bangun Aplikasi Kompresi Data Pada Handphone Menggunakan Algoritma Huffman Berbasis Android “

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 29 November 2011

Kami yang membuat pernyataan


Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 101.880.0189

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.

*) Coret yang tidak perlu

Form S-3b

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Dinasti Cahyo Prasetyo

NIM : 07.12.648

Semester : IX (Sembilan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bersedia/~~tidak bersedia~~*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

“ Rancang Bangun Aplikasi Kompresi Data Pada Handphone Menggunakan Algoritma Huffman Berbasis Android “

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 29 November 2011

Kami yang membuat pernyataan


Michael Ardita, ST, MT
NIP.P. 1031000434

Catatan :

Setelah disetujui agar formulir ini

Diserahkan mahasiswa/i yang bersangkutan

Kepada Jurusan untuk diproses lebih lanjut.


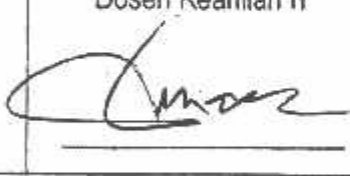
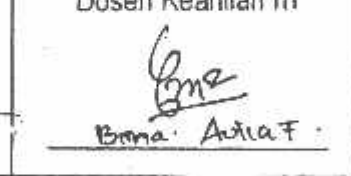

*) Coret yang tidak perlu

Form S-3b



BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/ Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: <u>DINASTI CAHYO PRASETYO</u>		Nim: <u>0712648</u>	
2.	Keterangan	Tanggal	Waktu	Tempat:
	Pelaksanaan	<u>06 DESEMBER 2011</u>		Ruang:
Spesifikasi Judul (berilah tanda silang)**)				
3.	a. Sistem Tenaga Elektrik		e. Elektronika & Komponen	
	b. Energi & Konversi Energi		f. Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran		g. Elektronika Komunikasi	
	d. Sistem Kendali Industri		h. lainnya	
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>RAJANG BAMBUN APLIKASI KOMPRESI DATA PADA HANDPHONE MENGGUNAKAN ALGORITMA HUFFMAN BERBASIS ANDROID.</u>		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian		
6.	Catatan:			
Catatan:				
Persetujuan Judul Skripsi				
7.	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
			 <u>Bona. Aulia F.</u>	
Mengetahui, Ketua Jurusan.		Disetujui, Calon Dosen Pembimbing ybs		
 <u>Ir. Yusuf Ismail Makhoda, MT</u> NIP. Y. 1018800183		Pembimbing I	Pembimbing II	

Perhatian:

1. Keterangan: *) Coret yang tidak perlu

**) dilingkari a, b, c, atau g sesuai bidang keahlian



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 9 Desember 2011

Nomor : ITN- 901/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI
Kepada : Yth. Sdr/I. **IR. YUSUF ISMAIL NAKHODA, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : DINASTI CAHYO P
Nim : 0712648
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

5 Desember 2011 s/d 5 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1,
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan
Teknik Elektro S-1

(Signature)
Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018300189

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 9 Desember 2011

Nomor : ITN- 902/L.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **MICHAEL ARDITA, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : DINASTI CAHYO P
Nim : 0712648
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

5 Desember 2011 s/d 5 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1,
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan
Teknik Elektro S-1

[Signature]
Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a



FORMULIR BIMBINGAN SKRIPSI

Nama : DINASTI CAHYO PRASETYO
Nim : 07.12.648
Masa Bimbingan : 5 Desember 2011 s/d 5 Juni 2012
Judul Skripsi : **RANCANG BANGUN APLIKASI KOMPRESI DATA
PADA HANDPHONE MENGGUNAKAN
ALGORITMA HUFFMAN BERBASIS ANDROID**

No	Tanggal	Uraian	Paraf Pembimbing
1	09 - 01 - 2012	Demo Program	
2	11 - 01 - 2012	Revisi Tampilan Program	
3	18 - 01 - 2012	ACC Program	
4	20 - 01 - 2012	Revisi Bab 1, Bab 2	
5	23 - 01 - 2012	ACC Bab 1, Bab 2	
6	31 - 02 - 2012	Revisi Penulisan Kata-Kata Asing	
7	06 02 - 2012	Revisi Bab3, Bab 4, Bab 5	
8	13 - 02 - 2012	ACC Bab 3, Bab 4, Bab 5	
9			
10			

Malang,

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP. N.1018800189

Form S-4b



FORMULIR BIMBINGAN SKRIPSI

Nama : DINASTI CAHYO PRASETYO
Nim : 07.12.648
Masa Bimbingan : 5 Desember 2011 s/d 5 Juni 2012 *84*
Judul Skripsi : RANCANG BANGUN APLIKASI KOMPRESI
DATA PADA HANDPHONE MENGGUNAKAN
ALGORITMA HUFFMAN BERBASIS ANDROID

No.	Tanggal	Uraian	Paraf Pembimbing
1.	06-01-2012	DEMO PROGRAM	<i>Ar</i>
2.	17-01-2012	ACC PROGRAM	<i>Ar</i>
3.	18-01-2012	REVISI BAB I, BAB II	<i>Ar</i>
4.	20-01-2012	ACC BAB I, BAB II	<i>Ar</i>
5.	25-01-2012	REVISI BAB III	<i>Ar</i>
6.	28-01-2012	ACC BAB III	<i>Ar</i>
7.	30-01-2012	REVISI BAB IV	<i>Ar</i>
8.	03-02-2012	ACC BAB IV	<i>Ar</i>
9.			
10.			

Malang, 8 - Mei - 2012
Dosen Pembimbing II

Ar
Michael Ardita, ST, MT
NIP.P. 103.1000.434

Form.S-4b