

TUGAS AKHIR

PEMBANGUNAN PENGARSIP FILE SEDERHANA



Rio Aurelio Sumantri

NPM: 2017730004

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2024

FINAL PROJECT

BUILDING A SIMPLE FILE ARCHIVER



Rio Aurelio Sumantri

NPM: 2017730004

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2024**

LEMBAR PENGESAHAN

PEMBANGUNAN PENGARSIP FILE SEDERHANA

Rio Aurelio Sumantri

NPM: 2017730004

Bandung, 21 Juni 2024

Menyetujui,

Pembimbing

Digitally signed
by Raymond
Chandra Putra

Raymond Chandra Putra, M.T.

Ketua Tim Penguji

Digitally signed
by Gede Karya

Gede Karya, M.T.

Anggota Tim Penguji

Digitally signed
by Cecilia Esti
Nugraheni

Dr.rer.nat. Cecilia Esti Nugraheni

Mengetahui,

Ketua Program Studi

Digitally signed
by Lionov

Lionov, Ph.D.

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa tugas akhir dengan judul:

PEMBANGUNAN PENGARSIP FILE SEDERHANA

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 21 Juni 2024



Rio Aurelio Sumantri
NPM: 2017730004

ABSTRAK

Pengarsipan *file* adalah proses penyimpanan dan pengelolaan *file* digital dalam suatu struktur tertentu agar dapat diakses, diorganisasi, dan dipertahankan dengan mudah. Pengarsipan *file* merupakan kegiatan penting dalam era digital untuk menjaga integritas, aksesibilitas, dan keberlanjutan informasi. Skripsi ini membahas pembangunan perangkat lunak pengarsip *file* sederhana yang mampu mengarsipkan, mengekstrak, dan mengelola *file* serta direktori dengan efisien. Perangkat lunak ini dirancang untuk mempertahankan struktur direktori dan metadata *file*, yang meliputi deskripsi, identifikasi, dan konteks data. Penggunaan algoritma *hashing* MD5 memastikan integritas *file* dengan mencatat nilai *hash* saat pengarsipan dan membandingkannya saat pengekstrakan.

Fitur utama yang ditawarkan perangkat lunak ini meliputi kemampuan untuk mengarsipkan beberapa direktori dan *file* menjadi satu arsip, menambah *file* atau direktori ke dalam arsip yang sudah ada, mengekstrak direktori atau *file* dari arsip, melakukan pencarian dalam arsip, serta menghapus *file* atau direktori dari arsip. Pengembangan perangkat lunak ini menggunakan bahasa pemrograman Java dan berbasis terminal, menjadikannya ringan dan mudah digunakan.

Pengujian dilakukan untuk mengevaluasi fungsionalitas perangkat lunak, termasuk pengarsipan dan pengekstrakan *file* serta direktori, dan hasilnya menunjukkan perangkat lunak berfungsi dengan baik sesuai dengan spesifikasi yang ditentukan. Hasil analisis dan pengujian mengindikasikan bahwa perangkat lunak ini mampu mengelola *file* dan direktori dengan efisien dan aman.

Dengan demikian, perangkat lunak pengarsip *file* sederhana yang dibangun dalam penelitian ini dapat menjadi solusi praktis dan efektif dalam mengelola arsip digital, serta dapat diimplementasikan dalam berbagai konteks baik untuk keperluan bisnis maupun pendidikan.

Kata-kata kunci: Pengarsipan *file*, *Metadata*, Integritas Data, *Hashing* MD5, Java, Pengelolaan Arsip Digital

ABSTRACT

File archiving is the process of storing and managing digital files in a specific structure to ensure easy access, organization, and preservation. File archiving is an essential activity in the digital era to maintain the integrity, accessibility, and sustainability of information. This thesis discusses the development of a simple file archiving software capable of efficiently archiving, extracting, and managing files and directories. The software is designed to preserve the directory structure and file metadata, including descriptions, identifications, and data context. The use of the MD5 hashing algorithm ensures file integrity by recording the hash value during archiving and comparing it during extraction.

The main features offered by this software include the ability to archive multiple directories and files into a single archive, add files or directories to an existing archive, extract directories or files from the archive, search within the archive, and delete files or directories from the archive. The software development utilizes the Java programming language and is terminal-based, making it lightweight and easy to use.

Testing was conducted to evaluate the functionality of the software, including the archiving and extraction of files and directories, and the results show that the software functions well according to the specified requirements. Analysis and testing results indicate that this software can efficiently and securely manage files and directories.

Thus, the simple file archiving software developed in this study can provide a practical and effective solution for managing digital archives and can be implemented in various contexts for both business and educational purposes.

Keywords: File Archiving, Metadata, Data Integrity, MD5 Hashing, Java, Digital Archive Management

*Dipersembahkan kepada keluarga, Teknik Informatika UNPAR,
teman-teman, dan diri sendiri.*

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus Allah yang Maha Esa atas seluruh berkat yang diberikan kepada penulis sehingga dapat menyelesaikan tugas akhir dengan judul **Pembangunan Pengarsip File Sederhana** dengan baik dan tepat waktu. Penulis juga berterima kasih kepada pihak-pihak yang telah memberikan dukungan dan bantuan kepada penulis dalam menyelesaikan tugas akhir ini. Penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua dan keluarga yang selalu memberikan dukungan dan doa.
2. Bapak Keenan Adiwijaya Leman selaku dosen pembimbing yang dengan sangat sabar membimbing hingga penulis berhasil menyelesaikan tugas akhir ini dengan baik.
3. Seluruh mahasiswa Teknik Informatika UNPAR angkatan 2017 yang telah membantu dalam tugas akhir ini.
4. Seluruh dosen Teknik Informatika UNPAR yang telah memberikan ilmu dari awal kuliah sampai penulis berhasil menyelesaikan tugas akhir ini.
5. Teman-teman 'GRBLG' yang telah setia menemani penulis dalam susah dan senang sejak awal kuliah : Dio Antares, David Christopher Sentosa, Reynard Rafferty Susilo, Reinalta Sugianto, Sterenlie, Fritz Humphrey Silalahi, Leonard Wang, Juan Nandriisa Redemptino, dan Melody Victorian Angel Johany.
6. Teman-teman Teknik Informatika UNPAR angkatan 2017 yang telah berbagi ilmu, memberikan dukungan, dan membantu penulis menyelesaikan tugas akhir ini.
7. Pihak-pihak lain yang telah membantu penulisan tugas akhir ini, yang terus memberikan doa dan semangat kepada penulis.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna karena keterbatasan pengetahuan dan kemampuan penulis. Oleh karena itu, penulis memohon maaf jika terdapat kekurangan pada tugas akhir ini. Penulis juga mengharapkan kritik dan saran yang membangun untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi yang membaca atau membutuhkan.

Bandung, Juni 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR KODE PROGRAM	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 Pengarsipan	5
2.2 File	6
2.3 Folder	6
2.4 <i>Hashing</i>	7
2.4.1 MD5	7
2.5 Bahasa Pemrograman Java	8
2.5.1 <i>Library</i> java.io	8
2.5.2 <i>Library</i> java.security	10
2.5.3 <i>Library</i> java.util	10
2.6 UNIX tar	11
3 ANALISIS	13
3.1 Spesifikasi Kebutuhan Perangkat Lunak	13
3.2 Eksperimen Teknik Baca Tulis <i>File</i>	13
3.2.1 Menulis <i>File</i>	13
3.2.2 Membaca <i>File</i>	14
3.3 Rancangan Struktur Data	15
3.3.1 Struktur Data Linear	15
3.3.2 Struktur Data Menggunakan <i>header</i> dan <i>body</i>	16
3.4 Rancangan Algoritma	17
3.4.1 Mengarsip <i>File</i> /Folder	17
3.4.2 Mengarsip <i>File</i> /Folder pada Arsip yang Sudah Ada	17
3.4.3 Mengekstrak Sebuah Arsip	17
3.4.4 Mengekstrak Sebuah <i>File</i> dalam Arsip	18
3.4.5 Mengekstrak Sebuah Folder dalam Arsip	18
3.4.6 Menghapus <i>File</i> /Folder dalam Arsip	18

3.4.7	Melihat Isi Arsip	18
3.5	Analisis Use Case Diagram dan Use Case Scenario	19
4	PERANCANGAN	23
4.1	Perancangan Proses Aplikasi Pengarsipan	23
4.2	Perancangan Kelas	28
4.3	Diagram <i>Sequence</i> Perangkat Lunak	29
5	IMPLEMENTASI DAN PENGUJIAN	39
5.1	Implementasi	39
5.1.1	Lingkungan Implementasi	39
5.1.2	Hasil Implementasi	39
5.2	Pengujian	48
5.2.1	Pengujian Fungsional	48
6	KESIMPULAN DAN SARAN	59
6.1	Kesimpulan	59
6.2	Saran	59
	DAFTAR REFERENSI	61
	A KODE PROGRAM	63

DAFTAR GAMBAR

3.1	Hasil Tulis File	14
3.2	Hasil Baca File	15
3.3	Diagram Struktur Data Linear	15
3.4	Diagram Struktur Data Header and Body 1	16
3.5	Diagram Struktur Data Header and Body 2	16
3.6	Use Case Diagram	19
4.1	Flowchart untuk mengarsip	25
4.2	Flowchart untuk mengekstrak/menghapus	26
4.3	Flowchart untuk melihat isi arsip	27
4.4	Diagram kelas Arsip	28
4.5	Diagram Sequence	30
4.6	Diagram Sequence Mengarsip File	31
4.7	Diagram Sequence Mengekstrak File	32
4.8	Diagram Sequence Melihat Isi Arsip	33
4.9	Diagram Sequence Mengekstrak File Spesifik dari Arsip	34
4.10	Diagram Sequence Mengekstrak Folder Spesifik dari Arsip	35
4.11	Diagram Sequence Menghapus File/Folder pada Arsip	36
4.12	Diagram Sequence Memperbarui Arsip	37
5.1	Pengujian Mengarsip Bagian 1	49
5.2	Pengujian Mengarsip Bagian 2	49
5.3	Pengujian Mengarsip Bagian 3	49
5.4	Pengujian Mengarsip File pada Arsip yang Sudah Ada Bagian 1	50
5.5	Pengujian Mengarsip Folder pada Arsip yang Sudah Ada Bagian 1	50
5.6	Pengujian Mengekstrak Bagian 1	51
5.7	Folder Extracted Setelah Ekstraksi 1	52
5.8	Pengujian Mengekstrak Sebuah File dari Sebuah Arsip Bagian 1	52
5.9	Folder Extracted Setelah Ekstraksi 2	52
5.10	Pengujian Mengekstrak Sebuah Folder dari Sebuah Arsip Bagian 1	53
5.11	Folder Extracted Setelah Ekstraksi 3	53
5.12	Pengujian Menghapus Sebuah File dari Sebuah Arsip Bagian 1	54
5.13	Pengujian Menghapus Sebuah Folder dari Sebuah Arsip Bagian 1	54
5.14	Pengujian Melihat Isi Arsip Bagian 1	55
5.15	Pengujian Melihat Isi Arsip Bagian 2	55
5.16	Pengujian Melihat Isi Arsip Bagian 3	56
5.17	Pengujian Melihat Isi Arsip Bagian 4	56
5.18	Pengujian Mengekstrak Seluruh Isi dari Sebuah Arsip dengan Salah Satu Jumlah Karakter Alamat File yang Tidak Sesuai	56
5.19	Pengujian Mengekstrak Seluruh Isi dari Sebuah Arsip dengan Salah Satu Konten File yang Tidak Sesuai	57

DAFTAR KODE PROGRAM

2.1	Contoh BufferedInputStream dan BufferedOutputStream pada Java	8
2.2	Contoh BufferedReader pada Java	9
2.3	Contoh DataInputStream and DataOutputStream pada Java	9
2.4	Contoh File and FileInputStream/FileOutputStream pada Java	9
2.5	Contoh MessageDigest pada Java	10
2.6	Contoh ArrayList pada Java	10
2.7	Contoh Scanner pada Java	11
3.1	Potongan kode untuk membuat dan menulis pada <i>file</i> menggunakan <i>library</i> Java .	14
3.2	Potongan kode untuk menulis bilangan bulat pada <i>file</i>	14
3.3	Potongan kode untuk membaca bilangan bulat pada <i>file</i>	14
5.1	Modul Menu Utama	39
5.2	Modul Mengarsip <i>File</i> /Folder	41
5.3	Modul Mengarsip <i>File</i> /Folder pada Arsip yang Sudah Ada	42
5.4	Modul Mencatat Nama <i>file</i> /Folder pada Sumber Dirketori	43
5.5	Modul Mengekstrak Sebuah Arsip	43
5.6	Modul Mengekstrak Sebuah <i>File</i> dalam Arsip	44
5.7	Mengekstrak Sebuah Folder dalam Arsip	45
5.8	Menghapus <i>File</i> /Folder dalam Arsip	46
5.9	Modul Melihat Isi Arsip	47
5.10	Modul Menghitung <i>Checksum</i> Sebuah <i>File</i>	48
A.1	Kode Eksperimen Baca Tulis File	63
A.2	Kode Program Aplikasi Pengarsip	64

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengarsipan *file* merupakan suatu kegiatan yang umum dilakukan pada era digital yang penuh dengan informasi dan data. Dalam berbagai konteks, baik itu di lingkungan bisnis atau pendidikan, pengarsipan *file* merupakan hal penting dalam menjaga integritas, aksesibilitas, dan keberlanjutan informasi dan data. Pengarsipan *file* sederhana adalah salah satu pendekatan yang dapat digunakan untuk mengatur dan menjaga berkas dan dokumen dengan lebih baik. Berikut adalah beberapa alasan mengapa pembangunan pengarsip *file* sederhana cukup penting: Agar praktis untuk dipindahkan, untuk keperluan pembuatan *backup*, dan agar mudah diorganisasi.

Pada sebuah *file*, terdapat metadata. Metadata adalah "data yang memberikan informasi tentang data lainnya", namun bukan tentang konten data itu sendiri, seperti teks pesan atau gambar itu sendiri. Metadata dapat berupa deskripsi, identifikasi, dan konteks suatu data, yang membantu dalam manajemen, penemuan, dan penggunaan data. Metadata dapat memberikan informasi tentang asal usul data, siapa yang membuatnya, tujuan pembuatan data, hak akses, kebijakan keamanan, dan lain sebagainya. Dengan pentingnya metadata, maka pengarsipan *file* sederhana perlu mempertahankan metadata dan agar tidak berubah saat dilakukan pengarsipan dan pengestrakan.

Dengan memahami pentingnya pengarsipan *file* sederhana, maka pembangunan sistem pengarsipan yang efisien dan mudah digunakan akan menjadi solusi yang relevan. Berikut adalah fitur-fitur yang akan dibangun pada sistem pengarsipan *file* sederhana.

- Dapat mengarsipkan beberapa direktori dan *file* menjadi sebuah *file* arsip serta mempertahankan struktur direktori.
- Dapat menambah *file* atau direktori ke *file* arsip yang telah ada sebelumnya.
- Dapat mengekstrak direktori atau *file* dari *file* arsip.
- Dapat melakukan pencarian *file* dan direktori yang berada di dalam *file* arsip.
- Dapat menghapus *file* atau direktori dari *file* arsip.
- Dapat mempertahankan metadata dan struktur folder agar tidak berubah saat dilakukan pengarsipan dan pengestrakan.
- Dapat menjaga integritas *file* arsip.

Untuk menjaga integritas *file* arsip, setiap *file* yang akan diarsip akan dicatat nilai *hash* yang dihasilkan oleh algoritma *hashing* MD5. Lalu saat sebuah *file* diekstrak, akan dibandingkan nilai *hash*-nya dengan nilai *hash* saat *file* diarsip pertama kali. Jika perbandingan nilai *hash* file tersebut sama, maka integritas *file* tersebut terjaga.

Pada tugas akhir ini, akan dibangun sebuah perangkat lunak sederhana yang dapat mengarsipkan *file* dan direktori. Dengan menggunakan perangkat lunak tersebut, beberapa *file* dan direktori dapat dijadikan satu arsip. Perangkat lunak akan berbasis *terminal* dan dibangun dengan bahasa pemrograman Java. Setelah perangkat lunak dibangun, akan dilakukan pengujian terhadap perangkat lunak tersebut dan hasil dari pengujian ini akan dinalisis sehingga dapat ditarik kesimpulan dari analisis tersebut.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan teknik baca tulis *file* menggunakan Java API?
2. Bagaimana cara merancang struktur data untuk mengarsipkan *file*?
3. Bagaimana cara kerja teknik *hashing* untuk menjaga integritas *file*?
4. Bagaimana cara membangun perangkat lunak untuk pengarsip *file* sederhana?

1.3 Tujuan

Tujuan dari penulisan tugas akhir ini adalah sebagai berikut:

1. Mengimplementasikan teknik baca tulis *file* menggunakan Java API.
2. Mengetahui cara merancang struktur data untuk mengarsipkan *file*.
3. Mengetahui cara kerja teknik *hashing* untuk menjaga integritas *file*.
4. Membangun perangkat lunak untuk pengarsip *file* sederhana.

1.4 Batasan Masalah

Batasan yang dibuat terkait dengan pengerjaan tugas akhir ini adalah *file* arsip yang dapat dijalankan oleh perangkat lunak hanya *file* arsip yang dihasilkan oleh perangkat lunak ini sebelumnya.

1.5 Metodologi

Metodologi yang dilakukan pada tugas akhir ini adalah sebagai berikut:

1. Melakukan studi literatur mengenai struktur direktori dan *file* menggunakan Java API.
2. Melakukan studi literatur mengenai teknik baca tulis *file* menggunakan Java API.
3. Melakukan studi literatur mengenai struktur data untuk mengarsipkan *file*.
4. Melakukan studi literatur mengenai teknik *hashing* MD5 untuk menjaga integritas *file*.
5. Merancang dan membangun perangkat lunak berbasis *terminal*.
6. Melakukan pengujian terhadap perangkat lunak.
7. Menganalisis hasil pengujian perangkat lunak.
8. Mengambil kesimpulan dari hasil analisis.
9. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematika penulisan setiap bab tugas akhir ini adalah sebagai berikut :

1. Bab 1 Pendahuluan
Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan yang digunakan untuk menyusun tugas akhir ini.
2. Bab 2 Dasar Teori
Bab ini berisi teori-teori yang digunakan dalam pembuatan tugas akhir ini. Teori yang digunakan yaitu pengarsipan, *file*, folder, teknik *hashing* MD5, bahasa pemrograman Java.
3. Bab 3 Analisis Masalah
Bab ini berisi analisis yang digunakan pada tugas akhir ini, analisa kebutuhan perangkat lunak, eksperimen teknik baca tulis *file*, analisis struktur data, analisis algoritma dan analisis diagram *use case* dan skenario.
4. Bab 4 Perancangan
Bab ini berisi perancangan perangkat lunak yang akan dibuat, meliputi perancangan proses perangkat lunak, diagram kelas dan diagram *sequence*.

5. Bab 5 Implementasi dan Pengujian

Bab ini berisi implementasi dan pengujian program, meliputi lingkungan implementasi, hasil implementasi, pengujian fungsional.

6. Bab 6 Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pembangunan program beserta saran untuk pengembangan selanjutnya.