

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi telah berkembang dengan pesat, terutama dalam pengolahan data. Semakin banyak data yang tersedia, semakin besar pula kebutuhan akan kapasitas penyimpanan (Arizki et al., 2024). Hal ini juga berlaku di dunia pendidikan. Pendidikan merupakan alat strategis untuk meningkatkan kualitas suatu bangsa, sehingga kemajuan suatu negara seringkali diukur dari perkembangan sektor pendidikannya. Indonesia berupaya meningkatkan kualitas pendidikan melalui tiga jalur utama, yaitu pendidikan formal, nonformal, dan informal. Di tingkat menengah, salah satu bentuk pendidikan formal di Indonesia adalah Sekolah Menengah Kejuruan (SMK) (Saputra et al., 2023).

Sekolah Menengah Kejuruan(SMK) berfungsi sebagai subsistem pendidikan yang secara nyata mendukung persiapan siswa untuk memasuki dunia kerja. Bursa Kerja Khusus(BKK) menjadi komponen utama dalam mengukur keberhasilan pendidikan di SMK, karena perannya dalam mengoptimalkan distribusi lulusan SMK dan sebagai sumber informasi bagi pencari kerja. Melalui BKK, penyebaran lulusan SMK dapat dioptimalkan, serta menjadi tempat bagi para pencari kerja untuk memperoleh informasi. Pemanfaatan internet dalam pelaksanaan BKK di SMK merupakan metode terbaik untuk memperkenalkan lulusan SMK sekaligus menyediakan informasi yang dibutuhkan oleh pencari kerja(Saputra et al., 2023).

Berdasarkan hasil wawancara di SMK Negeri 3 Kuningan dengan bapak Adi selaku koordinator BKK dan bapak Idi selaku staf, proses pendaftaran peserta BKK menghadapi sejumlah tantangan. Salah satunya adalah volume berkas pendaftaran yang semakin meningkat setiap tahunnya, yang disimpan

dalam penyimpanan internal komputer di BKK SMK Negeri 3 Kuningan. Berkas-berkas ini berukuran besar, sehingga membutuhkan ruang penyimpanan yang besar. Selain itu, dalam sistem pendaftaran yang ada saat ini, peserta sering kali harus melakukan kompresi berkas secara manual sebelum mengunggahnya, yang memakan waktu dan bisa kebingungan, terutama bagi pengguna yang kurang terbiasa dengan teknologi. Keterbatasan ruang penyimpanan dan proses manual tersebut dapat memperlambat kinerja sistem BKK, sehingga memunculkan kebutuhan akan solusi yang lebih baik. Salah satu solusinya yaitu implementasi algoritma kompresi untuk mengurangi ukuran berkas tanpa mengorbankan kualitas dokumen pdf.

Kompresi data adalah proses mengubah aliran data masukan (input data stream) menjadi aliran data lain (output atau bitstream) dengan ukuran yang lebih kecil. Data stream ini bisa berupa file atau buffer di memori (Hasibuan et al., 2023). Berdasarkan paparan permasalahan tersebut, penelitian ini berfokus pada penerapan algoritma Lempel Ziv Welch (LZW) dalam sistem pengelolaan berkas pendaftaran BKK di SMK Negeri 3 Kuningan. Algoritma LZW merupakan teknik kompresi berbasis pembaruan kamus, yang menyimpan elemen-elemen data dalam bentuk kode dan menggunakan kode tersebut saat string yang sama muncul kembali. Ketika kamus penuh, setiap elemen di dalamnya akan dihapus agar bisa menambahkan entri baru. Berbeda dengan metode konvensional, teknik ini tidak memperhatikan frekuensi penggunaan string, tetapi menghapus semua entri di kamus (Riskiono et al., 2020).

Berdasarkan permasalahan di atas, penulis akan melakukan penelitian dengan judul "Implementasi Algoritma Lempel Ziv Welch Pada Sistem Bursa Kerja Khusus (BKK) Untuk Kompresi Berkas Pendaftaran". Penelitian ini diharapkan dapat membantu SMK Negeri 3 Kuningan dalam mengelola proses pendaftaran kerja lulusannya, sehingga pengelolaan dan penyimpanan berkas dapat dilakukan dengan terstruktur dan memiliki ruang penyimpanan yang besar.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah di paparkan di atas dapat disimpulkan bahwa ditemukannya permasalahan, yaitu :

1. Jumlah berkas pendaftaran terus bertambah setiap tahun yang berformat PDF berukuran besar, sehingga sulit dikelola.
2. Berkas yang besar memerlukan banyak ruang penyimpanan, yang dapat menyebabkan penyimpanan di komputer cepat penuh dan sulit menampung berkas tambahan.
3. Peserta harus mengompres berkas secara manual sebelum mengunggahnya, hal ini memperlambat proses pengunggahan dokumen pdf.

1.3 Rumusan Masalah

Berdasarkan masalah yang telah diuraikan pada latar belakang diatas. maka dapat di ambil sebuah rumusan masalah, yaitu sebagai berikut:

1. Bagaimana membangun aplikasi pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan?
2. Bagaimana cara mengimplementasikan algoritma *Lempel Ziv Welch (LZW)* sebagai kompresi file dalam aplikasi pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan?

1.4 Batasan Masalah

Dalam pembahasan dan permasalahan yang muncul perlu adanya pembatasan masalah atau ruang lingkup kajian sehingga penyajian lebih terarah dan saling berkaitan satu dengan yang lainnya. Adapun Batasan dari permasalahan berikut:

1. Aplikasi pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) ini ditujukan untuk mengelola pendaftaran pelamar kerja, khususnya di SMK Negeri 3 Kuningan.

2. Aplikasi BKK yang di bangun berbasis *website*.
3. Algoritma yang digunakan pada proses kompresi adalah algoritma *Lempel Ziv Welch(LZW)*.
4. Tipe file yang akan di kompresi berekstensi PDF.
5. Ukuran file yang dapat dikompresi oleh sistem ini dibatasi pada ukuran minimal 50 KB dan maksimal 10 MB.
6. Besaran file yang menjadi fokus dalam pengujian sebesar 2 sampai 7 MB.
7. Yang berhak mengakses sistem ini adalah:
 - Peserta
 - Staf admin

1.5 Pertanyaan Penelitian

Berdasarkan uraian dari rumusan masalah, tujuan serta manfaat diatas, maka pertanyaan dari penelitian ini diantaranya:

1. Apakah aplikasi pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan dapat dibangun untuk mendukung kebutuhan sekolah?
2. Apakah algoritma *Lempel Ziv Welch (LZW)* dapat diimplementasikan sebagai kompresi file dalam aplikasi pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan?

1.6 Tujuan Penelitian

Berdasarkan perumusan masalah yang telah di uraikan diatas, maka di dapat tujuan sebagai berikut:

1. Membangun Aplikasi Pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan.

2. Mengimplementasikan algoritma *Lempel Ziv Welch (LZW)* pada Aplikasi Pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan untuk proses kompresi.

1.7 Manfaat Penelitian

Manfaat dari perancangan sistem pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan ini diantaranya yaitu:

- a. Manfaat *Teoritis*

Mengimplementasikan Algoritma *Lempel Ziv Welch* pada proses kompresi yang ada pada seleksi Pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan.

- b. Manfaat *Praktis*

1. Bagi Penulis

Menambah pengetahuan mengenai algoritma *Lempel Ziv Welch* dalam penggunaannya pada aplikasi pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan. Peneliti juga dalam hal ini dapat mengimplementasikan ilmu pengetahuan yang dimiliki dan di dapat dari bangku perkuliahan.

2. Bagi Staf Admin

- a) Memudahkan staf admin dalam mengelola data peserta pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan yang sudah lolos seleksi.
- b) Memudahkan staf dalam melakukan pengelolaan data pendaftar kerja.
- c) Mengurangi penyimpanan yang banyak diperlukan.

3. Bagi peserta

- a) Memudahkan untuk melakukan peserta pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) dimanapun dan kapanpun.
- b) Memudahkan dalam memperoleh informasi terkait pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK).

- c) Memperoleh transparansi hasil peserta pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK).

1.8 Hipotesis Penelitian

Penerapan algoritma *Lempel Ziv Welch (LZW)* untuk kompresi file pada aplikasi pendaftaran Bursa Kerja Khusus (BKK) di SMK Negeri 3 Kuningan akan secara signifikan meningkatkan ketersediaan ruang penyimpanan dan pengelolaan berkas. Algoritma ini diharapkan mampu mengurangi ukuran file tanpa mengurangi kualitas dokumen pdf, sehingga meminimalkan kebutuhan ruang penyimpanan yang tersedia di BKK. Dengan penerapan kompresi *LZW*, proses pendaftaran dan pengelolaan data di BKK akan menjadi lebih mudah, memberikan manfaat yang nyata bagi pengguna dan mempermudah administrasi pendaftaran penyaluran kerja di SMK Negeri 3 Kuningan.

1.9 Metodologi Penelitian

Metodologi penelitian adalah proses atau metode ilmiah untuk mengumpulkan data yang akan digunakan untuk tujuan penelitian. *Metodologi* juga merupakan analisis *teoritis* dari suatu cara atau metode. Penelitian merupakan penyelidikan yang sistematis untuk meningkatkan pengetahuan, juga merupakan usaha yang *sistematis* dan terorganisir untuk menyelidiki masalah-masalah tertentu yang memerlukan jawaban.

1.9.1 Metode Pengumpulan Data

Pada metode ini membahas mengenai cara memperoleh data yang dibutuhkan untuk melakukan penelitian, terdapat beberapa metode yang digunakan yaitu:

1. Observasi

Metode Observasi merupakan teknik pengumpulan data yang dilakukan melalui suatu pengamatan, dengan disertai pencatatan-pencatatan terhadap keadaan atau perilaku objek

sasaran(Hasibuan et al., 2023). Adapun teknik observasi yang dilakukan oleh peneliti yaitu pengumpulan data dengan cara pengamatan dan pencatatan yang ada di Bursa Kerja Khusus(BKK) di SMK Negeri 3 Kuningan untuk mengambil objek yang diteliti dan mencatat setiap keadaan yang diamati.

2. Wawancara

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara pengumpul data maupun peneliti terhadap nara sumber atau sumber data(Riskiono et al., 2020). Pada metode wawancara ini, peneliti melakukan tanya jawab dengan bapak Adi Dwifera, S.Pd selaku koordinator BKK dan bapak Idi Walidi selaku staf IT BKK mengenai proses pendaftaran penyaluran kerja di Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan, kesulitan yang saat ini di hadapi dan harapan yang di ingin kan Bursa Kerja Khusus (BKK) SMK Negeri 3 Kuningan mengenai proses pendaftaran penyaluran kerja di Bursa Kerja Khusus(BKK) SMK Negeri 3 Kuningan. Hasil wawancara ini digunakan untuk penyusunan penelitian.

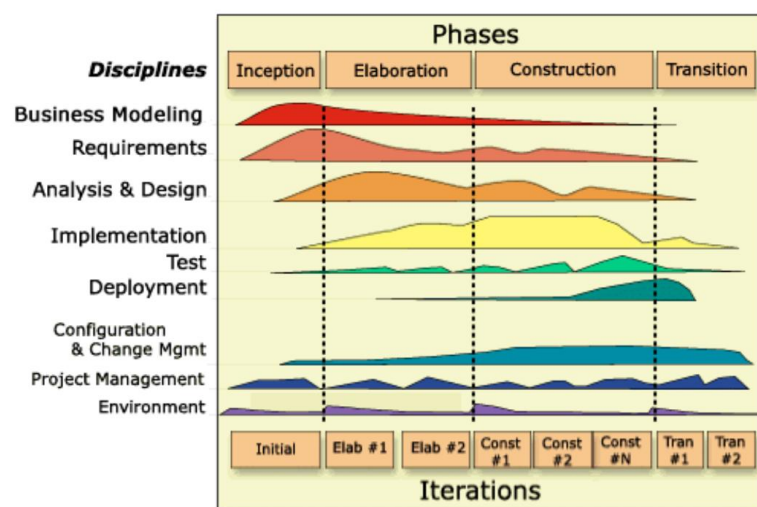
3. Studi Pustaka

Studi pustaka adalah penelitian yang mengkombinasikan berbagai literatur sehingga tersusunlah konsep pemikiran secara komprehensif berdasarkan hasil pemikiran - pemikiran terdahulu yang sudah ada(Faiz et al., 2022). Studi pustaka ini dilakukan dengan menggunakan sumber-sumber seperti jurnal. Studi pustaka ini berguna untuk mengetahui landasan teori pengetahuan dan informasi pada penelitian ini, seperti jurnal referensi yang sesuai dengan judul penelitian yang diambil, jurnal algoritma Lempel Ziv Welch(LZW) dan jurnal yang berhubungan dengan sistem yang akan dibuat. Penulis berharap melalui studi pustaka

ini mendapatkan banyak informasi yang bersagkutan dengan penelitian ini.

1.9.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang akan digunakan pada penelitian ini yaitu metode *Rational Unified Process (RUP)*. Metode *RUP* merupakan metode pembangunan perangkat lunak yang *iterative* dan *incremental* serta berfokus pada arsitektur. Metode *RUP* dapat menangani risiko yang berhubungan dengan pengembangan kebutuhan sistem berdasarkan perubahan yang diinginkan oleh klien. Untuk mengurangi risiko tersebut dilakukan dengan pengujian pada setiap akhir tahapan *RUP*, sehingga akan mudah melakukan perubahan sebelum mencapai tahap akhir. Metode *RUP* juga mengutamakan kepuasan pengguna sehingga lebih sering melakukan interaksi dengan pengguna. *RUP* memiliki empat tahap yaitu *inception*, *elaboration*, *contruction*, dan *transition* seperti pada gambar 1.



Gambar 1. Fase RUP (Perwitasari et al., 2020)

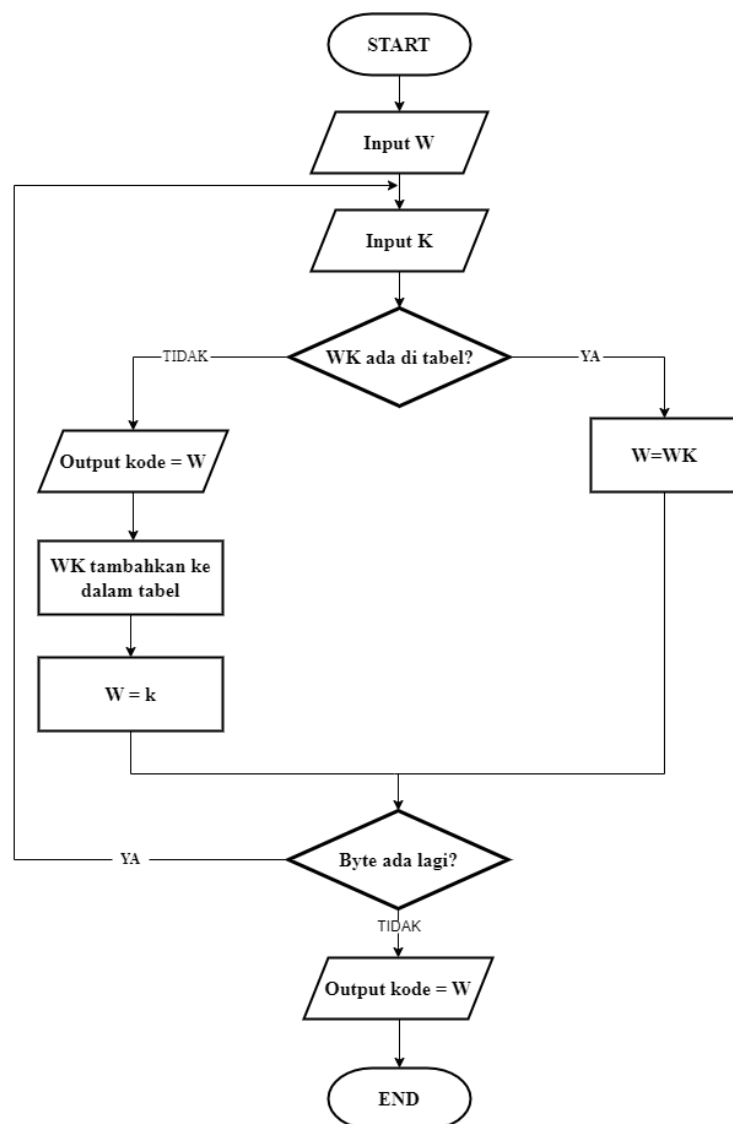
Fase-fase dalam *metodologi RUP* terdiri dari:

1. Fase *Inception* (Permulaan) Fase ini merupakan permulaan tahap pengembangan yaitu mendefinisikan batasan kegiatan, melakukan analisis kebutuhan pengguna dengan mengumpulkan informasi yang akan diolah serta dianalisa untuk memenuhi kebutuhan sistem dengan cara observasi, wawancara dan studi pustaka.
2. Fase *Elaboration* (Perencanaan) Pada fase ini dilakukan perancangan sistem. Tahap ini menggunakan *flowchart* untuk menyatakan aliran data atau proses. Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilsan *prototipe* versi *Beta* dari perangkat lunak.
3. Fase *Construction* (Konstruksi) Pada fase ini dilakukan pembangunan sistem sesuai dengan perancangan yang telah dibuat dengan menggunakan bahasa pemrograman yang telah tentukan. Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.
4. Fase *Transition* (Transisi) Pada fase ini dilakukan proses uji coba dan memperoleh umpan balik dari pengguna yang dapat dimanfaatkan menjadi perbaikan perangkat lunak(Perwitasari et al., 2020).

1.9.3 Metode Penyelesaian Masalah

Metode penyelesaian masalah yang akan digunakan dalam penelitian ini yaitu menggunakan algoritma *Lempel Ziv Welch* (LZW). Algoritma LZW menggunakan teknik *adaptif* dan berbasis “kamus”. Prosedur pemecahan ini melakukan kompresi *menggunakan dictionary*, dimana *fragmen-fragmen* teks diganti

menggunakan *indeks* yang diperoleh oleh sebuah “kamus”. Prinsip *homogen* juga digunakan 22 pada kode *Braille*, dimana kode-kode khusus dipergunakan untuk merepresentasikan istilah-kata yg ada. Pendekatan ini bersifat *adaptif* dan *efektif* karena banyak karakter dapat dikodekan dengan mengacu pada string yang sudah timbul sebelumnya dalam teks. Prinsip kompresi tercapai apabila disurat keterangan simpan pada jumlah *bit* yang lebih sedikit (Arizki et al., 2024).



Gambar 2. Flowchart Kompresi LZW (Oktaviani et al., n.d.)

Urutan langkah prosedur pemecahan kompresi *LZW* artinya sebagai berikut:

- a. *Dictionary* diinisialisasikan pada jumlah *bit* yang lebih sedikit dibandingkan *string* aslinya.
- b. *W* artinya karakter pertama dalam *stream* karakter.
- c. *K* adalah karakter berikutnya dalam *stream* karakter.
- d. Apakah *string* (*W* + *K*) ada pada *dictionary* ?
 - 1) Bila “ya” maka $W = W + K$ (gabungkan *W* serta *K* menjadi *string* baru)
 - 2) Jika “tidak” maka: *output* sebuah kode buat menggantikan *string* *W*, lalu masukkan *string* (*W* + *K*) ke pada *dictionary* serta berikan angka/kode berikutnya yg belum diganti dalam *dictionary* buat *string* tersebut. Dengan ($W = K$)
- e. Apakah masih terdapat karakter berikutnya pada *stream* karakter? Bila “ya” maka kembali ke langkah dua, dan Jika “tidak” maka *output* kode yg menggantikan *string* *W*, lalu proses (stop)(Arizki et al., 2024).

Sebagai contoh, *string* “ABBABABAC” akan dikompresi dengan *LZW*. Isi *dictionary* pada awal proses diatur dengan tiga karakter dasar yang ada: “A”, “B”, “C”, *bit dictionary* nya yaitu 8 *bit*. Tahapan proses kompresi ditunjukkan pada Tabel 1.

Tabel 1. Proses Kompresi(Yuliastuti et al., 2023)

Langkah	Input	String Gabungan	Kamus Baru (Keluaran)	Output
1	A	AB	4 = AB	1
2	B	BA	5 = BA	2
3	B	AB	-	2

4	AB	ABA	6 = ABA	4
5	B	BA	-	2
6	BA	BAC	7 = BAC	5
7	C	-	-	3

$$\begin{aligned}
 \text{Total awal bit disimpan tanpa kompresi} &= \text{Total input} * \text{Total input bit dictionary} \\
 &= 9 \times 8 \text{ bit} \\
 &= 72 \text{ bit}
 \end{aligned}$$

$$\begin{aligned}
 \text{Besar file setelah dikompresi} &= \text{Total output} * \text{bit dictionary} \\
 &= 7 \times 8 \text{ bit} \\
 &= 56 \text{ bit}
 \end{aligned}$$

Hasil Kompresi dari *string* ABBABABAC yaitu :

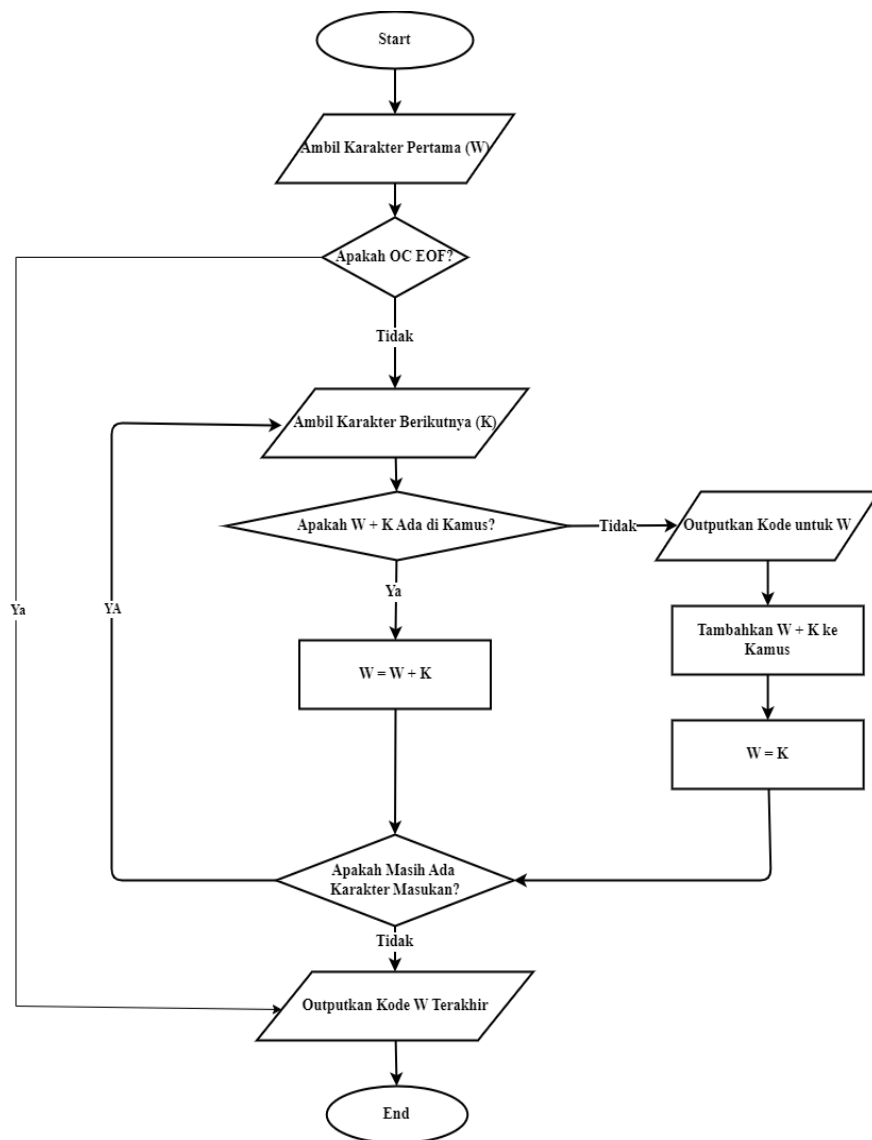
Decimal : 65 66 66 65 66 65 66 65 67
 Heksa : 41 42 42 41 42 41 42 41 43
 Biner : 01000001 01000010 01000010 01000001
 01000010 01000001 01000010 01000001
 01000011

Hasil Kompresi

$$\begin{aligned}
 \text{Rasio} &= \frac{\text{Ukuran file terkompres} \times 100\%}{\text{Ukuran file asli}} \\
 &= \frac{56 \times 100\%}{72} \\
 &= 77,78 \% (\text{Yuliastuti et al., 2023}) \\
 \text{Space Saving} &= 1 - \frac{\text{Ukuran file terkompres} \times 100\%}{\text{Ukuran file asli}} \\
 &= 1 - \frac{56 \times 100\%}{72}
 \end{aligned}$$

= 23 %(Fauzan et al., 2022)

Kolom posisi menyatakan posisi sekarang dari *stream* karakter dan kolom karakter menyatakan karakter yang terdapat pada posisi tersebut. Kolom *dictionary* menyatakan *string* baru yang sudah ditambahkan ke dalam *dictionary* dan nomor *indeks* untuk *string* tersebut ditulis dalam kurung siku. Kolom *output* menyatakan kode *output* yang dihasilkan oleh langkah kompresi(Yuliastuti et al., 2023).



Gambar 3. Flowchart Dekompresi LZW(Yuliastuti et al., 2023)

Prinsip umum kerja Algoritma *LZW* yakni memeriksa setiap karakter yang muncul kemudian menggabungkan dengan karakter selanjutnya menjadi sebuah *string*. Jika *string* baru tersebut tidak berada dalam *dictionary* atau belum dimasukkan ke dalam *indeks*, maka *string* baru tersebut akan dimasukkan ke dalam *indeks* pada *dictionary*. Adapun alur dekompresi dari Algoritma *LZW* sebagai berikut:

- a. Proses dekompresi dimulai.
- b. Algoritma mengambil karakter pertama dari input dan menetapkan ke variabel *W*. Langkah ini menginisialisasi proses dekompresi dengan karakter input pertama.
- c. Algoritma memeriksa apakah telah mencapai *End of File (EOF)* pada data input. Jika sudah mencapai *EOF*, proses berakhir. Jika belum, algoritma dilanjutkan.
- d. Algoritma mengambil karakter berikutnya dari *stream input* dan menetapkan ke variabel *K*. Ini berarti algoritma terus membaca data *input*.
- e. Algoritma memeriksa apakah kombinasi $W + K$ sudah ada di kamus (atau tabel yang berisi urutan karakter yang sudah pernah muncul). Kamus ini menyimpan urutan karakter, dan algoritma berusaha melihat apakah urutan baru ini sudah pernah ditemukan sebelumnya.
- f. Jika Ya, artinya kombinasi $W + K$ sudah ada di kamus, algoritma memperbarui *W* menjadi $W + K$, membentuk urutan yang lebih panjang untuk dicari lebih lanjut.
- g. Jika Tidak, algoritma mengeluarkan kode untuk *W*, yaitu kode kompresi untuk urutan karakter *W*. Setelah itu, algoritma menambahkan $W + K$ ke kamus dan mengatur ulang $W = K$.

- h. Setelah memproses W dan K, algoritma memeriksa apakah masih ada karakter input yang tersisa untuk diproses. Jika masih ada, langkah-langkah pencarian di kamus dan pengeluaran kode untuk W diulangi.
- i. Setelah semua karakter diproses, algoritma mengeluarkan kode akhir untuk nilai terakhir dari W.
- j. Proses dekompresi berakhir.

Sebagai contoh, string “ABBABABAC” akan didekompresi dengan *LZW*. Tahapan proses dekompresi ditunjukkan pada Tabel 2.

Tabel 2. Proses Dekompresi(Yulianti et al., 2023)

Langkah	Input (Kode)	String Gabungan	Kamus Baru (Keluaran)	Output
1	1	A	-	A
2	2	AB	4 = AB	B
3	2	B	-	B
4	4	ABA	5 = BA	AB
5	2	B	-	B
6	5	BAB	6 = ABA	BA
7	3	BAC	-	C

Hasil dekompresi dari kode 65 66 66 65 66 65 66 65 67 yaitu ABBABABAC(Yulianti et al., 2023).

- **Metode Pengujian waktu dan kecepatan kompresi**

Salah satu tantangan utama dalam manajemen data adalah penyimpanan dan kecepatan pemrosesan. Pada sistem yang menangani *volume* data besar, kecepatan kompresi dan dekompresi menjadi metrik penting dalam mengevaluasi kinerja algoritma

kompresi. Kecepatan kompresi (*Compression Speed*, CS) diukur dalam satuan *bytes* per detik (*bytes/s*), yang merepresentasikan jumlah data tidak terkompresi yang dapat diproses dalam waktu tertentu. Waktu yang diperlukan untuk mengompresi dan mendekompresi data menjadi faktor krusial yang mempengaruhi kinerja keseluruhan sistem. Waktu kompresi (T_c) merupakan *metrik* yang mengukur waktu dari sebuah algoritma kompresi. Rumus untuk menghitung waktu kompresi dan dekompresi adalah sebagai berikut:

$$T_c = \frac{N_{events} \times 8}{CS}$$

Di mana N_{events} adalah ukuran awal file (dalam *byte*), CS adalah kecepatan kompresi (*Compression Speed*) dalam *bytes/detik*. Dengan rumus ini, semakin kecil T_c , semakin baik algoritma dalam memproses data (Khan et al., 2020).

Berikut adalah tabel lengkap dengan perhitungan kecepatan kompresi berdasarkan waktu kompresi yang telah diketahui:

Tabel 3. Hasil Perhitungan Waktu (Oktaviani et al., n.d.)

Ukuran Asli (byte)	Ukuran Hasil (byte)	Kecepatan Kompresi (byte/s)	Waktu Kompresi (s)
1.000.000	491.469	767.136	10,422
2.000.000	1.671.831	564.387	28,339
5.000.000	4.682.098	539.186	74,177
10.000.000	9.489.495	325.364	246,037

Berdasarkan hasil perhitungan, terlihat bahwa semakin besar ukuran data yang dikompresi, semakin lama waktu yang dibutuhkan untuk proses kompresi. Waktu kompresi bertambah secara signifikan saat ukuran data meningkat, yang berarti bahwa algoritma

BAB I : PENDAHULUAN

Bab ini memuat latar belakang penelitian, identifikasi masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, pertanyaan penelitian, metodologi penelitian, serta sistematika penulisan.

BAB II : LANDASAN TEORITIS

Bab ini mencakup pembahasan teori-teori yang relevan sebagai dasar pengetahuan untuk mendukung proses pengembangan sistem. Uraian meliputi landasan teori yang digunakan dalam penyusunan penelitian, kajian penelitian sebelumnya sebagai referensi, dan kerangka teoritis yang merupakan sintesis dari teori dan penelitian terkait untuk menjadi solusi dalam penelitian.

BAB III : ANALISA DAN PERANCANGAN

Bab ini membahas analisis dan perancangan sistem bursa kerja khusus. Materi yang diuraikan meliputi analisis sistem yang sedang berjalan, identifikasi masalah, kebutuhan sistem, perencanaan sistem, serta desain input dan output pada program.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan proses implementasi dan pengujian sistem berdasarkan hasil perancangan. Pembahasan mencakup analisis sistem, perencanaan, serta desain input dan output yang diterapkan pada program.

BAB V : KESIMPULAN DAN SARAN

Bab terakhir memuat kesimpulan yang dirangkum dari pembahasan di bab sebelumnya, serta memberikan saran untuk pengembangan lebih lanjut berdasarkan hasil penelitian. Saran-saran ini diharapkan dapat mendukung peningkatan sistem di masa depan.