TUGAS 1

DASAR MULTIMEDIA

Dosen Pengampu : A. Ais Prayogi Alimuddin, S.T., M.Eng.



Disusun Oleh:

Nama : Ady Ulil Amri

NIM : D121231080

Kelas : B

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK UNIVERSITAS HASANUDDIN

2024

**TUGAS DASMUL**

Berikut tabel mengenai perbedaan antara ANSI, UTF-16 LE, UTF-16 BE, UTF-8, dan UTF-8 with BOM:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Karakteristik | ANSI (Windows-1252) | UTF-16 LE | UTF-16 BE | UTF-8 | UTF-8 with BOM |
| Encoding Type | Single-byte | Variable-width (2 atau 4 byte) | Variable-width (2 atau 4 byte) | Variable-width (1-4 byte) | Variable-width (1-4 byte) |
| Byte Order | N/A (Single-byte encoding) | Little Endian (LE) | Big Endian (BE) | N/A | N/A |
| Byte Order Mark (BOM) | Tidak ada | Opsional (0xFFFE di awal) | Opsional (0xFEFF di awal) | Tidak ada | Ada (0xEF, 0xBB, 0xBF) |
| Character Coverage | Terbatas (karakter Latin) | Unicode | Unicode | Unicode | Unicode |
| Backward Compatibility | Tidak kompatibel dengan Unicode | Kompatibel dengan ASCII dalam bentuk LE | Kompatibel dengan ASCII dalam bentuk BE | Kompatibel dengan ASCII dan ANSI | Kompatibel dengan ASCII dan ANSI |
| Primary Usage | Dokumen lama dan aplikasi Windows | Aplikasi modern dan format data | Aplikasi modern dan format data | Web, aplikasi, dan format data modern | Web, aplikasi, dan format data modern yang membutuhkan BOM |
| Size Efficiency | Efisien untuk teks Latin | Kurang efisien untuk teks Latin, lebih efisien untuk karakter Asia | Kurang efisien untuk teks Latin, lebih efisien untuk karakter Asia | Sangat efisien untuk teks Latin dan variabel untuk teks lainnya | Mirip dengan UTF-8, tapi ada overhead kecil di awal |
| Supported Characters | | Sekitar 256 karakter | 1.112.064 karakter Unicode | 1.112.064 karakter Unicode | 1.112.064 karakter Unicode | 1.112.064 karakter Unicode |
| Typical File Size | Lebih kecil untuk teks bahasa Inggris | Lebih besar untuk teks bahasa Inggris | Lebih kecil untuk teks bahasa Inggris, bervariasi untuk lainnya | Lebih besar untuk teks bahasa Inggris | Sedikit lebih besar dari UTF-8 karena BOM |
| Compatibility with Web | Tidak digunakan di web | Jarang digunakan di web | Jarang digunakan di web | Standar de facto untuk web | Jarang digunakan di web |
| Encoding/Decoding Speed | Sangat cepat | Cepat (kurang efisien karena ukuran lebih besar) | Cepat (kurang efisien karena ukuran lebih besar | Sangat cepat | Cepat |
| ASCII Compatibility | Tidak kompatibel | Kompatibel dalam mode LE | Kompatibel dalam mode BE | Kompatibel secara langsung | Kompatibel secara langsung |
| Error Handling | Kesalahan pembacaan bisa menyebabkan karakter salah tampil | Kesalahan byte order bisa menyebabkan karakter salah tampil | Kesalahan byte order bisa menyebabkan karakter salah tampil | Kesalahan byte encoding bisa menyebabkan karakter salah tampil atau gagal | Kesalahan byte encoding bisa menyebabkan karakter salah tampil atau gagal |
| Historical Context | Dikembangkan untuk sistem Windows lama | Dikembangkan untuk mendukung teks multibahasa | Dikembangkan untuk mendukung teks multibahasa | Dikembangkan untuk mengatasi keterbatasan encoding lain dan mendukung teks multibahasa secara efisien | Dikembangkan untuk menandai file sebagai UTF-8 |

Penjelasan Tambahan:

1. ANSI (Windows-1252): Sebuah standar lama yang terutama digunakan di Windows untuk karakter Latin. Terbatas dalam hal dukungan karakter dibandingkan dengan encoding Unicode.

2. UTF-16 LE (Little Endian): Menggunakan dua atau empat byte untuk mewakili karakter. Little Endian berarti byte yang kurang signifikan disimpan terlebih dahulu.

3. UTF-16 BE (Big Endian): Sama dengan UTF-16 LE tetapi dengan urutan byte terbalik. Big Endian berarti byte yang lebih signifikan disimpan terlebih dahulu.

4. UTF-8: Encoding Unicode yang sangat efisien dan kompatibel luas. Menggunakan satu hingga empat byte untuk karakter, sehingga sangat hemat untuk teks berbasis ASCII.

5. UTF-8 with BOM: Sama dengan UTF-8, tetapi memiliki Byte Order Mark di awal file untuk menunjukkan bahwa file tersebut menggunakan UTF-8. Meskipun BOM tidak diperlukan dalam banyak konteks modern, ia bisa membantu beberapa aplikasi mengenali encoding dengan benar.