

2106708311

output:

```
StandardPostings
byte hasil encode:
b'"\\x00\\x00\\x00\\x00\\x00\\x00\\x00C\\x00\\x00\\x00\\x00\\x00\\x00Y\\x00\\x00\\x00\\x00\\x00\\x00\\xc6\\x01\\x00\\x00\\x00\\x00\\x00\\x00\\n\\xcb#\\x00\\x00\\x00\\x00\\x00\\xd2\\xbfo\\x0b\\x00\\x00\\x00\\x00'
ukuran encoded postings: 48 bytes
hasil decoding: [34, 67, 89, 454, 2345738, 191872978]

VBEPostings
byte hasil encode: bytearray(b'\\xa2\\xa1\\x96\\x02\\xed\\x01\\x0f\\x12\\xc4Z/i\\xc8')
ukuran encoded postings: 13 bytes
hasil decoding: [34, 67, 89, 454, 2345738, 191872978]

EliasGammaPostings
byte hasil encode: b'\\x04@\\x84,\\x01m\\x00\\x00\\x04y(\\x80\\x00\\x00\\x02\\xd2\\xfd2\\x00'
ukuran encoded postings: 19 bytes
hasil decoding: [34, 67, 89, 454, 2345738, 191872978]

Uji coba dengan postings list yang lebih besar
StandardPostings
ukuran encoded postings: 8000000 bytes
Waktu indexing: 0.026751279830932617 detik

VBEPostings
ukuran encoded postings: 1000000 bytes
Waktu indexing: 0.6105461120605469 detik

EliasGammaPostings
ukuran encoded postings: 125001 bytes
Waktu indexing: 0.7308619022369385 detik
```

Berdasarkan hasil yang diperoleh, berikut adalah analisis yang saya buat:

1. **StandardPostings:** Seperti yang diharapkan, StandardPostings menghasilkan ukuran yang jauh lebih besar karena menggunakan representasi standar. Untuk posting_list kecil, ukuran encoded-nya adalah 48 bytes, sedangkan untuk posting_list yang besar ukurannya 8 juta bytes. Ini menunjukkan bahwa StandardPostings menggunakan 8 byte (64 bit) untuk setiap integer yang disimpan, sesuai dengan representasi int standar dalam Python untuk platform 64-bit.
2. **VBEPostings:** Teknik kompresi VBE sangat efisien untuk posting_list kecil, menghasilkan ukuran hanya 13 bytes. Namun, untuk posting_list yang besar, ukurannya meningkat menjadi 1 juta bytes. Perbedaan ini menggambarkan bagaimana VBE lebih efisien untuk angka-angka kecil, tetapi saat angka bertambah besar, efisiensinya menurun.
3. **EliasGammaPostings:** Elias Gamma, sebagai bit-level coding, memiliki efisiensi yang variatif. Untuk posting_list kecil, ia membutuhkan 19 bytes, sedangkan untuk posting_list yang besar hanya membutuhkan 125001 bytes. Faktanya, untuk posting_list yang besar, Elias Gamma jauh lebih efisien dibandingkan VBE, sesuai dengan teori yang menyatakan bahwa Elias Gamma seharusnya lebih kecil karena merupakan bit-level coding.

Mengapa VBE lebih kecil untuk posting_list yang lebih kecil tetapi lebih besar untuk posting_list yang lebih besar dibandingkan dengan EliasGamma?

1. **Overhead EliasGamma:** Untuk posting_list kecil, bisa jadi overhead dari Elias Gamma (seperti padding bit) membuatnya kurang efisien dibandingkan VBE. Namun, saat data menjadi lebih besar, overhead ini menjadi sepele, dan efisiensi asli Elias Gamma mulai muncul.
2. **Karakteristik Data:** posting_list yang besar (0 hingga 9.999.999) memiliki karakteristik yang sangat berbeda dari posting_list kecil yang diberikan. Perbedaan ini dapat mempengaruhi efisiensi dari teknik kompresi.

Sebagai kesimpulan, walaupun Elias Gamma adalah bit-level coding dan seharusnya lebih efisien, realitasnya tergantung pada data yang sedang kita kompres. VBE bisa menjadi pilihan yang lebih baik untuk data tertentu, sedangkan Elias Gamma mungkin lebih sesuai untuk data lain. Penting untuk selalu melakukan pengujian empiris untuk menentukan metode kompresi mana yang paling sesuai untuk dataset tertentu.

VBE saat indexing collection:

```
100%|██████████| 20/20 [01:44<00:00, 5.24s/it]
Waktu yang diperlukan untuk menjalankan kode: 105.64 detik.
```

Elias Gamma saat indexing collection:

```
100%|██████████| 20/20 [12:20<00:00, 37.01s/it]
Waktu yang diperlukan untuk menjalankan kode: 745.46 detik.
```