

# Kelola Bandwidth Dengan Menerapkan Algoritma Per Connection Queue Pada Jaringan Komputer Rajawali Kupie

Agung Nanda Pratama

<sup>1</sup>Fakultas Ilmu Komputer, Prodi Teknik Informatika Universitas Budi Darma, Medan, Indonesia

Email: agung@gmail.com

Email Penulis Korespondensi: agung@gmail.com

**Abstrak**– Dijaman sekarang umumnya pada tempat-tempat usaha menggunakan wifi internet sebagai daya tarik pengunjung untuk usahanya, seperti usaha coffee shop. Akan tetapi tak jarang dijumpai fasilitas wifi yang disediakan warung kopi atau coffee shop tidak dapat di gunakan, biasanya hal ini disebabkan oleh banyaknya pengguna jaringan wifi, belum lagi ada salah satu pengguna yang melakukan aktifitas download atau pun streaming. Hal ini menyebabkan alokasi bandwidth yang tersedia diraup seluruhnya oleh pengguna yang melakukan aktifitas tersebut, yang mengakibatkan tidak adanya alokasi bandwidth yang tersisa untuk pengguna lainnya. Oleh karena itu perlu adanya suatu pengolahan bandwidth untuk mengatur alokasi bandwidth untuk pengguna jaringan wifi. Salah satu solusi untuk permasalahan tersebut ialah melakukan manajemen bandwidth dengan menggunakan routerboard mikrotik, routerboard mikrotik akan melakukan kelola bandwidth terhadap jaringan dengan menggunakan algoritma per connection queue (PCQ), algoritma ini akan mengalokasikan bandwidth secara merata sesuai jumlah pengguna pada jaringan wifi, sehingga pembagian bandwidth pada jaringan dilakukan secara adil dan sama rata jumlahnya. Dengan adanya kelola bandwidth ini diharapkan dapat membantu pihak pengelola coffee shop dalam mengatasi permasalahan pada jaringan wifi.

**Kata Kunci:** Bandwidth; PCQ; Queue; Mikrotik

**Abstract**– In this day and age generally in places of business using the internet wifi as a tourist attraction for endings, such as coffee shop business. However, it is not uncommon to find wifi facilities provided by coffee shops or coffee shops that cannot be used, usually this is caused by the many users of the wifi network, not to mention one of the users who performs downloading or streaming activities. This causes the allocation of available bandwidth to be captured entirely by users who carry out these activities, resulting in no allocation of the remaining bandwidth for other users. Therefore it is necessary to have a bandwidth processing to regulate bandwidth allocation for wifi network users. One solution to this problem is to manage bandwidth using a proxy routerboard, a proxy routerboard will manage bandwidth to the network using the algorithm per connection queue (PCQ), this algorithm will allocate bandwidth evenly according to the number of users on the wifi network, so the bandwidth distribution the network is carried out fairly and equally. With this bandwidth management it is expected to help the coffee shop management in overcoming problems in the wifi network.

**Keywords:** Bandwidth; PCQ; Queue; Mikrotik.

## 1. PENDAHULUAN

Rajawali kupie merupakan warung kopi yang menjual berbagai jenis kopi olahan aceh salah satunya kopi ulee kareng beserta produk turunannya sebagai minuman utama dan berbagai jenis minuman (sampingan) lainnya seperti teh dan coklat, serta menyediakan jenis makanan ringan yang juga ditawarkan sebagai kudapan pendamping minum kopi. Rajawali Kupie juga merupakan tempat berkumpulnya orang-orang yang sekedar bersantai atau melakukan aktifitas lainnya seperti diskusi atau obrolan, membaca media cetak, online atau buku, menyelesaikan beberapa tugas akademik atau non-akademik.

Rajawali Kupie memanfaatkan layanan internet sebagai fasilitas cafe untuk menarik pengunjung, saat ini Rajawali Kupie sudah menggunakan layanan internet dari suatu provider dengan kecepatan 30 mbps, seperti kita ketahui kecepatan internet segitu seharusnya sudah cukup untuk melayani kebutuhan internet para pengunjung cafe, namun kenyataannya tidaklah cukup disebabkan karena tidak adanya pengolahan bandwidth yang baik sehingga penggunaan bandwidth nya saling tarik menarik yang menyebabkan internet jadi tidak stabil.

Per Connection Queue (PCQ) adalah algoritma untuk mengelola jaringan lalu lintas kualitas layanan (QoS) dengan melakukan bandwidth sharing otomatis dan merata ke multiclient, dan juga untuk menyeimbangkan bandwidth yang digunakan pada tiap klien. Per Connection Queue merupakan penyempurnaan dari metode Stochastic Fairness Queuing (SFQ). Cara kerja kedua metode ini yaitu menyeimbangkan traffic dengan membuat beberapa sub stream (sub queue). Pada Per Connection Queue, parameter yang dapat dipilih untuk menjadi classifier adalah src-address, dst-address, src-port maupun dst-port. Fungsi dari parameter itu adalah sebagai patokan atau standar yang dapat digunakan untuk dijadikan tolak ukur pengujian metode antrian Per Connection Queue[1]

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi data

Kompresi merupakan teknik memperkecil file yang berukuran besar dan mengurangi kebutuhan ruang penyimpanan. Proses kompresi merupakan proses yang mendekati pada minimisasi jumlah bit untuk representasi digital seperti gambar, audio, dan video, yang menghasilkan ukuran data yang lebih kecil namun tetap menjaga kuantitas informasi dalam data tersebut. Kompresi data adalah proses memperkecil atau meminimalisasi jumlah tiap

bit yang merepresentasikan suatu data yang menjadi lebih kecil. Berdasarkan kemungkinan data yang sudah dikompresi dapat dikembalikan ke bentuk aslinya atau data sebelum dikompresi, maka teknik kompresi data dibagi menjadi dua jenis, sebagai berikut:

1. **Loseless Data Compression**  
Loseless Data Compression adalah teknik kompresi data dimana data kompresi dapat didekompres kembali. Selain itu hasil kompresi data yang dihasilkan sama persis seperti data asli sebelum dilakukan kompresi.
2. **Lossy Data Compression**  
Lossy Data Compression adalah teknik pemampatan data yang hasilnya tidak sama dengan data aslinya, namun tidak menjadikannya masalah. Teknik Lossy Data Compression ini contohnya adalah MP3, Streaming Media, JPEG, MPEG dan WMA. Penggunaan teknik Lossy Data Compression data ini dinilai lebih efisien dibandingkan dengan Loseless Data Compression. Alasannya yaitu teknik. Lossy Data Compression hanya membuang bagian-bagian data yang kurang penting. Oleh sebab itu data yang dihasilkan masih bisa digunakan meskipun beberapa bagian dari data aslinya ada yang dihilangkan.

**2.2 Algoritma Stout Codes**

Dalam makalah pendek 1980-nya [Stout 80], Quentin Stout memperkenalkan dua keluarga RI dan SI rekursif, kode panjang variabel untuk bilangan bulat, mirip dengan dan lebih umum dari pada Elias omega dan kode Even – Rodeh. Stout Codes bersifat universal dan asimtotik optimal. Bahwa panjang L dari bilangan bulatn diberikan oleh  $L = 1 + \log_2 n$  (Persamaan (2.1)). Dua kelompok kode tergantung pada pilihan parameter integer l. Nilai a (lebih besar atau sama dengan 2) untuk l telah dipilih, kata sandi pertamaterdiri dari awalan RI (n) dan akhiran 0 n. Akhiran adalah nilai biner dari n dalam bit L, didahului oleh pemisah 0 tunggal. Itu awalan RI (n) terdiri dari grup panjang. Dimulai dengan panjang L dari n, yaitu prepended panjang L1 dari L, maka panjang L2 dari L1, dan seterusnya sampai panjang Li tercapai yang cukup pendek untuk masuk dalam kelompok l-bit. Perhatikan bahwa kelompok panjang (Kecuali mungkin yang paling kiri) mulai dengan 1. [4]-

Dekoder dimulai dengan membaca bit pertama. Ini adalah panjang grup berikutnya. Semakin banyak grup panjang dibaca, hingga grup ditemukan yang diikuti oleh 0. Ini menunjukkan bahwa grup berikutnya adalah n itu sendiri. Dengan latar belakang ini, definisi rekursif dari awalan RI mudah dibaca dan memahami. Kami menggunakan notasi  $L = 1 + \log_2 n$  dan dilambangkan dengan B (n, l) biner l- bit representasi (kode beta) bilangan bulat n. Dengan demikian,  $B(12, 5) = 01100$ . Untuk  $l \geq 2$ , awalan didefinisikan oleh [4].

$$RI(n) = \begin{cases} B(n,l), & \text{for } 0 \leq n \leq 2^l - 1 \\ R_l(L - l - 1)B(n,L), & \text{for } n \geq 2^l \end{cases}$$

Keluarga kedua kode Stout serupa, tetapi dengan awalan yang berbeda dilambangkan dengan SI (n). Untuk nilai l yang kecil, keluarga ini menawarkan beberapa peningkatan dibandingkan RI kode. Secara khusus, menghapus sedikit redundansi yang ada dalam kode RI karena grup panjang tidak boleh 0 (itulah sebabnya grup panjang dalam kode omega mengkodekan  $L_i - 1$  dan bukan  $L_i$ ). Awalan SI mirip dengan awalan RI dengan perbedaan itu grup panjang untuk  $L_i$  mengkodekan  $L_i - 1 - l$ . Jadi, S2 (985), awalan integer 985-bit n, dimulai dengan grup panjang 10-bit 1111011001 dan menambahkannya ke grup panjang untuk  $10 - 1 - 2 = 7 = 1112$ . Untuk ini ditambahkan grup panjang untuk  $3 - 1 - 2 = 0$  sebagai dua bit 00. Hasilnya adalah awalan 15-bit 00 | 111 | 1111011001, lebih pendek dari 19 bit dari R2 (985). Contoh lain adalah S3 (985), yang dimulai dengan 1111011001 yang sama dan bergantung padanya kelompok panjang untuk  $10 - 1 - 3 = 6 = 1102$ . Rekursi berhenti pada ini titik karena 110 adalah grup l-bit. Hasilnya adalah 13-bit codeword 110 | 1111011001,

lagi lebih pendek dari 17 bit R3 (985). Awalan SI (n) didefinisikan secara rekursif oleh

$$SI(n) = \begin{cases} B(n,l), & \text{for } 0 \leq n \leq 2^l - 1 \\ R_l(L - l - 1)B(n,L), & \text{for } n \geq 2^l \end{cases}$$

Tabel 2.1 berisi daftar beberapa awalan S2 (n) dan S3 (n) dan menggambarkan keteraturannya. Melihat bahwa kolom paling kiri mencantumkan nilai L, mis., panjang bilangan bulat yang dikodekan, dan bukan bilangan bulat itu sendiri. Grup panjang mempertahankan nilainya hingga grup yang mengikutinya menjadi semua 1, di mana titik grup bertambah 1 dan grup yang mengikuti diatur ulang ke 10 ... 0. Semua grup panjang, kecuali mungkin yang paling kiri, mulai dengan 1. Perilaku reguler ini adalah hasil dari pilihan  $L_i - 1 - l$ . [5]

**Tabel 1.** Kode S2(n) dan Kode S3(n) David Salomon dan Giovani Motta, 2010[4]

<i>L</i>	<i>S2(n)</i>	<i>S3(n)</i>
1	01	001
2	10	010
3	11	011
4	00 100	100
5	00 101	101
6	00 110	110
7	00 111	111
8	01 1000	000 1000
15	01 1111	000 1111
16	10 10000	001 10000
32	11 100000	010 100000
64	00 100 1000000	0111000000
128	00 101 10000000	10010000000
256	00110100000000	101 100000000
512	00 111 1000000000	110 1000000000
1024	01 1000 10000000000	111 10000000000
2048	01 1001 100000000000	000 1000 100000000000

Awalan S2 (64), misalnya, dimulai dengan grup 7-bit 1000000 = 64 dan bergantung padanya S2 (7-1-2) = S2 (4) = 00 | 100

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Penerapan Algoritma Stout Code

Berdasarkan analisa, *kompresi file* memiliki ukuran lebih besar ukuran yang lainnya. Dengan melakukan kompresi file, file yang berukuran besar akan dikompresi menjadi ukuran yang kecil dan akan mengurangi alokasi penyimpanan. Dalam penelitian ini, akan dibahas 2 teknik proses utama yaitu proses kompresi dan dekompresi, dan peneliti akan mengkompresi sebuah *file* dengan menggunakan sebuah algoritma *stout codes*.

**Tabel 2.** Tabel Frekuensi Kemunculan Karakter

n	Hexsa	ASCII Biner	Bit	Frekuensi	Bit x Frekuensi
1	00	00000000	8	4	32
2	70	01110000	8	3	24
3	39	00111001	8	1	8
4	2B	00101011	8	1	8
5	A0	10100000	8	1	8
Jumlah					80

Mencari codeword dari setiap karakter pada data yang akan dikompresikan, untuk mencari codeword, pertama kita akan menentukan nilai l, pada penyelesaian ini kita akan menggunakan nilai l=2. Setelah kita menentukan nilai l, maka kita akan memasukan rumus yang ada sebagai berikut :

- Untuk nilai  $0 = n = 2^l - 1$   
karena nilai l=2, maka :  $0 = n = 2^2 - 1 \Rightarrow 0 = n = 3$

n = 1, maka codeword yang dihasilkan adalah nilai biner dari n yaitu 1 dan diambil sebanyak nilainya dari l adalah 2 sehingga kata kodenya adalah 01

$n = 2$ , maka codeword yang dihasilkan adalah nilai biner dari  $n$  yaitu 10 dan diambil sebanyak nilainya dari 1 yang 2 sehingga codeword adalah 10  
 $n = 3$ , maka codeword yang dihasilkan adalah nilai biner dari  $n$  yaitu 11 dan diambil sebanyak nilainya dari 1 yang 2 sehingga codeword adalah 11

2. Untuk nilai  $n \geq 2^1$

karena nilai  $l=2$ , maka  $n = 2^2 \Rightarrow n = 4$

$n = 4$  dengan nilai biner 100, maka nilai  $L = 3$  diambil dari panjang bit biner nilai  $n$  dan langkah selanjutnya mencari nilai  $R$ .

$R_2 (3-1-2) = 0$ , nilai biner adalah 0 dan diambil sebanyak nilai 1, yaitu 2 sehingga nilainya dari  $R = 00$ .

$R_1 (L-1-1) B (n,L) = 00100$

$n=5$  dengan nilai biner 101, maka nilai  $L=3$

$R_2(3-1-2) = 0$ , nilai binernya 0 dan diambil sebanyak nilai 1, yaitu 2 sehingga nilai  $R=00$ .

Setelah membentuk codeword, langkah kelima adalah mengganti karakter yang ada dengan codeword itu telah dihasilkan, prosesnya dapat dilihat pada table.

**Tabel 3.** Pergantian Karakter Menjadi Codeword

N	Hexsa	Codeword	Bit	Frekuensi	Bit x Frekuensi
1	00	01	2	4	8
2	70	10	2	3	6
3	39	11	2	2	2
4	2B	00100	5	1	5
5	A0	00101	5	1	5
Jumlah					26

Langkah selanjutnya adalah untuk mengatur ulang codeword yang dalam urutan

**Tabel 4.** Codeword Yang Dalam Urutan Awal

39	2B	70	70	00	00	A0	00	70	00
11	00100	10	10	01	01	00101	01	10	01

Dari proses di atas, string yang dihasilkan sedikit "11001001010010100101011001"

Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *stringbit* itu sendiri yaitu *padding* bit dan *flag* bit. Menambahkan padding dilakukan jika string bit yang dihasilkan tidak dapat dibagi oleh 8.

Dalam string yang dihasilkan, ada 26bit, di mana 26 tidak habis dibagi 8 dan sisanya 2 diekspresikan dalam  $n$ . Menambahkan padding dilakukan dengan menambahkan 0 sebanyak  $7 - n + "1"$  di akhir bit string,  $7-2 + "1"$  lalu menambahkan padding "000110". Menambahkan padding dilakukan dengan rumus  $9-n$ , yaitu  $9-2 = 7$  yang dinyatakan dalam bentuk angka 8-bit biner sehingga menghasilkan "00000111". Bit string yang telah ditambahkan ke padding dan flagging menjadi "1100100101001010010101100100011000000111" dengan total panjang bit 40 bit.

a) Rasio kompresi (CR).

Compression Ratio (Cr) adalah rasio persentase antara data yang telah dikompresi oleh datayang belum dikompresi.

$$Cr = \frac{UkuranDataSetelahDikompresi}{UkuranDataSebelumDikompresi} \times 100\%$$

$$Cr = \frac{40bit}{80bit} \times 100\% = 50\%$$

Untuk mengembalikan binary menjadi *string bit* semula dapat dilakukan melalui langkah berikut ini. Lakukan pembacaan pada 8 bit terakhir, hasil pembacaan berupa bilangan desimal. Nyatakan hasil pembacaan dengan  $n$ . Hilangkan bit pada bagian akhir sebanyak  $7+n$ . Setelah dilakukan perhitungan pembacaan bit akhir. Nilai biner yang dihilangkan sebanyak 8 bit pada akhir.  $n = 1$ . Hilangkan  $7 + n$  atau  $7 + 1 = 14$ . Penjelasan diatas menunjukan bahwa bit akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula dapat dilihat sebagai berikut ini: "11001001010010100101011001"

**Tabel 5.** Pengecekan Bit

Indeks	Nilai	Keterangan
1	1	Tidak Ada
2	10	Ada pada tabel
3	0	Tidak Ada

4	10	Ada pada tabel
5	0	Tidak Ada
6	10	Ada pada tabel
7	1	Tidak Ada
8	00	Ada pada tabel
9	1	Tidak Ada
10	01	Tidak Ada
11	001	Tidak Ada
12	0101	Tidak Ada
13	11001	Ada pada tabel
14	0	Tidak Ada
15	10	Ada pada tabel
16	0	Tidak Ada
17	10	Ada pada tabel
18	1	Tidak Ada
19	00	Ada pada tabel
20	1	Tidak Ada
21	01	Tidak Ada
22	001	Tidak Ada
23	0101	Tidak Ada
24	11001	Ada pada tabel
25	0	Tidak Ada
26	10	Ada pada tabel

**Tabel 6.** Tabel *Hexa* dan *stout codes*

n	Hexsa	Codeword	Bit	Frekuensi	Bit x Frekuensi
1	00	01	2	4	8
2	70	10	2	3	6
3	39	11	2	2	2
4	2B	00100	5	1	5
5	A0	00101	5	1	5
		Jumlah			26

#### 4. KESIMPULAN

Kesimpulan yang dapat diambil setelah melakukan kompresi databases pada aplikasi pinjaman online dengan algoritma stout codes adalah :

1. Prosedur dari kompresi dan dekompresi sebuah databases dimulai dari memilih data yang akan dikompresi dari databases tersebut akan dikompresi dengan menggunakan algoritma stout codes dan dapat dihasilkan menjadi sebuah databases yang sudah memiliki ukuran lebih kecil dari ukuran sebelumnya. Prosedur dari dekompresi dari

- data tersebut dimulai dari input data yang sudah dikompresi dan akan didekompresikan sehingga mendapatkan hasil data yang berukuran seperti semula tanpa menghilangkan suatu data tertentu.
2. Penerapan algoritma stout codes untuk kompresi databases yaitu untuk membaca sebuah nilai heksadesimal yang didapatkan dari suatu databases, kemudian dengan cara mengubah nilai heksadesimal tersebut menjadi sebuah nilai bit yang baru dan berupa bilangan biner, kemudian menyusun kembali nilai biner yang sudah menjadi karakter baru.
  3. Dengan dibangunnya aplikasi kompresi databases ini dengan menggunakan stout codes maka dapat menghasilkan hingga 50%. Untuk aplikasi pemrograman yaitu Microsoft Visual Basic 2008 supaya mempermudah suatu proses kompresi, dengan cara menginput sebuah databases yang akan dikompresi

## UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

## REFERENCES

- [1] K. Mahesa, "Rancang Bangun Aplikasi Kompresi Dan Dekompresi Pada Citra Digital Menggunakan Metode Huffman," vol. 12, no. 1, pp. 948–963, 2017.
- [2] M. R. Pane, "Perancangan Aplikasi Kompresi Menggunakan Metode Shannon Fano dan Unary Coding pada File Teks," *Inf. dan Teknol. Ilm.*, vol. 12, no. September, pp. 306–311, 2017.
- [3] G. Urva, H. F. Siregar, J. Prof, M. Y. Kisaran, and S. Utara, "Pemodelan UML E- Marketing Minyak Goreng," no. 9, pp. 92–101, 2015.
- [4] R. D. Pratiwi, S. D. Nasution, and F. Fadlina, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Fixed Length Binary Encoding (Flbe)," *J. Media Inform. Budidarma*, vol. 2, no. 1, pp. 10–14, 2018, doi: 10.30865/mib.v2i1.813.
- [5] D. Salomon and G. Motta, *Handbook of data compression*. 2010.
- [6] S. D. Nasution, "Data Compression Using Stout Codes," vol. 3, no. 1, pp. 28–33, 2019.
- [7] S. Yohanlis, "Perancangan Data Warehouse pada Profil Mahasiswa di Universitas XYZ," *J. Tek. dan Ilmu Komput.*, vol. 3, no. 10, pp. 192–206, 2014.
- [8] H. Hidayat, T. Pamungkas, and W. Zarman, "Implementasi Algoritma Kompresi Lzw Pada Database Server," *Komputa J. Ilm. Komput. dan Inform.*, vol. 2, no. 1, 2015, doi: 10.34010/komputa.v2i1.74.
- [9] P. Daerah, K. Paser, and J. I. Mulawarman, "Jurnal Informatika Mulawarman," vol. 2, no. 1, p. 2017, 2015.
- [10] M. Sadeli, *7 Jam Belajar Interaktif Visual Basic.net 2008 Untuk Orang Awam*. Palembang: Maxikom, 2009.
- [11] A. Kadir, *REKAYASA PERANGKAT LUNA*. 2013.