

ANALISIS PERANCANGAN DESAIN KOMPUTASI PARAREL DENGAN ALGORITMA LEMPEL ZIV WELCH (LZW)

Achmad Aries Pirnando^{1*)}, Rido Febryansyah²⁾

¹Informatika

²Sistem Informasi

*) aries.pirnando@gmail.com

Abstrak

Metode kompresi menggunakan algoritma Lempel Ziv Welch (LZW) merupakan algoritma yang sering digunakan dalam proses kompresi data pada beberapa penelitian sebelumnya tentang teknologi kompresi data berbasis kamus LZW. Algoritma LZW dianggap sebagai algoritma kompresi yang lebih baik daripada algoritma kompresi dasar lainnya. Menurut penelitian lain, proses kompresi data menggunakan algoritma LZW menghasilkan rasio kompresi 60–70 untuk data tertentu, baik data gambar maupun data teks. Sedangkan pada penelitian lain, algoritma LZW menghasilkan waktu kompresi 55-6ms untuk data teks yang berisi 200-1000 karakter. Berdasarkan penelitian sebelumnya dijelaskan bahwa komputasi paralel dengan prosesor yang berbeda dapat meningkatkan efisiensi algoritma dalam hal volume data yang besar. Penggunaan komputasi paralel pada algoritma Lempel Ziv-Welch (LZW) berimplikasi pada peningkatan kinerja algoritma dalam proses kompresi data sesuai dengan kecepatan proses kompresi data dan ukuran kompresi data yang dihasilkan.

Kata Kunci: Kompresi, Paralel, Algoritma LZW.

PENDAHULUAN

Metode kompresi menggunakan algoritma Lempel Ziv Welch (LZW) merupakan algoritma yang sering digunakan dalam proses kompresi data pada beberapa penelitian sebelumnya tentang teknologi kompresi data berbasis kamus LZW (Darwis et al., 2022), (Ahdan et al., 2021). Algoritma LZW dianggap sebagai algoritma kompresi yang lebih baik daripada algoritma kompresi dasar lainnya. Menurut penelitian lain, proses kompresi data menggunakan algoritma LZW menghasilkan rasio kompresi 60–70 untuk data tertentu, baik data gambar maupun data teks (Neneng et al., 2021; Surahman et al., 2021; Wicaksono et al., n.d.). Sedangkan pada penelitian lain, algoritma LZW menghasilkan waktu kompresi 55-6ms untuk data teks yang berisi 200-1000 karakter.

Secara umum, algoritma LZW tetap menggunakan proses perhitungan sekuensial. Kecepatan pemrosesan algoritma LZW dipengaruhi oleh kinerja perangkat keras prosesor (Kurniati et al., 2015), (Rossi et al., 2018), (Soraya & Wahyudi, 2021). Komputasi paralel

adalah cara untuk meningkatkan kinerja proses komputasi serial (Aziz & Fauzi, 2022), (Jamaaluddin & Sumarno, 2017). Komputasi paralel adalah metode komputasi di mana proses komputasi dibagi dan dieksekusi secara bersamaan pada beberapa komputer secara paralel (Alita & Isnain, 2020), (Reza & Putra, 2021) .

Berdasarkan penelitian sebelumnya dijelaskan bahwa komputasi paralel dengan prosesor yang berbeda dapat meningkatkan efisiensi algoritma dalam hal volume data yang besar (Nurkholis & Sitanggang, 2020), (Ahdan & Susanto, 2021), (Putra et al., 2019). Penggunaan komputasi paralel pada algoritma Lempel Ziv-Welch (LZW) berimplikasi pada peningkatan kinerja algoritma dalam proses kompresi data sesuai dengan kecepatan proses kompresi data dan ukuran kompresi data yang dihasilkan (Abidin & Permata, 2021), (Susanto et al., 2021).

KAJIAN PUSTAKA

Komputasi Pararel

Pada dasarnya, komputasi paralel membagi suatu masalah menjadi lebih kecil sehingga masing-masing prosesor (CPU) dapat bekerja pada waktu yang sama atau pada waktu yang sama (simultan), yang disebut konkurensi (Ahdan et al., 2020), (Saipulloh Fauzi1, 2020), (Rusliyawati & Sinaga, 2017).

Algoritma Pararel

Algoritma paralel menjelaskan langkah-langkah komputer paralel dalam menyelesaikan tugas. Hal-hal yang terjadi pada algoritma paralel adalah:

1. Identifikasi terhadap beban permasalahan yang akan dikerjakan secara paralel.
2. Pemetaan porsi pekerjaan yang dibebankan kepada tiap-tiap prosesor.
3. Distribusi data input, output dan prantara yang terkait dengan program.
4. Pengaturan data yang diakses bersamaan oleh beberapa prosesor
5. Menyelaraskan fungsi prosesor pada setiap langkah pekerjaan.

Speed-Up

Dalam hal komputasi paralel, speedup menunjukkan seberapa cepat komputasi paralel dibandingkan dengan komputasi sekuensial(Ahmad et al., 2021), (Wantoro, Syarif, et al., 2021), (Wantoro, Rusliyawati, et al., 2021). Pada dasarnya percepatan cepat dirumuskan sebagai berikut:

$$S N = \frac{T_p(1)}{T_p(N)}$$

Executor Framework

Executor Framework adalah opsi lain yang digunakan untuk membuat utas di Java. Executing Framework menggunakan thread untuk mengeksekusi thread. Mulai kerangka kerja eksekutor menggunakan antarmuka Layanan Pelaksana dalam paket `java.util.concurrent`. Objek Ace Executor Service dibuat menggunakan objek Executors(Setiawan & Pasha, 2020), (LIA FEBRIA LINA, 2019).

Kompresi Data

Kompresi data mengacu pada pengkodean data menggunakan bit atau unit pembawa informasi orde rendah lainnya daripada representasi data yang tidak dikodekan menggunakan skema pengkodean tertentu. Kompresi data proses adalah persyaratan umum di sebagian besar aplikasi(Darwis et al., 2018), (Darwis & Pamungkas, 2021).

Rasio Komperesi

Rasio kompresi adalah rasio antara ukuran data asli sebelum kompresi dan ukuran data terkompresi(Widodo et al., 2020), (Styawati et al., 2021). Banyaknya tahapan kompresi menunjukkan format gambar yang dicapai dalam satu kali proses kompresi, semakin rendah nilai tahapan kompresi maka semakin memuaskan hasil kompresi(Samsugi et al., 2018). Berikut adalah rumus rasio kompresi:

$$\text{Rasio Kompresi} = \frac{\text{Ukuran Setelah Kompresi}}{\text{Ukuran Sebelum Kompresi}}$$

Algoritma Lempel Ziv Welch (LZW)

Lempel Ziv Welch (LZW) adalah algoritma kompresi lossless umum yang dibuat oleh Abraham Lempel, Jacob Ziv, dan Terry Welch. Algoritma ini melakukan kompresi menggunakan kamus, di mana - fragmen teks diganti dengan indeks yang diperoleh dari kamus. Pendekatan ini fleksibel dan efisien, karena banyak karakter dapat dikodekan dengan mengacu pada string yang sebelumnya muncul dalam teks(Satria & Haryadi, 2018), (*Strategi Pengembangan Bisnis Usaha Mikro Kecil Menengah Keripik Pisang Dengan Pendekatan Business Model Canvas*, 2020). Prinsip kompresi dicapai ketika referensi dalam bentuk pointer dapat disimpan dalam lebih banyak bit daripada string asli(Alita, 2021). Lempel Ziv Welch (LZW) adalah algoritma kompresi umum yang dapat menangani semua jenis data(Arrahman, 2022).

Kompresi LZW

Proses kompresi LZW dimulai dengan menginisialisasi kamus yang berisi urutan karakter input yang mungkin. Algoritma ini memindai string input, yang diubah menjadi substring, hingga menemukan atau gagal menemukan karakter dalam kamus(Alita et al., 2020), (Bakri & Darwis, 2021). Ketika substring ini ditemukan, indeks string dari setiap karakter terakhir (yaitu substring terpanjang dalam kamus) dicari dan dikeluarkan, dan string baru (termasuk karakter terakhir) ditambahkan ke kamus(Alita et al., 2020), (Dita et al., 2021). dengan kode yang tersedia berikut. Carriage return kemudian digunakan sebagai titik awal untuk mencari substring berikutnya(Hendrastuty et al., 2021).

Dekompresi LZW

Algoritma decoding bekerja dengan membaca nilai input yang dikodekan dan mencetak kamus format string(Mertania & Amelia, 2020). Pada saat yang sama, dapatkan nilai berikutnya dari input dan tambahkan kamus gabungan dari string yang dihasilkan dan karakter pertama dari string yang diekstraksi ke nilai input berikutnya(Ahluwalia, 2020).

METODE

Kerangka Penelitian

Kerangka penelitian berawal dari adanya sebuah masalah yaitu ukuran data yang semakin besar dan bagaimana melakukan proses pemampatan data. Kemudian, untuk melakukan proses pemampatan data digunakan pendekatan kompresi lossless dengan menggunakan algoritma Lempel Ziv welch dan proses komputasi paralel dengan menggunakan teknik dekomposisi data. Proses selanjutnya dilakukan dengan melakukan indentifikasi kombinasi antara komputasi paralel dengan teknik data dekomposisi dan algoritma Lempel Ziv Welch yang digunakan untuk melakukan proses kompresi data, yang diharapkan dapat meningkatkan performa dari algoritma Lempel Ziv Welch itu sendiri. Proses ini bertujuan untuk menggabungkan kedua metode serta menghitung performansi dari proses – proses tersebut. Dalam proses selanjutnya dilakukan pengujian pada setiap data, yaitu uji hasil rasio, uji kecepatan proses kompresi maupun dekompresi data, dan speed up atau peningkatan kecepatan paralelisasi. Setelah dilakukan pengujian maka dapat dihasilkan sebuah desain komputasi paralel pada proses algoritma Lempel ziv welch (LZW).

Alat Penelitian

1. Perangkat Keras
 - a. Prosesor Intel Pentium 2.20 GHz (2 core), Intel I5 2.50 GHz (2 core, Hyper – threading 4 core), Intel I7 2.50 GHz (4 core,Hyper – threading 8 core), dan
 - b. Memori RAM (Random Akses Memori) 4 GB.
2. Perangkat Lunak
 - a. Sistem operasi Windows
 - b. Bahasa pemrograman java (jdk versi 8) dengan tambahan concurrentAPI dan Framework Executor
 - c. Eclipse Nano

Studi Pustaka dan Literatur

Metode pengumpulan data yang dilakukan penelitian ini yaitu studi pustaka. Studi pustaka dilakukan untuk mendapatkan referensi tentang metode yang akan digunakan dalam melakukan penggabungan atau mendesain komputasi paralel pada algoritma Lempel ziv welch (LZW). Studi pustaka ini didapatkan dari berbagai sumber seperti jurnal, buku, thesis, dan skripsi.

Desain Konsep

Desain konsep disajikan dengan menggunakan flowchart diagram untuk menggambarkan proses kompresi data dengan menggunakan komputasi paralel yang akan dibangun. Berikut desain konsep implementasi atau pengujian komputasi paralel pada proses kompresi algoritma Lempel ziv welch.

HASIL DAN PEMBAHASAN

Proses komputasi pada algoritma Lempel ziv welch diimplementasikan pada beberapa laptop yang memiliki prosesor Intel Pentium 2.20 GHz (2 core), Intel I5 2.50 GHz (2 core, Hyper – threading 4 core), dan Intel I7 2.50 GHz (4 core, Hyper – threading 8 core) dan menggunakan jumlah tread sebanyak 2 kali dari jumlah core prosesor yaitu Intel Pentium 4 thread, Intel I5 8 thread dan Intel I7 16 thread. Tujuan dari pengujian ini adalah mendapatkan waktu proses dari masing – masing prosesor yang digunakan dengan jumlah core yang berbeda serta untuk mengetahui besarnya rasio kompresi dan speedup yang dihasilkan. Variable analisis yang digunakan terhadap hasil pengujian adalah waktu inialisasi thread, perbandingan kecepatan, speed-up, dan rasio kompresi.

Hasil Pengujian

Hasil pengujian waktu inialisasi thread

Hasil pengujian ini menerangkan hasil yang diperoleh pada pengujian inialisasi thread dengan menggunakan prosesor yang berbeda-beda, satuan pengujian ini menggunakan millisecond.

Prosesor	Waktu inisialisasi (ms)
Intel Pentium 2.20 GHZ (2 core) dengan 4 thread	2,70315
Intel I5 2.50 GHZ (2 core, Hyper 4) dengan 8 thread	1,84204
Intel I7 2.50 GHZ (4 core, Hyper 8) dengan 16 thread	1,34103

Hasil pengujian berdasarkan waktu proses

Hasil pengujian ini menerangkan table hasil yang diperoleh pada pengujian komputasi pada algoritma Lempel ziv welch berdasarkan waktu proses satuan nilai yang dipakai yaitu milisecond.

Data (bytes)	Intel Pentium 2.20 GHz (2 core)	Intel I5 2.50 GHz (2 core, Hyper – threading 4 core)	Intel I7 2.1 GHz (4 core, Hyper – threading 8 core)
+100000	73,93163	62,87196	19,93295
+200000	111,97508	73,39544	37,12057
+300000	119,19573	83,62374	47,13755
+400000	175,64832	98,27015	63,56976
+500000	184,66861	119,65497	72,87525

Hasil pengujian berdasarkan rasio kompresi

Hasil pengujian ini menerangkan table hasil yang diperoleh pada pengujian komputasi pada algoritma Lempel ziv welch berdasarkan rasio kompresi yang dihasilkan.

Data (bytes)	Hasil Kompresi Paralel (bytes)	Hasil Kompresi Serial (bytes)	Rasio Kompresi Paralel	Rasio Kompresi Serial
+100000	24834	16167	0,24213	0,15631
+200000	40468	25158	0,19728	0,12264
+300000	53478	32403	0,17381	0,10531
+400000	64664	38543	0,15762	0,09395
+500000	74759	44010	0,14578	0,08582

Data (bytes)	Hasil Kompresi Paralel (bytes)	Hasil Kompresi Serial (bytes)	Rasio Kompresi Paralel	Rasio Kompresi Serial
+100000	30286	16167	0,29952	0,15631
+200000	49681	25158	0,24219	0,12264
+300000	66155	32403	0,21501	0,10531
+400000	80936	38543	0,19728	0,09395
+500000	94392	44010	0,18401	0,08582

Data (bytes)	Hasil Kompresi Paralel (bytes)	Hasil Kompresi Serial (bytes)	Rasio Kompresi Paralel	Rasio Kompresi Serial
+100000	36420	16167	0,35511	0,15763
+200000	60561	25158	0,29524	0,12264
+300000	81014	32403	0,26331	0,10532
+400000	99369	38543	0,24221	0,09395
+500000	116396	44010	0,22697	0,08582

Hasil Pengujian berdasarkan speed-up

Hasil pengujian ini menerangkan table hasil yang diperoleh pada pengujian komputasi pada algoritma Lempel ziv welch berdasarkan speed-up.

Data (bytes)	Kecepatan Waktu Kompresi Serial (ms)	Kecepatan Waktu Kompresi Paralel (ms)	SpeedUp
+100000	83,67891	73,93163	1,13184
+200000	121,83338	111,97508	1,08804
+300000	147,64961	119,19573	1,23871
+400000	178,97904	175,64832	1,01901
+500000	203,60006	184,66861	1,10251

Data (bytes)	Kecepatan Waktu Kompresi Serial (ms)	Kecepatan Waktu Kompresi Paralel (ms)	SpeedUp
+100000	71,59979	62,87196	1,13882
+200000	92,08267	73,39544	1,25461
+300000	110,56424	83,62374	1,32216
+400000	145,23621	98,27015	1,47792
+500000	145,89881	119,65497	1,21932

Data (bytes)	Kecepatan Waktu Kompresi Serial (ms)	Kecepatan Waktu Kompresi Paralel (ms)	SpeedUp
+100000	31,96121	19,93295	1,60343
+200000	59,40885	37,12057	1,60042
+300000	83,52161	47,13755	1,77186
+400000	103,65983	63,56976	1,63064
+500000	111,55797	72,87525	1,53081

Analisis

Setelah mendapatkan hasil pengujian algoritma paralel pada algoritma Lempel ziv welch maka dapat dilakukan analisis terhadapnya.

Analisis waktu inisialisasi thread

Berdasarkan data yang didapatkan, bila digambarkan ke dalam grafik makadiketahui kecepatan pembangunan thread terjadi selama lebih dari 2 ms untuk prosesor Intel Pentium, 1 ms untuk prosesor Intel i5, dan kecepatan pembangunan thread tertinggi

terjadi pada prosesor Intel i7 0,74103. Kecepatan ini tidak dipengaruhi oleh berapa besarnya data, tetapi dipengaruhi oleh clock speed oleh beberapa prosesor yang digunakan.



Gambar 1 Grafik Inisialisasi Thread

Analisis kecepatan proses kompresi parallel

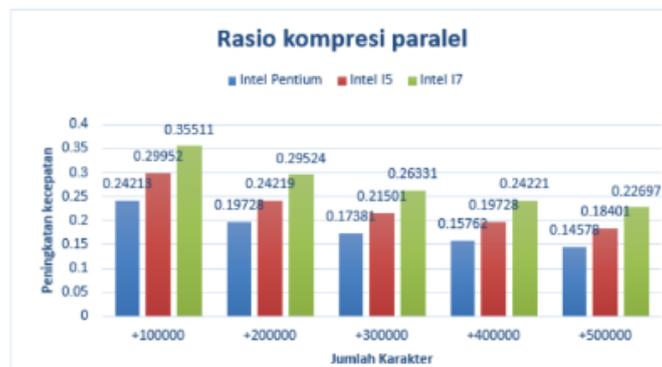
Implementasi komputasi paralel pada algoritma Lempel ziv welch terhadap beberapa jumlah core yang diuji menghasilkan kecepatan yang lebih signifikan dibandingkan dengan algoritma serialnya. Berdasarkan data yang didapat, pengujian dengan menggunakan prosesor Intel I7 2.50 GHz (4 core, Hyper – threading 8 core) menghasilkan waktu kecepatan yang lebih cepat untuk setiap 5 ukuran data dibandingkan dengan pengujian menggunakan prosesor lainnya, sedangkan pada proses komputasi paralel menggunakan prosesor intel Pentium mengalami peningkatan waktu yang cukup besar untuk setiap 5 data yang diuji. Kondisi ini tampak pada grafik berikut ini :



Gambar 2 Kecepatan Komputasi Pararel

Analisis hasil rasio kompresi parallel

Berdasarkan hasil pengujian rasio kompresi dengan menggunakan komputasi paralel, prosesor intel Pentium (2 core) menghasilkan rasio kompresi yang lebih kecil dibandingkan dengan yang lainnya. Sedangkan hasil rasio kompresi tertinggi berada pada saat pengujian menggunakan prosesor intel I7 (quadcore) saat data yang diuji +100000 bytes dengan nilai sebesar 0,35511. Kondisi ini tampak pada grafik berikut ini :



Gambar 3 Rasio Kompresi Pararel

Berdasarkan grafik yang telah digambarkan, rasio kompresi memiliki ukuran-ukuran yang berbeda. Hal ini disebabkan pada saat proses implementasi, kamus data yang digunakan masih dilakukan secara sendiri-sendiri pada setiap proses. Hal ini mengakibatkan data hasil kompresi mengalami perubahan sesuai dengan jumlah core pada saat pembagian data saat akan melakukan proses kompresi. Secara umum, bila data hasil rasio kompresi serial dibandingkan dengan rasio kompresi paralel, maka hasil rasio kompresi paralel akan mengalami peningkatan 2 hingga 4 kali lipat dari ukuran hasil rasio kompresi serial.



Gambar 4 Perbandingan Rasio

Analisis peningkatan kecepatan paralel (SpeedUp)

Berdasarkan hasil yang diujikan, maka didapatkan bahwa setiap proses komputasi paralel mengalami peningkatan kecepatan terhadap komputasi serial hal ini didukung oleh jenis dan jumlah core yang dimiliki oleh setiap prosesor. Peningkatan kecepatan paralel tertinggi didominasi oleh prosesor berjenis quadcore intel I7. Peningkatan tertinggi terjadi pada prosesor intel I7 quadcore dengan peningkatan sebesar 1.772, sedangkan untuk peningkatan terendah terjadi pada prosesor intel Pentium dengan peningkatan sebesar 1.019.



Gambar 5 Peningkatan Kecepatan

SIMPULAN

Berdasarkan uraian yang telah disampaikan bahwa desain komputasi paralel algoritma lempel ziv welch dapat diambil kesimpulan sebagai berikut :

1. Rasio kompresi yang dihasilkan oleh komputasi paralel mengalami peningkatan untuk setiap prosesor yang diujicoba. Hal ini dipengaruhi karna, desain komputasi pada saat pembuatan kamus dilakukan sendirisendiri oleh setiap proses. Hal ini mengakibatkan banyaknya data yang berulang pada saat proses hasil kompresi paralel digabungkan kembali.
2. Prosesor Intel I7 2.1 GHz (4 core, Hyper – threading 8 core) menghasilkan waktu kecepatan yang lebih cepat untuk setiap 5 ukuran data dibandingkan dengan pengujian menggunakan prosesor lainnya, sedangkan pada proses komputasi paralel

menggunakan prosesor intel pentium mengalami peningkatan waktu yang cukup lambat dibandingkan dengan yang lainnya.

3. Peningkatan kecepatan (SpeedUp) paralel tertinggi didominasi oleh prosesor berjenis quadcore intel I7. Peningkatan tertinggi terjadi pada prosesor intel I7 quadcore dengan nilai sebesar 1.772, sedangkan untuk peningkatan terendah terjadi pada prosesor intel Pentium dengan nilai sebesar 1.019.

REFERENSI

- Abidin, Z., & Permata, P. (2021). Pengaruh Penambahan Korpus Paralel Pada Mesin Penerjemah Statistik Bahasa Indonesia Ke Bahasa Lampung Dialek Nyo. *Jurnal Teknoinfo*, 15(1), 13. <https://doi.org/10.33365/jti.v15i1.889>
- Ahdan, S., Putri, A. R., & Sucipto, A. (2020). Aplikasi M-Learning Sebagai Media Pembelajaran Conversation Pada Homey English. *Sistemasi*, 9(3), 493. <https://doi.org/10.32520/stmsi.v9i3.884>
- Ahdan, S., Sucipto, A., Priandika, A. T., & ... (2021). Peningkatan Kemampuan Guru SMK Kridawisata Di Masa Pandemi Covid-19 Melalui Pengelolaan Sistem Pembelajaran Daring. *Jurnal ABDINUS ...*, 5(2), 390–401. <http://ojs.unpkediri.ac.id/index.php/PPM/article/view/15591>
- Ahdan, S., & Susanto, E. R. (2021). IMPLEMENTASI DASHBOARD SMART ENERGY UNTUK PENGONTROLAN RUMAH PINTAR PADA PERANGKAT BERGERAK BERBASIS INTERNET OF THINGS. *Jurnal Teknoinfo*, 15(1), 26–31.
- Ahluwalia, L. (2020). EMPOWERMENT LEADERSHIP AND PERFORMANCE: ANTECEDENTS. *Angewandte Chemie International Edition*, 6(11), 951–952., 7(1), 283. [http://www.nostarch.com/javascriptforkids%0Ahttp://www.investopedia.com/terms/i/in_specie.asp%0Ahttp://dspace.ucuenca.edu.ec/bitstream/123456789/35612/1/Trabajo de Titulacion.pdf%0Ahttps://educacion.gob.ec/wp-content/uploads/downloads/2019/01/GUIA-METODOL](http://www.nostarch.com/javascriptforkids%0Ahttp://www.investopedia.com/terms/i/in_specie.asp%0Ahttp://dspace.ucuenca.edu.ec/bitstream/123456789/35612/1/Trabajo%20de%20Titulacion.pdf%0Ahttps://educacion.gob.ec/wp-content/uploads/downloads/2019/01/GUIA-METODOL)
- Ahmad, I., Borman, R. I., Caksana, G. G., & Fakhrurozi, J. (2021). IMPLEMENTASI STRING MATCHING DENGAN ALGORITMA BOYER-MOORE UNTUK MENENTUKAN TINGKAT KEMIRIPAN PADA PENGAJUAN JUDUL SKRIPSI/TA MAHASISWA (STUDI KASUS: UNIVERSITAS XYZ). *SINTECH (Science and Information Technology) Journal*, 4(1), 53–58.
- Alita, D. (2021). Multiclass SVM Algorithm for Sarcasm Text in Twitter. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(1), 118–128. <https://doi.org/10.35957/jatisi.v8i1.646>
- Alita, D., Fernando, Y., & Sulistiani, H. (2020). Implementasi Algoritma Multiclass SVM pada Opini Publik Berbahasa Indonesia di Twitter. *Jurnal Tekno Kompak*, 14(2), 86–91.

- Alita, D., & Isnain, A. R. (2020). Pendeteksian Sarkasme pada Proses Analisis Sentimen Menggunakan Random Forest Classifier. *Jurnal Komputasi*, 8(2), 50–58.
- Arrahman, R. (2022). Rancang Bangun Pintu Gerbang Otomatis Menggunakan Arduino Uno R3. *Jurnal Portal Data*, 2(2), 1–14. <http://portaldata.org/index.php/portaldata/article/view/78>
- Aziz, M., & Fauzi, A. (2022). *CNN UNTUK DETEKSI BOLA MULTI POLA STUDI KASUS : LIGA HUMANOID ROBOCUP CNN For Multi Pattern Ball Detection Case Study : RoboCup Humanoid League*. 5(1), 23–34.
- Bakri, M., & Darwis, D. (2021). *PENGUKUR TINGGI BADAN DIGITAL ULTRASONIK BERBASIS ARDUINO DENGAN LCD DAN OUTPUT*. 2, 1–14.
- Darwis, D., & Pamungkas, N. B. (2021). Comparison of Least Significant Bit, Pixel Value Differencing, and Modulus Function on Steganography to Measure Image Quality, Storage Capacity, and Robustness. *Journal of Physics: Conference Series*, 1751(1), 12039.
- Darwis, D., Prabowo, R., & Hotimah, N. (2018). Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman Untuk Meningkatkan Keamanan Data. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 5(4), 389–394.
- Darwis, D., Sulistiani, H., Isnain, A. R., Yasin, I., Hamidy, F., & Mega, E. D. (2022). *Pelatihan pengarsipan secara elektronik (e-filling) bagi perangkat desa di pekon sukanegeri jaya*. 3(1), 108–113.
- Dita, P. E. S., Al Fahrezi, A., Prasetyawan, P., & Amarudin, A. (2021). Sistem Keamanan Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 121–135.
- Hendrastuty, N., Ihza, Y., Ring Road Utara, J., & Lor, J. (2021). Rancang Bangun Aplikasi Monitoring Santri Berbasis Android. *Jdmsi*, 2(2), 21–34.
- Jamaaluddin, J., & Sumarno, S. (2017). Perencanaan Sistem Pentanahan Tenaga Listrik Terintegrasi Pada Bangunan. *JEEE-U (Journal of Electrical and Electronic Engineering-UMSIDA)*, 1(1), 29–33. <https://doi.org/10.21070/jeee-u.v1i1.375>
- Kurniati, I. D., Setiawan, R., Rohmani, A., Lahdji, A., Tajally, A., Ratnaningrum, K., Basuki, R., Reviewer, S., & Wahab, Z. (2015). *Buku Ajar Basis Data*.
- LIA FEBRIA LINA, B. P. (2019). *KREDIBILITAS SELEBRITI MIKRO PADA NIAT BELI PRODUK DI MEDIA SOSIAL*. 1(2), 41–50. *A NIAT BELI PRODUK DI MEDIA SOSIAL*. 1(2), 41–50.
- Mertania, Y., & Amelia, D. (2020). Black Skin White Mask: Hybrid Identity of the Main Character as Depicted in Tagore's The Home and The World. *Linguistics and Literature Journal*, 1(1), 7–12. <https://doi.org/10.33365/lj.v1i1.233>
- Neneng, N., Puspaningrum, A. S., & Aldino, A. A. (2021). Perbandingan Hasil Klasifikasi

- Jenis Daging Menggunakan Ekstraksi Ciri Tekstur Gray Level Co-occurrence Matrices (GLCM) Dan Local Binary Pattern (LBP). *SMATIKA JURNAL*, 11(01), 48–52.
- Nurkholis, A., & Sitanggang, I. S. (2020). Optimization for prediction model of palm oil land suitability using spatial decision tree algorithm. *Jurnal Teknologi Dan Sistem Komputer*, 8(3), 192–200. <https://doi.org/10.14710/jtsiskom.2020.13657>
- Putra, A. D., Ardiansyah, T., Latipah, D., & Hidayat, S. (2019). *Data Extraction Using The Web Crawler As A Media For Information On The Popularity Of Lampung Province Tourism For The Development Of Rides And Abstract* : 6(2).
- Reza, F., & Putra, A. D. (2021). Sistem Informasi E-Smile (Elektronik Service Mobile)(Studi Kasus: Dinas Kependudukan dan Pencatatan Sipil Kabupaten Tulang Bawang). *Jurnal Teknologi Dan Sistem Informasi*, 2(3), 56–65. <http://jim.teknokrat.ac.id/index.php/sisteminformasi/article/view/909>
- Rossi, F., Aizzuddin, A., & Rahni, A. (2018). *Joint Segmentation Methods of Tumor Delineation in PET – CT Images : A Review*. 7, 137–145.
- Rusliyawati, & Sinaga, I. (2017). Pengaruh Self-Efficacy Komputer Jurusan Sia (Studi Kasus Mahasiswa Bidang Keahlian Sia Stmik Teknokrat Lampung). *Prosiding Seminar Nasional Darmajaya*, 1(1), 56–89. <https://jurnal.darmajaya.ac.id/index.php/PSND/article/view/750%0Ahttps://jurnal.darmajaya.ac.id/index.php/PSND/article/viewFile/750/484>
- Saipulloh Fauzi1, L. F. L. (2020). PERAN FOTO PRODUK, ONLINE CUSTOMER REVIEW, ONLINE CUSTOMER RATING PADA MINAT BELI KONSUMEN. *Jurnal Muhammadiyah Manajemen Bisnis*, 1(1), 37–47. <https://jurnal.umj.ac.id/index.php/JMMB/article/view/5917>
- Samsugi, S., Neneng, N., & Aditama, B. (2018). *IoT: kendali dan otomatisasi si parmin (studi kasus peternak Desa Galih Lunik Lampung Selatan)*.
- Satria, M. N. D., & Haryadi, S. (2018). Effect of the content store size to the performance of named data networking: Case study on Palapa Ring topology. *Proceeding of 2017 11th International Conference on Telecommunication Systems Services and Applications, TSSA 2017, 2018-Janua*, 1–5. <https://doi.org/10.1109/TSSA.2017.8272911>
- Setiawan, A., & Pasha, D. (2020). Sistem Pengolahan Data Penilaian Berbasis Web Menggunakan Metode Pieces (Studi Kasus : Badan Pengembangan Sumber Daya Manusia Provinsi Lampung). *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, 1(1), 97–104. <http://jim.teknokrat.ac.id/index.php/sisteminformasi>
- Soraya, A., & Wahyudi, A. D. (2021). Rancang bangun aplikasi penjualan dimsun berbasis web. *Teknologi Dan Sistem Informasi (JTSI)*, 2(4), 43–48.
- Styawati, Andi Nurkholis, Zaenal Abidin, & Heni Sulistiani. (2021). Optimasi Parameter Support Vector Machine Berbasis Algoritma Firefly Pada Data Opini Film. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(5), 904–910.

<https://doi.org/10.29207/resti.v5i5.3380>

- Surahman, A., Wahyudi, A. D., Putra, A. D., Sintaro, S., & Pangestu, I. (2021). Perbandingan Kualitas 3D Objek Tugu Budaya Saibatin Berdasarkan Posisi Gambar Fotogrametri Jarak Dekat. *InfoTekJar : Jurnal Nasional Informatika Dan Teknologi Jaringan*, 2, 296–301.
- Susanto, E. R., Puspaningrum, A. S., & Neneng, N. (2021). Model Rekomendasi Penerima Bantuan Sosial Berdasarkan Data Kesejahteraan Rakyat. *Jurnal Tekno Kompak*, 15(1), 1–12.
- Strategi Pengembangan Bisnis Usaha Mikro Kecil Menengah Keripik Pisang Dengan Pendekatan Business Model Kanvas, 19 *Journal Management, Business, and Accounting* 320 (2020).
- Wantoro, A., Rusliyawati, R., & Wantoro, A. (2021). *Model sistem pendukung keputusan menggunakan FIS Mamdani untuk penentuan tekanan udara ban Decision support system model using FIS Mamdani for determining tire*. 9(November 2020), 56–63. <https://doi.org/10.14710/jtsiskom.2020.13776>
- Wantoro, A., Syarif, A., Berawi, K. N., Muludi, K., Sulistiyanti, S. R., Lampung, U., Komputer, I., Lampung, U., Masyarakat, K., Kedokteran, F., Lampung, U., Elektro, T., Teknik, F., Lampung, U., Lampung, U., Meneng, G., & Lampung, B. (2021). *METODE PROFILE MATCHING PADA SISTEM PAKAR MEDIS UNTUK*. 15(2), 134–145.
- Wicaksono, G. W., Mahmudah, N., Studi, P., Sipil, T., Teknik, F., & Yogyakarta, U. M. (n.d.). *Analisis Hubungan Rasio Volume dan Kapasitas Jalan dengan Potensi Kecelakaan di Ruas Jalan Kyai Mojo Yogyakarta*. d, 1–11.
- Widodo, T., Irawan, B., Prastowo, A. T., & Surahman, A. (2020). Sistem Sirkulasi Air Pada Teknik Budidaya Bioflok Menggunakan Mikrokontroler Arduino Uno R3. *Jurnal Teknik Dan Sistem Komputer*, 1(2), 1–6.