**Rabin-Karp & Knuth-Morris Pratt (KMP)**

Vincent Arianto1, Jonathan Tiong2,  Jovan Marian Winarko3, Mikhail Keene Grady4,

Informatika, Fakultas Teknologi Informasi, Universitas Pelita Harapan

1 01082220009@student.uph.edu

2 01082220017@student.uph.edu

3 01082220003@student.uph.edu

4 01082220012@student.uph.edu

**Abstract.** The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be set in 9-point font size and should be inset 1.0 cm from the right and left margins. There should be two blank (10-point) lines before and after the abstract. This document is in the required format.

**1 Introduction**

Penelitian ini membahas implementasi dan analisis performa algoritma pencocokan string menggunakan CUDA untuk komputasi paralel. Pencocokan string adalah salah satu permasalahan mendasar dalam ilmu komputer dengan berbagai aplikasi, termasuk dalam bioinformatika, keamanan siber, dan pencarian teks. Dalam bioinformatika, pencocokan string sangat penting untuk menganalisis urutan DNA, RNA, dan protein guna mengidentifikasi pola spesifik yang memiliki peran biologis. Seiring dengan bertambahnya ukuran dataset bioinformatika, efisiensi algoritma pencocokan string menjadi semakin krusial.

Metode pemrosesan sekuensial pada CPU memiliki keterbatasan dalam menangani dataset besar. Untuk meningkatkan efisiensi, pendekatan komputasi paralel menggunakan CUDA dapat dimanfaatkan guna mempercepat proses pencocokan string. Penelitian ini mengeksplorasi bagaimana pemanfaatan CUDA dalam algoritma Knuth-Morris Pratt (KMP) dan Rabin-Karp (RK) dapat meningkatkan efisiensi pencocokan string dalam analisis urutan DNA.

**1.1 Background and motivation**

String-matching adalah salah satu permasalahan fundamental dalam komputasi, terutama dalam bidang bioinformatika, keamanan siber, dan pencarian teks. Dalam bioinformatika, pencocokan string digunakan untuk menganalisis urutan DNA, RNA, dan protein guna mengidentifikasi pola spesifik yang berperan dalam fungsi biologis. Seiring dengan meningkatnya ukuran dataset bioinformatika, efisiensi algoritma pencocokan string menjadi semakin penting.

Namun, metode konvensional yang berjalan secara sekuensial pada CPU memiliki keterbatasan dalam menangani dataset berskala besar. Untuk meningkatkan efisiensi, pendekatan komputasi paralel menggunakan CUDA dapat digunakan untuk memanfaatkan kekuatan GPU dalam mempercepat proses pencocokan string. Penelitian ini mengeksplorasi bagaimana pemanfaatan CUDA dalam algoritma Knuth-Morris Pratt (KMP) dan Rabin-Karp (RK) dapat meningkatkan efisiensi pencocokan string dalam urutan DNA.

**1.2 Problem Statement**

Seiring bertambahnya jumlah data bioinformatika, metode pencocokan string konvensional sering kali menjadi tidak efisien karena keterbatasan pemrosesan sekuensial pada CPU. Beberapa algoritma pencocokan string seperti Knuth-Morris Pratt dan Rabin-Karp memiliki keunggulan dalam aspek yang berbeda, namun masih terdapat kekurangan dalam kajian yang membandingkan performa kedua algoritma tersebut dalam lingkungan pemrosesan paralel berbasis GPU.

Penelitian ini berangkat dari permasalahan mengenai bagaimana implementasi algoritma Knuth-Morris Pratt dan Rabin-Karp menggunakan CUDA dapat meningkatkan efisiensi pencocokan string pada dataset DNA. Selain itu, penelitian ini juga menyoroti sejauh mana peningkatan performa yang dapat dicapai dibandingkan dengan implementasi sekuensial, serta bagaimana penggunaan memori dan efisiensi waktu eksekusi kedua algoritma pada berbagai ukuran dataset.

**1.3 Research Objectives**

Penelitian ini bertujuan untuk:

Penelitian ini bertujuan untuk mengimplementasikan algoritma Knuth-Morris Pratt dan Rabin-Karp menggunakan CUDA dalam proses pencocokan string pada dataset DNA. Selain itu, penelitian ini juga berfokus pada analisis performa kedua algoritma, khususnya dalam hal waktu eksekusi dan penggunaan memori pada berbagai ukuran dataset. Dengan membandingkan keunggulan dan kelemahan masing-masing algoritma dalam konteks pemrosesan paralel berbasis GPU, penelitian ini diharapkan mampu memberikan gambaran yang komprehensif mengenai efektivitas pendekatan CUDA. Proses benchmarking terhadap beragam ukuran dataset dilakukan untuk mengidentifikasi batas optimal kinerja masing-masing algoritma. Secara keseluruhan, penelitian ini diharapkan dapat memberikan wawasan baru mengenai potensi akselerasi komputasi dalam bidang bioinformatika melalui pemanfaatan GPU.

**1.4 Contributions and paper organizations**

Penelitian ini memberikan kontribusi sebagai berikut:

1. Implementasi paralel dari algoritma Knuth-Morris Pratt dan Rabin-Karp menggunakan CUDA untuk pencocokan string dalam urutan DNA.
2. Analisis komparatif mengenai efisiensi kedua algoritma dalam berbagai ukuran dataset, termasuk dalam aspek waktu eksekusi dan penggunaan memori.
3. Evaluasi penggunaan GPU dalam mempercepat pencocokan string dalam konteks bioinformatika.
4. Benchmarking dengan berbagai ukuran dataset untuk memahami batas optimal masing-masing algoritma.
5. Penyajian hasil eksperimen yang dapat dijadikan referensi bagi penelitian lebih lanjut di bidang pemrosesan paralel dan optimasi algoritma string-matching berbasis GPU.

**2 CUDA and String Matching**

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

Use 10-point type for the name(s) of the author(s) and 9-point type for the address(es) and the abstract. For the main text, please use 10-point type and single-line spacing. We recommend the use of Computer Modern Roman or Times. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided.

Papers not complying with the LN style will be reformatted. This can lead to an increase in the overall number of pages. We would therefore urge you not to squash your paper.

**Headings.** Headings should be capitalized (i.e., nouns, verbs, and all other words except articles, prepositions, and conjunctions should be set with an initial capital) and should, with the exception of the title, be aligned to the left. Words joined by a hyphen are subject to a special rule. If the first word can stand alone, the second word should be capitalized. The font sizes are given in Table 1.

Here are some examples of headings: "Criteria to Disprove Context-Freeness of Collage Languages", "On Correcting the Intrusion of Tracing Non-deterministic Programs by Software", "A User-Friendly and Extendable Data Distribution System", "Multi-flip Networks: Parallelizing GenSAT", "Self-determinations of Man".

**Table 1.** Font sizes of headings. Table captions should always be positioned *above* the tables.

| Heading level | Exa | Font size and style |
| --- | --- | --- |
| Title (centered) | **Lecture Notes …** | 14 point, bold |
| 1st-level heading | **1 Introduction** | 12 point, bold |
| 2nd-level heading | **2.1 Printing Area** | 10 point, bold |
| 3rd-level heading | **Headings.** Text follows … | 10 point, bold |
| 4th-level heading | *Remark.* Text follows … | 10 point, italic |

**Lemmas, Propositions, and Theorems.** The numbers accorded to lemmas, propositions, and theorems, etc. should appear in consecutive order, starting with Lemma 1, and not, for example, with Lemma 11.

**2.1 CUDA Overview**

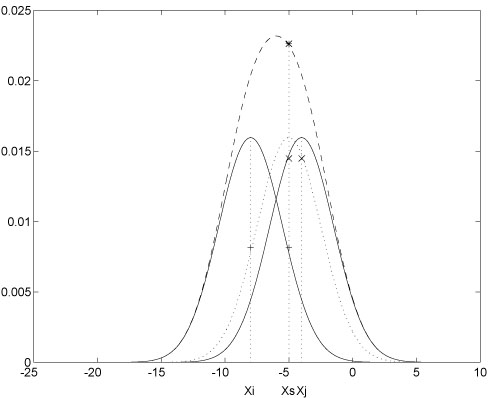
Compute unified Device Architecture (CUDA) merupakan platform komputasi paralel dan model pemrograman yang dikembangkan NVIDIA. CUDA memungkinkan pengembangan menggunakan kekuatan pemrosesan paralel dari GPU  
dengan Menggunakan CUDA aplikasi dapat dipercepat secara signifikan dengan menjalankan ribuan thread secara simultan. Dengan memanfaatkan GPU dapat membantu pengembang untuk rendering grafis dan juga untuk komputasi secara umum.

Please check that the lines in line drawings are not interrupted and have a constant width. Grids and details within the figures must be clearly legible and may not be written one on top of the other. Line drawings should have a resolution of at least 800 dpi (preferably 1200 dpi). The lettering in figures should have a height of 2 mm (10-point type). Figures should be numbered and should have a caption, which should always be positioned *under* the figures, in contrast to the caption belonging to a table, which should always appear *above* the table. Please center the captions between the margins and set them in 9-point type (Fig. 1 shows an example). The distance between text and figure should be about 8 mm, the distance between the figure and caption about 6 mm.

To ensure that the reproduction of your illustrations is of a reasonable quality, we advise against the use of shading. The contrast should be as pronounced as possible.

If screenshots are necessary, please make sure that you are happy with the print quality before you send the files.

*Remark 1.* In the printed volumes, illustrations are generally black and white (halftones), and only in exceptional cases, and if the author is prepared to cover the extra costs involved, are colored pictures accepted. Colored pictures are welcome in the electronic version free of charge. If you send colored figures that are to be printed in black and white, please make sure that they really are legible in black and white. Some colors show up very poorly when printed in black and white.



**Fig. 1.** One kernel at *xs* (*dotted kernel*) or two kernels at *xi* and *xj* (*left and right*) lead to the same summed estimate at *xs*. This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in italics, in parentheses, as shown in this sample caption.

**2.2 Rabin-Karp Algorithm**

Rabin-karp Algorithm merupakan sebuah algoritma string matching yang digunakan untuk hashing sebagai perbandingan antara string pencarian. Nilai hash diperoleh dari substring pertama hingga terakhir dalam sebuah teks dengan menggeser substring dengan panjang k ke kanan.

**2.3 Knuth-morris pratt Algorithm**

The superscript numeral used to refer to a footnote appears in the text either directly after the word to be discussed or – in relation to a phrase or a sentence – following the punctuation mark (comma, semicolon, or period). Footnotes should appear at the bottom of the normal text area, with a line of about 5cm set immediately above them[[1]](#footnote-0).

**2.4 Program Code**

Program listings or program commands in the text are normally set in typewriter font, e.g., CMTT10 or Courier.

Example of a Computer Program from Jensen K., Wirth N. (1991) Pascal user manual and report. Springer, New York

program Inflation (Output)  
 {Assuming annual inflation rates of 7%, 8%, and  
 10%,... years};  
 const MaxYears = 10;  
 var Year: 0..MaxYears;  
 Factor1, Factor2, Factor3: Real;  
 begin  
 Year := 0;  
 Factor1 := 1.0; Factor2 := 1.0; Factor3 := 1.0;  
 WriteLn('Year 7% 8% 10%'); WriteLn;  
 repeat  
 Year := Year + 1;  
 Factor1 := Factor1 \* 1.07;  
 Factor2 := Factor2 \* 1.08;  
 Factor3 := Factor3 \* 1.10;  
 WriteLn(Year:5,Factor1:7:3,Factor2:7:3,  
 Factor3:7:3)  
 until Year = MaxYears  
end.

**2.5 Citations**

The list of references is headed “References” and is not assigned a number. The list should be set in small print and placed at the end of your contribution, in front of the appendix, if one exists. Please do not insert a pagebreak before the list of references if the page is not completely filled. An example is given at the end of this information sheet. For citations in the text please use square brackets and consecutive numbers: [1], [2], [3], etc.

**2.6 Page Numbering and Running Heads**

Please do not set running heads or page numbers.

**Acknowledgments.** The heading should be treated as a 3rd level heading and should not be assigned a number.

**3 Methodology**

**4 Result and Analysis**

**5 Conclusion**

**References**

1. **https://www.researchgate.net/publication/338501840\_Comparative\_Study\_of\_String\_Matching\_Algorithms\_for\_DNA\_dataset**

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121

2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438

3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)

4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)

**Appendix: Checklist of Items to be Sent to Conference Proceedings Editors (see instructions at conference webpage)**

1. A final Word file
2. A final PDF file
3. A copyright form, signed by one author on behalf of all the authors of the paper. Digital Signature is valid.
4. A readme giving the first name(s) and the surname(s) of all the authors of the paper as well as the name and email address of the corresponding author

1. The footnote numeral is set flush left and the text follows with the usual word spacing. [↑](#footnote-ref-0)