

C12_Agus

by Agus Maman Abadi

Submission date: 28-Feb-2020 08:14AM (UTC+0700)

Submission ID: 1265618418

File name: C12_Agus.pdf (623.48K)

Word count: 2566

Character count: 14422

TEKNIK REGENERATING CODE PADA PENYIMPANAN DISTRIBUSI

REGENERATING CODE TECHNIQUE IN DISTRIBUTED STORAGE

Agus Maman Abadi*, Karyati, Musthofa dan Emut

25

Jurusan Pendidikan Matematika, Fakultas MIPA, Universitas Negeri Yogyakarta

*email: agusmaman@uny.ac.id

Diterima 15 Januari 2016 disetujui 4 Maret 2016

Abstrak

Peningkatan kebutuhan akan sebuah cara untuk menyimpan data dalam jumlah yang besar menghadirkan sebuah tantangan baru. Salah satu cara untuk menghadapi tantangan ini adalah menggunakan sistem penyimpanan data terdistribusi (*Distributed Storage Systems*). Salah satu strategi yang diterapkan dalam sistem penyimpanan data terdistribusi adalah menggunakan teknik *regenerating code*. Kode yang digunakan pada teknik ini didasarkan pada struktur aljabar berupa field. Beberapa penelitian juga telah dilakukan untuk membuat kode yang didasarkan pada struktur aljabar lain, yaitu modul. Pada penelitian ini dicoba untuk mengkaji penerapan kode modul pada teknik *regenerating code*. Hasil pengkajian menunjukkan ada potensi yang dapat dimanfaatkan dari kode atas modul pada teknik *regenerating code*.

Kata kunci: penyimpanan data terdistribusi, teknik *regenerating code*, kode modul

Abstract

The Increasing need of storing large amounts of data presents a new challenge. One way to address this challenge is to use distributed data storage system. One of the strategies implemented in the distributed data storage system is using the technique of regenerating code. The code used in this technique is based on the algebraic structure of fields. Some studies have also been carried out to create code that is based on the other algebraic structure namely module. In this study, we attempted to assess the implementation of the code module at regenerating technique code. The study showed there is a potential properties code based on module that can be used in regenerating code technique.

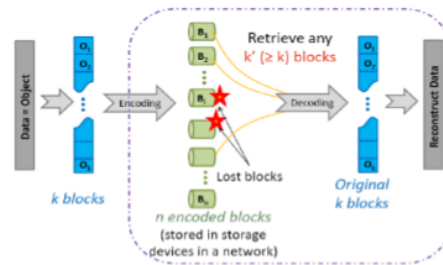
Keywords: Distributed storage, regenerating code technique, module code

Pendahuluan

Perkembangan teknologi informasi khususnya dalam hal penggunaan data berukuran besar seperti video, gambar dan jejaring sosial telah melahirkan tantangan baru, yaitu bagaimana strategi untuk mengelola data yang sedemikian besar tersebut. Selain memanfaatkan teknologi media penyimpanan seperti harddisk dan DVD, penyimpanan data dalam jumlah besar dilakukan dengan teknik *distributed storage system* (DSS).

Melalui DSS, data dipecah-pecah dan kemudian disimpan dalam sistem penyimpanan terdistribusi yang terkoneksi melalui sistem jaringan komunikasi. Masalah yang terjadi dalam DSS adalah selalu terjadi bagian penyimpan data (*node*) yang gagal atau error. Oleh karena itu proses perbaikan data secara sistematis menjadi perhatian

utama. Skema sistem DSS disajikan dalam Gambar 1 [1].



Gambar1. Sistem Penyimpanan terdistribusi (DSS)

Berdasarkan penelitian yang dilakukan peneliti sebelumnya [11], struktur aljabar yang lebih umum seperti modul dan semimodul. Namun dalam penelitian tersebut belum dilakukan uji coba menggunakan teknik pengkodean seperti regenerating code. Seperti dibahas dalam [10], salah satu teknik pengkodean yang digunakan dalam komunikasi data secara terdistribusi melalui jaringan internet adalah *regenerating code*. Beberapa penelitian yang telah dilakukan, menunjukkan bahwa dengan teknik ini bandwidth yang digunakan lebih optimal.

Penyimpanan Data Terdistribusi

Meningkatnya penggunaan media penyimpanan data seiring dengan meningkatnya penggunaan email, foto, video dan data-data berukuran besar lainnya membutuhkan solusi yang tidak mudah. Jika data-data tersebut disimpan dalam satu tempat, maka akan sangat berbahaya dikarenakan media penyimpanan dapat rusak sehingga mengakibatkan hilangnya data.

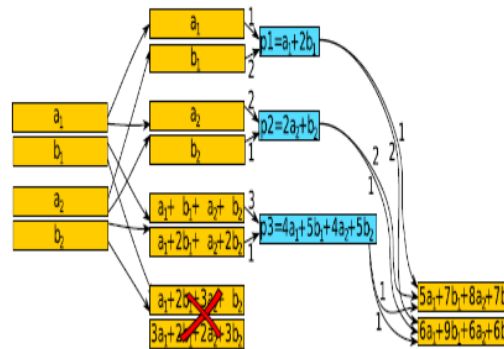
Solusi yang dapat dilakukan untuk mengatasi hal tersebut antara lain [6]: menyimpan data dengan menyediakan cadangan data dalam banyak media penyimpan (disk). Jika salah satu disk mengalami kerusakan, maka tinggal mengganti disk tersebut dengan yang baru sehingga data bisa terselamatkan. Teknik ini dinamakan sistem penyimpanan terdistribusi (*distributed storage system* (DSS)).

Namun, masalah yang terjadi tidaklah sesederhana ini. Pusat data yang akan menyimpan data dalam jumlah besar membutuhkan banyak sekali piranti penyimpan. Akibatnya kerusakan pada piranti penyimpan menjadi hal yang tak terhindarkan yang dapat menimbulkan masalah baru. Oleh karena itu diperlukan strategi dalam mengelola penyimpanan data.

Network Coding

Network coding adalah suatu pendekatan untuk meningkatkan efisiensi dari proses komunikasi [2]. *Network coding* adalah suatu bagian (node) dalam proses komunikasi yang terletak di antara sumber data(*source*) dan penerima (*receiver*) yang selain mampu menyimpan dan meneruskan data, juga mampu mengkombinasikan secara independen data yang masuk menjadi data yang siap dikirim.

Melalui *network coding*, bagian penyimpan data yang rusak dalam sistem DSS, diperbaiki dengan langkah-langkah tertentu. Skema dari *network coding* dalam [4] adalah sebagai berikut:



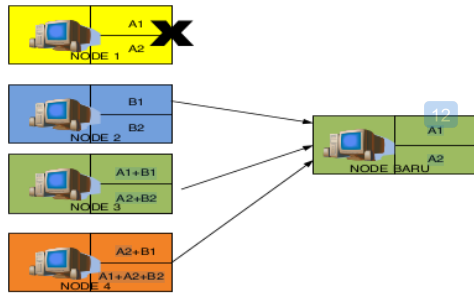
Penggunaan *Network Coding* dapat meningkatkan meningkatkan *Troughput* , yaitu kapasitas informasi yang dialirkan melalui jaringan yang diukur dengan satuan waktu tertentu dan kondisi jaringan tertentu.

Erasure Code

Dalam proses komunikasi melalui jaringan atau chanel, seringkali terjadi beberapa error atau gagalnya suatu proses pengiriman data. Suatu teknik pengkodean yang berkaitan dengan komunikasi melalui jaringan atau chanel yang bersifat demikian dinamakan *erasure code*. Dengan teknik *erasure code*, ketika salah satu pihak akan mengakses suatu data tertentu melalui jaringan yang dapat mengalami gagal akses, data tersebut dipecah pecah dalam k bagian. Setiap k bagian *encoding*, kemudian disimpan secara terdistribusi pada n node. Untuk mengakses data asli dapat dilakukan dengan mengakses sebarang k node.

Ketika terjadi kegagalan atau error dalam proses rekonstruksi data, maka salah satu strategi yang dapat dilakukan adalah menggunakan *MDS code*, yaitu sejumlah misalnya terjadi gagal akses pada node 1, maka dibuat node baru yang dikonstruksi dari node yang lain. Menggunakan cara ini, bandwidth yang dibutuhkan dalam proses rekonstruksi data berukuran sama dengan ukuran data semula.

Oleh karena itu dalam [10], diperkenalkan teknik *regenerating code* yang bisa mengurangi kebutuhan bandwidth dalam proses rekonstruksi data. Secara umum proses rekonstruksi data dalam teknik ini dilakukan dengan mengakses node lain yang tidak mengalami error untuk kemudian dihasilkan suatu node berisi kode yang sama dengan node yang mengalami gagal akses. Dalam [5] hal ini dapat diilustrasikan sebagai berikut:



15
Metode Penelitian

Dalam penelitian ini digunakan metode penelitian studi literatur berupa jurnal-jurnal ilmiah yang terkait dengan topik penelitian, dan buku-buku referensi yang mendukung. Pada tahap awal dipelajari konsep-konsep dasar tentang *distributed storage system* (DSS) dan *network coding* serta pengkonstruksian kode menggunakan *MDS Code* dan *Regenerating Code* yang didasarkan ada struktur aljabar berupa field . Konsep-konsep ini nantinya digunakan sebagai dasar untuk mengkontruksi kode pada struktur aljabar yang lebih umum, yaitu pada modul.

Selanjutnya, dipelajari beberapa teknik pengkodean menggunakan modul. Langkah berikutnya adalah mempelajari penerapan kode yang didasarkan atas modul apakah dapat diterapkan untuk melakukan rekontruksi data dengan teknik *regenerating code*.

Hasil dan Pembahasan

Teknik Regenerating Code atas Field.

Berikut adalah skema dari teknik regenerating code dengan teknik *exact repair* yang didasarkan pada struktur aljabar atas lapangan hingga F_q . Suatu file dengan ukuran B dalam penyimpanan terdistribusi dipecah-pecah dan disimpan dalam n buah *node* dengan ketentuan setiap node dapat menyimpan sejumlah α . Suatu data collector dapat terkoneksi dengan setiap node, dan kemudian merekonstruksi file semula. Untuk merekonstruksi file, data collector cukup terhubung ke sejumlah $k < n$ node yang ada. Ketika misalnya ada node yang error, maka data collector akan terhubung ke sejumlah $d > k$ dari $n-1$ node yang tersisa untuk membentuk node baru menggantikan node yang error. Pada teknik regenerating code, secara umum node yang baru tidak harus sama persis dengan node yang error, tetapi memiliki sifat yang sama, yaitu untuk merekonstruksi data semula, cukup

dibentuk dari sejumlah $k < n$ node yang ada. Dalam [Rashmi], telah dikenalkan suatu teknik *exact repair*, yaitu ketika membentuk node yang baru pada saat ada node yang error, maka node yang baru tersebut identik dengan node yang error tersebut.

Secara umum teknik regenerating code untuk meminimalkan penyimpanan (minimum storage regeneration) yang didasarkan atas struktur aljabar lapangan hingga adalah sebagai berikut.

- a) File atau objek $O = (o_1, o_2, \dots, o_B) \in F_q^B$ dipecah menjadi 2, yaitu $O_1 = (o_1, o_2, \dots, o_k)$ dan $O_2 = (o_{k+1}, o_{k+2}, \dots, o_B)$. Dalam hal ini B merupakan bilangan genap sehingga O_1 dan O_2 mempunyai ukuran yang sama.
- b) Node ke- i akan menyimpan $\alpha = (O_1 p_i^T, O_2 p_i^T + O_1 v_i^T)$, dengan p_i dan v_i merupakan vektor baris yang digunakan untuk melakukan encoding, $i = 1, 2, \dots, n$. Lebih khusus, p_i^T merupakan vektor yang merupakan generator untuk G dari suatu (n,k) *MDS code*, yaitu $G' = [p_1^T \dots p_n^T]$
- c) Secara keseluruhan kode yang dihasilkan adalah $(2n, 2k)$ kode yang disajikan sebagai $[O_1, O_2] \begin{bmatrix} G' & V \\ 0 & G' \end{bmatrix}$ dengan $V = [v_1^T \dots v_n^T]$ adalah suatu matriks.
- d) Untuk merekonstruksi data/file, data collector mendownload α dari k buah node. Karena G' merupakan generator untuk (n,k) *MDS code*, maka diperoleh O_1 . Karena O_1 dan v_i diketahui, maka data collector tinggal menyisakan $O_2 p_i^T$. Karena G' generator, maka dapat diperoleh O_2 .
- e) Misalkan node ke- j error, yaitu $(O_1 p_j^T, O_2 p_j^T + O_1 v_j^T)$ tidak dapat diakses. Node baru kemudian mendownload $\beta=1$ simbol dari $d = k+1$ buah node, yaitu $w_i = a_i (O_1 p_i^T) + O_2 p_i^T + O_1 v_i^T$, dengan $a_i \in F_q$. Node baru menghitung 2 simbol dari d symbol yang didownload, dengan membuat kombinasi linear dari $(\sum_{i=1}^d \delta_i w_i, \sum_{i=1}^d \rho_i w_i)$ dengan δ_i dan ρ_i dipilih sedemikian sehingga memenuhi $\sum_{i=1}^d \rho_i p_i^T = p_j^T$, dan $\sum_{i=1}^d \delta_i p_i^T = 0$.

Contoh :

Untuk $n = 5, k = 3, d = k+1 = 4$, maka $B = 2k = 6$.
Objek/file = $(o_1, \dots, o_6) \in F_q^6$.
Dipilih $q = 8$, jadi $F_q = F_8 = \{ 0, 1, w, w+1, w^2, w^2 + 1, w^2+w, w^2+w + 1 \}$ dan $w^3 = w+1$.
Diperoleh enkodingnya adalah :

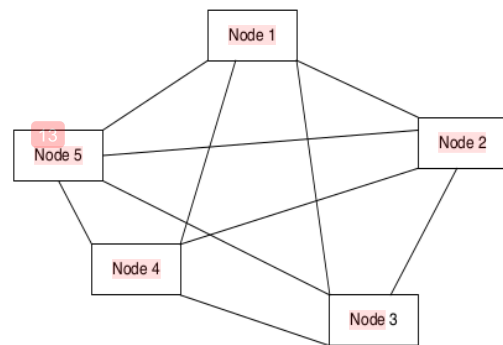
$$\begin{bmatrix} o_1 & & o_2 & & o_3 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & w & w+1 & w^2 & w^2+1 \\ 1 & w^2 & (w+1)^2 & w^4 & (w^2+1)^2 \end{bmatrix}$$

Suatu teknik konstruksi yang lain adalah sebagaimana yang telah dibahas dalam [12] adalah sebagai berikut.

- 1) File berukuran B, dipecah pecah ke dalam bagian-bagian (f_0, f_1, \dots, f_{B-1}).
- 2) Didefinisikan $d = n-1$ dan matriks V berukuran $n \times \theta$ dengan $\theta = \frac{1}{2} d (d+1)$ dan memenuhi:
 - (i) Elemen dari V adalah 0 atau 1;
 - (ii) Setiap baris mempunyai elemen 1 tepat sebanyak d;
 - (iii) Setiap kolom mempunyai elemen 1 tepat sebanyak dua;
 - (iv) Sebarang dua baris, mempunyai irisan 1, tepat sebanyak 1.

Berdasarkan ketentuan tersebut, matriks V merupakan graf matriks insidensi dari graf tak berarah dengan n titik. Himpunan vektor $\{v_1, v_2, \dots, v_\theta\}$ membentuk MDS code dengan dimensi B. elemen-elemen dari setiap v_i merupakan elemen dari lapangan hingga F_q . Node j menyimpan $f^j(v_i)$ jika dan hanya jika $V(j,i) = 1$.

Contoh : Misal $n = 5, k = 3$. Diperoleh $d = n - 1 = 4$ dan $\theta = \frac{1}{2} \times 4 \times 5 = 10$.



Gambar 3 Skema encoding

Matriks V merupakan matriks insidensi dari graf tak berarah pada gambar 3 di atas, yaitu :

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
n1	1	1	1	1	0	0	0	0	0	0
n2	1	0	0	0	1	1	1	0	0	0
n3	0	1	0	0	1	0	0	1	1	0
n4	0	0	1	0	0	1	0	1	0	1
n5	0	0	0	1	0	0	1	0	1	1

Berdasarkan teknik konstruksi di atas, maka tiap node akan menyimpan :

Node 1: $\{f^1v1; f^1v2; f^1v3; f^1v4\}$

Node 2: $\{f^1v1; f^1v5; f^1v6; f^1v7\}$

Node 3: $\{f^1v2; f^1v5; f^1v8; f^1v9\}$

Node 4: $\{f^1v3; f^1v6; f^1v8; f^1v10\}$

Node 5: $\{f^1v4; f^1v7; f^1v9; f^1v10\}$

Untuk proses rekonstruksi data, misal data collector menghubungi ke node 1, 2 dan 3. Dari ketiga node tersebut akan diperoleh $f^1v1, f^1v2, \dots, f^1v9$. Demikian pula jika misalnya node 2 bermasalah, maka node 1 akan memberikan informasi f^1v1 , node 3 akan memberikan informasi f^1v5 , node 4 akan memberikan informasi f^1v6 , dan node 4 akan memberikan informasi f^1v7 .

Teknik Regenerating Code atas Modul

Pada bagian ini akan dikaji tentang pemanfaatan struktur aljabar berupa modul pada teknik regenerating code. Di awal bagian ini akan terlebih dahulu dibahas tentang kode modul seperti yang dibahas dalam [2].

Definisi 1

Misalkan M_1 dan M_2 adalah suatu modul atas ring R dan $E: M_1 \rightarrow M_2$ adalah suatu fungsi encoding. Suatu kode $C = E(M_1)$ dinamakan kode modul jika $E(M_1)$ merupakan submodul dari M_2 .

Contoh 2.

Diberikan modul $B^2 = \{00, 01, 10, 11\}$ dan $B^5 = \{00000, 10000, 10001, 10011, 10010, 10101, 10110, 10111, 11000, 11001, 11010, 11011, 11100, 11100, 11101, 11110, 00001, 00011, 00010, 00100, 00110, 00111, 01000, 01001, 01010, 01011, 01100, 01101, 01110, 01110, 01111, 11111\}$.

terhadap operasi penjumlahan modulo 2 dan pergandaan skalar atas $B = \{0,1\}$.

Selanjutnya didefinisikan fungsi encoding $E: B^2 \rightarrow B^5$ sebagai :

$$E(00) = 00000$$

$$E(01) = 01110$$

$$E(10) = 10101$$

$$E(11) = 11011$$

$E(B^2)$ merupakan suatu submodul dari B^5 sebab :

- 1) Tertutup terhadap penjumlahan, yaitu:

$$01110 + 10101 = 11011$$

$$01110 + 11011 = 10101$$

$$10101 + 11011 = 01110$$

$$01110 + 01110 = 00000$$

$$10101 + 10101 = 00000$$

$$11011 + 11011 = 00000$$

$$11011 + 11011 = 00000$$

2) Tertutup terhadap perkalian skalar $\{0, 1\}$, yaitu

$$0 \times (10101) = 00000 \text{ dan}$$

$$1 \times (10101) = 10101$$

Jadi, $E(B^2) = \{00000, 01110, 10101, 11011\}$ di atas adalah kode modul sebab merupakan submodul dari B^5 .

Contoh 3

Diberikan B^2 dan B^4 seperti pada contoh 4.2. Didefinisikan fungsi enkoding $E : B^2 \rightarrow B^4$ sebagai berikut :

$$E(00) = 0000$$

$$E(01) = 0101$$

$$E(10) = 1010$$

$$E(11) = 1111$$

$E(B^2) = \{0000, 0101, 1010, 1111\}$ merupakan submodul dari B^4 sebab:

1) Tertutup terhadap penjumlahan, yaitu:

$$0101 + 1010 = 1111$$

$$1010 + 1111 = 0101$$

$$0101 + 1111 = 1010$$

$$0101 + 0101 = 0000$$

$$1010 + 1010 = 0000$$

$$1111 + 1111 = 0000$$

2) Tertutup terhadap perkalian skalar $B = \{0, 1\}$, yaitu :

$$1 \times (1010) = 1010$$

$$0 \times (1010) = 0000$$

Jadi, $E(B^2) = \{0000, 1010, 0101, 1111\}$ merupakan kode modul.

Selanjutnya, gagasan untuk menerapkan kode modul pada teknik regenerating kode adalah sebagai berikut:

1) File dipecah-pecah menjadi, misalnya $\{00, 01, 10, 11\} \in B^2$. Selanjutnya file tersebut disimpan dalam 4 node $\{0000, 0101, 1010, 1111\} \in B^4$.

2) Untuk proses rekonstruksi file, data collector mengontak sebarang tiga node tersebut, misalnya node 1, 2 dan 3, maka data pada node ke 4 akan diperoleh dari node 2 dan 3.

Bila ada node yang bermasalah, misalnya node 2 (0101), maka node 2 dapat direkonstruksi dengan mengambil informasi dari node 3 dan 4, yaitu $1010 + 1111 = 0101$ (node 2).

Simpulan

Penelitian ini mencoba suatu gagasan untuk menerapkan kode modul pada teknik regenerating code untuk penyimpanan terdistribusi. Hasil yang diperoleh melalui pengkajian menunjukkan ada

sebuah potensi untuk mengimplementasikan kode atas modul pada teknik regenerating code. Hal ini perlu terus dikembangkan dengan menyelidiki sifat yang lebih lanjut pada kode atas modul, aspek kepraktisan dan kemudahan dalam teknik komputasi.

19

Ucapan Terima Kasih

Tim Peneliti mengucapkan terimakasih kepada Universitas Negeri Yogyakarta, khususnya Fakultas MIPA yang telah mendanai kegiatan penelitian ini.

Pustaka

- [1] F. Oggier. 2013, *On Coding Techniques for Networked Distributed Storage Systems*. First European Training School on Network Coding, Barcelona February.
- [2] F.Oggier., A. Datta. 2012 *Coding Techniques for Repairability in Networked Distributed Storage Systems*.
- [3] T. Chareonvisal. 2012 *Implementing Distributed Storage System by Network Coding in Presence of Link Failure. Master Thesis Report*.
- [4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, 2010, Network coding for distributed storage systems," *IEEE Trans. on Info. Theory*, Sep.
- [5] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Su, 2010, A Survey on Network Codes for Distributed Storage, *The Proceedings of the IEEE*, vol. 99, pp.476–489
- [6] V. Venkatesan, 2011, *Distributed storage systems*, IBM Zurich Research Lab.EPFL
- [7] C. Fragouli and E. Soljanin, 2007, *Network coding fundamentals*, Now Publishers, Jun.
- [8] R.Dougherty, C.Freiling, dan K. Zeger. 2005, Insufficiency of Linear Coding in Network Information Flow. *IEEE transactions on information theory*, vol. 51, no. 8, August
- [9] M. Gadouleau, 2009, Algebraic Codes for Random Linear Network Coding. *P.hD dissertation*, Lehigh University.
- [10] K. V. Rashmi , N. B. Shah , K. Ramchandran and P. V. Kumar, 2012, Regenerating codes for errors andasures in distributed storage, Proc. IEEE ISIT, pp.1202 -1206
- [11] Agus Maman Abadi, Musthofa, Emut. (2014)Studi Awal Penerapan Aljabar Max Plus pada Sistem Penyimpanan Terdistribusi (Distributed Storage System) Melalui

Network Coding. *Penelitian DIPA FMIPA UNY.*

- [12] K. V. Rashmi, N. B. Shah, P. Vijay Kumar, K. Ramchandran, 2009, *Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage*, Allerton

ORIGINALITY REPORT

17%

SIMILARITY INDEX

13%

INTERNET SOURCES

11%

PUBLICATIONS

11%

STUDENT PAPERS

PRIMARY SOURCES

1	people.eecs.berkeley.edu Internet Source	3%
2	diva-portal.org Internet Source	2%
3	sands.sce.ntu.edu.sg Internet Source	1%
4	kth.diva-portal.org Internet Source	1%
5	Submitted to Queen Mary and Westfield College Student Paper	1%
6	Ankit Singh Rawat, Onur Ozan Koyluoglu, Natalia Silberstein, Sriram Vishwanath. "Optimal Locally Repairable and Secure Codes for Distributed Storage Systems", IEEE Transactions on Information Theory, 2014 Publication	1%
7	Submitted to Universitas Sebelas Maret Student Paper	1%

8

Internet Source

<1%

9

Y. L. BORISSOV. "Fast Decoding of the p-Ary First-Order Reed-Muller Codes Based On Jacket Transform", IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, 03/01/2008

Publication

<1%

10

Kenneth W. Shum, Yuchong Hu. "Cooperative Regenerating Codes", IEEE Transactions on Information Theory, 2013

Publication

<1%

11

Submitted to Universitas Jember

Student Paper

<1%

12

Submitted to University of Melbourne

Student Paper

<1%

13

etd.lib.nsysu.edu.tw

Internet Source

<1%

14

Shuang-Hong Yang. "Reformulated Parametric Learning Based on Ordinary Differential Equations", Lecture Notes in Computer Science, 2006

Publication

<1%

15

journal.widyadharma.ac.id

Internet Source

<1%

16	Submitted to Universitas Mercu Buana Student Paper	<1%
17	greget13.blogspot.com Internet Source	<1%
18	www.webriche.fr Internet Source	<1%
19	docobook.com Internet Source	<1%
20	eprints.ulm.ac.id Internet Source	<1%
21	jhp.ui.ac.id Internet Source	<1%
22	hanglekiumc.com Internet Source	<1%
23	Submitted to University of Southampton Student Paper	<1%
24	www.scribd.com Internet Source	<1%
25	adoc.tips Internet Source	<1%
26	Jin Soo Park, Jung-Hyun Kim, Ki-Hyeon Park, Hong-Yeop Song. "Average Repair Read Cost of Linear Repairable Code Ensembles", The Journal of Korea Information and	<1%

Communications Society, 2014

Publication

Exclude quotes Off

Exclude matches Off

Exclude bibliography On

C12_Agus

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6
