

Pengembangan *Tiling database* untuk Penyimpanan Data Penginderaan Jauh pada Pembangunan LAPAN *Engine*

(Database Tiling Development for Remote Sensing Data Storage in LAPAN Engine Construction)

Ayom Widipaminto^{1a}, Yuvita Dian Safitri^{1a}, Wisnu Sunarmodo^{1a}, Rokhmatullah^{2b}

¹Pusat Teknologi dan Data Penginderaan Jauh, LAPAN

²FMIPA, Universitas Indonesia

e-mail: ayom.widipaminto@lapan.go.id

^aKontributor Utama, ^bKontributor anggota

Diterima: 24 Juni 2019; Direvisi: 27 Juni 2019; Disetujui: 8 Juli 2019

ABSTRACT

Remote sensing image data is included in the unstructured data category which is characterized by large volumes of data and is regularly updated. Special techniques are needed in large capacity data storage and supported by high-capacity data processing machines. This study aims to find a design representation of remote sensing image data that is more efficient in storage and processing than conventional methods. The design proposed is with the concept of tiling databases, namely the method of breaking down image data into small size pieces with certain identities and then entering them into a database. The test results compared to the conventional method found that the storage volume can be reduced by up to 25%, the speed of reading the data also increases by about 21%. This system can support the development of LAPAN Engine because it offers a storage strategy that is more effective in terms of volume, and efficient in terms of the speed of reading data even though the tiling process into the database takes pretty long time.

Keywords: tiling, database, storage, image, remote sensing

ABSTRAK

Data citra penginderaan jauh termasuk dalam kategori data *unstructured* yang dicirikan dengan volume data besar dan diperbaharui secara berkala. Diperlukan teknik khusus dalam penyimpanan data berkapasitas besar serta didukung mesin pengolah data berkemampuan tinggi. Penelitian ini bertujuan untuk menemukan desain representasi data citra penginderaan jauh yang lebih efisien dalam penyimpanan dan pengolahan dibanding metode konvensional yang digunakan saat ini. Desain yang diajukan menerapkan konsep *tiling database*, yaitu metode memecah data citra menjadi potongan-potongan berukuran kecil dengan identitas tertentu, kemudian memasukkannya ke dalam *database*. Hasil pengujian metode *tiling database* dibanding metode konvensional didapatkan bahwa volume penyimpanan dapat ditekan hingga 25%, dan kecepatan pembacaan data meningkat sekitar 21%. Sistem ini mampu mendukung pengembangan LAPAN *Engine* karena menawarkan strategi penyimpanan yang lebih efektif dari segi volume, dan efisien dalam segi kecepatan pembacaan data, meskipun proses *tiling* ke dalam *database* memerlukan waktu yang cukup lama.

Kata Kunci : *tiling, database, penyimpanan, citra, penginderaan jauh*

1 PENDAHULUAN

Pertumbuhan volume data penginderaan jauh yang terus meningkat disertai kehadiran satelit-satelit baru yang terus bermunculan menuntut sistem penyediaan yang lebih cepat, handal dan akurat. Pemanfaatan data penginderaan jauh semakin beragam, seiring semakin banyak tersedianya jenis sensor baru. Perkembangan tersebut membuat aplikasi data penginderaan jauh berkembang pesat, dan dimanfaatkan untuk kemaslahatan yang lebih luas.

Namun, bertambahnya jumlah data primer maupun data proses dari satelit juga menimbulkan tantangan baru. Jumlah data citra yang tidak terstruktur (*unstructured*) terus meningkat dan menyebabkan beban penyimpanan juga meningkat. Hal ini berimbas pada meningkatnya beban pengolahan data, terutama proses untuk ekstraksi informasi yang bersifat *multi-temporal*, karena banyak dan besarnya data yang diolah (Bargellini *et al.*, 2013). Tidak hanya persoalan penyimpanan, banyaknya data juga menimbulkan masalah dalam manajemen penyimpanan file, pencarian data, serta menampilkan data yang dibutuhkan (Jewell *et al.*, 2014). Sehingga permasalahan yang muncul sekarang bukanlah lagi bagaimana untuk mendapatkan data, tetapi bagaimana untuk dapat mengeksploitasi

data citra penginderaan jauh yang ada (Lee & Kang, 2015).

Diperlukan strategi untuk manajemen penyimpanan data citra yang tidak hanya lebih efisien namun juga dapat mempercepat proses pencarian, pengolahan, dan visualisasi data citra. Penelitian ini mengevaluasi dan mengusulkan sebuah sistem yang dinamakan LAPAN *Engine*. Sistem ini memungkinkan pengguna untuk mengakses dan mengolah data penginderaan jauh yang diakuisisi oleh LAPAN secara *online*. LAPAN *Engine* menawarkan akses volume dan kecepatan yang lebih baik daripada pengolahan di perangkat pribadi. Pengguna tidak perlu lagi menyimpan data mentah yang dapat menghabiskan kapasitas penyimpanan. Pengguna cukup memilih data olah citra apa yang dibutuhkan, dan sistem akan secara otomatis menyiapkannya. Sistem ini nantinya diharapkan mampu menyediakan dan membantu pengolahan data dengan lebih efisien.

Pengguna dapat bekerja bersama dan berkontribusi secara kolaboratif dalam infrastruktur *client-server online* untuk mengekstrak informasi data observasi bumi dengan volume sangat besar yang disimpan dalam bentuk *3D spatio-temporal data cubes* (Sudmanns, *et al.*, 2018). LAPAN *Engine* memiliki tahap-tahap pengembangan, mulai dari penyimpanan data citra, pembuatan sistem pengolahan data citra

penginderaan jauh, sampai pembangunan *Application Program Interface* (API) untuk menghubungkan *user* pada sistem penyimpanan dan pengolahan.

Fokus pembahasan makalah ini ada di tahap pertama strategi penyimpanan data citra penginderaan jauh menggunakan *tiling database*. *Tiling* data didasarkan pada *zoom level* yang dimiliki tiap resolusi spasial citra. Disebut *tiling database* karena penyimpanan data citra setelah proses *tiling* pada *LAPAN Engine* disimpan pada *database* jenis NoSQL (*not only SQL*). NoSQL berguna untuk menyimpan data tidak terstruktur yang berkembang jauh lebih cepat daripada data terstruktur dan tidak sesuai dengan skema relasional yang bergantung pada tabel, baris, dan kolom (Basho, 2018). Hasil *tiling* disimpan dalam bentuk *data cube* multidimensi berupa dimensi *x*, *y*, dan *temporal*. Program *tiling* citra sendiri dibuat menggunakan bahasa pemrograman python dengan memanfaatkan GDAL (*Geospatial Data Abstraction Library*) yang mampu memanipulasi data *raster* geospasial (Butler, 2018).

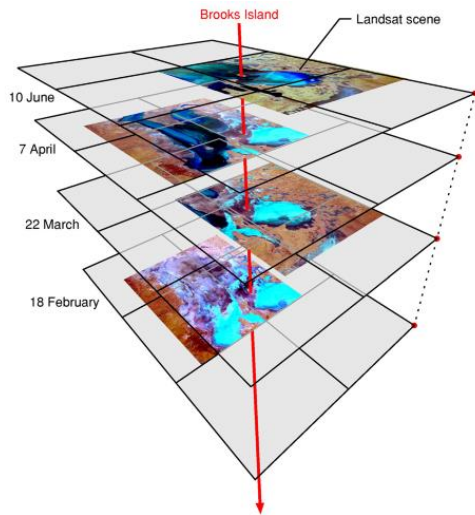
The Australian Geoscience Data cube, menggunakan *scene* Landsat yang di format ulang sebagai *data cube*, konsisten secara spasial namun berubah secara *temporal*. (Lewis et al., 2017). Penelitian sebelumnya telah membahas pemanfaatan pemodelan *data cube* untuk metode kompresi dan rekonstruksi *hyperspectral image* berdasarkan pendekatan pemrosesan data multi-dimensi atau tensor (Zhang, et al., 2015). Pemodelan *data cube* juga dimanfaatkan untuk analisis data penginderaan jauh secara *hyperspectral* (Bioucas-dias et al., 2013).

Makalah dari Lewis et al., (2017), Zhang, Zhang, Tao, Huang, & Du, (2015), dan Bioucas-dias et al., (2013) memanfaatkan *data cube* dalam penelitiannya, namun belum menggunakan *tiling* berdasarkan *zoom level* spasial datanya. Penyimpanan *tiling* pun bisa diasumsikan masih menggunakan penyimpanan konvensional, meskipun ada pula yang sudah menggunakan *database SQL* (*Structured Query Language*). Penelitian ini menerapkan strategi penyimpanan dengan memanfaatkan *database* NoSQL untuk menyimpan data citra penginderaan jauh multidimensi dalam bentuk *array* yang sudah di *tiling* berdasarkan *zoom level* sebagai unsur kebaruannya. Strategi penyimpanan data citra penginderaan jauh ini menggunakan data citra Landsat-8 untuk mengetahui secara jelas perbedaan yang dihasilkan dengan penyimpanan konvensional.

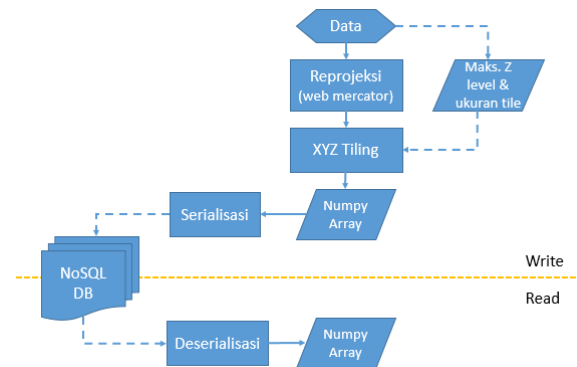
2 METODOLOGI

Metode yang digunakan dalam pengembangan sistem ini adalah eksperimen *prototype* dengan *data cube* sebagai wadah *tiling database*. Pengembangan penyimpanan dengan *data cube* dilakukan pada data citra Landsat-8 *multi-temporal* dengan pemodelan seperti tampak pada Gambar 2-1. Adapun skema diagram alir penyimpanan dan pengambilan data seperti pada Gambar 2-2.

Diagram alir pemrosesan data terbagi menjadi dua tahap, yaitu *write* dan *read*. Pada tahapan *write*, data citra diproyeksikan menggunakan reprojeksi *web Mercator*. Kumpulan data kemudian dipetakan pada maksimum *zoom level* (*Z*) untuk semua citra input beserta ukuran *tile* yang kemudian disimpan pada *database*.



Gambar 2-1: Pemodelan *data cube* dari *The Australian Geoscience Data cube*. Area yang ditampilkan adalah Pulau Brooks di Danau Eyre, Australia Tengah, pada tahun 2009. (Lewis *et al.*, 2017)



Gambar 2-2: Diagram alir proses penyimpanan dan pengambilan data.

Proses berlanjut pada *tiling*, dengan mengubah data per *tile raster* menjadi *numpy array* menggunakan program Python. Data kemudian di-serialisasi untuk direkam ke dalam NoSQL *database* yang telah ditentukan. Pada saat tahap *read* data, proses dilakukan dengan membaca data yang telah disimpan di *database* dalam bentuk *tile*. Data yang dibutuhkan

pengguna disusun kembali oleh sistem dari nilai yang tersedia dalam *database*.

Penelitian ini menggunakan data citra Landsat-8 sebagai *sample* untuk menunjukkan efisiensi yang dihasilkan dengan menggunakan penyimpanan *tiling database*. Sayar, *et al.*, (2014) menyebutkan tahapan proses *tiling* mulai dari pembacaan *longitude* dan *latitude* citra, perhitungan nilai piksel yang paling tepat, perhitungan lebar dan tinggi gambar yang akan diproses, sudut koordinat sebagai (x, y) dari setiap *tile*, perhitungan menggunakan metode *lighten*, dan akhirnya menyimpan gambar sebagai format JPEG. Segmentasi gambar berbasis *tile* juga diusulkan untuk rancangan berdasarkan algoritma penggabungan wilayah. (Patil & Shinde, 2018). Metode yang digunakan oleh Sayar *et al.*, (2014) dan Patil & Shinde, (2018) masih menggunakan penyimpanan yang konvensional sehingga memerlukan kapasitas *storage* yang besar. Strategi penyimpanan data dalam penelitian ini menekankan pada penyimpanan berbasis *database* NoSQL yang memang dirancang untuk manajemen data tidak terstruktur. *Database* NoSQL yang digunakan disini adalah MongoDB yang memiliki fitur kompresi tersendiri sehingga data yang disimpan menjadi lebih kecil dari ukuran aslinya pada penyimpanan konvensional. Keunggulan metode penyimpanan yang digunakan dalam penelitian ini adalah kecepatan pembacaan data (*load data*) citra yang telah di-*tiling*, meskipun proses *write* ke dalam *database*-nya memerlukan waktu cukup lama.

Tabel 2-1: PERBANDINGAN VOLUME ANTARA DATA RASTER DAN TILING

Jumlah Scene	Metode		
	Raster	Tiling	Efisiensi Penggunaan Storage (%)
1 scene	704,6 MB	523 MB	25,7734
10 scene	7 GB	5,2 GB	25,7142
2928 scene	2,0027 TB	1,1150 TB	55,6748

Sumber: pengujian

Tabel 2-2: PERBANDINGAN WAKTU PROSES *LOAD DATA*

No Pengujian	<i>Read Raster</i> (detik)	<i>Read Tiling</i> (detik)	Peningkatan Kecepatan <i>Load data</i> (%)
1	2,3872	1,9080	20,07363
2	2,3977	1,9673	17,9484
3	2,3940	1,9653	17,9045
4	2,3238	1,8786	19,1578
5	2,4413	1,8477	24,3170
6	2,3118	1,9073	17,4963
7	2,4025	1,8434	23,2715
8	2,4019	1,9052	20,6816
9	2,3372	1,8614	20,3571
10	2,3212	1,5652	32,5698
Mean	2,3719	1,8649	21,3778

Sumber: pengujian

Tabel 2-3: PERBANDINGAN WAKTU PROSES MULTI-*TEMPORAL* PERIODE MARET-MEI 2018

No Pengujian	Maret-Mei <i>Raster</i> (detik)	Maret-Mei <i>Tiling</i> (detik)	Efisiensi (%)
1	1,4549	0,9022	37,9870
2	1,4589	0,9340	35,9761
3	1,4581	1,2793	12,2612
4	1,4627	1,3331	8,8593
5	1,4291	1,2104	15,2970
6	1,4590	0,9670	33,7167
7	1,4333	0,9620	32,8818
8	1,4699	0,9658	34,2917
9	1,4433	1,3448	6,8297
10	1,4428	1,0808	25,0894
Mean	1,4512	1,0979	24,3190

Sumber: pengujian

3 HASIL PEMBAHASAN

Pengujian kinerja sistem dilakukan menggunakan data citra Landsat-8 sejumlah 10 *scene*, disajikan secara *temporal* dari tanggal 21 Februari sampai 15 Juli 2018. Alat uji utama adalah program python, serta MongoDB *database*. Proses *tiling* dilakukan berdasarkan *zoom* maksimal yang dimiliki Landsat-8. Perbandingan hasil uji dilakukan pada metode konvensional (*raster*) dan *data cube* (*tiling database*) dengan menggunakan parameter volume, kecepatan *write* data, serta kecepatan *read* data. Ukuran volume

antara *raster* dan *tiling* bisa dilihat pada Tabel 2-1. Tampak bahwa ukuran data bisa ditekan sampai 25% dengan menggunakan metode *tiling database*. Semakin banyak *scene* yang digunakan, maka semakin besar pula efisiensi penggunaan *storagenya*. Saat volume data diperbesar hingga 2928 *scene*, efisiensi penggunaan *storage* meningkat hingga mencapai 55%. Selain menekan ukuran volume, metode *tiling* ini juga menjadikan proses *read* data menjadi lebih cepat dari pembacaan biasa pada *raster*. Representasi yang digunakan untuk pengujian pembacaan data

adalah citra Landsat-8 pada Band-4 (Band Red) secara *temporal*.

Waktu proses *read* data dalam bentuk *array* pada Tabel 2-2 menunjukkan bahwa kecepatan yang dihasilkan *tiling* 21% lebih efisien dari *raster*. Kecepatan ini dikarenakan parameter pencarian secara *query* pada *database* menjadikan sistem lebih spesifik dalam penyajian data. Namun demikian ukuran volume dan kecepatan *read* data yang lebih baik pada metode *tiling* menjadikan kecepatan *write* data ke dalam *database* lebih lama dari kecepatan *write raster (copying)*. Sebagai catatan, *tiling database* menggunakan satu *scene* saja memerlukan waktu sekitar 316,69 detik, sedangkan pada proses *copying raster* hanya memerlukan waktu kurang dari 20 detik.

Volume data hasil proses yang lebih kecil pada *tiling database* dikarenakan MongoDB memiliki proses kompresi bawaan menggunakan *snappy compression*. MongoDB yang merupakan NoSQL memudahkan *developer* dalam memodifikasi data yang sangat mungkin untuk berkembang ke depannya. Kecepatan *read* data besar juga mempengaruhi proses pengolahan yang akan dilakukan pada LAPAN *Engine*.

Kemudahan yang ditawarkan oleh metode *tiling* ini adalah *user* dapat melakukan pemrosesan secara *multi-temporal* dengan waktu lebih cepat dan tidak menghabiskan *space disk* terlalu banyak karena lingkungan kerjanya berupa sistem *online*. Sebagai simulasi, misal *user* ingin memproses satu band citra yang sama dari Landsat-8 pada kurun waktu Maret-Mei 2018. Tabel 2-3 memperlihatkan perbandingan waktu pembacaan data antara bulan Maret-Mei 2018. Tampak disana bahwa pemrosesan secara *tiling* dalam periode tersebut menghasilkan efisiensi proses *read* mencapai 24% jika dibandingkan dengan metode konvensional *raster*. Metode ini tentu menawarkan hasil yang menarik dari segi volume dan

kecepatan proses jika dihadapkan pada kenyataan bahwa data besar adalah salah satu tantangan pengolahan.

Ketika pengujian ini dilakukan menggunakan jumlah data hanya 10 *scene*, efisiensi penyimpanan tampak tidak terlalu tinggi. Namun saat pemrosesan dilakukan menggunakan data yang lebih besar sebanyak 2928 *scene* menjadikan perbedaan waktu proses terlihat signifikan. Metode *tiling database* untuk penyimpanan pada pembangunan LAPAN *Engine* terbukti mampu meningkatkan efisiensi kapasitas penyimpanan yang diperlukan pada sistem.

4 KESIMPULAN

Perbandingan uji menunjukkan bahwa volume data pada pembangunan LAPAN *Engine* dapat ditekan menggunakan metode *tiling database*. Meskipun memerlukan waktu yang lebih lama dalam proses *writing to database*, namun metode ini menawarkan kecepatan proses pembacaan yang lebih efisien untuk keperluan pengolahan daripada pengolahan langsung menggunakan *raster*.

UCAPAN TERIMAKASIH

Penulis berterima kasih kepada Pusat Teknologi dan Data Peninginderaan Jauh - LAPAN yang telah menyediakan data dan dukungan dalam proses penyelesaian makalah ini. Penelitian didanai oleh Kementerian Riset, Teknologi, dan Perguruan Tinggi dengan Program Insentif Riset Sistem Inovasi Nasional Gelombang I Tahun Anggaran 2018 Nomor 11/INS-1/PPK/E4/2018.

DAFTAR RUJUKAN

Bargellini, P., Cheli, S., Desmos, Y. L., Greco, B., Guidetti, V., Marchetti, P. G., ... Sawjer, G. (2013). *Big Data from Space: Event Report, European Space Agency Publication. ESA-ESRIN*. Retrieved from <http://www.congrexprojects.com/docs/default->

- source/13c10_docs/13c10_event_report.pdf?sfvrsn
- Basho. (2018). NoSQL Databases Explained. Retrieved December 9, 2018, from <http://basho.com/resources/nosql-databases/>
- Bioucas-dias, J. M., Plaza, A., Camps-valls, G., Scheunders, P., Nasrabadi, N. M., & Chanussot, J. (2013). Hyperspectral Remote Sensing Data Analysis and Future Challenges. *IEEE Geoscience and Remote Sensing Magazine*, (June). <https://doi.org/10.1109/MGRS.2013.2244672>
- Butler, H. (2018). GDAL. Retrieved December 9, 2018, from <https://pypi.org/project/GDAL/>
- Jewell, D., Portilla, I., Barros, R. D., Saptarshi, M., Diederichs, S., Seera, H. P., ... Plach, O. O. A. (2014). Performance and Capacity Implications for Big Data. *International Business Machines Corporation*, 1–36.
- Lee, J. G., & Kang, M. (2015). Geospatial Big Data: Challenges and Opportunities. *Big Data Research*, 2(2), 74–81. <https://doi.org/10.1016/j.bdr.2015.01.003>
- Lewis, A., Oliver, S., Lymburner, L., Evans, B., Wyborn, L., Mueller, N., ... Wang, L. W. (2017). The Australian Geoscience Data Cube — Foundations and lessons learned. *Remote Sensing of Environment*, 202, 276–292. <https://doi.org/10.1016/j.rse.2017.03.015>
- Patil, V. V., & Shinde, S. G. (2018). EFFICIENT TILE BASED REGION MERGING REMOTE SENSING IMAGE SEGMENTATION. *IJRET*, 7–12. <https://doi.org/10.15623/ijret.2018.0705003>
- Sayar, A., Eken, S., & Mert, U. (2014). Tiling of Satellite Images to Capture an Island Object. *Springer International Publishing Switzerland*, 195–204.
- Sudmanns, M., Tiede, D., Lang, S., & Baraldi, A. (2018). Semantic and syntactic interoperability in online processing of big Earth observation data. *International Journal of Digital Earth*, 11(1), 95–112. <https://doi.org/10.1080/17538947.2017.1332112>
- Zhang, L., Zhang, L., Tao, D., Huang, X., & Du, B. (2015). Compression of hyperspectral remote sensing images by tensor approach. *Neurocomputing*, 147(1), 358–363. <https://doi.org/10.1016/j.neucom.2014.06.052>

