

APLIKASI PENGELOLAAN SKRIPSI DI STMIK AKAKOM YOGYAKARTA MENGGUNAKAN ARSITEKTUR MICROSERVICE DENGAN Node.js

Heri Purnama¹⁾, Indra Yatini B²⁾

Program Studi Teknik Informatika, STMIK AKAKOM Yogyakarta^{1,2)}

Jalan Raya Janti 143 Karang Jambe Yogyakarta 5519812

e-mail : herii.sinatra@gmail.com¹⁾ indrayatini@akakaom.ac.id²⁾

ABSTRAK

Koleksi Skripsi Mahasiswa STMIK AKAKOM YOGYAKARTA sudah berjumlah ribuan judul, namun angka tepatnya tidak ada yang tahu secara pasti. STMIK AKAKOM YOGYAKARTA sebagai salah satu perguruan tinggi Informatika sudah menaruh perhatian khusus dalam mengelola data Skripsi dari Mahasiswa yang telah menyelesaikan studinya dikampus pada <http://perpus.akakom.ac.id/> yang disajikan dalam bentuk format yang dapat dilihat dan diunduh yang mencakup Intisari, Persetujuan, Pengesahan, Daftar Isi, Bab.I, Bab.V, dan Daftar Pustaka yang akan disajikan dengan cara melakukan pencarian berdasarkan kata kunci dan akan ditampilkan bukan hanya dari skripsi dari Mahasiswa STMIK AKAKOM YOGYAKARTA saja namun juga jurnal atau penelitian yang telah di tulis oleh Dosen STMIK AKAKOM YOGYAKARTA.

Data Skripsi disajikan seperti sekarang ini dikhawatirkan akan terjadi plagiat atau kesamaan Skripsi di Institusi. Sebagai contoh : Mengambil judul dan isi Skripsi yang sama dengan yang sudah dibuat oleh Mahasiswa STMIK AKAKOM YOGYAKARTA pada tahun-tahun yang sudah berlalu kemudian mempresentasikan lagi dihadapan Dosen Penguji untuk dijadikan sebagai Skripsinya. Hal inilah yang dikhawatirkan dikarenakan akan terjadi plagiat dalam hal Skripsi, oleh karena itu akan dibuat pengelolaan Skripsi untuk menghindari hal tersebut. Pada pembuatan aplikasi pengelolaan Skripsi ini akan menggunakan Arsitektur Microservice dengan perangkat lunak node.js, dan untuk penyimpanan datanya menggunakan databases NoSQL yaitu MongoDB pada pembuatannya menggunakan sistem operasi Linux : Ubuntu 14.04 LTS serta dalam pengimplementasian dalam Private Cloud menggunakan platform docker.

Microservice adalah sistem yang dapat digunakan untuk membagun layanan secara terpisah-pisah. Aplikasi ini dapat melakukan pencarian berdasarkan kata kunci yang berasal dari deskripsi dari abstrak. Dan juga dapat melakukan pencarian berdasarkan judul dari skripsi yang dibuat.

Kata Kunci : Docker, Linux, Microservice, mongoDB, node.js, Skripsi

ABSTRACT

Thesis collection Students STMIK AKAKOM YOGYAKARTA already amounted to thousands of titles, but exact figures no one knows for sure. Thesis collection Students STMIK AKAKOM YOGYAKARTA is one of the intellectual property assets STMIK AKAKOM YOGYAKARTA be maintained as a responsibility and accountability STMIK AKAKOM YOGYAKARTA. STMIK AKAKOM YOGYAKARTA as one of the colleges of Informatics has been put special attention in managing data Thesis of students who have completed their studies at the university in <http://perpus.akakom.ac.id/> presented in formats that can be viewed and downloaded which includes Digest , Approval, Validation, Table of Contents, Chapter.I, Chapter.V, and Bibliography to be presented in a way to search by keywords and will appear not only on the thesis of a student STMIK AKAKOM YOGYAKARTA, but also journals or research that has been in written by lecturers STMIK AKAKOM YOGYAKARTA.

If data Thesis presented as now feared would happen plagiarism or similarity Thesis diInstitusi. For example: Taking the title and contents Thesis similar to those already made by the Student STMIK AKAKOM YOGYAKARTA in years gone by then present again before Lecturer Examiners to serve as thesis. It is what is feared will happen because of plagiarism in terms of thesis, therefore, be made Thesis managing to avoid it. In making this thesis management application will use the software architecture MICROSERVICE with node.js, and for storage of data using NoSQL databases MongoDB on making use the Linux operating system: Ubuntu 14:04LTS as well as in the implementation of the Private Cloud use docker platforms.

Microservice is a system that can be used to build their services separately. This application can perform a search based on keywords derived from the description of the abstract. And also can search by title of thesis made.

Keywords : Docker, Linux, Microservice, MongoDB, Node.js. Thesis

I. PENDAHULUAN

STMIK AKAKOM YOGYAKARTA sebagai salah satu perguruan tinggi Informatika sudah menaruh perhatian khusus dalam mengelola data Skripsi dari Mahasiswa yang telah menyelesaikan studinya dikampus pada <http://perpus.akakom.ac.id/> yang disajikan dalam bentuk format yang dapat dilihat dan diunduh yang mencakup Intisari, Persetujuan, Pengesahan, Daftar Isi, Bab.I, Bab.V, dan Daftar Pustaka yang akan disajikan dengan cara melakukan pencarian berdasarkan kata kunci dan akan ditampilkan bukan hanya dari skripsi dari Mahasiswa STMIK AKAKOM YOGYAKARTA saja namun juga jurnal atau penelitian yang telah di tulis oleh Dosen STMIK AKAKOM YOGYAKARTA.

Apabila data Skripsi disajikan seperti sekarang ini dikhawatirkan akan terjadi plagiat atau kesamaan Skripsi diInstitusi. Sebagai contoh : Mengambil judul dan isi Skripsi yang sama dengan yang sudah dibuat oleh Mahasiswa STMIK AKAKOM YOGYAKARTA pada tahun-tahun yang sudah berlalu kemudian mempresentasikan lagi dihadapan Dosen Penguji untuk dijadikan sebagai Skripsinya. Hal inilah yang dikhawatirkan dikarenakan akan terjadi plagiat dalam hal Skripsi, oleh karena itu akan dibuat pengelolaan Skripsi untuk menghindari hal tersebut.

Pada pembuatan aplikasi pengelolaan Skripsi ini akan menggunakan *Arsitektur Microservice* dengan perangkat lunak node.js, dan untuk penyimpanan datanya menggunakan *databases NoSQL* yaitu MongoDB. *Arsitektur Microservices* berbeda dengan *web service*, *Arsitektur Microservice* digunakan untuk melakukan setiap *service* yang akan fokus pada satu fungsionalitas tertentu. Satu *service* dan *service* lainnya saling berkomunikasi untuk menghasilkan *business value*. *Business value* disini diartikan, setiap *service* yang telah dibuat pada aplikasi tidak terpengaruh pada *service* yang akan dibuat. Artinya pembuat aplikasi atau *programer* tidak perlu merubah keseluruhan aplikasi sehingga tinggal menambah *service* baru kedalam aplikasi tersebut. Inti dari Nilai bisnis adalah menghemat waktu pengerjaan aplikasi pada saat penambah *service* baru pada aplikasi. *Web service* sendiri mempunyai pengertian sebuah perangkat lunak yang tidak terpengaruh oleh *platform*, yang akan menyediakan method-method yang dapat diakses oleh network. *Web service* menggunakan XML untuk pertukaran data, khususnya pada dua entities yang berbeda.

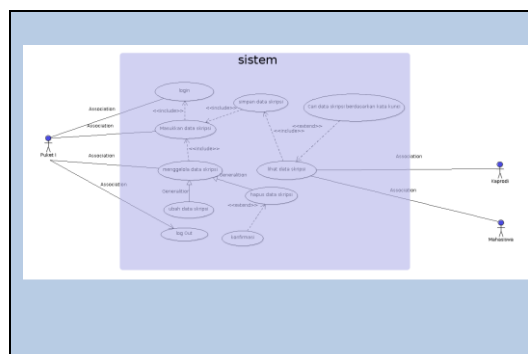
Pada implementasinya *web service* bebeda dengan *web site* dikarenakan *web service* memiliki ciri tidak memiliki antar muka yang bagus dan dibuat untuk berinteraksi langsung dengan aplikasi yang lain beda os atau konsep. *Web service* pun berbeda dengan REST, REST adalah arsitektur perangkat lunak yang digunakan untuk pendistribusian sistem hipermedia seperti *www (world wide web)* . REST sendiri proses transfer data dimulai ketika klient melakukan *request* ke *server*, kemudian *server* memproses permintaan klient dan merespon balik dengan mengirimkan data sesuai permintaan klient. REST pada dasarnya adalah representasi dari objek. Konten-konten objek menggunakan (*GET, POST, PUT, DELETE*).

II. METODE PENELITIAN

Aplikasi Pengelolaan Skripsi Di STMIK AKAKOM YOGYAKARTA Menggunakan Arsitektur Microservice Dengan Node.js, meliputi analisis dan perancangan sistem. Analisis digunakan untuk menganalisa kebutuhan perangkat lunak. Sedangkan perancangan akan digunakan untuk menggambarkan sistem yang akan dibuat.

A. Perancangan Sistem

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasikan, menspesikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis *OO (Object-Oriented)*. Berikut perancangan sistem yang ada pada aplikasi ini.



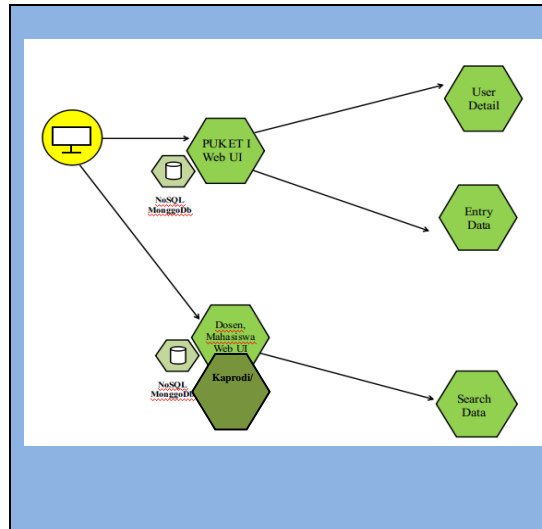
Gambar 1. Use Case Diagram Sistem

Pada kebutuhan perancangan sistem, pemodelan menggunakan *use case*. Diagram *use case* dapat menjadi teknik yang cukup baik untuk menganalisa kebutuhan pengguna sistem karena selain bagus dalam unsur pemahamannya, diagram *use case* juga dapat mendokumentasikan persyaratan sistem dengan baik, yang tampak pada gambar berikut.

Pada Diagram *Use Case* diatas ada beberapa *use case* yang saling terhubung satu sama lain yang mana setiap *use case* mendokumentasikan sistem sendiri.

B. Skema *Microservice* Pada Aplikasi Web

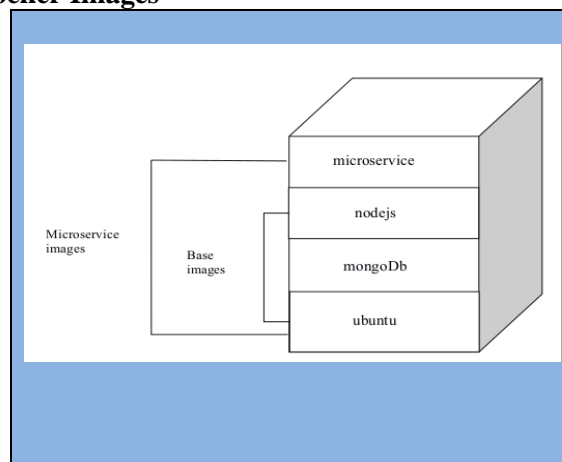
Skema Arsitektur *Microservice* digambarkan dalam aplikasi web. Dapat digambarkan sebagai berikut :



Gambar 2. Skema Arsitektur *Microservice*

Pada pembuatan aplikasi ini menggunakan dua buah *web UI*, yang pertama untuk PUKET I sebagai admin dalam aplikasi ini dan yang kedua adalah Dosen dan Mahasiswa sebagai User. Pada pembuatan aplikasi ini PUKET I mempunyai hak akses dalam memasukkan data skripsi, sedangkan Dosen, Mahasiswa mencari data skripsi yang telah ada dalam *databases*. Pada aplikasi ini *databases* yang digunakan adalah *databases* noSQL dengan MongoDB.

C. Skema Private Cloud Docker Images

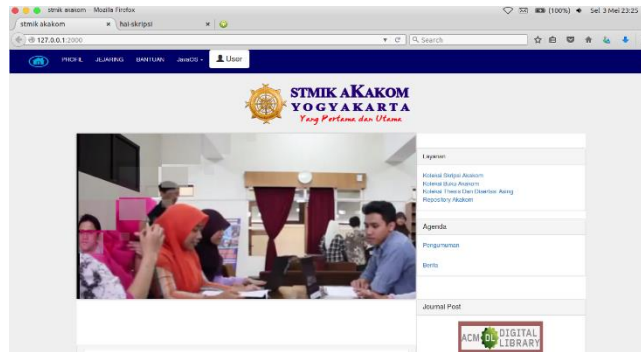


Gambar 3. *Microservice* Images Docker

Untuk implementasi dalam *private cloud* menggunakan docker, pada Gambar 3 *Base Images* yang digunakan adalah ubuntu, mongoDb dan nodejs sedangkan untuk aplikasinya berada paling atas dan semuanya dibungkus dalam satu images yang bernama *microservice images*, *images* kemudian dijalankan pada komputer.

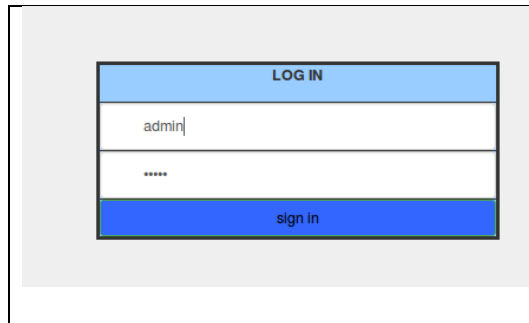
III. HASIL

Pembuatan dan uji coba aplikasi dengan menggunakan Arsitektur Microservice atau bisa juga disebut *Microservices* dengan menggunakan *platform node.js*, *databases mongoDB* dan menerapkannya kedalam *private cloud* dengan *docker*. Pembahasan pada sistem ini merupakan sebuah prosedur yang harus dilakukan guna menyelesaikan rancangan sistem.



Gambar 4. Gambar Halaman Utama

Pada gambar 4 merupakan halaman awal yang pertama kali diakses oleh pengguna. Dimana pengguna memilih layanan yang diinginkan.



Gambar 5. Gambar Halaman Login

Untuk layanan pemasukan data pada aplikasi ini juga diberi fasilitas login yang mana dipergunakan oleh pengguna yang dimaksud adalah Puket I dalam memasukkan data.

Pada halaman inilah nanti si admin akan melakukan memasukkan data dan pada halaman ini sudah dibuat kondisi apabila ada kesamaan judul maka data tidak akan dimasukkan kedalam sistem.

Pada halaman inilah user Kaprodi atau Mahasiswa mengecek tentang skripsi apa yang akan dibuat. Pada pencarian ini menggunakan dua buah fitur pencarian yaitu pencarian sederhana dan pencarian Advance berdasarkan judul yang dibuat.



Gambar 6. Halaman Pemasukan Data

Ketika pencarian menggunakan pencarian sederhana berdasarkan kata kunci maka data yang mengandung kata kunci akan ditampilkan. Kemudian dapat melihat detail dari skripsi yang dibuat.

Aplikasi ini menerapkan *Private Cloud* yang diterapkan ke sebuah *platform* yaitu docker. Sebelum menjalankan docker komputer atau pc harus terkoneksi ke internet dikarenakan untuk menggunakan docker harus mengaktifkan docker daemon dahulu.

```
heriipurnama@halloween:~$ docker images
Cannot connect to the Docker daemon. Is
heriipurnama@halloween:~$ |
```

Bila komputer sudah terkoneksi ke internet, cek koneksi internet dengan menggunakan perintah 'ifconfig'. Perintah yang menunjukkan docker sudah terkoneksi ke internet, docker0 adalah ipdefault dari docker. Jalankan docker images dengan menggunakan perintah “ \$docker run -i -t <username/images> ”

IV. PEMBAHASAN

Aplikasi ini dibangun menggunakan arsitektur *microservice* dengan menggunakan node.js dalam pembangunannya, untuk *databases* menggunakan *databases* noSQL dengan menggunakan mongoDb. Sedangkan untuk penerapan dalam *cloud computing* dengan menggunakan docker sebagai wadah dalam aplikasi. Pada aplikasi ini menggunakan seneca.js dalam membangun aplikasi dengan Microservices.

```
heriipurnama@halloween:~/Documents/node/hai-skripsi node index.js
2016-05-08T08:51:52.538Z
rtl9r5vwu8o6/1462697512481/162/-
INFO hello
Seneca/0.6.2/rtl9r5vwu8o6/1462697512481/162/-
2016-05-08T08:51:52.539Z
vsk7hb3i56jz/1462697512416/163/-
INFO hello
Seneca/0.6.2/vsk7hb3i56jz/1462697512416/163/-
2016-05-08T08:51:53.479Z
rtl9r5vwu8o6/1462697512481/162/-
INFO listen 10201
2016-05-08T08:51:53.596Z
rtl9r5vwu8o6/1462697512481/162/-
```

Gambar 7. Server Microservice

Pada Gambar di atas server yang digunakan untuk melihat setiap *service* yang berjalan pada console. Dari hasil uji coba aplikasi dengan menggunakan *microservice* ada beberapa keunggulan dalam menggunakan *microservice* diantaranya adalah :

1. Arsitektur memungkinkan setiap layanan untuk dikembangkan secara *independent* oleh tim yang difokuskan pada layanan tersebut.
2. Memungkinkan layanan untuk digunakan secara mandiri.
3. Setiap layanan dapat dipantau, diperbaharui secara *independent* untuk mencocokkan tautan.
4. Setiap layanan dapat dikembangkan dan diperbaharui secara mandiri.

Pada aplikasi ini dalam menggunakan *Private Cloud* docker, docker harus terkoneksi ke internet dalam menjalankan docker images. Images Docker bisa di *publish* dengan layanan dari docker dengan docker hub.

V. SIMPULAN DAN SARAN

Berdasarkan dari hasil analisa, perancangan, implementasi, dan pengujian didapatkan kesimpulan:

1. *Microservice* adalah sistem yang dapat digunakan untuk membangun layanan secara terpisah-pisah.
2. Aplikasi ini dapat melakukan pencarian berdasarkan kata kunci yang berasal dari deskripsi dari abstrak. Dan juga dapat melakukan pencarian berdasarkan judul dari skripsi yang dibuat.

3. Aplikasi ini diterapkan dalam *private cloud* dengan menggunakan docker, yang mana dari *base images* ubuntu, node.js dan mongoDB.
4. *Service-service* yang dijalankan ada yang saling berkomunikasi satu sama-lain dan ada yang berdiri sendiri. *Service* yang saling berkomunikasi bisa dilakukan melalui http atau tcp. *Service* yang saling berkomunikasi adalah *service* yang tidak menggunakan mode GUI atau dalam hal ini berhubungan dengan pengguna.
5. Pada penerapan docker penggunaan *private cloud*, digunakan teknologi docker dalam menyetest aplikasi ini berjalan atau tidak. Pada penggunaan docker harus terkoneksi internet dalam pendeploy-an dan meruning *images-images* yang ada dikarenakan dalam running awal docker harus mengaktifkan docker daemon dahulu.

Hasil dari pengujian yang dilakukan, masih banyak kekurangan sehingga perlu dikembangkan untuk meningkatkan kinerja dari aplikasi yang dibangun, adapun saran yang diberikan:

1. Perlu adanya pemberitahuan atau validasi data yang sudah dimasukkan sudah ada dalam databases dalam hal ini mongoDB. Pemberitahuan data sudah ada “already exist”.
2. Dalam pencarian perlu dilengkapi fasilitas download dan *sugestion* atau saran pada saat melakukan pencarian data.

REFERENSI

- [1] Yatini B, 2014, *Aplikasi Pengolahan Citra Berbasis web Menggunakan Javascript dan JQuery* . Jurnal Teknik.Volume empat, Halaman 62-68.
- [2] Kartika Putri Pangesti, 2015, *Implementasi Databases-relasional Pada Sistem Informasi Kearsipan Sekolah (Studi Kasus SMA Internasional Budi Mulia Dua Yogyakarta)*.. Program Studi Komputer. Diterbitkan Jurusan Ilmu Komputer Dan Elektronika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
- [3] Sneps-Snepe Manfred, Namiot Dmtry, 2014, *Micro-service Architecture for Emerging Telecom Applications*. International Journal of Open Information Technologies. Volume Dua, Halaman 34-38.
- [4] Villamor Jose., Ignacio F., Carlos A. Iglesias, Mercedes Garijo, 2010, *Microservices: Lightweight Service Descriptions For REST Architectural Style*. Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid.
- [5] Zanuvar Rizqi Prasetyo, 2015, *Pemanfaatan Google Maps API Untuk Mencari Lokasi SPBU Terdekat Di Kota Jepara dan Kudus dengan Teknologi Node.js*. Diterbitkan diterbitkan Program Studi Teknik Informatika Fakultas Teknik Universitas Kudus.
- [6] Agus Kurniawan, 2014, *Node.js Succintly.[Online]*.Aerial Center Parkway Suite 200 Morrisville, NC 27560 USA: Syncfusion Inc
- [7] Bambang Purnomosidi D.P, 2013, *Buku Cloud Node.js*, <https://github.com/bpdp/buku-cloud-nodejs>, diakses pada tanggal 02 Oktober 2015 pukul 06:13 WIB