

Perancangan Sistem Otomatisasi Continuous Deplo...

By: FIKRI ANGGAWIE

As of: Mar 23, 2022 8:58:12 AM
2,949 words - 16 matches - 16 sources

Similarity Index

9%

Mode: Similarity Report ▼

paper text:

Perancangan Sistem Otomatisasi Continuous Deployment berbasis Monolithic Repository Fikri Anggawie¹, Dana Sulisty Kusumo², Donny Richasdy³ Fakultas Informatika, Universitas Telkom, fikrianggawie@students.telkomuniversity.ac.id, Dana Sulisty Kusumo danakusumo@telkomuniversity.ac.id, Donni Richasdy donnir@telkomuniversity.ac.id Abstrak Otomatisasi adalah hal yang sudah seharusnya diterapkan sebuah perusahaan yang mengerjakan proyek besar terdiri dari dua orang atau lebih dalam pengembangan aplikasi mereka[1]. Tanpa adanya Otomatisasi maka dalam melakukan perubahan code meskipun berskala kecil akan memakan banyak waktu dan biaya tentunya[5]. Otomatisasi menggunakan metode continuous deployment dan aplikasi jenkins, docker dan menerapkan prinsip monolithic repository maka akan banyak membantu dalam penerapan otomatisasi. Tugas akhir kali ini berfokus pada pengembangan dan integrasi menggunakan metode continuous deployment dan monolithic repository serta menggunakan tools jenkins dan docker dalam pengerjaannya. Penerapan metode continuous deployment dan monolithic repository menggunakan jenkins guna mempermudah dalam melakukan integrasi dikarenakan jenkins bersifat open source dan memiliki banyak opsi plug-in. docker sendiri berperan sebagai container guna meletakkan semua kebutuhan integrasi aplikasi dan deployment. Adanya sistem otomatisasi adalah bentuk penghematan sumber daya seperti waktu dan penggunaan yang lebih mudah. Jadi

integrasi dan penerapan menggunakan jenkins dan docker kita dapat menghemat banyak waktu dan juga

11

jika ada perubahan pada code kita tidak memerlukan integrasi ulang dan dapat dirubah dengan mudah[1]. Abstract Automation is something that should be implemented by a company working on a large project consisting of two or more people in developing their application[1]. Without automation, making code changes even though it is small

will take a lot of time and money

15

, of course[5]. Automation using the continuous deployment method and the Jenkins, Docker application and applying the monolithic repository principle will help a lot in the application of automation. This final project focuses on development and integration using the continuous deployment method and monolithic repository as well as using Jenkins and Docker tools in the process. The application of the continuous deployment method and monolithic repository uses jenkins to facilitate

integration because jenkins is open source and has many plug-in options. docker itself acts as a container to place all application integration and deployment needs. The existence of an automation system is a form of saving resources such as time and easier use. So integration and deployment using jenkins and docker we can save

a lot of time and also if there are changes to

10

the code we don't need re-integration and can be changed easily[1]. 1. Pendahuluan Latar Belakang Banyak aplikasi yang diharuskan bersifat cepat dalam peningkatan pengembangan produknya, hal tersebut dikarenakan pembaruan dibutuhkan berdasarkan permintaan pasar ataupun dalam segi bisnis. Deployment tanpa otomatisasi akan memakan banyak waktu untuk melakukan perubahan pada code yang akan di ubah, serta melakukan deploy kembali code tersebut[1]. Continuous deployment (CD) berperan sebagai metode yang memastikan bahwa perangkat lunak dapat dirilis kapan saja dengan mengotomatiskan deployment dan rilis alur kerja[2]. tim perangkat lunak perusahaan besar pun seperti mozilla, google, microsoft telah mengadopsi CD sebagai bentuk peningkatan development serta mempercepat pembangunan dan meningkatkan kualitas perangkat lunak[2]. Proses deployment yang menerapkan monolithic repository akan lebih mempermudah dalam banyak aspek. beberapa aspek tersebut adalah dengan menyederhanakan management ketergantungan dari repository lain dengan cara melakukan penyimpanan secara terpusat dan selain menyederhanakan ketergantungan antar repository akan sulit apabila merilis versi baru dengan tidak menyebabkan kerusakan pada bagian lain karena semua penelponnya harus di perbarui pada saat yang bersamaan[8]. Jenkins dalam otomatisasi sendiri berperan sebagai pendukung proses integrasi dalam membangun auto deployment dan docker berperan sebagai container yang akan berisikan semua kebutuhan otomatisasi seperti jenkins dan perangkat lunak itu sendiri. Sistem otomatisasi yang menerapkan metode continuous deployment dan penerapan monolithic repository akan banyak memotong waktu dalam melakukan deploy code pada server. Kebutuhan otomatisasi akan menjadi penting karena banyak keuntungan yang akan di dapatkan selain memangkas waktu dalam proses deployment. Topik dan Batasannya

Dari latar belakang di atas, maka permasalahan yang ditentukan **dalam penelitian** kali **ini adalah** 8

penerapan proses deployment secara otomatis dengan mengadaptasi penerapan monolithic repository. Penerapan ini menawarkan konfigurasi manajemen yang lebih baik dari cara yang digunakan selama ini yaitu proses deployment dengan yang tidak diotomatisasi, cara tersebut akan banyak memakan waktu lama dan menimbulkan potensi kesalahan pada proses deployment dari sisi developers ataupun sistem seperti muncul bug ataupun, gagal deployment karena ketidakcocokan pada sistem. Proses deployment sendiri mencakup bagaimana sebuah code yang akan di deploy ke production dengan cara otomatis, sedangkan monolithic repository berperan dalam manajemen repository yang di mana code aplikasi akan tersimpan dalam satu tempat atau satu repository, hal tersebut akan mempermudah pengerjaan karena code tersimpan dalam satu tempat dan developer pun bisa melihat sejauh apa pengerjaan tim lain sejauh apa pengerjaannya dan bisa menyesuaikan tahapan pengerjaan sehingga bisa selesai pada saat yang sama. Dalam proses pengerjaan sistem

otomatisasi kali ini mencakupi proses deployment itu sendiri dengan menggunakan tools docker sebagai container sebagai manajemen aplikasinya, jenkins guna melakukan proses deployment, virtual private server(VPS) sebagai penyimpanan os dan github sebagai penyimpanan code. Proses selanjutnya dalam pengerjaan sistem monolithic repository akan dilakukan dengan pembuatan code sendiri dan tidak melakukan tools apapun. Tujuan Melihat topik yang ada tujuan dalam penerapan ini adalah • Mengimplementasikan metode continuous deployment untuk menjadi sebuah sistem otomatisasi. • Membuat sistem otomatisasi yang menggunakan docker dan jenkins. • Menggabungkan metode continuous deployment dengan monolithic repository Organisasi Tulisan Pada penulisan kali ini topik bahasan selanjutnya akan dibagi menjadi beberapa bagian yang

akan dijelaskan secara singkat pada poin-poin di bawah ini: 1

9

. Pada bagian kedua yang berisikan penjelasan terkait hasil dari beberapa penelitian sebelumnya dan penjelasan metode-metode yang ada. 2. Pada bagian selanjutnya yaitu bagian ketiga akan menjelaskan bagaimana sistem penelitian kali ini dibangun serta spesifikasi lengkap mengenai tools apa saja yang digunakan dalam penelitian ini. 3. Di bagian keempat menjelaskan bagaimana proses otomatisasi berjalan dan melakukan pengujian dan membandingkan dengan acuan yang telah ada. 4. Pada bagian terakhir adalah menyimpulkan hasil dari membandingkan kedua arsitektur yang ada. 2. Studi Terkait Point Penelitian Sebelumnya Dalam penelitian sebelumnya proses otomatisasi dengan menggunakan metode continuous deployment dan tools jenkins ansible mendapatkan hasil penelitian adalah setelah menerapkan otomatisasi waktu penyebaran dalam aplikasi tunggal berkurang secara signifikan yang akan meningkatkan dalam pemeliharaan aplikasi, selain itu dengan otomatisasi akan berguna apabila aplikasi melakukan perubahan code pada repository yang di mana akan mengurangi waktu dalam peninjauan code serta bisa melakukan perubahan code secara berulang untuk mengubah aplikasi[1]. Dalam pembahasan penelitian selanjutnya yang membahas tentang keuntungan dan kerugian dalam menerapkan monolithic repository(monorepo). Dalam penelitian kali ini adalah dengan melakukan survei kepada developers yang telah melakukan penerapan monorepo atau multi-repo ataupun telah mencoba keduanya[4]. Hasil dari survei dapat disimpulkan banyak developers lebih menyukai monorepo dikarenakan kemudahan karena tidak memiliki ketergantungan pada code-base lain akan tetapi pada penerapannya monorepo tidak disarankan dalam penggunaan berskala besar karena begitu banyak code-base yang akan disimpan dalam satu tempat[4]. beberapa developers lain menyarankan pula apabila penerapan monorepo disandingkan dengan manajemen yang baik maka code dalam jumlah besar pun bukan menjadi masalah yang berarti[4]. Otomatisasi Penerapan otomatisasi bertujuan untuk mempermudah pekerjaan dalam sebuah project untuk melakukan deployment suatu aplikasi serta menghindari proses-proses seperti munculnya bug dalam deploying[2]. Otomatisasi perlu dilakukan karena begitu banyak kemudahan yang akan didapatkan serta tidak perlu melakukan integrasi berulang ulang dalam setiap pembuatan aplikasi baru. Continuous Deployment continuous deployment merupakan jantung devops dan membentuk bagian terpenting pada bagian optimasi pengiriman software secara menyeluruh. Sebagian besar perusahaan teknologi informasi dan komunikasi mengalami sisi operasi kontributor yang signifikan tetapi mengalami keterlambatan dalam pengiriman software[3]. Continuous deployment merupakan proses rilis perangkat lunak menggunakan pengujian otomatis guna memvalidasi apakah perubahan code sudah sesuai dan stabil

untuk penerapan dalam lingkungan produksi. Berikut adalah gambaran tahapan dalam proses continuous deployment. Gambar 2-1. Proses Continuous Deployment Docker Docker adalah

perangkat lunak yang memungkinkan untuk membuat, menguji, dan menerapkan aplikasi dengan 1
cepat[6]. **Docker** dapat **mengemas perangkat lunak ke dalam unit standar yang disebut container** .
Container sendiri berisikan **semua** kebutuhan **perangkat lunak agar dapat berfungsi. Termasuk pustaka, alat sistem** , code, **dan waktu proses. Docker**

menerapkan dan melaksanakan **aplikasi ke** environment apapun **dan yakin bahwa** code 13
akan berjalan

. [6] Jenkins Jenkins adalah tools

berbasis open- source yang ditulis menggunakan bahasa pemrograman 16

java[7]. jenkins sendiri berdiri sejak tahun 2011 yang menawarkan pangsa pasar dominan, jenkins digunakan tim dari semua ukuran dalam sebuah proyek[7]. jenkins menjadi populer dalam kalangan pengembang dikarenakan jenkins mudah digunakan serta pengguna sederhana[7], intuitif, dan menarik secara visual. jenkins juga bersifat fleksibel dan mudah beradaptasi untuk tujuan para pengembang. kemudahan lain yang dimiliki jenkins adalah memiliki ratusan plugin open source dan plugin yang tersedia mencakup semua hal termasuk sistem kontrol versi, alat build, metrik kualitas code, build notifikasi, integrasi dengan sistem external, kustomisasi ui serta banyak plugin lain yang mudah dalam penginstalannya[7].
FVirtual

Private Server virtual private server (VPS) adalah server yang 12

menggunakan teknologi virtualisasi hardware server. Vps disebut virtual di karenakan pembagian setiap bagian nya dilakukan dengan sebuah software sehingga dalam satu server bisa terdapat beberapa vps. Git adalah sebuah penyimpanan historikal yang di mana setiap perubahan code yang telah kita lakukan akan tersimpan dalam git sehingga memudahkan kita untuk mengetahui perubahan apa saja yang telah dilakukan oleh para pengembang. Monolithic Repository Monolithic repository(monorepo) atau biasa disebut dengan monorepo adalah model repository yang ditawarkan oleh google dan telah diterapkan juga oleh perusahaan besar lainnya yaitu sebuah repo yang menyimpan semua code nya dalam satu repository saja [5]. Monorepo memberikan keuntungan selain semua code tersimpan dalam sebuah repository

adalah apabila ada sebuah project atau taks yang memerlukan project lain sebagai acuan dasarnya maka akan lebih mudah karena ketika semua tersimpan dalam satu repository maka developers akan dapat melihat pekerjaan tim lain sejauh mana dan dapat menyesuaikan alur pengerjaan nya [5]. keunggulan lain dari monorepo ialah manajemen ketergantungan yang disederhanakan, mampu mengkolaborasi lintas tim, tidak ada batasan kepemilikan code, visibilitas code dan meminimalisir kerusakan ketika melakukan pembaruan code dikarenakan masalah ketergantungan code-base yang dapat dilihat seperti perumpamaan di bawah ini antara monorepo dan multi-repo. Gambar 2-1. Proses Ketergantungan Perubahan Code[5] 3.

Sistem yang Dibangun Deskripsi Umum Implementasi sistem menggunakan metode continuous deployment dan monorepo guna menerapkan otomatisasi deploy software kepada production. Dalam penerapan ini menggunakan beberapa aplikasi sebagai pendukung proses deployment yaitu vps yang akan berjalan sebagai server serta tempat penyimpanan semua aplikasi kebutuhan deployment. Aplikasi yang akan berjalan dalam vps adalah docker sebagai container dan jenkins sebagai aplikasi yang menyimpan kebutuhan deployment. Berikut adalah gambaran bagaimana proses otomatisasi deploy berjalan.

Setup Docker Penginstalan docker pada virtual private server dilakukan pada komputer lokal melalui ssh kepada virtual private server yang telah di sewa. Pada pengerjaan kali ini menggunakan docker versi 20.10.8 spesifikasi virtual private server menggunakan ubuntu versi 21.04 dengan 2 vcpu ram 2048 MB dan penyimpanan 40 GB. Setup Jenkins Jenkins di instal pada container dalam docker yang telah di buat dan melakukan beberapa penginstalan sebagai berikut :

- Jenkins container Setup jenkins dalam docker container serta menentukan port yang akan diakses yaitu port 8080. proses selanjutnya adalah mengakses jenkins sesuai ip yang ada lalu melakukan registrasi dan penginstalan jenkins pada virtual private server.
- Membuat branch pipline dengan github Membuat pipline pada jenkins serta menghungkannya kepada github sebagai sumber semua code tersimpan. Tools jenkins yang ada menggunakan jenkins versi 2.303.1 Setup Otomatisasi Deploy Pada bagian setup ini dilakukan hard code pada github yang telah dikonfigurasi dengan jenkins yang terinstall pada virtual private server. Setup Monorepo Setup repository dilakukan bersamaan dengan pembuatan sistem otomatisasi dengan menambahkan beberapa code. Penambahan code tersebut adalah fitur pengecekan apabila terdapat aktivitas commit pada github maka sistem akan melakukan build dan folder yang berubah saja. (penjelasan flow app)

Pengukuran Tingkat Keberhasilan Pengukuran tingkat keberhasilan dalam penerapan ini adalah dengan membandingkan hasil dari proses deployment 4. Evaluasi Hasil pengujian A. Struktur Sistem otomatisasi Sebelum memasuki tahapan hasil dari pengujian, berikut adalah tahapan bagaimana proses sistem otomatisasi berjalan yang awali dengan mendeteksi adanya perubahan code yang ada pada Repository dan memasuki 3 tahapan, yaitu:

1. Prepare Pada tahapan ini sistem bekerja dengan mendeteksi adanya commit pada penyimpanan yang ada dan melakukan get change list melihat bagian penyimpanan yang berubah serta melakukan pengecekan apakah semua kondisi telah memenuhi syarat untuk dilanjutkan kepada proses build.
2. Build Dalam proses build melakukan build list yang telah didapatkan dari proses prepare, setelah itu membuat docker image dan dalam tahapan terakhir melakukan pengecekan kembali apakah ada list yang tidak terbaca ataupun memenuhi syarat untuk lanjut ke tahapan selanjutnya.
3. Deployment Pada Proses ini adalah melakukan pengecekan apakah terdapat docker service dalam penyimpanan yang diubah, selanjutnya apabila telah adanya docker service maka akan dilanjutkan ke dalam tahapan update docker service dengan perubahan yang baru dan apabila belum memiliki docker service maka akan melakukan pembuatan docker service dan melakukan deployment aplikasi. Proses prepare, build dan deployment yang dapat di gambarkan oleh

gambar di bawah ini: Gambar 4-1. Proses

14

sistem otomatisasi berjalan B. Perbandingan Sistem Proses pengujian selanjutnya adalah dengan membandingkan hasil dari kedua framework yang ada dengan membandingkan hasil stages (proses akhir deployment). 1. Framework pertama Sistem otomatisasi yang menjadi pembanding pada bahasan penelitian kali ini dibangun dengan menggunakan metode continuous deployment dan integration serta menggunakan tools jenkins ansible. Jenkins berperan sebagai tools otomatisasi dan ansible sendiri digunakan sebagai penyebaran aplikasi yang berisikan alamat ip dari sistem yang manghost aplikasi[1]. kemudian mendeklarasikan variabel seperti nama pengguna sistem tempat meng-host kemudian kata sandi dan menentukan nomor port bila dibutuhkan[1]. Tujuan dalam pembangunan sistem otomatisasi ini bertujuan mengurangi waktu yang dibutuhkan dalam menghosting aplikasi[1]. selain waktu yang telah berkurang banyak tools yang digunakan pun memberi banyak kemudahan karena bersifat open source dan memiliki banyak plugin[1]. hasil dari pembuatan sistem otomatisasi kali ini adalah seperti gambar di mana tahapan jenkins setelah otomatisasi di continuous integration dan continuous deployment untuk waktu penyebaran aplikasi tunggal telah berkurang secara signifikan yang akan meningkatkan pemeliharaan aplikasi[1]. otomatisasi juga akan sangat berguna bagi aplikasi yang sering berubah[1]. Gambar 4-2. Hasil stages dari acuan yang ada[1] 2. Framework Kedua Pembangunan sistem otomatisasi ini dibangun dengan menggunakan metode continuous deployment berbasis monolithic repository dengan menggunakan tools docker dan jenkins yang sama-sama menawarkan berbagai kemudahan seperti acuan pada pembangunan sistem pertama tetapi menambahkan basis monolithic repository sebagai penyimpanan source codenya. Tujuan penerapan basis monolithic repository adalah menawarkan kemudahan yang lebih lagi karena semua penyimpanan tersimpan pada satu wadah besar serta kemudahan lain yang tidak perlu melakukan konfigurasi sistem otomatisasi baru apabila ingin melakukan pembuatan aplikasi baru yang berbeda dari aplikasi sebelumnya. Berikut adalah gambaran pengukuran waktu deployment sistem otomatisasi pada jenkins. Gambar 4-3. Hasil dari stages penerapan yang baru Analisis Hasil Pengujian Gambar 4-3. Proses waktu Deployment kedua Framework Skenario pengujian dilakukan dengan cara melakukan proses deployment web dengan melakukan deploy web sebanyak lima kali dan menggunakan spesifikasi serta isi perubahan web yang sama dengan tujuan mendapatkan hasil yang setara dan optimal. Dalam proses perbandingan juga menggunakan dua virtual private server yang mengatur dari setiap framework yaitu ansible dan monolithic. Tujuan dari penempatan masing masing virtual private server agar mendapatkan hasil yang optimal dan setara dari kedua framework. Dan berikut hasil waktu proses deployment antara kedua framework dapat dilihat seperti tabel di atas dan rata-rata keseluruhan proses deployment setiap framework. Hasil perbandingan antara perhitungan framework ansible dan monolithic repository yang dapat dilihat seperti pada tabel di atas adalah proses waktu deployment yang menunjukkan bahwa framework dengan ansible lebih baik dalam proses deployment dalam skenario pengujian yang sama dari segi spesifikasi dan isi deployment nya. Hal tersebut dapat terjadi karena ansible hanya memiliki satu aplikasi di setiap penyimpanannya sedangkan monolithic memiliki beberapa aplikasi dalam satu penyimpanan dan dalam proses deploymentnya terdapat beberapa proses pengecekan standar deployment yang harus dilewati sehingga akan memakan waktu lebih lama. Sistem otomatisasi menggunakan metode continuous deployment dan monolithic repository dengan tools docker dan jenkins adalah suatu proses otomatisasi yang baik digunakan pada masa ini. Selain berbagai kemudahan yang didapatkan, tools yang adapun mudah digunakan karena bersifat open source dan gratis.

Sistem otomatisasi dengan metode continuous deployment yang mengadaptasi monolithic repository memang mendapatkan hasil yang jauh berbeda dengan jurnal acuan sebelumnya dikarenakan monolithic repository menawarkan kemudahan dalam bentuk yang lain. Kemudahan menerapkan metode monolithic repository adalah selain tidak adanya ketergantungan pada code yang ada dengan monolithic repository juga apabila kita ingin membuat sebuah aplikasi baru maka tidak perlu melakukan konfigurasi ulang terkait framework yang ada karena semua code pada dasarnya hanya tersimpan pada satu wadah yang sama. 5. Kesimpulan Perbandingan antara penerapan sistem menggunakan monolithic repository dengan ansible memang menunjukkan perbedaan yang jauh dalam waktu deployment yang di mana ansible menunjukkan waktu deployment lebih baik, tetapi penerapan monolithic repository memiliki kelebihan lain yaitu dengan adanya penyimpanan secara terpusat maka keseluruhan code dapat dilihat oleh developers dari berbagai divisi untuk menyesuaikan proses pengembangan aplikasi sehingga berjalan secara beriringan sampai semua proses pembuatan aplikasi selesai. Apabila code penyimpanan tersimpan di banyak tempat maka framework ansible akan lebih memudahkan karena hanya perlu melakukan ssh dari satu tempat yang sama untuk mengatur server yang lain. Sedangkan framework monolithic penggunaannya lebih manajemen data terpusat yang tersusun dengan baik sehingga dengan satu repository saja bisa menyimpan begitu banyak aplikasi. [1]

Mysari, S . and Bejgam, V., 2020, February. Continuous Integration and Continuous Deployment Pipeline Automation Using Jenkins Ansible. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic- ETITE) (pp. 1-4). IEEE

2

. [2]Gallaba, K., 2019, September.

Improving the robustness and efficiency of continuous integration and deployment. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME

5

) (pp. 619-623). IEEE. [3]

Virmani, M., 2015, May. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In Fifth international conference on the innovative computing technology (intech 2015) (pp. 78- 82). IEEE . [4]Jaspan, C ., Jorde, M., Knight, A

3

Sadowski, C., Smith, E.K., Winter, C. and Murphy-Hill, E., 2018. Advantages and Disadvantages of a Monolithic Repository

6

. [5]Laanti,

Potvin, R . and Levenberg, J., 2016. Why Google stores billions of lines of code in a single repository. Communications of the ACM, 59(7

4

), pp.78-87. [6]

Ahmed, A . and Pierre, G., 2018 , July. Docker container deployment in fog computing infrastructures. In 2018 IEEE

7

]

sources:

1 35 words / 1% - Internet from 19-Mar-2021 12:00AM
teknologi.bisnis.com

2 34 words / 1% - Internet from 24-Nov-2021 12:00AM
tesis.udea.edu.co

3 32 words / 1% - Internet from 19-Jul-2021 12:00AM
www.ijeat.org

4 23 words / 1% - Internet from 16-Jan-2020 12:00AM
www.groundai.com

5 21 words / 1% - Internet from 08-Aug-2020 12:00AM
researchr.org

6 19 words / 1% - Internet from 03-Nov-2021 12:00AM
qspace.library.queensu.ca

7 15 words / < 1% match - Crossref

[Shubha Brata Nath, Subhrendu Chattopadhyay, Raja Karmakar, Sourav Kanti Addya, Sandip Chakraborty, Soumya K Ghosh. "Containerized deployment of micro-services in fog devices: a reinforcement learning-based approach", The Journal of Supercomputing, 2021](#)

8 12 words / < 1% match - Internet from 01-Jul-2021 12:00AM

otaksakti.wordpress.com

9 11 words / < 1% match - Crossref

[Paulus Bawole. "The development of urban kampong as one the alternatives special interest tourism", ARTEKS : Jurnal Teknik Arsitektur, 2020](#)

10 11 words / < 1% match - Internet from 14-Jul-2021 12:00AM

www.ijariit.com

11 10 words / < 1% match - Internet from 30-Dec-2021 12:00AM

repository.ub.ac.id

12 9 words / < 1% match - Internet from 06-Aug-2021 12:00AM

docplayer.info

13 9 words / < 1% match - Internet from 22-Jul-2021 12:00AM

eprints.akakom.ac.id

14 8 words / < 1% match - Internet from 29-Dec-2020 12:00AM

123dok.com

15 8 words / < 1% match - Internet from 11-May-2019 12:00AM

jaiandhraclub.blogspot.com

16 8 words / < 1% match - Internet from 24-Apr-2021 12:00AM

stt-wastukencana.ac.id
