**RabbitMQ**

RabbitMQ adalah *software* yang digunakan untuk antrian pesan atau sebagai perantara pesan (*message broker*) atau manajer antrian. Sederhananya adalah perangkat lunak tempat antrian ditentukan, yang menghubungkan aplikasi untuk mentransfer pesan.

Pesan yang dikirim dapat berbagai jenis informasi. Misalnya, memiliki informasi tentang proses atau tugas yang harus dimulai pada aplikasi lain (yang bahkan bisa berada di server lain), atau bisa juga hanya pesan teks sederhana. *Message broker* menyimpan pesan sampai aplikasi penerima terhubung dan mengeluarkan pesan dari antrian. Aplikasi penerima kemudian memproses pesan tersebut.

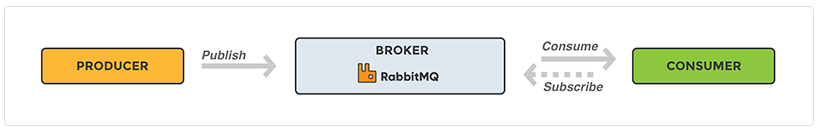
**Penggunaan *message broker***

Message broker bertindak sebagai perantara untuk berbagai layanan (misalnya aplikasi web). Message broker dapat digunakan mengurangi beban dan waktu pengiriman server aplikasi web dengan mendelegasikan tugas yang biasanya memakan banyak waktu atau sumber daya ke pihak ketiga yang tidak memiliki pekerjaan lain.

**Skenario**

Aplikasi web pengguna dapat mengunggah informasi ke situs web. Situs tersebut akan menangani informasi ini, menghasilkan PDF, dan mengirimkannya kembali ke pengguna melalui email. Menangani informasi, membuat PDF, dan mengirim email, dalam contoh kasus ini, memerlukan waktu beberapa detik. Itulah salah satu alasan mengapa antrian pesan akan digunakan untuk melakukan tugas.

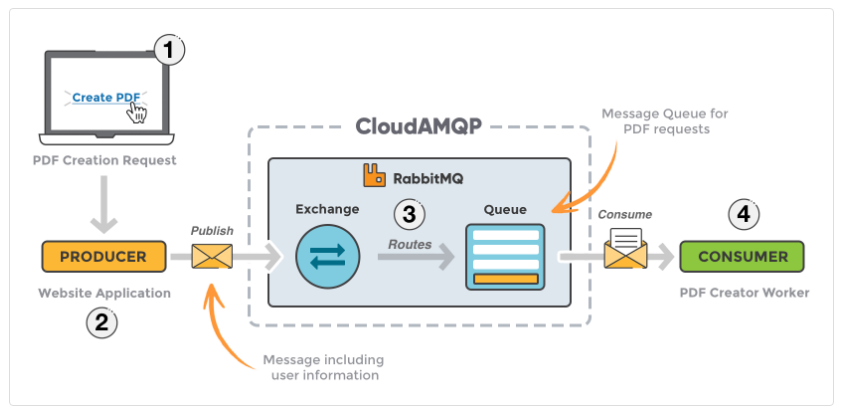
Saat pengguna memasukkan informasi pengguna ke antarmuka web, aplikasi web akan membuat pesan "pemrosesan PDF" yang mencakup semua informasi penting yang dibutuhkan pengguna ke dalam pesan dan menempatkannya ke antrean yang ditentukan di RabbitMQ.



Kapan dan mengapa harus menggunakan message broker?

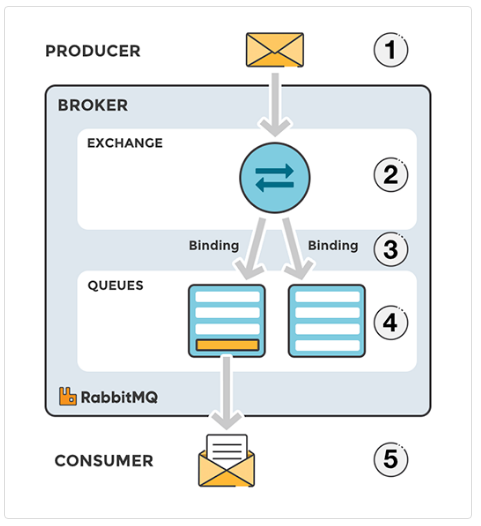
Antrian pesan memungkinkan server web untuk menanggapi permintaan dengan cepat alih-alih dipaksa untuk melakukan prosedur dengan banyak sumber daya di tempat yang dapat menunda waktu respons. Antrian pesan juga baik ketika Anda ingin mendistribusikan pesan ke banyak konsumen atau untuk menyeimbangkan beban antar pekerja.

Konsumen mengambil pesan dari antrian dan mulai memproses PDF. Pada saat yang sama, produser mengantri pesan baru. Konsumen dapat berada di server yang sama sekali berbeda dari produsen atau mereka dapat ditempatkan di server yang sama. Permintaan tersebut dapat dibuat dalam satu bahasa pemrograman dan ditangani dalam bahasa pemrograman lain. Intinya, kedua aplikasi hanya akan berkomunikasi melalui pesan yang mereka kirim satu sama lain, yang berarti pengirim dan penerima memiliki kopling yang rendah.



**Exchanges**

Pesan tidak dipublikasikan langsung ke antrian; sebagai gantinya, produser mengirim pesan ke *exchange*. *Exchange* bertanggung jawab untuk merutekan pesan ke antrian yang berbeda dengan bantuan binding dan *routing key*. *Binding* adalah hubungan antara *queue* dan *exchange*



Message flow RabbitMQ

1. *Producer* *publish* pesan ke *exchange*. Saat membuat sebuah *exchange* maka jenisnya terlebih dahulu harus ditentukan.
2. *Exchange* menerima pesan dan bertanggung jawab untuk melakukan *routing* terhadap pesan. *Exchange* tersebut memperhitungkan atribut pesan yang berbeda seperti *routing* *key* dan tergangung pada jenis *exchange*.
3. Binding harus dibuat dari exchange ke antrian. Dalam hal ini ada dua binding kedalam dua antrian berbeda dari exchange. Exchange merutekan pesan ke antrian tergantung pada atribut pesan.
4. Pesan akan tetap berada dalam antrian sampai di *consume* oleh *consumer*.
5. Konsumen *handle* pesan tersebut.

Jenis-jenis *exchange*

1. Direct

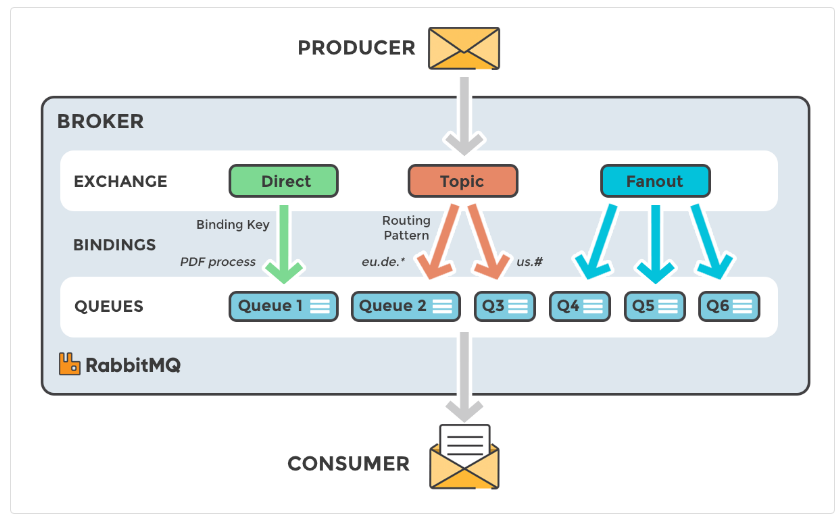
Pesan dirutekan ke antrian yang *binding key* sama persis dengan *routing key* pesan. Misalnya, jika antrean terikat ke *exchange* dengan proses pdf kunci yang mengikat, pesan yang *dipublish* ke *exchange* dengan proses pdf kunci perutean dirutekan ke antrean itu.

1. Topic

*Topic exchange* yakni melakukan pencocokan karakter pengganti antara *routing key* dan pola perutean yang ditentukan dalam *binding*.

1. Fanout

Fanout *exchange* yakni pesan rute ke semua antrian yang terikat padanya.



Istillah-istilah rabbitMQ.

* *Producer*

Aplikasi pengirim pesan.

* *Consumer*

Aplikasi penerima pesan.

* *Queue*

Antrian penyimpanan pesan.

* *Message*

Informasi yang dikirim melalui *producer* ke *consumer* melalui rabbitMQ.

* *Connection*

Koneksi TCP antara aplikasi Anda dan RabbitMQ.

* *Channel*

Koneksi virtual di dalam koneksi. Saat menerbitkan atau mengonsumsi pesan dari antrian - semuanya dilakukan melalui saluran.

* *Exchange*

Menerima pesan dari produsen dan mengirimkanya ke antrian tergantung pada aturan yang ditentukan oleh jenis pertukaran. Untuk menerima pesan, antrian harus terikat ke setidaknya satu pertukaran.

* *Binding*

Hubungan antara *queue* dan *exchange*

* *Routing key*

Sebuah key yang bertugas untuk merutekan pesan ke antrean. *Routing key* seperti alamat untuk pesan

* *AMQP* (*Advanced Message Queuing Protocol*)

Protokol yang digunakan oleh RabbitMQ untuk olahpesan.

* *Users*
* Vhost (*virtual host*)