# BIG DATA (TFC303)

Pertemuan 6 – HDFS (Hadoop Distributed File System)


**ALIFIA REVAN PRANANDA**

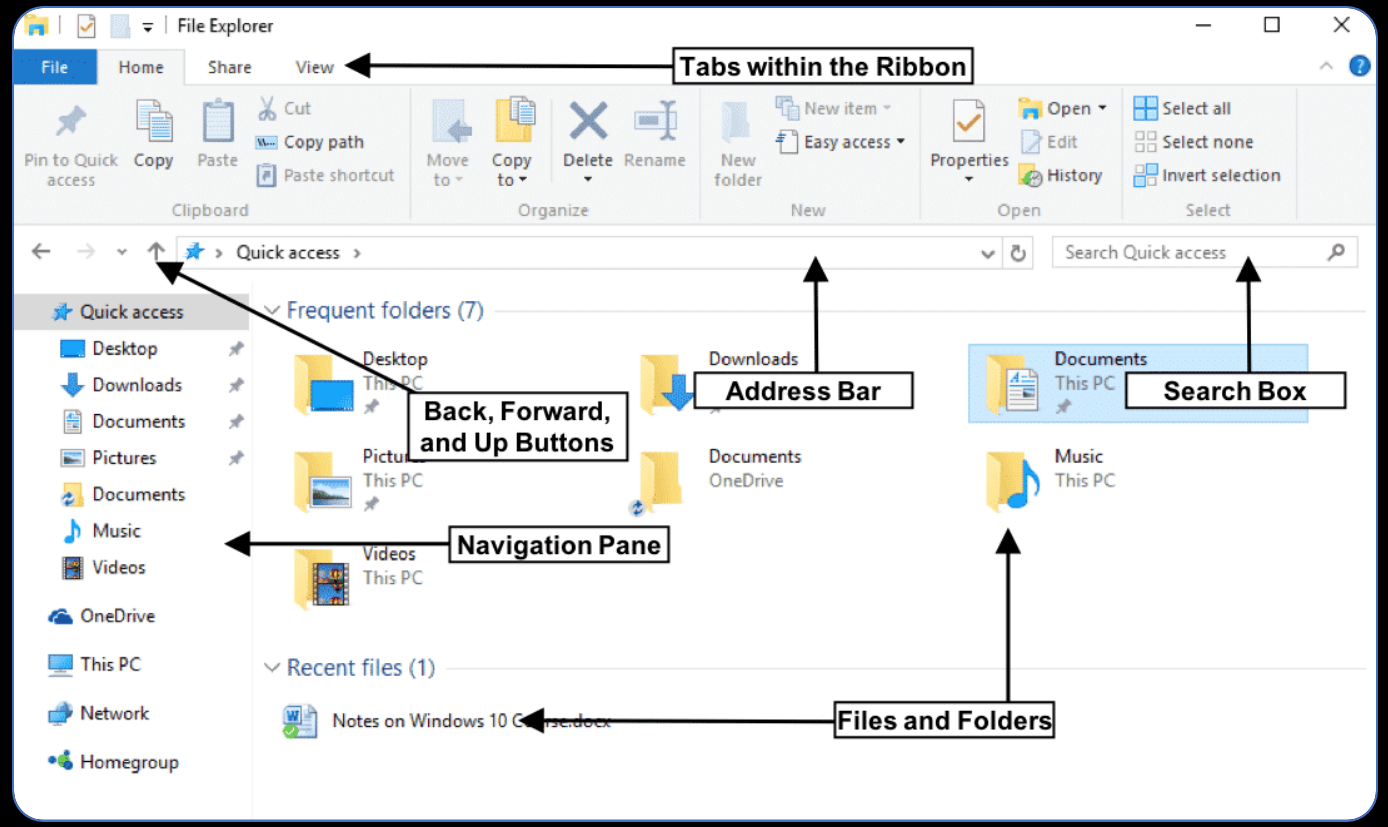
Department of Information Technology
Faculty of Engineering
Universitas Tidar

# HADOOP DISTRIBUTED FILE SYSTEM

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

What is "**FILE SYSTEM**" ?

File system mendefinisikan bagaimana file dikelola (seperti: diberi nama, disimpan dan dibuka kembali) dari sebuah media penyimpnan (seperti: hard drive, flashdisk dsb).



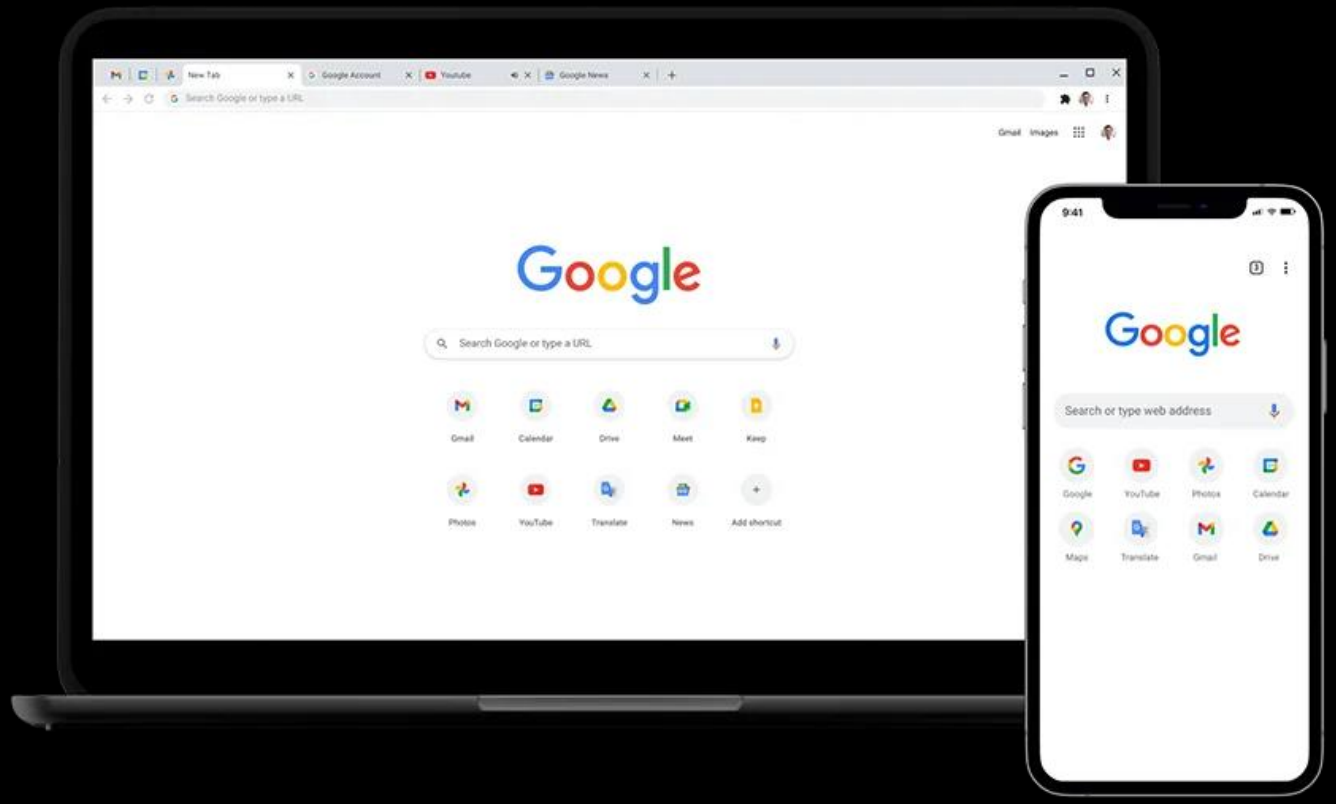

File explorer merupkan contoh bentuk file system. Dalam file explorer banyak proses pengelolaan yang dapat dilakukan.

Segala proses pengelolaan dalam file system sangat penting karena berguna untuk membantu dalam manajemen file.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

What is "**FILE SYSTEM**" ?

File system mendefinisikan bagaimana file dikelola (seperti: diberi nama, disimpan dan dibuka kembali) dari sebuah media penyimpnan (seperti: hard drive, flashdisk dsb).




Search engine seperti google juga merupakan salah satu bentuk file system.

Dalam search engine, kita mengakses data dari server. Dalam server tersebut juga memiliki file management system agar proses akses data oleh pengguna dapat dilakukan dengan mudah.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

What is "**FILE SYSTEM**" ?

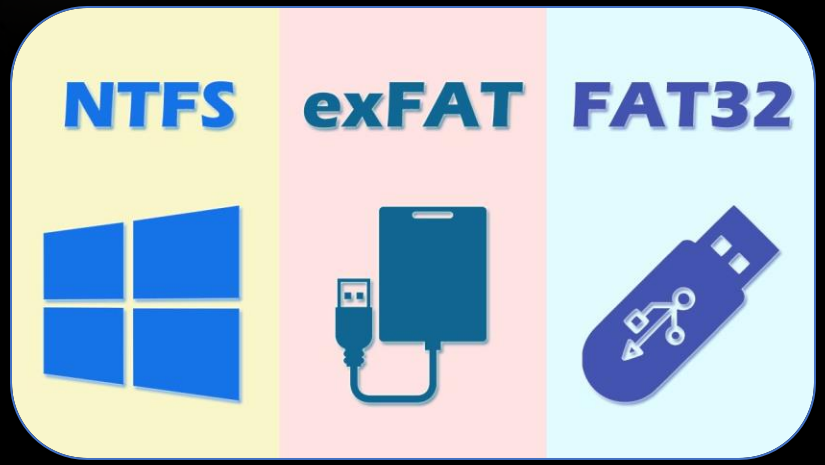
File system mendefinisikan bagaimana file dikelola (seperti: diberi nama, disimpan dan dibuka kembali) dari sebuah media penyimpnan (seperti: hard drive, flashdisk dsb).


Tidak hanya file explorer dan search engine, external drive seperti flashdisk, harddisk, dan SSD juga memiliki file system untuk melakukan manajemen file.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

What is "**FILE SYSTEM**" ?

File system memiliki standar yang berbeda pada setiap sistem operasi, seperti berikut:




Windows


Linux


MacOS

Big data juga harus memiliki file system, karena:

- Data tidak tersimpan di local drive
- File management dibutuhkan agar distribusi data dapat berjalan efisien

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

"**FILE SYSTEM**" in Big Data

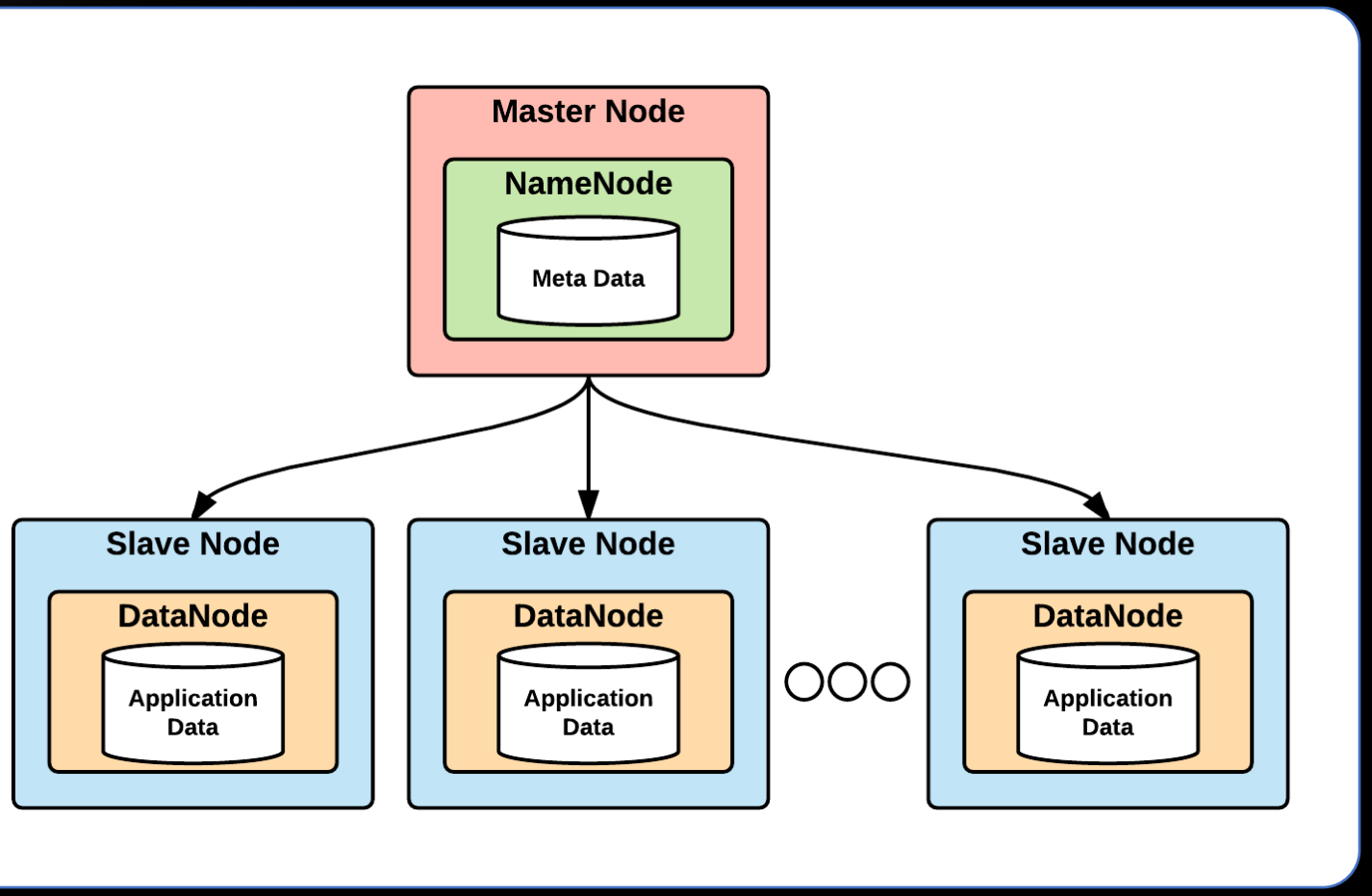
Di dalam big data kita mengenal istilah HDFS atau Hadoop Distributed File System.


HDFS berperan sebagai storage layer yang memiliki fungsi untuk mengatur dan mengelola data dalam server.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.




- **Master Node:**

  Berfungsi untuk mengendalikan distribusi data.

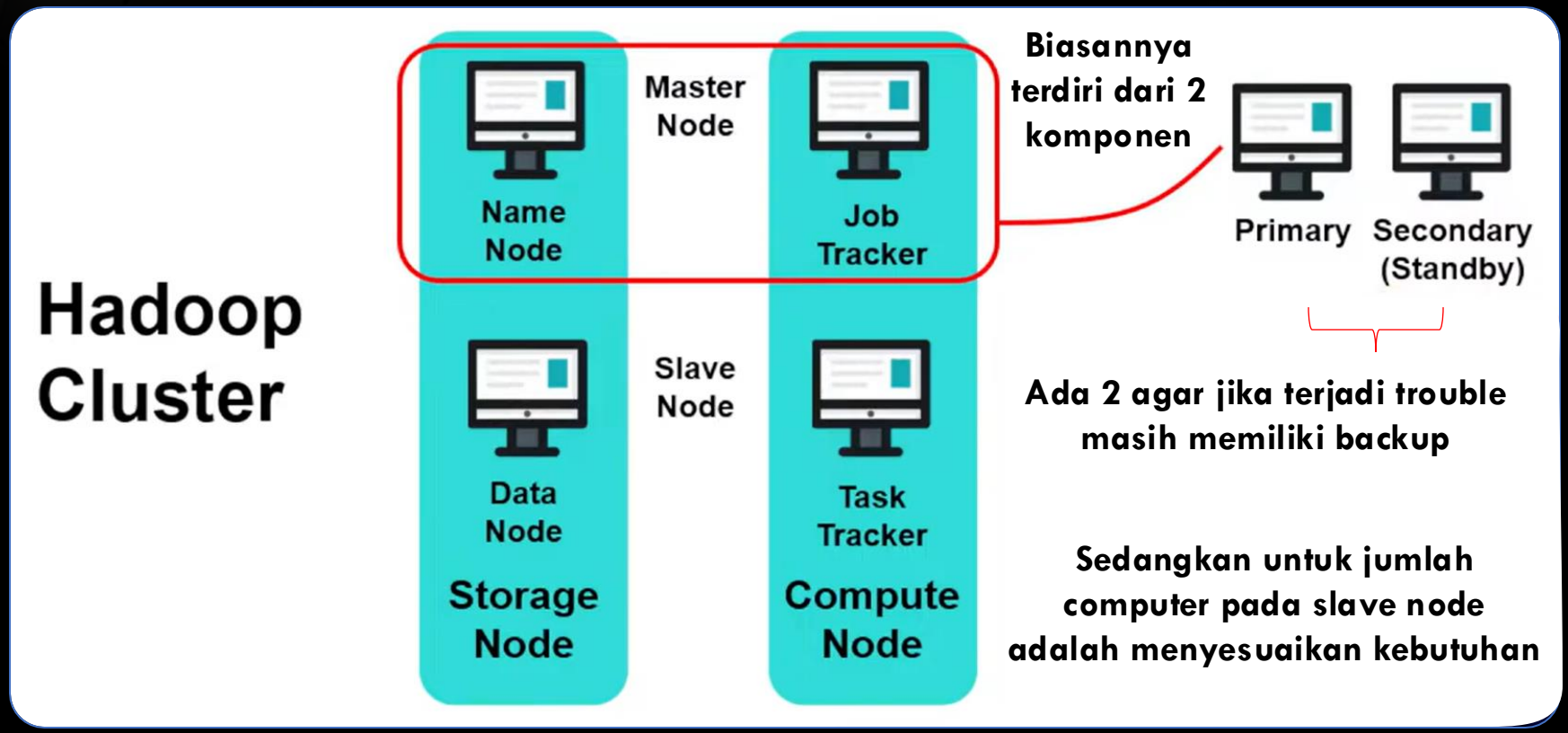
- **Slave Node:**

  Berfungsi untuk memproses task atau tugas dari Master Node.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.




- **Master Node**:

  Disebut dengan Name Node memiliki fungsi sebagai **Job Tracker**, yakni memberikan instruksi pada Slave Node

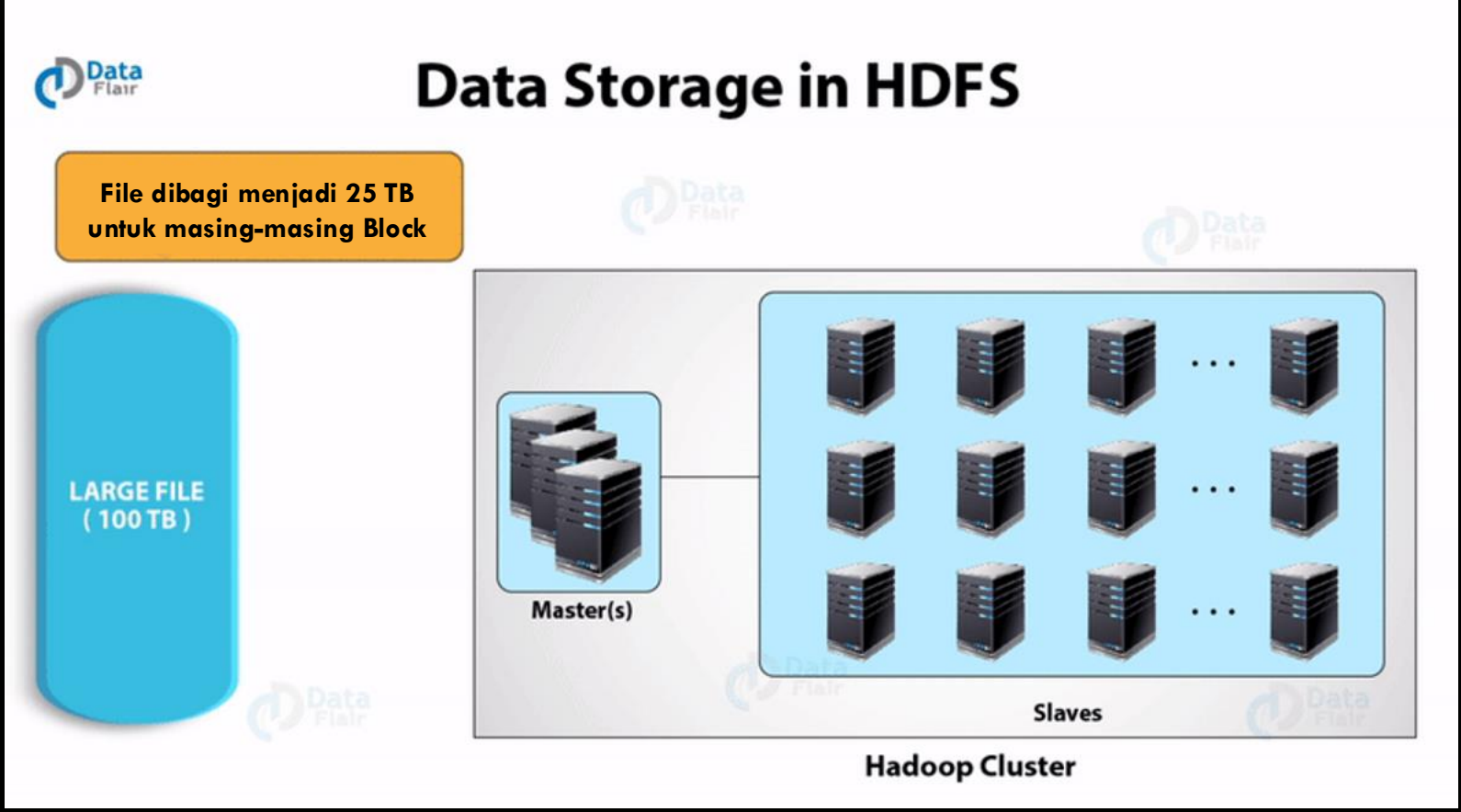
- **Slave Node**:

  Disebut dengan Data Node memiliki fungsi sebagai Task Tracker, yakni memastikan semua instruksi dari Master Node dikerjakan.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

## Architecture of HDFS

HDFS menggunakan arsitektur Master-Slave.



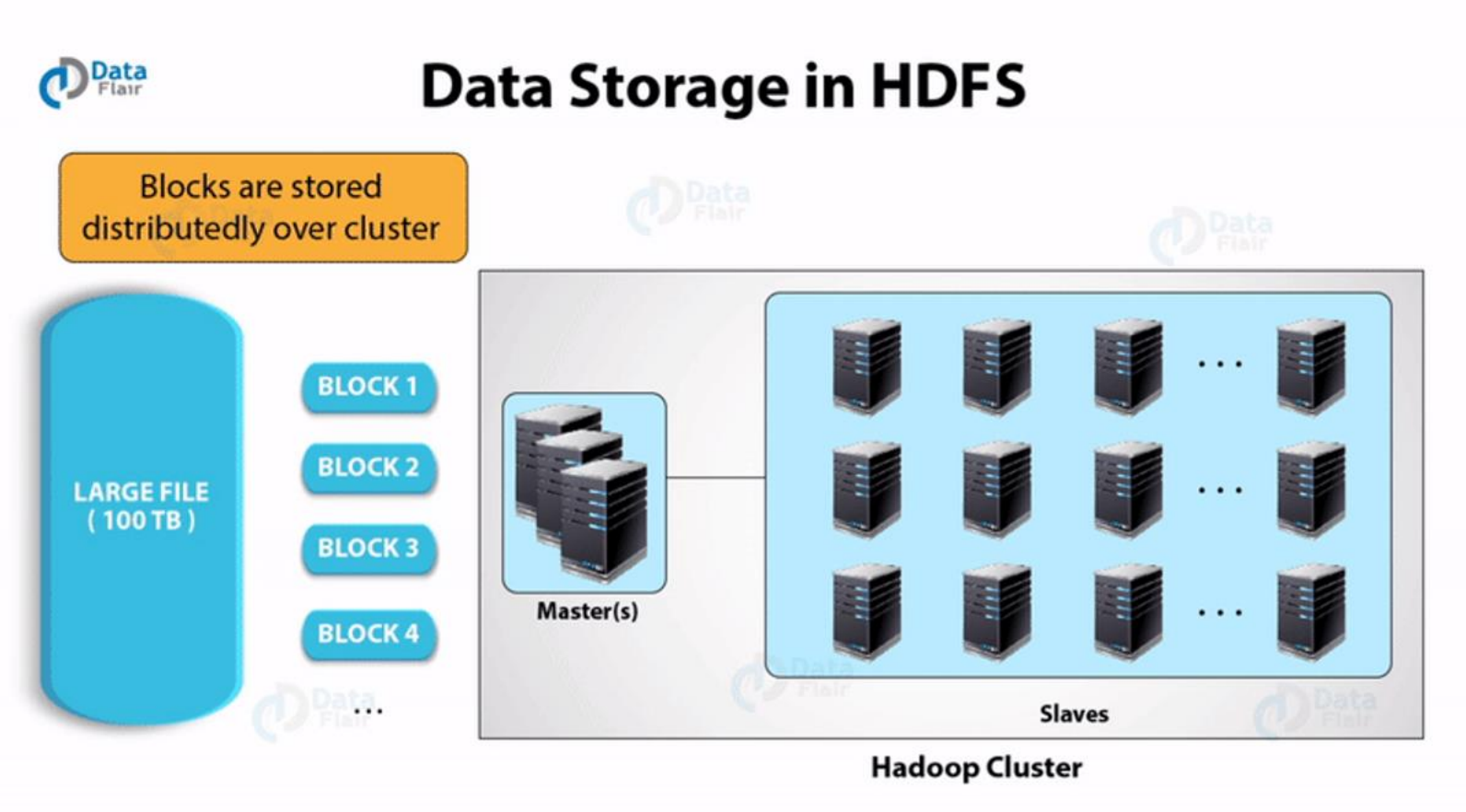

**Berikut adalah cara kerja HDFS:**

- Misal kita memiliki data 100 TB dan akan didistribusikan ke seluruh slave
- Langkah awal kita perlu untuk membuat "**Block**". Block berfungsi untuk membantu pembagian data sebelum masuk dalam masing-masing data node atau slave node.
- Dalam penentuan jumlah block kita perlu menentukan besar data default yang akan disimpan pada masing-masing block. Misal pada gambar di samping besar data masing-masing block adalah 25 TB.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.



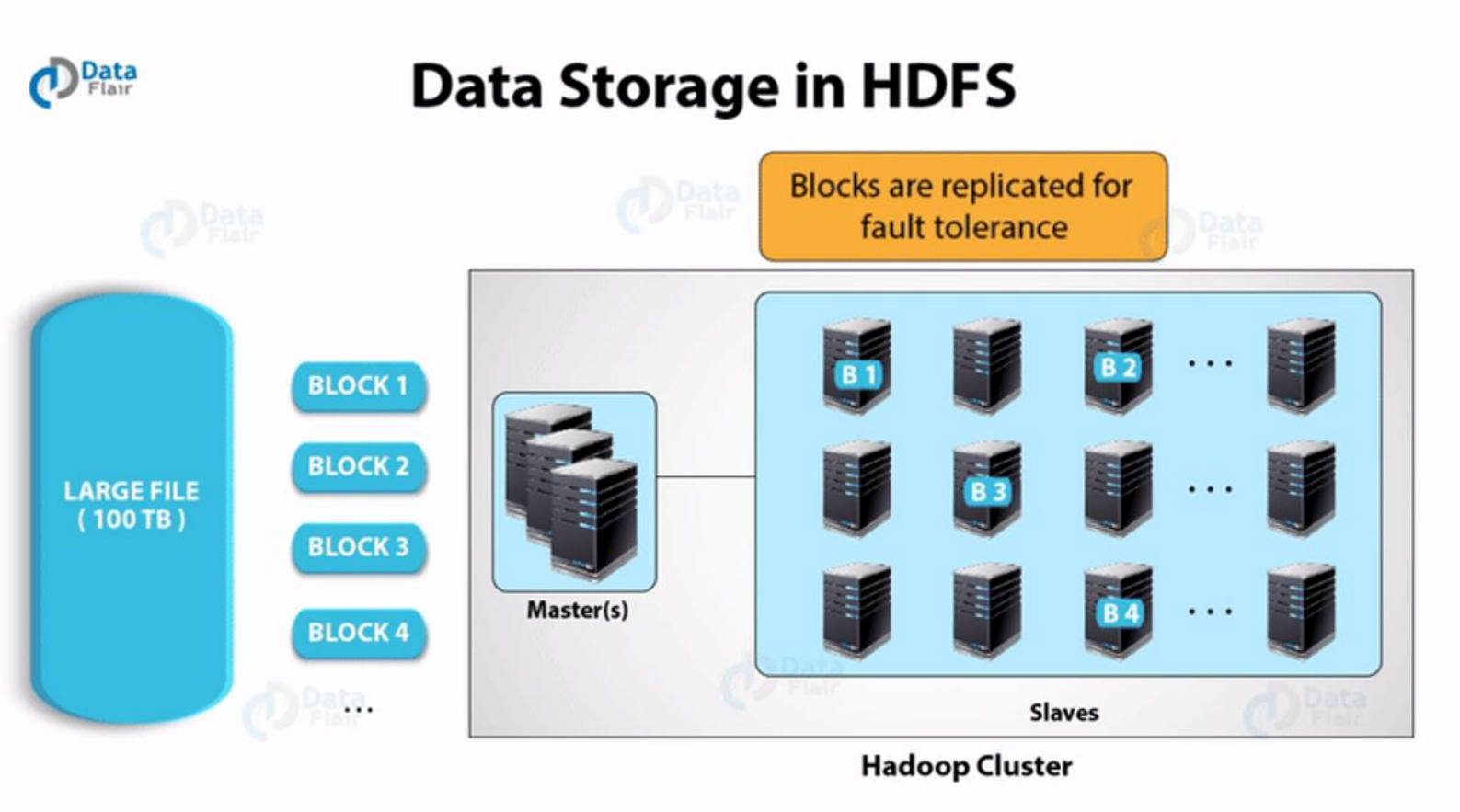

**Berikut adalah cara kerja HDFS:**

- Besar data setiap block dapat berbeda sesuai dengan kapasitas slave.
- Misalnya pada gambar di samping data dibagi dalam 4 block.
- 4 block tersebut didistribusikan pada masing-masing slave node.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.



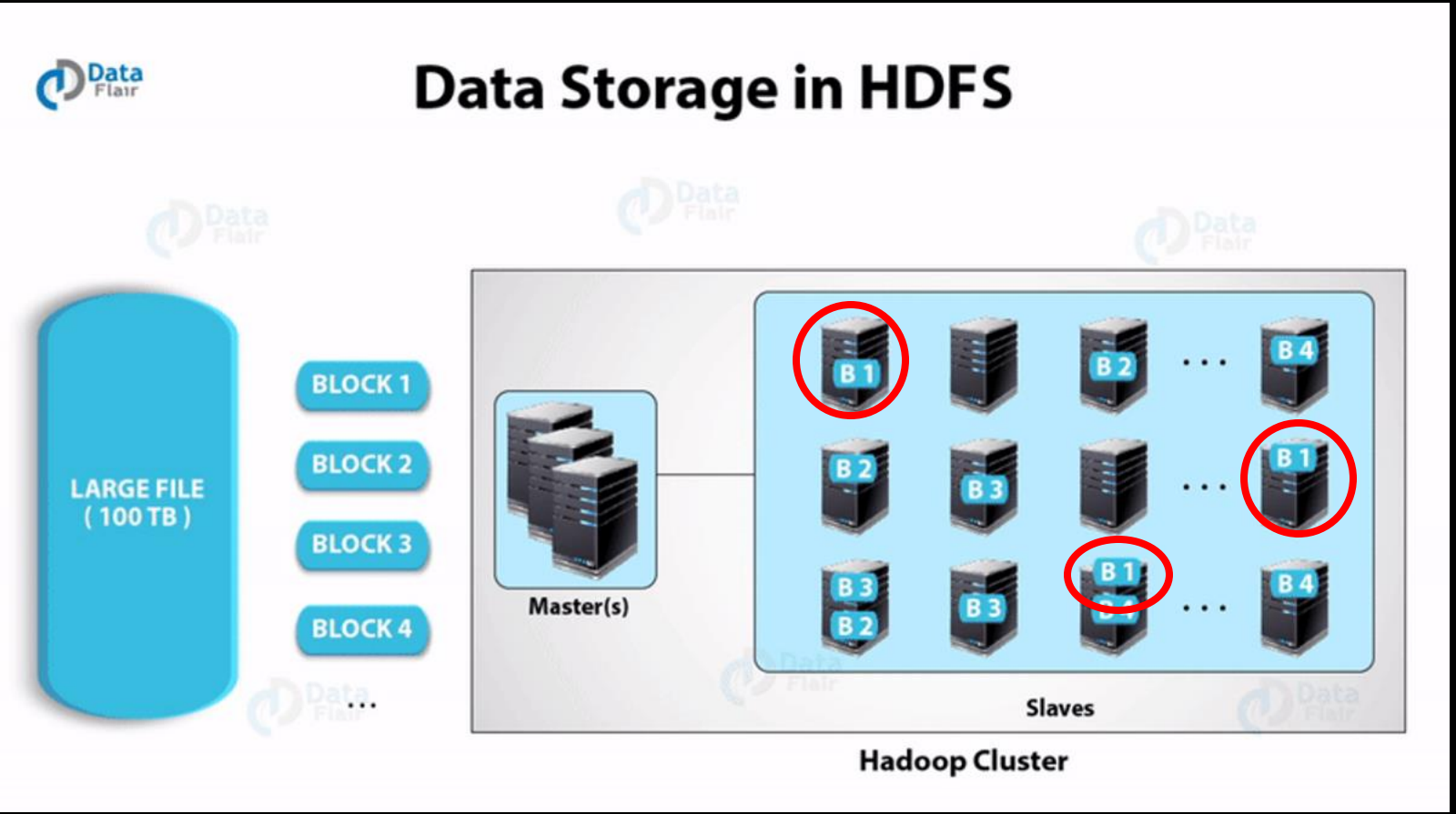

**Berikut adalah cara kerja HDFS:**

- Berikut adalah contoh pembagian pada masing-masing slave.
- Kemudian block di duplikasi.
- Tujuan dari proses duplikasi adalah agar jika terjadi kegagalan distribusi data pada salah satu node, kita masih memiliki backup pada block yang diduplikasi.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.



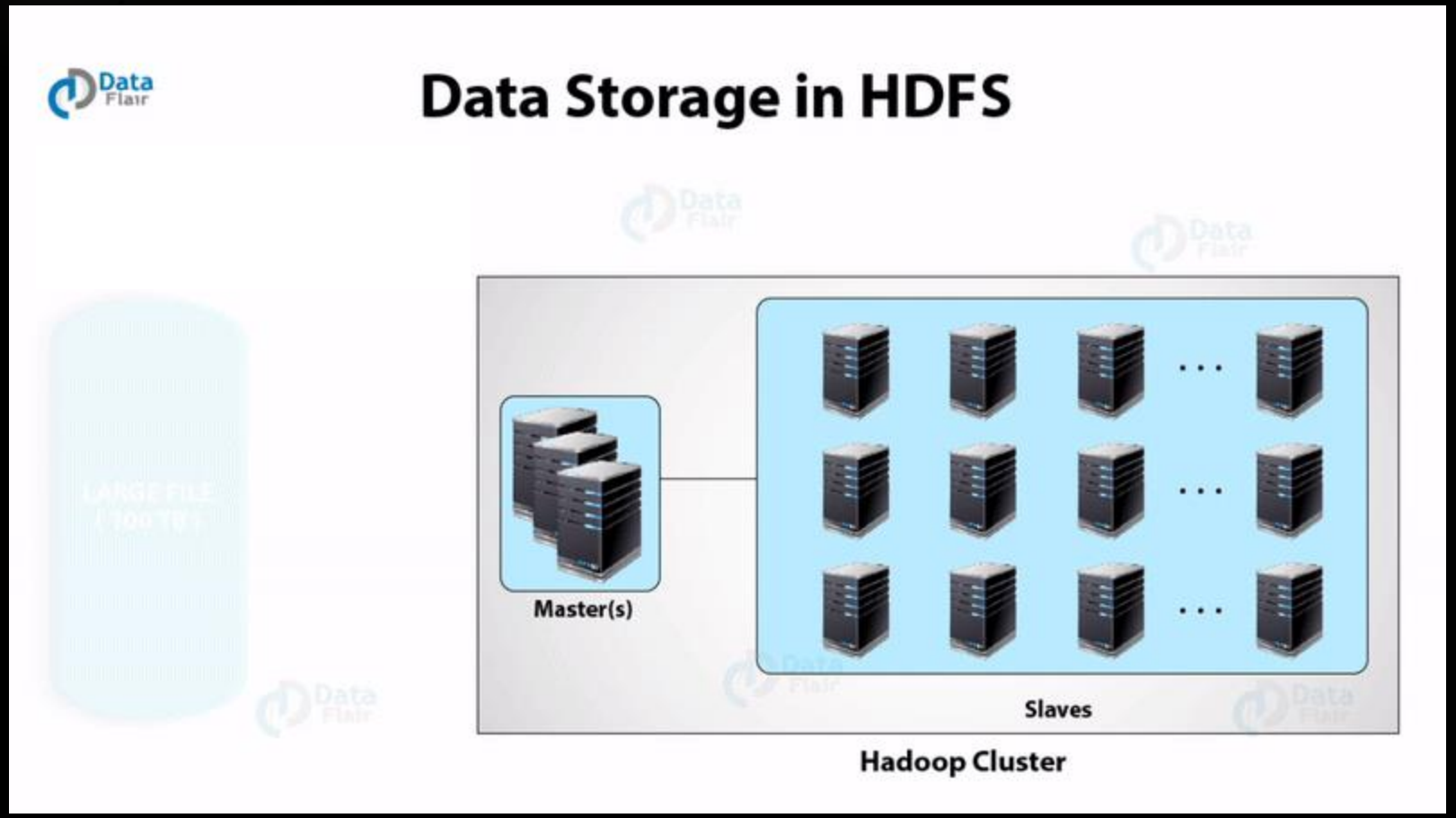

**Berikut adalah cara kerja HDFS:**

- Berikut adalah contoh hasil duplikasi masing-masing block.
- Pada saat menduplikasi block kita perlu menentukan berapa kali kita menduplikasi block.
- Misal pada gambar di samping masing-masing block di duplikasi sebanyak 3 kali

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.




Proses disamping kemudian akan menghasilkan meta data berikut (contoh):

File name and size : abcd.dat (100 TB)
File Permission : read, write, execute

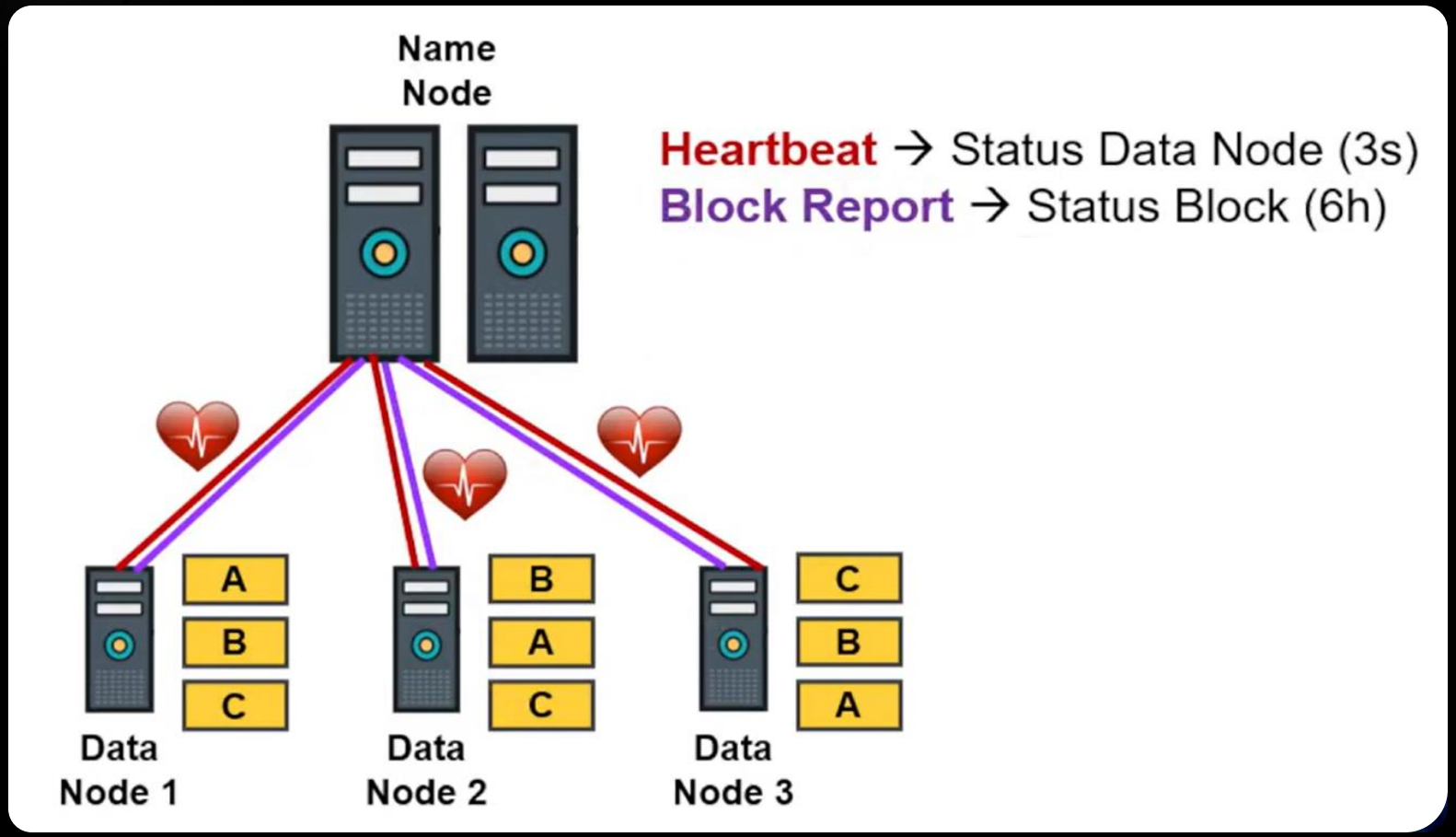
Replication factor : 3
Block : 1, 2, 3, 4
1 : DN1, DN3, DN4
2 : DN1, DN1, DN3
3 : DN1, DN2, DN2
4 : DN3, DN4, DN4

Total files/directories in cluster : 1
Total DataNodes in cluster : 5

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

## Architecture of HDFS

HDFS menggunakan arsitektur Master-Slave.




**Dalam HDFS terdapat dua proses pengecekan yang selalu berjalan, kedua proses tersebut adalah:**

- **Heartbeat**:

  Berfungsi untuk memastikan bahwa transfer data masih berjalan dengan normal. Pengecekan dilakukan setiap 3 detik.

- **Block report**:

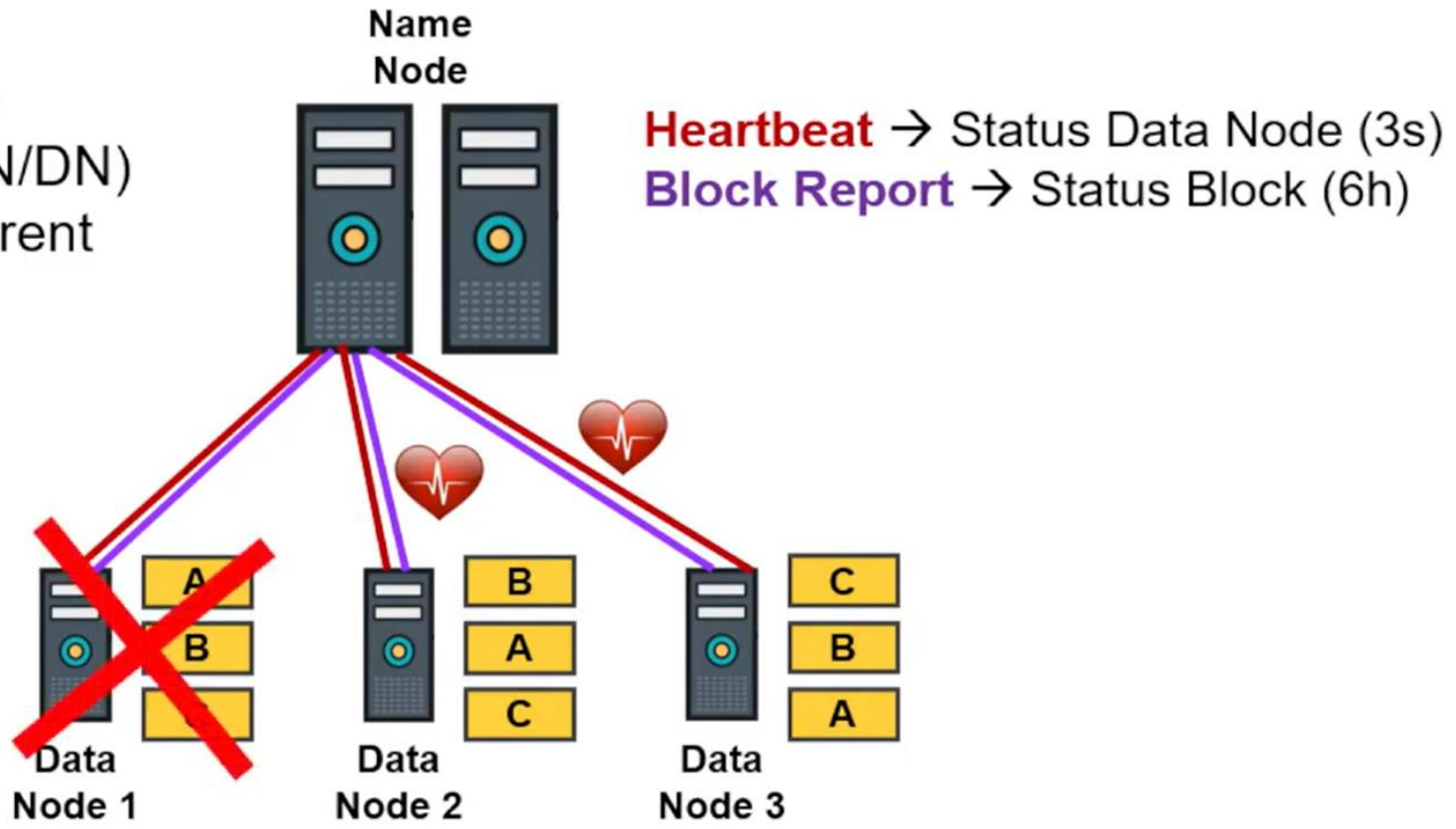
  Memberikan isyarat jika terjadi kesalahan pada proses transfer data. Proses non-aktif data node dilakukan setelah 6 jam tidak memberikan respon.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Architecture of HDFS**

HDFS menggunakan arsitektur Master-Slave.

- Network Failure
- Disk Failure (NN/DN)
- Block Size Different

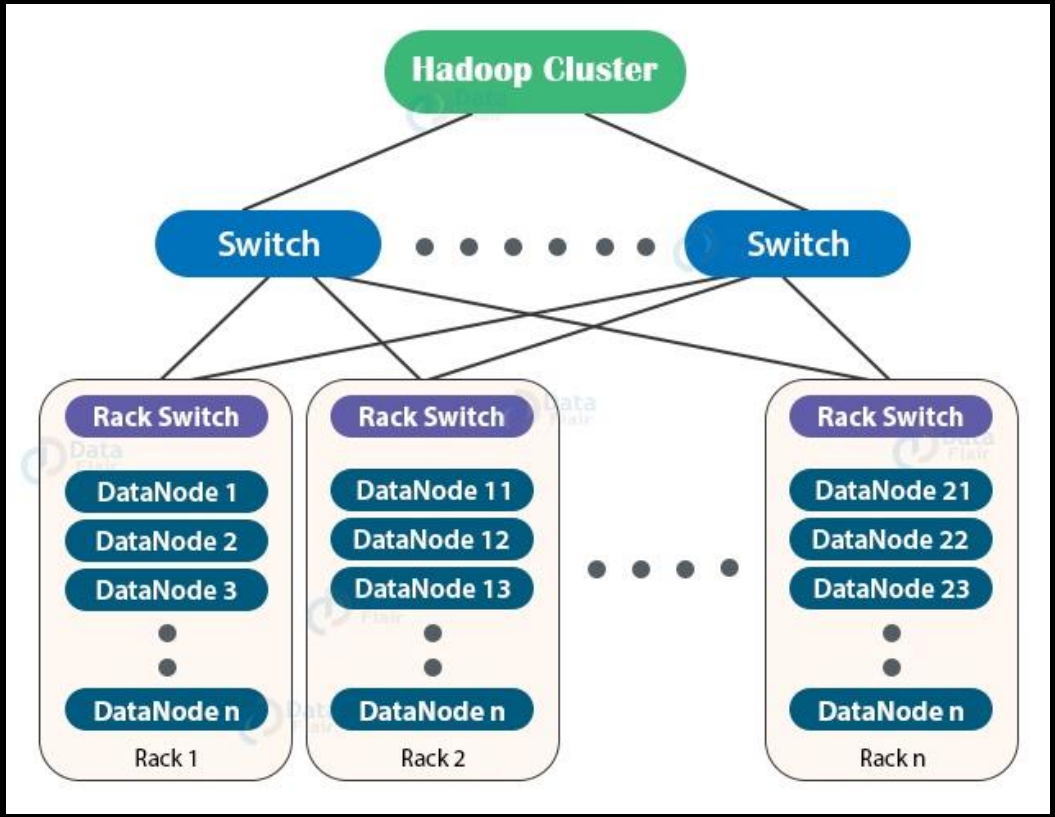



Pada saat terdapat data node yang di non-aktif kan maka proses transfer data pada node tersebut dihentikan.

# HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

**Rack Awareness**

Proses yang bertujuan untuk membuat data sedekatkan mungkin dengan master nodes, sehingga jika terjadi kegagalan transfer data maka dapat segera diselesaikan.




**The reasons for the Rack Awareness in Hadoop are:**

- To reduce the network traffic while file read/write, which improves the cluster performance.
- To achieve fault tolerance, even when the rack goes down.
- Achieve high availability of data so that data is available even in unfavorable conditions.
- To reduce the latency, that is, to make the file read/write operations done with lower delay.