

LEARNING PROGRESS REVIEW

WEEK 02

Scudetto Ciano Syam
Nugroho Wahyu Saputra
Radityo Fajar Pamungkas
Puteri Sakinah Mantikasari Ag
Yayes Kasnanda Bintang
Rima Fitrianti Azahra
Farhad Salim

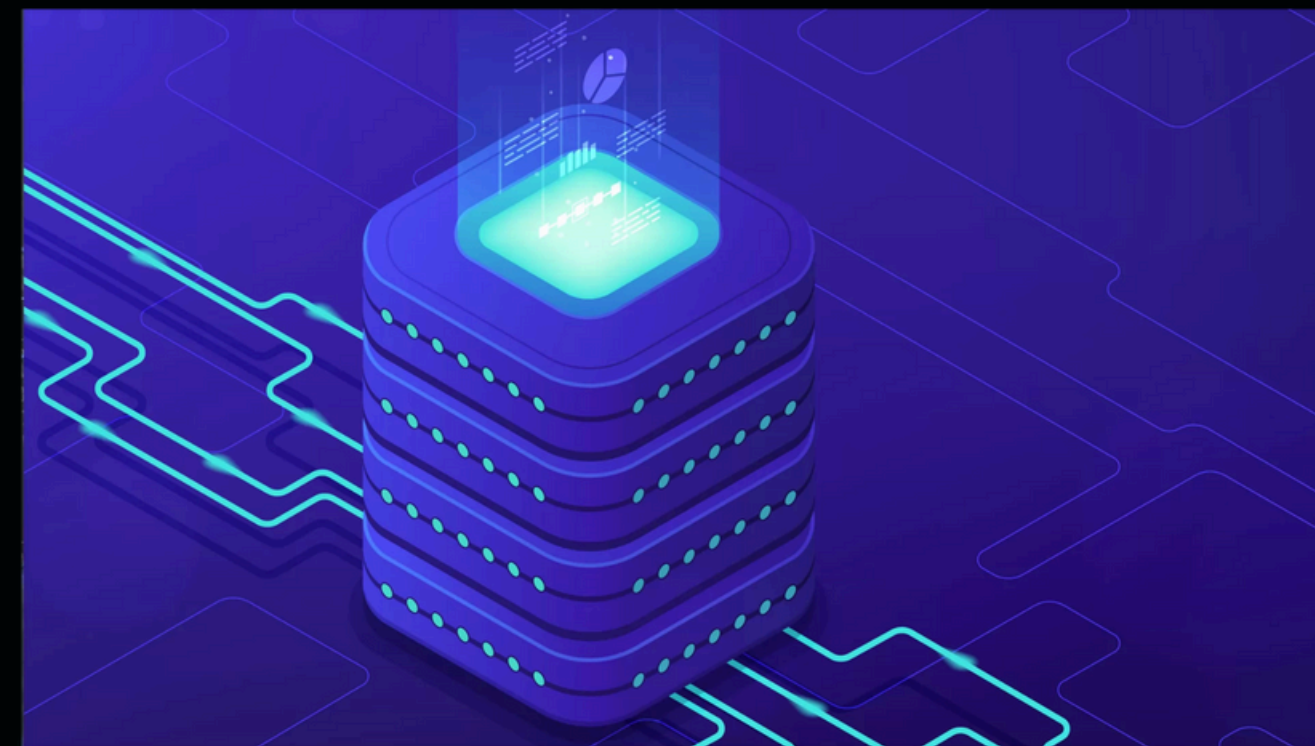


TABLE OF CONTENTS

01

SQL 1 (POSTGRE)

02

SQL 2

03

**GIT & VERSION
CONTROL SYSTEM**



SQL 1 (POSTGRE)

Penjelasan SQL beserta perintah-perintahnya

Apa itu SQL?

SQL (Structured Query Language) merupakan bahasa standard untuk sistem manajemen dengan basis data relasional untuk mengakses data.

- SQL dapat menjalankan query pada database.
- SQL dapat mengambil data dari database.
- SQL dapat memasukkan catatan ke dalam database.
- SQL dapat memperbarui catatan dalam database.
- SQL dapat menghapus catatan dari database.

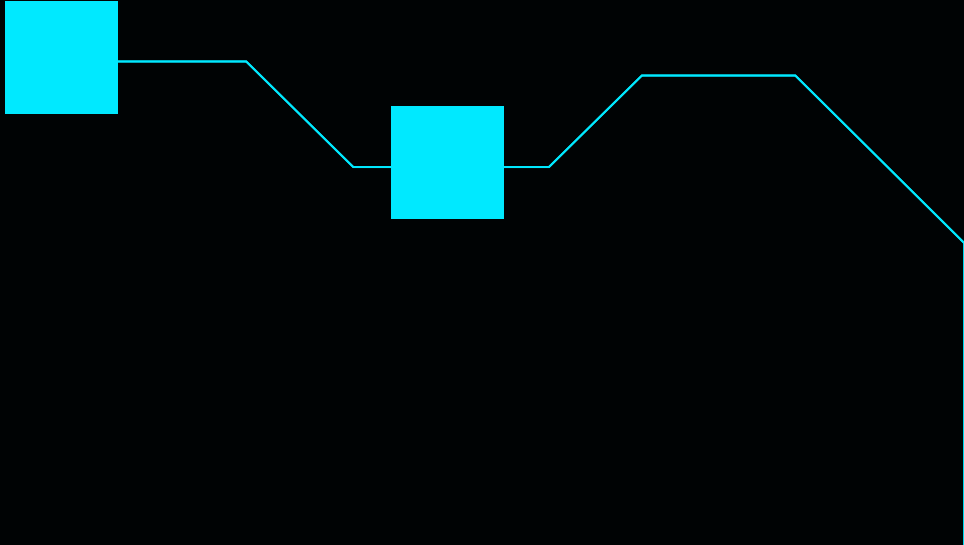
- SQL dapat membuat database baru.
- SQL dapat membuat tabel baru dalam database.
- SQL dapat membuat prosedur tersimpan dalam database.
- SQL dapat membuat tampilan dalam database.
- SQL dapat mengatur izin pada tabel, prosedur, dan tampilan.

Apa saja yang bisa SQL lakukan?



PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional (RDMS) open-source dimana sistemnya menggunakan dan terus mengembangkan bahasa pemrograman SQL. PostgreSQL juga menyediakan fitur-fitur lainnya untuk mengelola data agar lebih efisien.





PostgreSQL

Kelebihan postgresQL:

- **PostgreSQL gratis** : open-source yang dapat digunakan tanpa biaya dan tanpa memerlukan perjanjian lisensi.
- **PostgreSQL aman** : tidak mudah dibeli atau diambil alih oleh entitas manapun tanpa persetujuan dari semua perusahaan yang terlibat dalam pengelolaannya.
- **Dapat direplikasi**

JENIS COMMAND DALAM SQL

Data Definition Language (DDL)

Perintah untuk mendefinisikan data seperti membuat tabel database baru, mengubah dataset, dan menghapus data.

- **CREATE:** Untuk membuat tabel baru.
- **ALTER:** Untuk mengubah struktur tabel.
- **RENAME:** Untuk mengubah sebuah nama di sebuah tabel.
- **DROP:** Untuk menghapus database, table, dan kolom.
- **SHOW:** Untuk menampilkan sebuah tabel yang ada.

Data Manipulation Language (DML)

Perintah SQL untuk memanipulasi data.

- **INSERT:** Untuk memasukkan sebuah record baru didalam tabel database.
- **SELECT:** Untuk mengambil dan menampilkan sebuah data pada tabel.
- **UPDATE:** Untuk memperbaharui sebuah data pada tabel yang sudah ada di database.
- **DELETE:** Untuk menghapus sebuah record yang ada dalam tabel.

Data Control Language (DML)

Perintah SQL yang digunakan untuk mengatur hak akses pengguna terhadap objek dalam database.

- **GRANT:** Memberikan wewenang atau hak akses tertentu kepada pengguna untuk melakukan tindakan pada objek dalam basis data.
- **REVOKE:** Mencabut atau menghapus wewenang atau hak akses tertentu dari pengguna.



CREATE

CREATE digunakan untuk membuat sebuah objek baru dalam SQL (tabel baru, atau sebuah constraint, stored procedure, index, views, dan lainnya.)

```
CREATE TABLE table_name (  
  column1 datatype,  
  column2 datatype,  
  column3 datatype,  
  ....  
);
```

```
1 CREATE TABLE discovrio_employees (  
2   id INT PRIMARY KEY,  
3   first_name VARCHAR(50),  
4   last_name VARCHAR(50),  
5   email VARCHAR(100),  
6   phone_number VARCHAR(20),  
7   hire_date DATE,  
8   job_title VARCHAR(50),  
9   salary DECIMAL(10, 2)  
10 );
```

2 INSERT

INSERT dalam SQL biasa dituliskan bersamaan dengan **INTO**, perintah ini dalam SQL yang berfungsi untuk memasukkan data ke dalam sebuah tabel di dalam database.

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);
```

```
1 INSERT INTO discovrio_employees (id, first_name, last_name, email, phone_number, hire_date, job_title, salary)
2 VALUES
3 (1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '2020-01-15', 'Software Engineer', 70000.00),
4 (2, 'Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '2019-03-22', 'Project Manager', 85000.00);
```



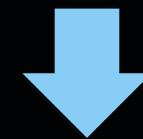
id	first_name	last_name	email	phone_number	hire_date	job_title	salary
1	John	Doe	john.doe@example.com	123-456-7890	2020-01-15	Software Engineer	70000.00
2	Jane	Smith	jane.smith@example.com	987-654-3210	2019-03-22	Project Manager	85000.00

3 UPDATE

UPDATE digunakan untuk mengubah data dalam tabel. Dan biasanya diikuti dengan perintah **WHERE** untuk mengubah data sesuai dengan kriteria yang kita inginkan.

UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;

```
1 UPDATE discovrio_employees
2 SET salary = 75000.00
3 WHERE id = 1;
```



id	first_name	last_name	email	phone_number	hire_date	job_title	salary
1	John	Doe	john.doe@example.com	123-456-7890	2020-01-15	Software Engineer	75000.00
2	Jane	Smith	jane.smith@example.com	987-654-3210	2019-03-22	Project Manager	85000.00



SELECT

SELECT merupakan fungsi SQL untuk mendapatkan data.

```
SELECT column1, column2, ...  
FROM table_name;
```

```
SELECT * FROM discovrio_employees WHERE job_title = 'Software Engineer';
```



id ▲	first_name ▲	last_name ▲	email ▲	phone_number ▲	hire_date ▲	job_title ▲	salary ▲
1	John	Doe	john.doe@example.com	123-456-7890	2020-01-15	Software Engineer	75000.00

5

ALTER

ALTER digunakan untuk **memodifikasi objek** yang sudah ada, bisa digunakan untuk menambahkan kolom baru, menambah atau menghapus constraint ataupun index, atau mengubah tipe data,

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
1 ALTER TABLE discovrio_employees  
2 ADD discovrio_department VARCHAR(50);
```



last_name	email	phone_number	hire_date	job_title	salary	discovrio_department
Doe	john.doe@example.com	123-456-7890	2020-01-15	Software Engineer	75000.00	(NULL)
Smith	jane.smith@example.com	987-654-3210	2019-03-22	Project Manager	85000.00	(NULL)

6 DELETE

DELETE berfungsi untuk menghapus data dalam tabel. apabila kita tidak menyertakan perintah **WHERE**, maka semua data dalam tabel akan terhapus.

DELETE FROM table_name
WHERE condition;

```
1 DELETE FROM discovrio_employees
2 WHERE id = 2;
```



id	first_name	last_name	email	phone_number	hire_date	job_title	salary
1	John	Doe	john.doe@example.com	123-456-7890	2020-01-15	Software Engineer	75000.00



SQL 2

Beberapa perintah SQL lainnya

1 DISTINCT

SELECT DISTINCT adalah perintah SQL yang digunakan untuk hanya mengembalikan nilai yang berbeda. Jika terdapat nilai yang sama atau duplikat di dalam tabel, dengan SELECT DISTINCT akan memunculkan nilai tersebut hanya sekali pada hasil query

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

```
1 SELECT DISTINCT job_title  
2 FROM discovrio_employees
```

job_title
Marketing Director
Web Developer
Manager
Project Manager
Sales Associate
Financial Analyst
Accountant
Data Analyst
HR Specialist
Sales Executive
Marketing Manager



WHERE

WHERE adalah perintah SQL yang digunakan untuk memfilter data berdasarkan kondisi tertentu.

Dengan perintah **SELECT**, hasil query akan menampilkan data dengan kondisi atau kriteria yang sesuai di dalam query.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
1 SELECT first_name, hire_date, job_title  
2 FROM discovrio_employees  
3 WHERE job_title = 'Data Analyst'
```

first_name ▲	hire_date ▲	job_title ▲
Charles	2023-07-08	Data Analyst
Samantha	2023-03-01	Data Analyst
Justin	2023-07-17	Data Analyst
Amanda	2024-07-05	Data Analyst



3 STRING FUNCTIONS

String Functions adalah kumpulan-kumpulan fungsi SQL yang digunakan untuk memanipulasi data tipe string. Berikut beberapa contoh fungsi string yang sering digunakan :

LOWER()	Mengubah string menjadi huruf kecil
UPPER()	Mengubah string menjadi huruf besar
LENGTH()	Menghitung panjang karakter dari sebuah string
CONCAT()	Menggabungkan beberapa string menjadi satu
SUBSTRING()	Mengekstrak atau memotong beberapa karakter string sesuai dengan kondisi yang ditentukan

```
1 SELECT first_name, LOWER(first_name)
2 FROM discovrio_employees
3 LIMIT 3
```

first_name ▲	lower ▲
John	john
Michael	michael
Sarah	sarah

LOWER()

Mengubah string menjadi huruf kecil

`LOWER(string)`

```
1 SELECT first_name, UPPER(first_name)
2 FROM discovrio_employees
3 LIMIT 3
```

first_name ▲	upper ▲
John	JOHN
Michael	MICHAEL
Sarah	SARAH

UPPER()

Mengubah string menjadi huruf besar

`UPPER(string)`

```
1 SELECT first_name, LENGTH(first_name)
2 FROM discovrio_employees
3 LIMIT 3
```

first_name ▲	length ▲
John	4
Michael	7
Sarah	5

LENGTH()

Menghitung panjang karakter dari sebuah string

`LENGTH(string)`

```
1 SELECT first_name, last_name, CONCAT(first_name, ' ', last_name)
2 FROM discovrio_employees
3 LIMIT 3
```

first_name ▲	last_name ▲	concat ▲
John	Doe	John Doe
Michael	Johnson	Michael Johnson
Sarah	Martinez	Sarah Martinez

CONCAT()

Menggabungkan beberapa string menjadi satu

`CONCAT(string1, string2, ..., string_n)`

```
1 SELECT first_name, SUBSTRING(first_name, 1,2)
2 FROM discovrio_employees
3 LIMIT 3
```

first_name ▲	substring ▲
John	Jo
Michael	Mi
Sarah	Sa

SUBSTRING()

Mengekstrak atau memotong beberapa karakter string sesuai dengan kondisi yang ditentukan

SUBSTRING(*string, start, length*)

4 AGGREGATE FUNCTIONS

Aggregate Functions adalah kumpulan-kumpulan fungsi SQL yang melakukan perhitungan atau kalkulasi pada sekumpulan nilai dari data, lalu mengembalikannya menjadi satu nilai tunggal

AVG()	Menghitung nilai rata-rata dari kolom yang dipilih
SUM()	Menghitung total nilai dari kolom yang dipilih
MAX()	Mengembalikan nilai terbesar dari kolom yang dipilih
MIN()	Mengembalikan nilai terkecil dari kolom yang dipilih
COUNT()	Menghitung jumlah baris pada suatu kolom

AVG()

Menghitung nilai rata-rata dari kolom yang dipilih

```
SELECT AVG(column_name)
FROM table_name
[GROUP BY column_name]
```

```
1 SELECT job_title, AVG(salary)
2 FROM discovrio_employees
3 GROUP BY job_title
```

job_title	avg
Web Developer	62600.000000000000
Manager	50000.000000000000
Project Manager	57500.000000000000
Sales Associate	70000.000000000000
Financial Analyst	53333.333333333333
Accountant	50000.000000000000
Data Analyst	58250.000000000000
HR Specialist	62600.000000000000
Sales Executive	80000.000000000000
Marketing Manager	47500.000000000000

SUM()

Menghitung total nilai dari kolom yang dipilih

```
SELECT SUM(column_name)
FROM table_name
[GROUP BY column_name]
```

```
1 SELECT job_title, SUM(salary)
2 FROM discovrio_employees
3 GROUP BY job_title
```

job_title	sum
Web Developer	626000.00
Manager	50000.00
Project Manager	115000.00
Sales Associate	70000.00
Financial Analyst	160000.00
Accountant	50000.00
Data Analyst	233000.00
HR Specialist	313000.00
Sales Executive	80000.00
Marketing Manager	95000.00

MAX()

Mengembalikan nilai terbesar dari kolom yang dipilih

```
SELECT MAX(column_name)
FROM table_name
[GROUP BY column_name]
```

```
1 SELECT job_title, MAX(salary)
2 FROM discovrio_employees
3 GROUP BY job_title
```

job_title	max
Web Developer	90000.00
Manager	50000.00
Project Manager	70000.00
Sales Associate	70000.00
Financial Analyst	60000.00
Accountant	50000.00
Data Analyst	75000.00
HR Specialist	85000.00
Sales Executive	80000.00
Marketing Manager	55000.00

MIN()

Mengembalikan nilai terkecil dari kolom yang dipilih

```
SELECT MIN(column_name)  
FROM table_name  
[GROUP BY column_name]
```

```
1 SELECT job_title, MIN(salary)  
2 FROM discovrio_employees  
3 GROUP BY job_title
```

job_title	min
Web Developer	35000.00
Manager	50000.00
Project Manager	45000.00
Sales Associate	70000.00
Financial Analyst	45000.00
Accountant	50000.00
Data Analyst	38000.00
HR Specialist	48000.00
Sales Executive	80000.00
Marketing Manager	40000.00

COUNT()

Menghitung jumlah baris
pada suatu kolom

```
SELECT COUNT(column_name)  
FROM table_name  
[GROUP BY column_name]
```

```
1 SELECT job_title, count(job_title)  
2 FROM discovrio_employees  
3 GROUP BY job_title
```

job_title	count
Web Developer	10
Manager	1
Project Manager	2
Sales Associate	1
Financial Analyst	3
Accountant	1
Data Analyst	4
HR Specialist	5
Sales Executive	1
Marketing Manager	2

5 GROUP BY

GROUP BY adalah klausa yang digunakan untuk melakukan pengelompokan terhadap data sesuai dengan kategorinya. Apabila dalam satu kelompok terdapat data yang duplikat, maka data yang ditampilkan hanya satu. Klausa ini biasanya digunakan bersamaan dengan klausa SQL yang lain. Contohnya :



WHERE



ORDER BY

GROUP BY, WHERE, ORDER BY

Penggunaan **GROUP BY** dengan
WHERE & ORDER BY

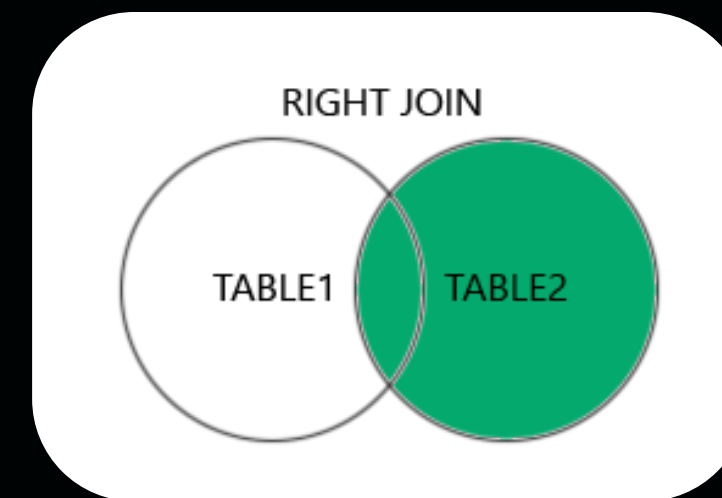
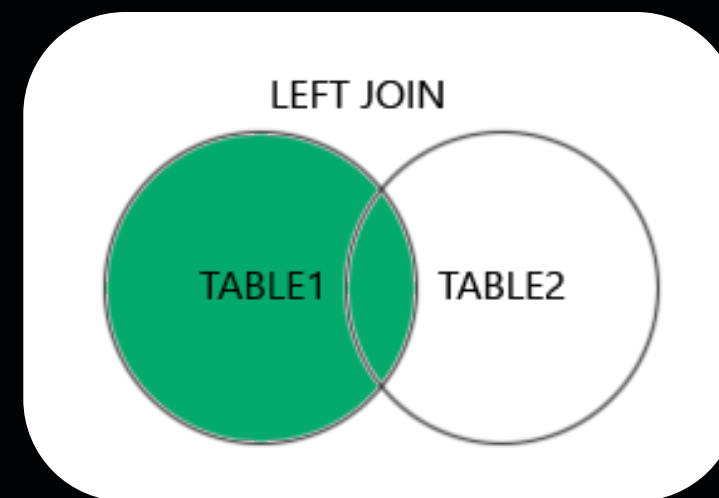
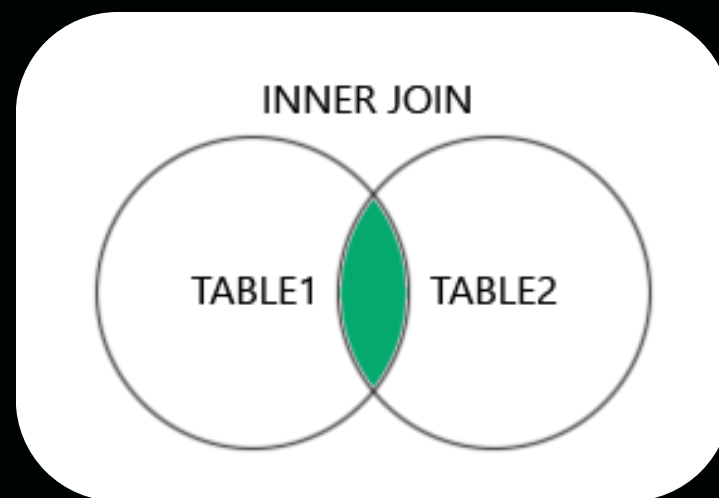
```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

```
1 SELECT job_title, AVG(salary) as avg_salary
2 FROM discovrio_employees
3 WHERE department_id = 3
4 GROUP BY job_title
5 ORDER BY avg_salary
```

job_title	avg_salary
Financial Analyst	53333.333333333333
HR Specialist	62600.000000000000
Sales Associate	70000.000000000000
Sales Executive	80000.000000000000

6 JOIN TABLE

JOIN adalah klausa yang digunakan untuk menggabungkan dua atau lebih tabel yang saling berelasi menjadi satu tabel. Ada beberapa tipe 3 JOIN dasar dalam SQL, diantaranya:



INNER JOIN

Menggabungkan kedua tabel yang berelasi dan menampilkan data yang sama atau saling terhubung

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name =
table2.column_name;
```

```
1 SELECT *
2 FROM discovrio_employees e
3 INNER JOIN discovrio_departments d
4 ON e.department_id = d.id
5
```

id	first_name	last_name	email	phone_number	hire_date	job_title	salary	department_id	id	name
1	John	Doe	johndoe@example.com	123-456-7890	2023-01-01	Manager	50000.00	1	1	Management
3	Michael	Johnson	michaeljohnson@example.com	555-123-4567	2023-03-10	Financial Analyst	45000.00	3	3	Finance
6	Sarah	Martinez	sarahmartinez@example.com	777-888-9999	2023-06-10	HR Specialist	48000.00	3	3	Finance
7	Christopher	Garcia	christophergarcia@example.com	222-333-4444	2023-07-15	Marketing Manager	55000.00	1	1	Management
8	Jessica	Lopez	jessicalopez@example.com	555-666-7777	2023-08-01	Web Developer	65000.00	2	2	Engineering
9	Kevin	Gonzalez	kevgonzalez@example.com	888-999-0000	2023-09-05	Financial Analyst	60000.00	3	3	Finance
10	Megan	Perez	meganperez@example.com	123-456-7890	2023-10-10	Project Manager	70000.00	1	1	Management
16	Stephanie	Rivera	stephanierivera@example.com	111-777-2222	2024-04-20	Accountant	50000.00	1	1	Management
22	Charles	Martinez	charlesmartinez@example.com	444-888-5555	2023-07-08	Data Analyst	55000.00	1	1	Management
24	Jose	Gonzalez	josegonzalez@example.com	888-111-4444	2023-03-12	Sales Executive	80000.00	3	3	Finance
25	Samantha	Hernandez	samanthahernandez@example.com	333-666-9999	2023-03-01	Data Analyst	65000.00	1	1	Management
28	Justin	Ortiz	justinortiz@example.com	444-777-2222	2023-07-17	Data Analyst	75000.00	1	1	Management
19	Amanda	King	amandaking@example.com	555-111-8888	2024-07-05	Data Analyst	38000.00	1	1	Management
11	Andrew	Rodriguez	andrewrodriguez@example.com	321-654-9876	2023-11-15	Web Developer	80000.00	2	2	Engineering

LEFT JOIN

Menggabungkan kedua tabel yang berelasi dan menampilkan data yang sama dan semua data yang ada di tabel kiri

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name =
table2.column_name;
```

```
1 SELECT *
2 FROM discovrio_employees e
3 LEFT JOIN discovrio_departments d
4 ON e.department_id = d.id
5 ORDER BY department_id DESC
6
```

id	first_name	last_name	email	phone_number	hire_date	job_title	salary	department_id	id	name
12	Maria	Hernandez	mariahernandez@example.com	999-888-7777	2023-12-20	HR Specialist	60000.00	(NULL)	(NULL)	(NULL)
15	James	Torres	jamestorres@example.com	666-333-9999	2024-03-15	Financial Analyst	55000.00	(NULL)	(NULL)	(NULL)
6	Sarah	Martinez	sarahmartinez@example.com	777-888-9999	2023-06-10	HR Specialist	48000.00	3	3	Finance
9	Kevin	Gonzalez	kevgonzalez@example.com	888-999-0000	2023-09-05	Financial Analyst	60000.00	3	3	Finance
21	Nicole	Perez	nicoleperez@example.com	777-222-9999	2024-09-15	Sales Associate	70000.00	3	3	Finance
3	Michael	Johnson	michaeljohnson@example.com	555-123-4567	2023-03-10	Financial Analyst	45000.00	3	3	Finance
24	Jose	Gonzalez	josegonzalez@example.com	888-111-4444	2023-03-12	Sales Executive	80000.00	3	3	Finance
18	William	Sanchez	williamsanchez@example.com	888-444-2222	2024-06-01	HR Specialist	60000.00	3	3	Finance
27	Emma	Torres	emmatorres@example.com	111-444-8888	2023-05-20	HR Specialist	60000.00	3	3	Finance
30	Tyler	Flores	tylerflores@example.com	222-999-6666	2023-06-22	HR Specialist	85000.00	3	3	Finance
8	Jessica	Lopez	jessicalopez@example.com	555-666-7777	2023-08-01	Web Developer	65000.00	2	2	Engineering
11	Andrew	Rodriguez	andrewrodriguez@example.com	321-654-9876	2023-11-15	Web Developer	80000.00	2	2	Engineering
14	Jennifer	Diaz	jenniferdiaz@example.com	222-888-4444	2024-02-10	Web Developer	38000.00	2	2	Engineering
2	Jane	Smith	ianesmith@example.com	987-654-3210	2023-02-15	Web Developer	60000.00	2	2	Engineering

RIGHT JOIN

Menggabungkan kedua tabel yang berelasi dan menampilkan data yang sama dan semua data yang ada di tabel kanan

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name =
table2.column_name;
```

```
1 SELECT *
2 FROM discovrio_employees e
3 RIGHT JOIN discovrio_departments d
4 ON e.department_id = d.id
5 ORDER BY department_id DESC
6
```

id	first_name	last_name	email	phone_number	hire_date	job_title	salary	department_id	id	name
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	5	Administration
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	4	Production
6	Sarah	Martinez	sarahmartinez@example.com	777-888-9999	2023-06-10	HR Specialist	48000.00	3	3	Finance
9	Kevin	Gonzalez	kevgonzalez@example.com	888-999-0000	2023-09-05	Financial Analyst	60000.00	3	3	Finance
21	Nicole	Perez	nicoleperez@example.com	777-222-9999	2024-09-15	Sales Associate	70000.00	3	3	Finance
3	Michael	Johnson	michaeljohnson@example.com	555-123-4567	2023-03-10	Financial Analyst	45000.00	3	3	Finance
24	Jose	Gonzalez	josegonzalez@example.com	888-111-4444	2023-03-12	Sales Executive	80000.00	3	3	Finance
18	William	Sanchez	williamsanchez@example.com	888-444-2222	2024-06-01	HR Specialist	60000.00	3	3	Finance
27	Emma	Torres	emmatorres@example.com	111-444-8888	2023-05-20	HR Specialist	60000.00	3	3	Finance
30	Tyler	Flores	tylerflores@example.com	222-999-6666	2023-06-22	HR Specialist	85000.00	3	3	Finance
8	Jessica	Lopez	jessicalopez@example.com	555-666-7777	2023-08-01	Web Developer	65000.00	2	2	Engineering
11	Andrew	Rodriguez	andrewrodriguez@example.com	321-654-9876	2023-11-15	Web Developer	80000.00	2	2	Engineering
14	Jennifer	Diaz	jenniferdiaz@example.com	222-888-4444	2024-02-10	Web Developer	38000.00	2	2	Engineering
2	Jane	Smith	jan smith@example.com	987-654-3210	2023-02-15	Web Developer	60000.00	2	2	Engineering



SUBQUERIES

Sederhananya, subqueries merupakan query di dalam query. Penggunaan subquery di dalam query utama bertujuan untuk menspesifikasikan data yang akan ditampilkan.

Contoh Penggunaan SUBQUERY

Misalnya untuk mencari daftar nama karyawan yang memiliki gaji di atas rata-rata dari seluruh gaji karyawan

```
1 SELECT first_name, salary
2 FROM discovrio_employees
3 WHERE salary > (SELECT avg(salary) FROM discovrio_employees)
```

first_name ▲	salary ▲
Jessica	65000.00
Kevin	60000.00
Megan	70000.00
Jose	80000.00
Samantha	65000.00
Justin	75000.00
Andrew	80000.00
Nicole	70000.00
Jane	60000.00

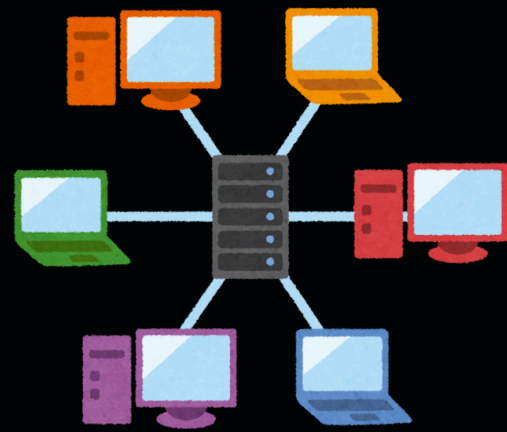


03

GIT & VERSION CONTROL SYSTEM

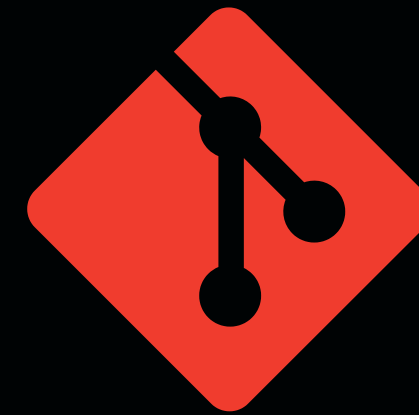
Understanding the Basics of Git

Definisi



Version Control System (VCS)

Sistem yang mencatat setiap perubahan yang terjadi pada sebuah/sekumpulan file dari waktu ke waktu.



Git (Cr: Linus Torvalds, 2005)

Salah satu version control system yang paling banyak digunakan untuk menangani untuk proyek kecil hingga besar .

Kegunaan VCS

1

Kolaborasi

Sifat terdistribusi yang memungkinkan banyak pengembang bekerja pada proyek yang sama secara bersamaan, mengelola perubahan, dan menggabungkan kontribusi mereka tanpa konflik.

3

Backup dan Pemulihan

menyimpan snapshot dari proyek pada berbagai titik waktu, VCS berfungsi sebagai solusi backup yang kuat, memungkinkan pemulihan kode dari berbagai versi jika terjadi kesalahan.

2

Manajemen Versi

menyimpan riwayat perubahan setiap file, memungkinkan pengembang untuk melihat, membandingkan, dan mengembalikan versi sebelumnya dari kode jika diperlukan.

4

Pengembangan Paralel

memungkinkan pengembang membuat cabang (branch) untuk mengembangkan fitur baru atau memperbaiki bug secara terpisah dari kode utama, kemudian menggabungkannya kembali setelah selesai.

5

Pemantauan Perubahan

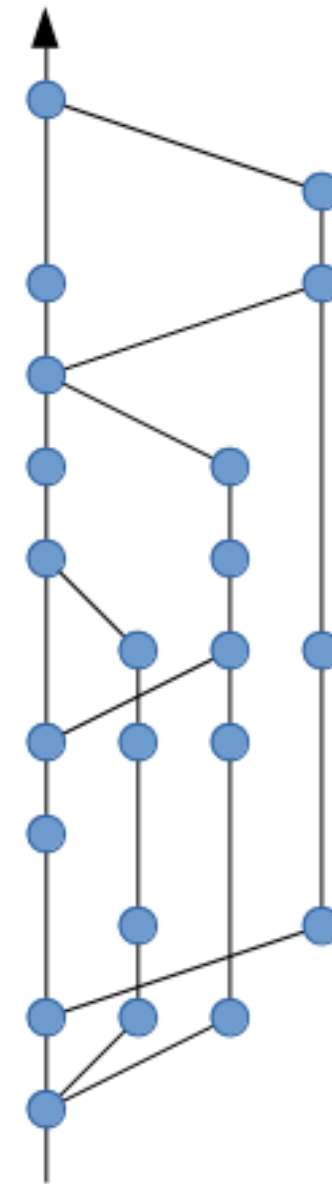
VCS mencatat siapa yang membuat perubahan, kapan, dan mengapa, memberikan jejak audit yang lengkap yang membantu dalam melacak dan memahami evolusi kode sepanjang waktu.

Fitur Git

History of Every Files

- Git menyimpan riwayat setiap perubahan dari sebuah repositori sebagai grafik snapshot, yang disebut commit.
- Setiap commit berisi petunjuk dari commit sebelumnya.
- Commit dapat memiliki banyak induk, sehingga membuat riwayat yang terlihat seperti grafik percabangan (branch).

Non-linear history



Linear history



Fitur Git

Traceability

- Setiap perubahan yang terjadi dapat dilacak secara mendetail, seperti: contributor, message, dan time.
- Fitur ini bermanfaat untuk memahami alur development sehingga dapat dengan mudah direplikasi dan dikembangkan di masa mendatang.

```
ShivAnI@ShivAnI-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git log --patch
commit 0d3835a746b82a4dc7ca97bcfbdbd4e39b26a680 (HEAD -> master)
Author: Shivani Goyal <shivanigoyal@gmail.com>
Date:   Fri Nov 8 15:49:51 2019 +0530

    newfile2 Re-added

diff --git a/newfile.txt b/newfile.txt
deleted file mode 100644
index d411be5..0000000
--- a/newfile.txt
+++ /dev/null
@@ -1,2 +0,0 @@
-new file to check git Head
-NEW COMMIT IN MASTER BRANCH.
diff --git a/newfile2.txt b/newfile2.txt
deleted file mode 100644
index e69de29..0000000

commit 56afce0ea387ab840819686ec9682bb07d72add6 (tag: -d, tag: --delete, tag: --d, tag: projectv1.1, origin/master, testing)
Author: Shivani Goyal <shivanigoyal@gmail.com>
Date:   Wed Oct 9 12:27:43 2019 +0530

    Added an empty newfile2

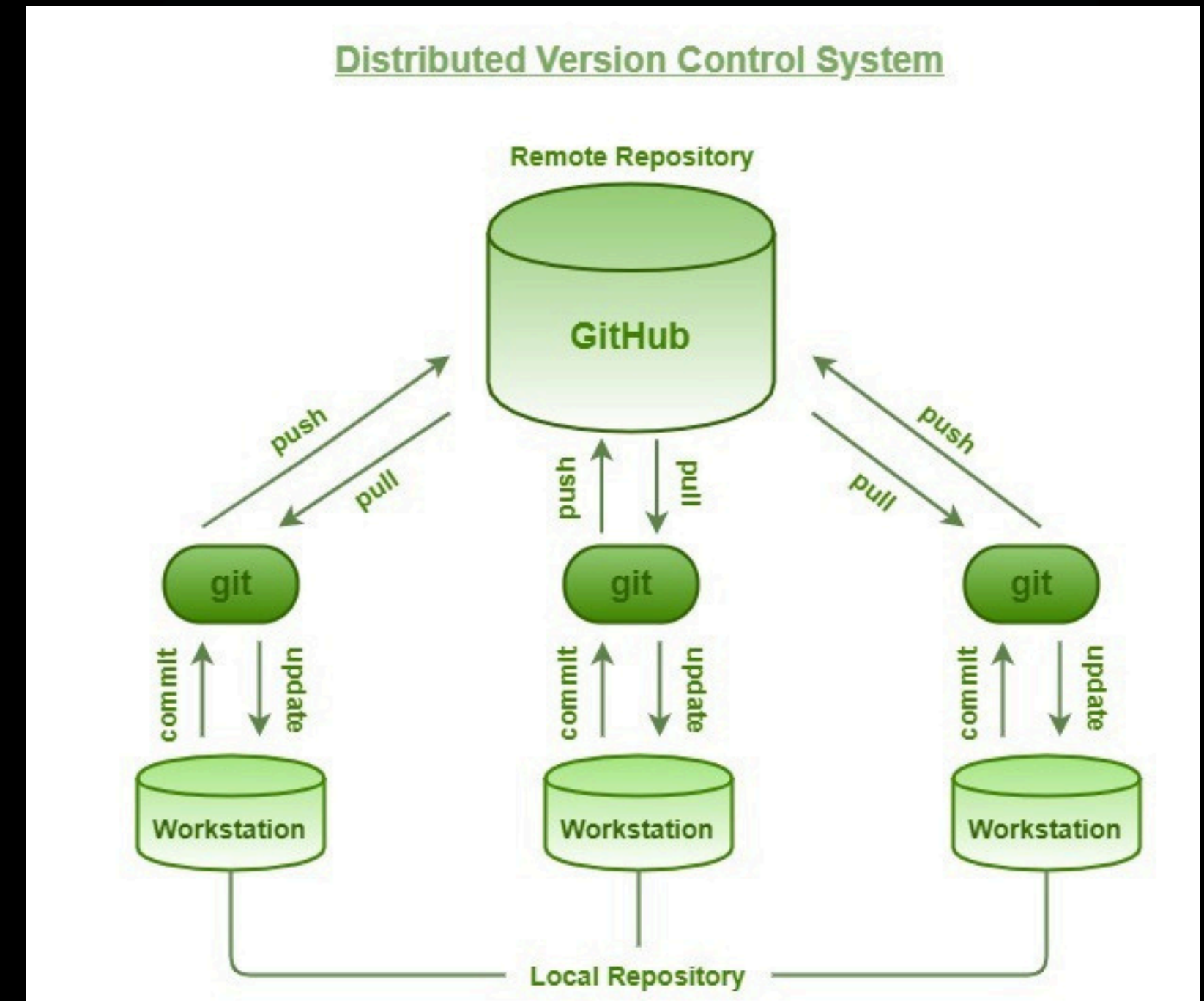
diff --git a/newfile2.txt b/newfile2.txt
new file mode 100644
index 0000000..e69de29

commit 0d5191fe05e4377abef613d2758ee0dbab7e8d95
Author: Shivani Goyal <shivanigoyal@gmail.com>
Date:   Sun Oct 6 17:37:09 2019 +0530
:
commit 0d3835a746b82a4dc7ca97bcfbdbd4e39b26a680 (HEAD -> master)
Author: Shivani Goyal <shivanigoyal@gmail.com>
```

Fitur Git

Sistem Distribusi

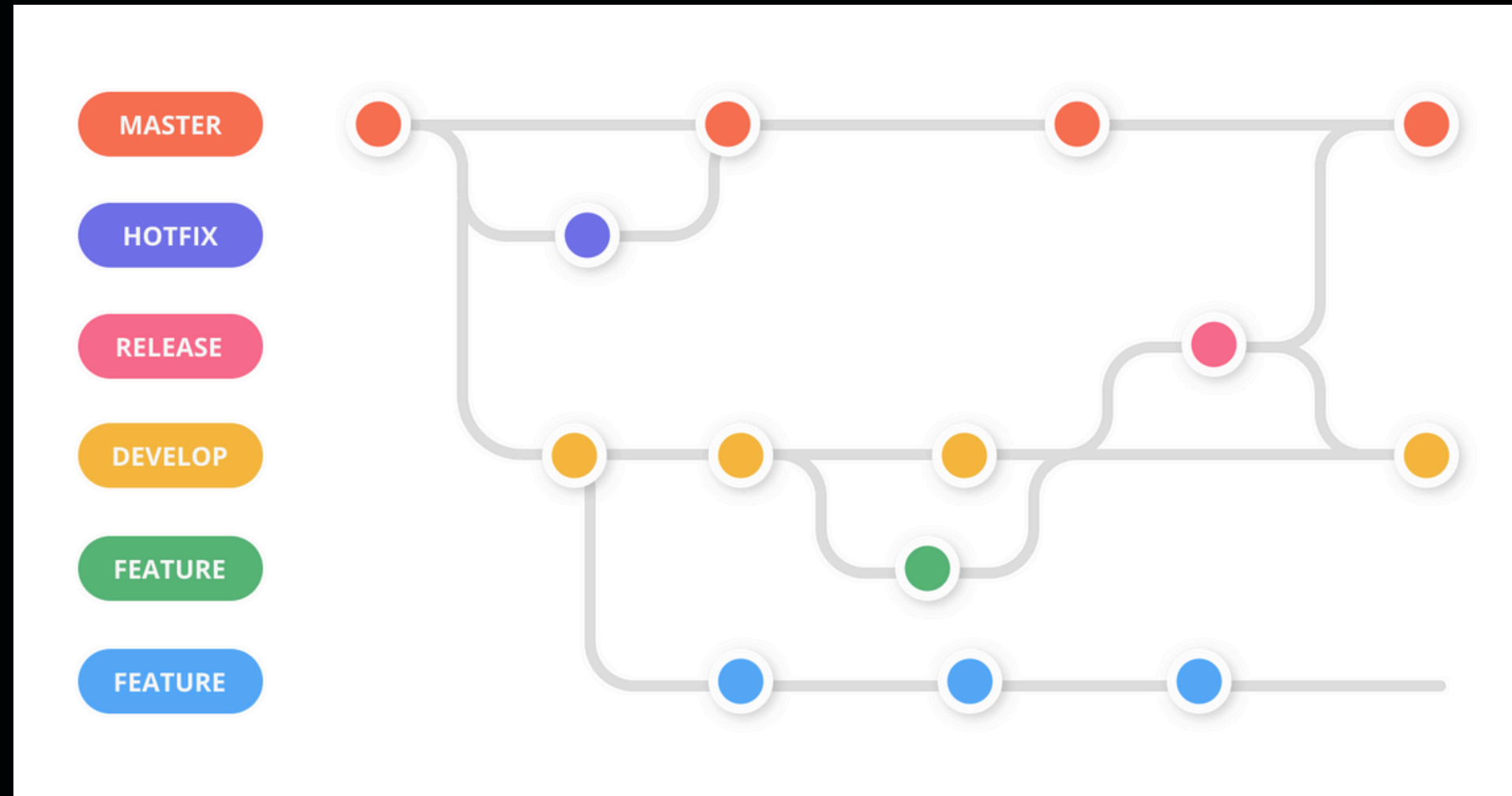
- Tidak bergantung pada server pusat untuk menyimpan seluruh versi file proyek.
- Dalam sistem ini, setiap kontributor memiliki salinan lokal atau *clone* dari repositori utama.
- Sehingga setiap kontributor dapat mengelola repositori lokalnya sendiri yang berisi semua file dan metadata yang ada di repositori utama.
- Sistem ini juga memungkinkan proses back up jauh lebih mudah.



Fitur Git

Branching Workflow

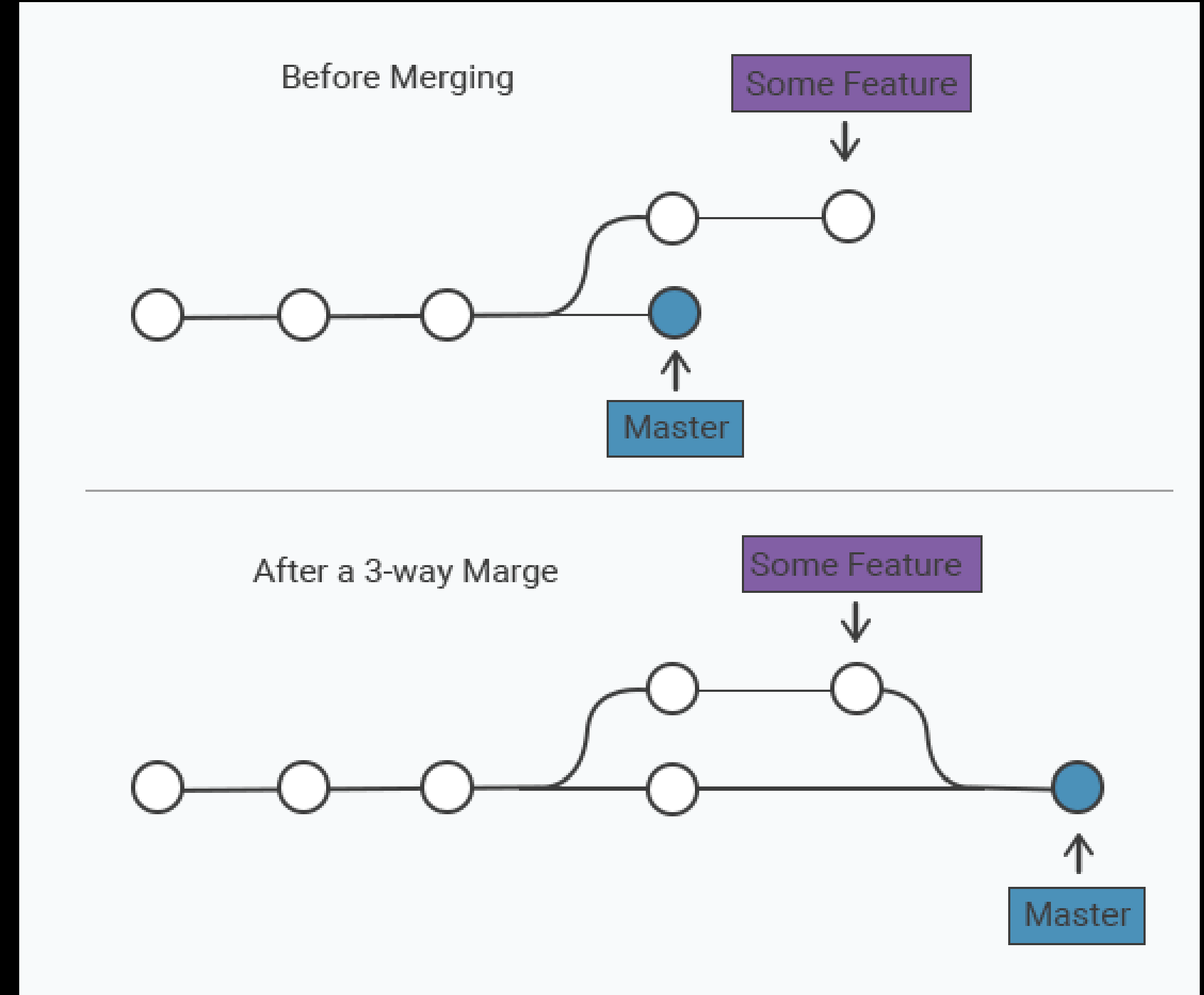
- Fitur dapat digunakan untuk mendefinisikan model percabangan secara jelas dan efisien.
- Fitur ini dapat mempermudah kolaborasi dalam mengerjakan proyek yang besar di dalam sebuah tim.



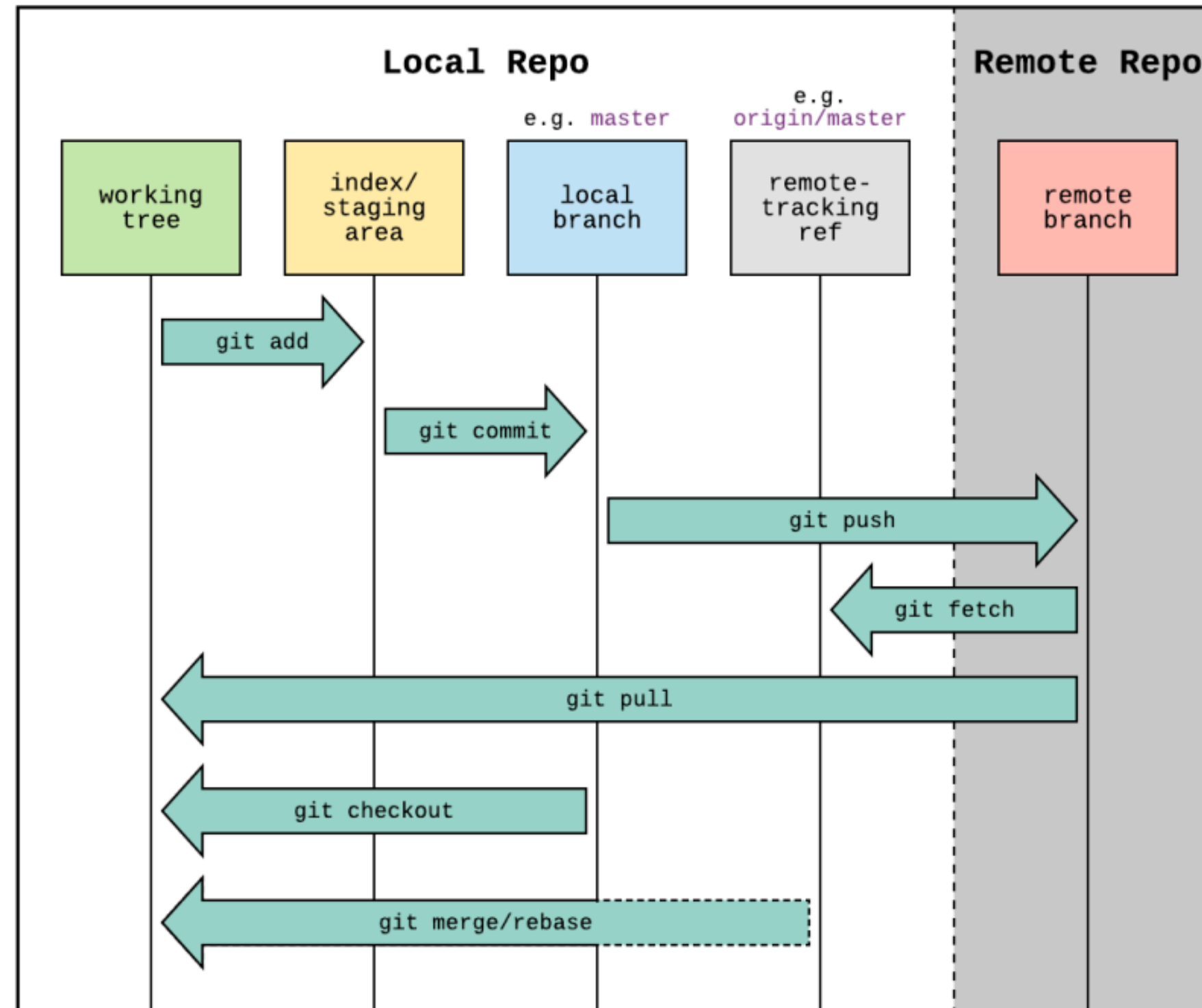
Fitur Git

Merging

- Memungkinkan contributor untuk mengambil pengembangan independen yang dibuat oleh git Branch dan mengintegrasikannya kembali kedalam satu cabang.



Flow Kerja Git



Instalasi Git



Windows

<https://git-scm.com/download/win>



Linux

```
# sudo apt-get install git
```



MacOS

```
$ brew install git
```

Konfigurasi dasar Git

Mengecek Git Version:

git -- version

Konfigurasi awal Git -- Username:

git config --global user.name "Nama"

Konfigurasi awal Git -- Email:

git config --global user.email "Email"

Mengecek Konfigurasi:

git config --list

Membuat repositori di local:

git init **Note:** Pastikan sudah berada di folder yang benar

Menambah file ke *staging area*:

git add "nama file" Or git add * (*semua file*)

Mengecek status local repositori:

git status

Menyimpan perubahan kedalam repositori:

git commit -m "keterangan terkait perubahan"

Mengirim perubahan dari local repository ke remote repository setelah perubahan di commit: git push origin master

Note: Jika sudah memiliki akun Github, GitLab atau lainnya, maka username dan email harus mengikuti akun tersebut agar lebih diintegrasikan.

Other Git Commands

Git Execution

Menghapus file dari *staging area*:

git rm --cached "nama file" Or
git rm --cached * (*semua file*)

Git Repo

Melihat perubahan setiap perubahan:

git log

Git Branching

Membuat branch baru:

git branch "branch baru"

Beralih ke branch lain:

git checkout "nama branch" Or
git switch "nama branch"

Mengecek seluruh branch yang aktif:

git branch

Menggabungkan branch baru:

git merge "branch baru"

Kembali ke commit sebelumnya:

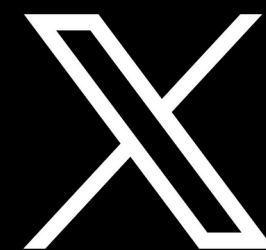
git reset --hard "commit id sebelumnya"



GitHub

Platform pengembangan software online berbasis website yang menyediakan layanan menyimpan repository Git, dan dilengkapi Version Control System.

Pengguna Layanan GitHub:



Perbedaan Git dan GitHub

Git

- Git adalah software yang dikelola Linux
- Git adalah command-line tool
- Diinstal secara lokal di sistem
- Berfokus pada kontrol versi dan berbagi kode
- Tidak memiliki fitur manajemen pengguna
- Sistem kontrol versi untuk mengelola riwayat kode sumber

GitHub

- Github adalah layanan yang dikelola Microsoft
- Merupakan Graphical User Interface (GUI)
- Di-hosting di web
- Berfokus pada hosting kode sumber terpusat
- Layanan hosting untuk repositori Git
- Memiliki fitur manajemen pengguna bawaan

<https://revou.co/revoupedia/kosakata>

Sumber: Geeks for Geeks

Git dan GitHub Commands

Clone Repositori dari GitHub

git clone "Link github clone"

Menyetel akun GitHub ke Git lokal

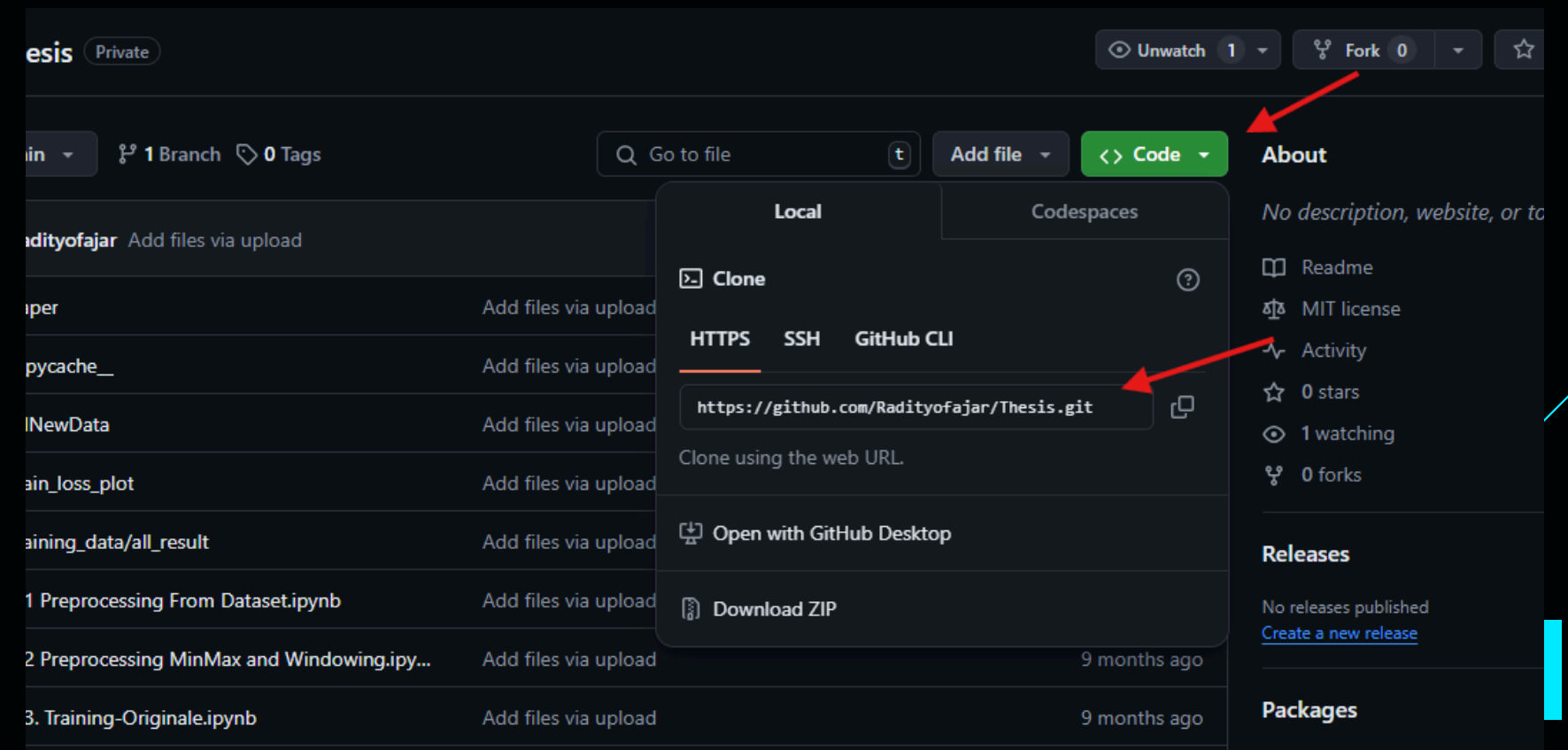
git remote add github "Link"

Mengambil perubahan terbaru GitHub repositori:

git fetch

Pull semua commit ke GitHub Repo (git fetch + git merge)

git config --list



REFERENCES

- W3Schools. (n.d.). SQL Tutorial. Retrieved July 16, 2024, from <https://www.w3schools.com/sql/default.asp>
- W3Schools. (n.d.). Git Tutorial. Retrieved July 17, 2024, from <https://www.w3schools.com/git/default.asp>

Thank You