



LEARNING PROGRESS REVIEW

WEEK 3

KELOMPOK 7: CITIZEN DATA SCIENTIST

DIAZ JUBIARY - HERMULIA HADIE PUTRA - DESI SULISTYOWATI - FARAHUL JANNAH

KEY PRESENTATION POINT

01 ADVANCED SQL

DATE FUNCTION – KEY – NORMALISASI DATABASE – JOIN & MERGE – JOIN FUNCTION
– MERGE/UNION – SUBQUERY – EFFECTIVE QUERY

02 VERSION CONTROL SYSTEM

PENGERTIAN – MANFAAT – GIT – GITHUB

03 PENGENALAN PYTHON

WHAT IS – WHY – USAGE – JUPYTER NOTEBOOK – VARIABLES – DATA TYPE –
ASSIGNMENT – ARITHMETIC PROGRAM – COMMENT – INDENTATION – WHITESPACE

ADVANCED SQL

Date Function

Digunakan untuk memanipulasi tanggal dan / atau waktu secara efektif

- `current_date` : untuk memunculkan informasi tanggal hari ini
select current_date
- `current_time` : untuk memunculkan informasi waktu saat ini
select current_time
- `Current_timestamp` : untuk memunculkan informasi tanggal dan waktu tepat di detik ini
select current_timestamp
- `Extract` : untuk memunculkan informasi spesifik dari waktu atau tanggal
select extract (month from current_timestamp)
- ----> menunjukkan bulan saat ini
- `Date_trunc` : untuk memotong informasi tanggal / waktu berdasarkan yang kita mau
select date_trunc ('month', current_timestamp)
- ----> Merujuk pada tanggal pertama di bulan ini

Date Function

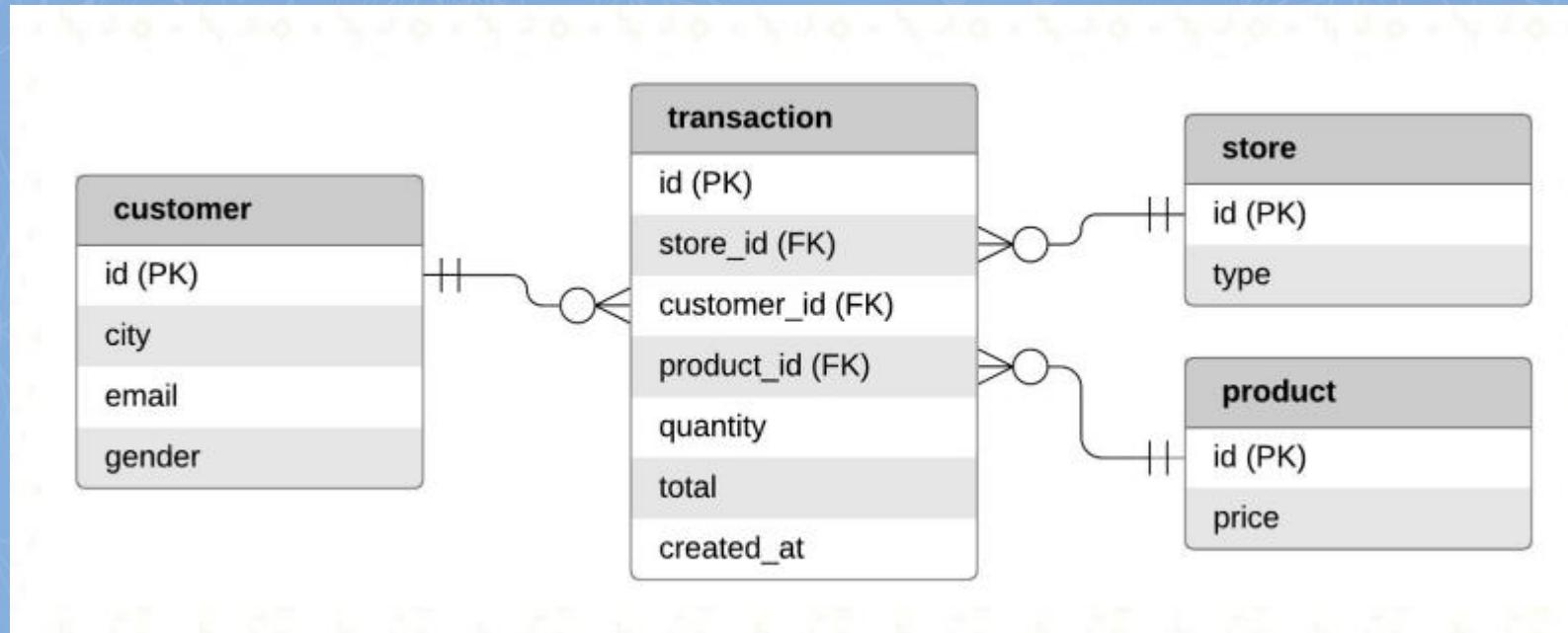
- date + integer → date
- date - date → integer
- date + interval → timestamp
- date + time → timestamp
- interval + interval → interval
- timestamp + interval → timestamp
- time + interval → time
- date - integer → date
- date - interval → timestamp
- time - time → interval
- time - interval → time
- timestamp - interval → timestamp
- interval - interval → interval
- timestamp - timestamp → interval
- interval * double precision → interval
- interval / double precision → interval

Key

Attribute atau set attribute pada tabel yang digunakan untuk mengidentifikasi sebuah row.

- Key dapat menunjukkan relasi antara dua tabel.
- Mengapa membutuhkan key:
 - Untuk mengidentifikasi row data pada tabel agar tidak terjadi double record
 - Untuk memudahkan kita mengidentifikasi relasi antar tabel pada schema database

Key



Key

- Super key : sekumpulan atribut yang mengidentifikasi baris dalam tabel secara unik. Super Key adalah superset dari Candidate Key.
- Candidate key : sekumpulan atribut yang secara unik mengidentifikasi tupel dalam sebuah tabel.
- Primary key : suatu atribut(bisa satu atau lebih) yang digunakan untuk mendefinisikan atau memastikan bahwa setiap record didalam tabel tersebut bersifat unique atau berbeda antara yang satu dengan yang lainya.
- Foreign key : suatu atribut (bisa satu atau lebih) yang digunakan sebagai penghubung antara tabel satu dengan tabel yang lainnya pada database relational
- Secondary / alternative key : candidate key yang tidak dipilih sebagai primary key

Normalisasi Database

- Normalisasi database adalah proses pengelompokan atribut data yang membentuk entitas sederhana, nonredundant, fleksibel, dan mudah beradaptasi. Sehingga dapat dipastikan bahwa database yang dibuat berkualitas baik.
- Tujuan utama dari normalisasi database adalah:
 - Menghilangkan dan mengurangi redundansi data.
 - Memastikan dependensi data (data berada pada tabel yang tepat).
- Contoh :

Kita ingin mengupdate nama customer pada tabel. Tabel kita terdapat beberapa nama customer yang sama, mungkin adalah orang yang sama dengan nama yang sama, atau mungkin orang yang berbeda. Hal ini dapat diatasi dengan database normalization

Normalisasi Database

Jika sebuah database belum dinormalisasi, kemungkinan terburuk yang akan merugikan sistem adalah:

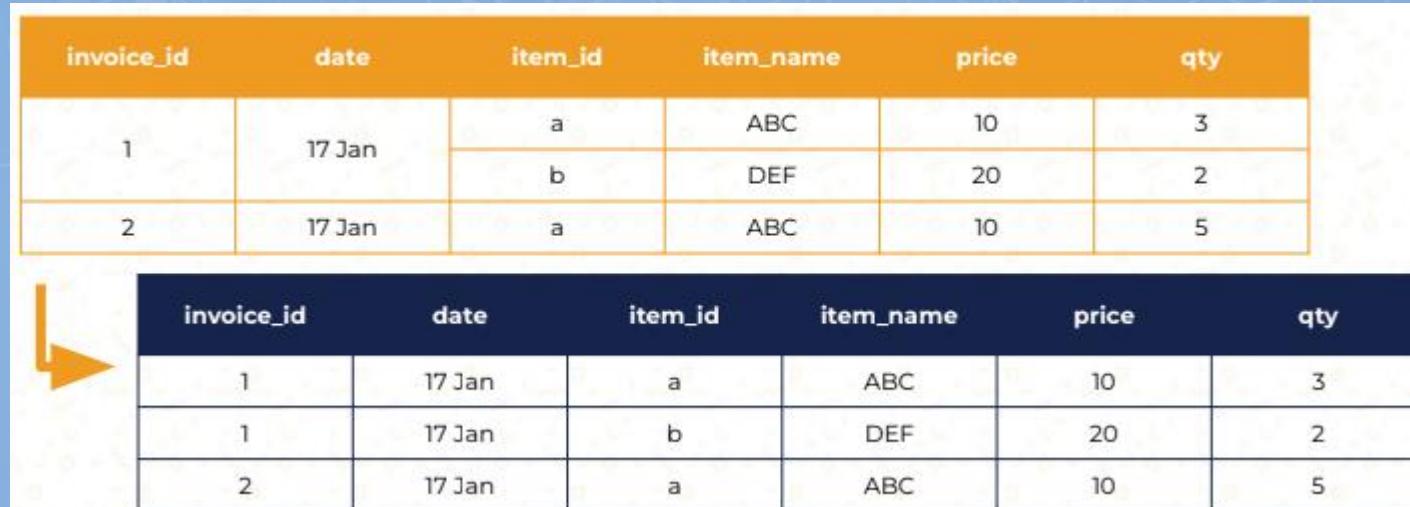
- **INSERT Anomali**
Situasi dimana tidak memungkinkan untuk memasukkan beberapa jenis data secara langsung di database.
- **UPDATE Anomali**
Situasi dimana nilai yang diubah mengakibatkan ketidakkonsistenan database, artinya data yang diubah tidak sesuai dengan yang diinginkan.
- **DELETE Anomali**
Penghapusan data tidak sesuai yang diharapkan, artinya data yang seharusnya tidak terhapus malah terhapus.

Normalisasi Database

1. 1NF / First Normal Form

1NF mensyaratkan beberapa kondisi dalam sebuah database. Berikut adalah fungsi dari bentuk normal pertama ini:

- Menghilangkan duplikasi kolom dari tabel yang sama.
- Buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (primary key).



The diagram illustrates the normalization process from a single-table invoice to two separate tables based on item_id.

Original Table (1NF Violation):

invoice_id	date	item_id	item_name	price	qty
1	17 Jan	a	ABC	10	3
		b	DEF	20	2
2	17 Jan	a	ABC	10	5

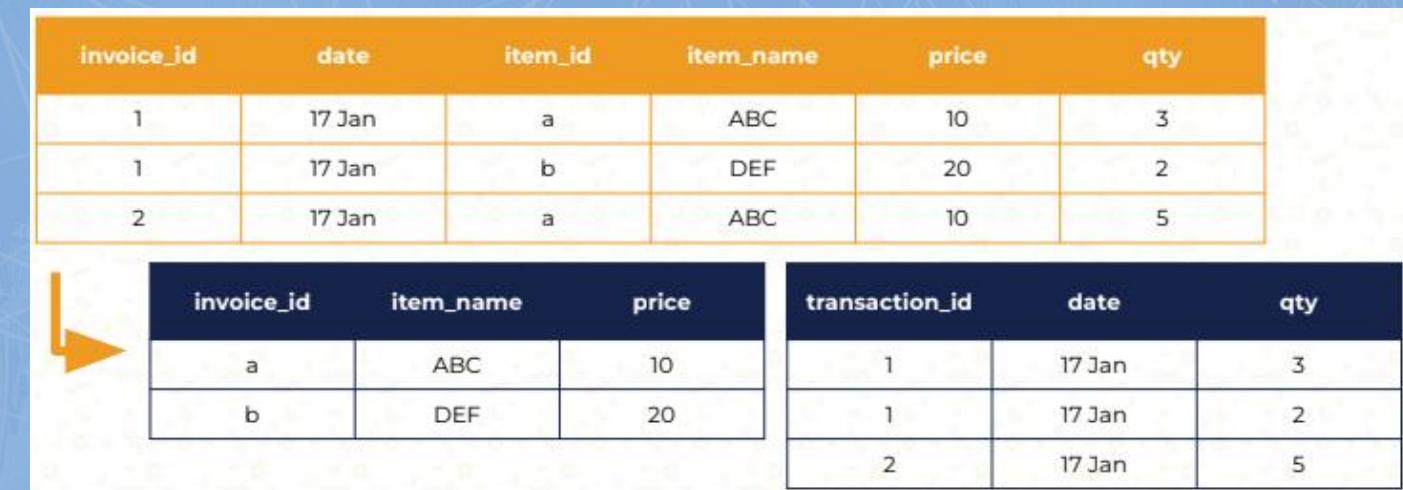
Normalized Tables (2 tables):

invoice_id	date	item_id	item_name	price	qty
1	17 Jan	a	ABC	10	3
1	17 Jan	b	DEF	20	2
2	17 Jan	a	ABC	10	5

Normalisasi Database

2. 2NF/Second Normal Form

- Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF.
- Menghapus beberapa subset data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
- Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan foreign key.
- Tidak ada atribut dalam tabel yang secara fungsional bergantung pada candidate key tabel tersebut.



The diagram illustrates the decomposition of a single table into three normalized tables. A large orange arrow points from the original table on the left to the three new tables on the right, indicating the transformation process.

invoice_id	date	item_id	item_name	price	qty
1	17 Jan	a	ABC	10	3
1	17 Jan	b	DEF	20	2
2	17 Jan	a	ABC	10	5

invoice_id	item_name	price
a	ABC	10
b	DEF	20

transaction_id	date	qty
1	17 Jan	3
1	17 Jan	2
2	17 Jan	5

Normalisasi Database

3.3NF/Third Normal Form

- Pada 3NF tidak diperkenankan adanya partial “transitive dependency” dalam sebuah tabel. Transitive dependency biasanya terjadi pada tabel hasil relasi, atau kondisi dimana terdapat tiga atribut A, B, C. Kondisinya adalah $A \Rightarrow B$ dan $B \Rightarrow C$. Maka C dikatakan sebagai transitive dependency terhadap A melalui B.
- Intinya pada 3NF ini, jika terdapat suatu atribut yang tidak bergantung pada primary key tapi bergantung pada field yang lain maka atribut-atribut tersebut perlu dipisah ke tabel baru.



The diagram illustrates the decomposition of a single-table into three separate tables to eliminate transitive dependency. An orange arrow points from the original table to the decomposed tables.

Original Table:

invoice_id	item_name	price
a	ABC	10
b	DEF	20

Decomposed Tables:

- Table 1:** invoice_id, item_name, price
- Table 2:** transaction_id, date, qty
- Table 3:** invoice_id, item_id, price, qty

The decomposition results in:

transaction_id	date	qty
1	17 Jan	3
1	17 Jan	2
2	17 Jan	5

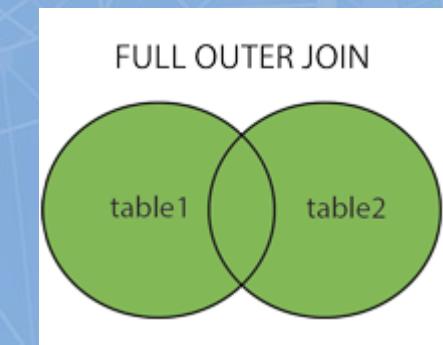
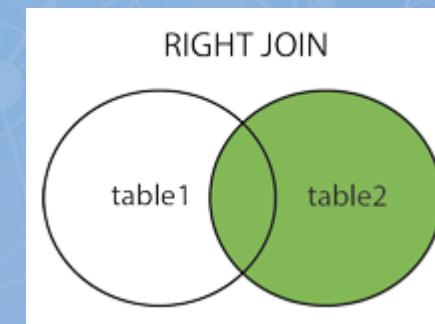
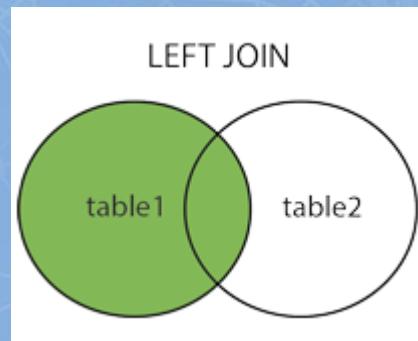
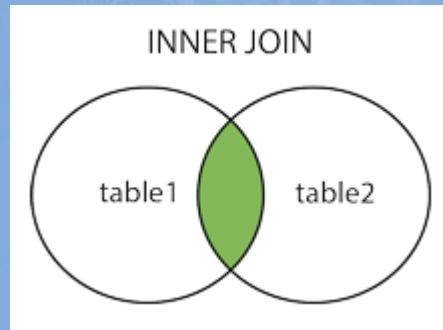
invoice_id	item_name	price
a	ABC	10
b	DEF	20

invoice_id	item_id	price	qty
1	a	10	3
1	b	20	2
2	a	10	5

Join & Merge

SQL memungkinkan kita untuk melakukan query dengan menggabungkan 2 tabel atau lebih. Kita dapat melakukan join (menambah kolom) atau merge (menambah value row).

Join Function



Join Function

Orders Table

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

Customers Table

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

Join Function

```
SELECT Orders.OrderID, Customers.CustomerName,  
Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.Custo  
merID
```

Merge / Union

Fungsi UNION digunakan untuk menggabungkan kumpulan hasil dari 2 atau lebih pernyataan SELECT. Ini menghapus duplikat baris antara berbagai pernyataan SELECT. Setiap pernyataan SELECT di dalam operator UNION harus memiliki jumlah field yang sama pada set hasil dengan tipe data yang sama.

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2
```

Subquery

- Subquery atau Inner query atau Nested query adalah query dalam query SQL lain dan tertanam dalam klausa WHERE. Sebuah subquery digunakan untuk mengembalikan data yang akan digunakan dalam query utama sebagai syarat untuk lebih membatasi data yang akan diambil. Subqueries dapat digunakan dengan SELECT, INSERT, UPDATE, dan DELETE statements bersama dengan operator seperti =, <,>,>=, <=, IN, BETWEEN dll.
- Tips menggunakan subquery
 - Gunakan query sesuai dengan kebutuhan
 - Gunakan alias jika menggunakan subquery dalam FROM atau JOIN FUNCTION
 - Gunakan alias yang merefleksikan nama tabel

```
SELECT column_name [, column_name ]  
FROM table1 [, table2 ]  
WHERE column_name OPERATOR  
      (SELECT column_name [, column_name ]  
       FROM table1 [, table2 ]  
      [WHERE])
```

Effective Query Technical Point of View

Tips

- Query yang berlebihan memerlukan resource yang besar
- Hindari menggunakan fungsi yang tidak diperlukan
- Lakukan query pada kolom yang dibutuhkan, hindari menggunakan select *
- Gunakan fungsi order by hanya saat diperlukan
- Kurangi menggunakan case when function jika memungkinkan
- Hindari bergantung pada subquery
- Gunakan fungsi limit jika hanya ingin crosscheck tabel

Effective Query Business Point of View

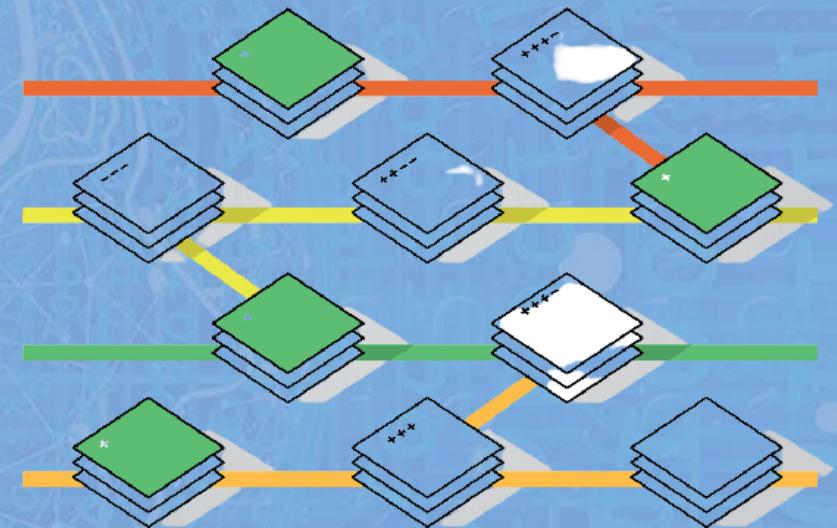
Tips

- Pastikan kembali pertanyaan bisnis yang akan dipecahkan
- Pilah kolom yang akan digunakan sesuai dengan tujuan awal masalah bisnis yang akan dipecahkan
- Uraikan semua sumber untuk kolom yang dibutuhkan
- Tulis skrip SQL per sumber untuk mendapatkan setiap kolom yang dibutuhkan
- Kompilasi semuanya menjadi satu skrip

VERSION CONTROL SYSTEM

VERSION CONTROL SYSTEM

Sebuah sistem software yang memungkinkan user untuk melacak dan merekam perubahan file atau projek dari waktu ke waktu. VCS juga memudahkan anggota tim untuk berkolaborasi dalam suatu projek. VCS juga memungkinkan untuk para user untuk melakukan memisahkan tugas-tugas mereka dengan menggunakan fitur "branch"



Manfaat VCS

- Memudahkan anggota tim untuk berkolaborasi dalam suatu projek.
- Memungkinkan para user melakukan pemisahan tugas-tugas dengan menggunakan fitur "branch"
- Mampu menyimpan setiap perubahan dalam repositori.
- Para user bisa membatalkan perubahan apabila terjadi suatu kesalahan.
- Membantu para user untuk membandingkan sebelum dan sesudah terjadinya perubahan.
- Mencegah banyaknya file dengan berbagai macam versi (misalnya, "sudah_fix"; "sudah_fix_bgt"; dan "sudah_fix_bgt_bgt")

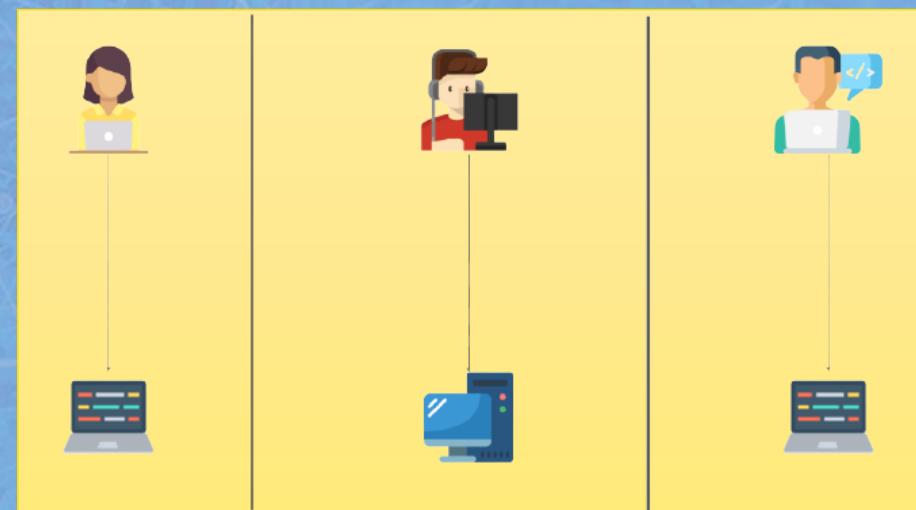
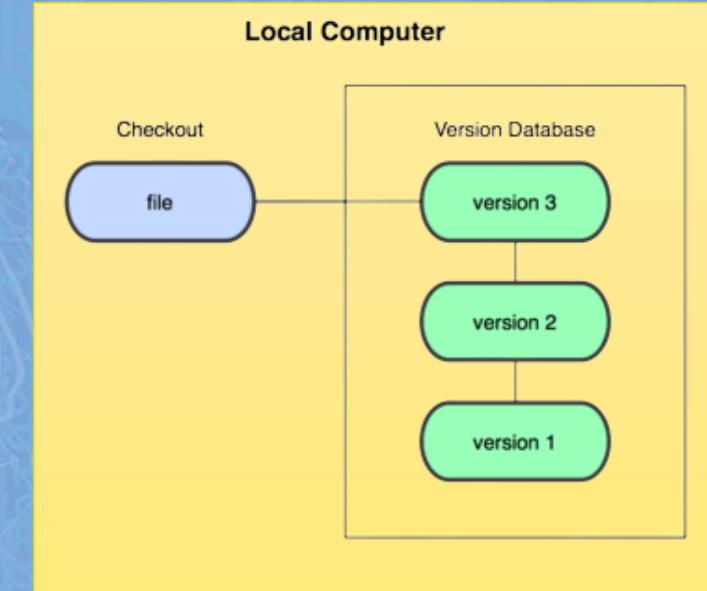
Jenis VCS

- Local version control
- Centralized version control
- Distributed version control

Jenis VCS

Local Version Control

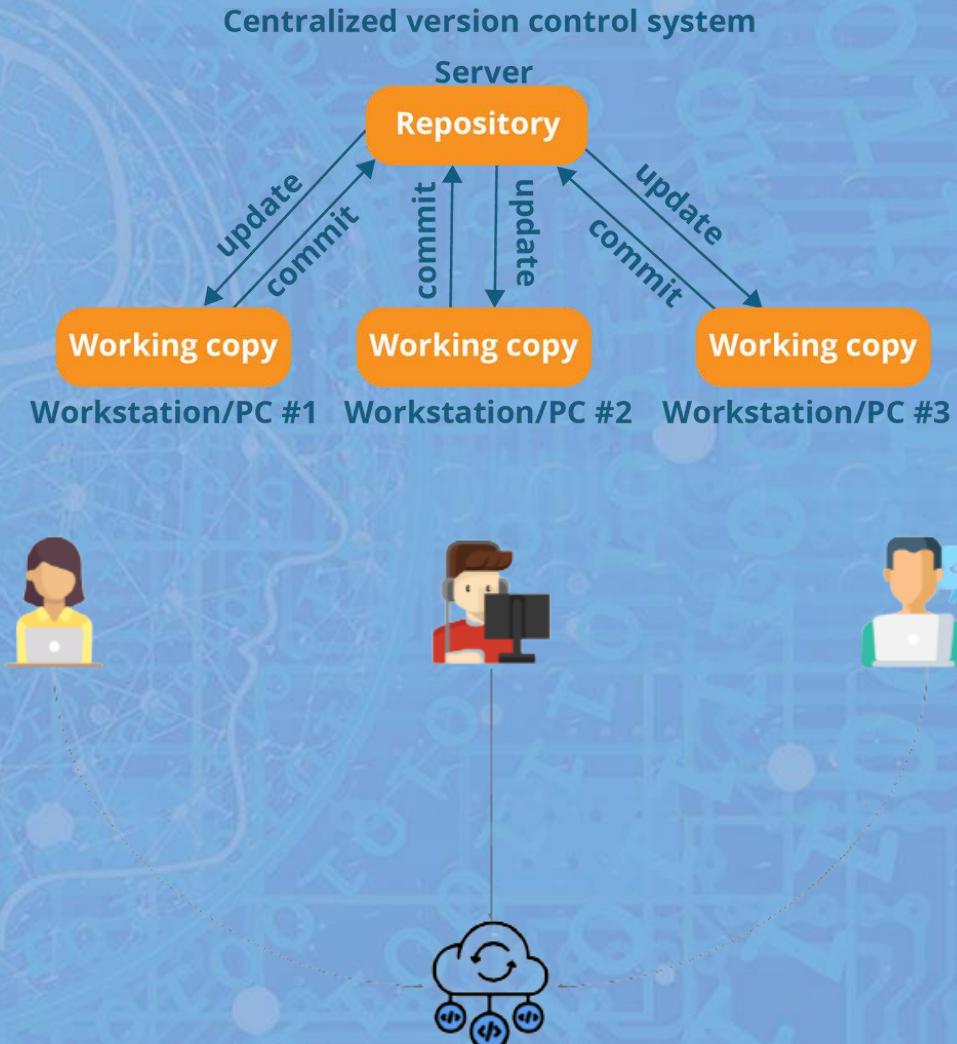
- Local Version Control System merupakan VCS yang tidak memiliki repository jarak jauh.
- Pengelolaan data hanya ada di sistem lokal (komputer, PC, atau laptop pribadi user).
- Perubahan data pada file pun hanya dicatat di database lokal.
- Gambar disamping mengilustrasikan bagaimana local version control system bekerja, dimana setiap user mempunyai sistem lokal sendiri dan tidak terhubung satu sama lain



Jenis VCS

Centralized Version Control

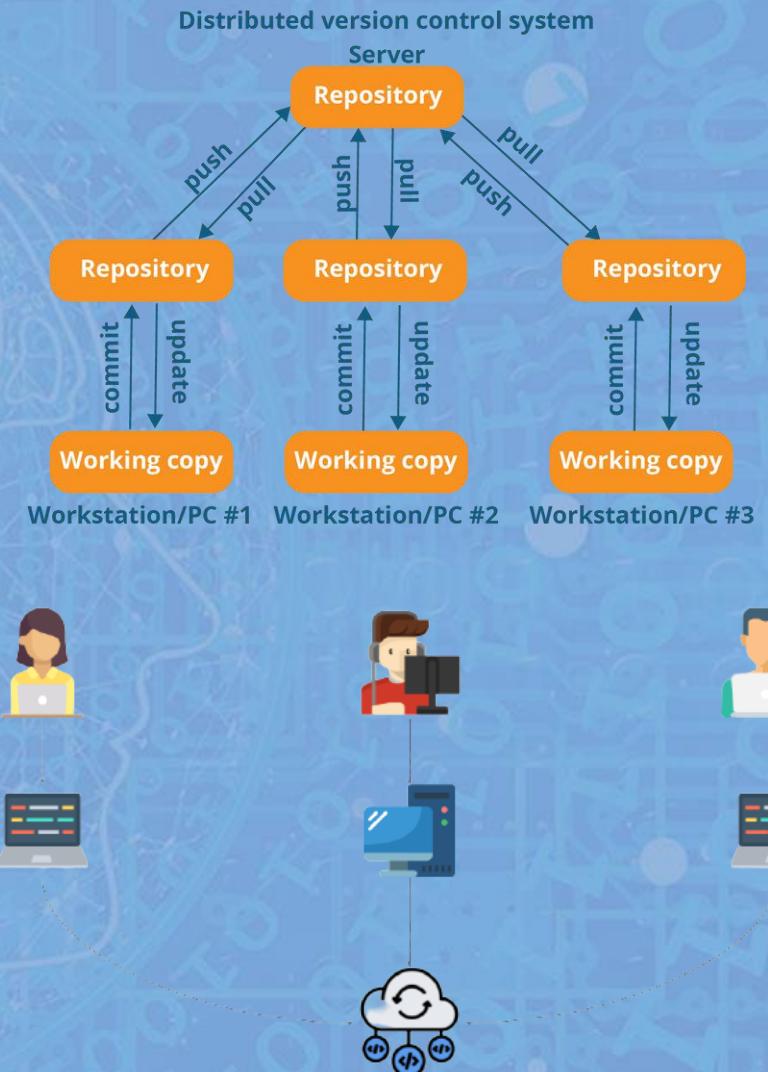
- Centralized Version Control System (CVCS) mempunyai sebuah repository pusat yang dibagikan kepada para user.
- Setiap user mendapatkan salinan (copy) pekerjaan masing-masing.
- Setiap perubahan yang dilakukan oleh user akan terekam di repository.
- Para user dapat berkolaborasi dalam mengerjakan suatu projek.
- Kelemahan dari CVCS ini, apabila server mati atau rusak maka perubahan yang sudah dikerjakan tidak akan tersimpan



Jenis VCS

Distributed Version Control

- Distributed version control mempunyai banyak repository.
- Setiap user mempunyai repository dan salinan pekerjaan masing-masing.
- Pada saat user melakukan modifikasi data di repository pusat tidak akan terpengaruhi.
- Perubahan di repository pusat hanya terjadi apabila user melakukan "push" data yang sudah berubah di repository lokal ke repository pusat.
- Sekarang VCS yang paling sering dipakai adalah distributed version control system ini.
- salah satu software VCS yang paling sering dipakai adalah Git



Git



Git merupakan software distributed (VCS) gratis dan open source dan dapat digunakan oleh programmer dan developer untuk mencatat perubahan seluruh file atau repository suatu projek.

GitHub



GitHub merupakan layanan cloud yang berguna untuk menyimpan dan mengelola sebuah repository. GitHub harus terkoneksi pada internet sehingga memberikan keringanan penyimpanan komputer yang kita gunakan karena file projek tersimpan oleh cloud GitHub.

Fitur Git

1. Riwayat Perubahan pada File

Git memungkinkan para user untuk melacak berbagai hal, misalnya:

- Kapan file dibuat;
- File yang dihapus;
- Perubahan yang dilakukan pada file (seperti penambahan baris, file yang dipindahkan ke folder lain, atau file yang diubah namanya)

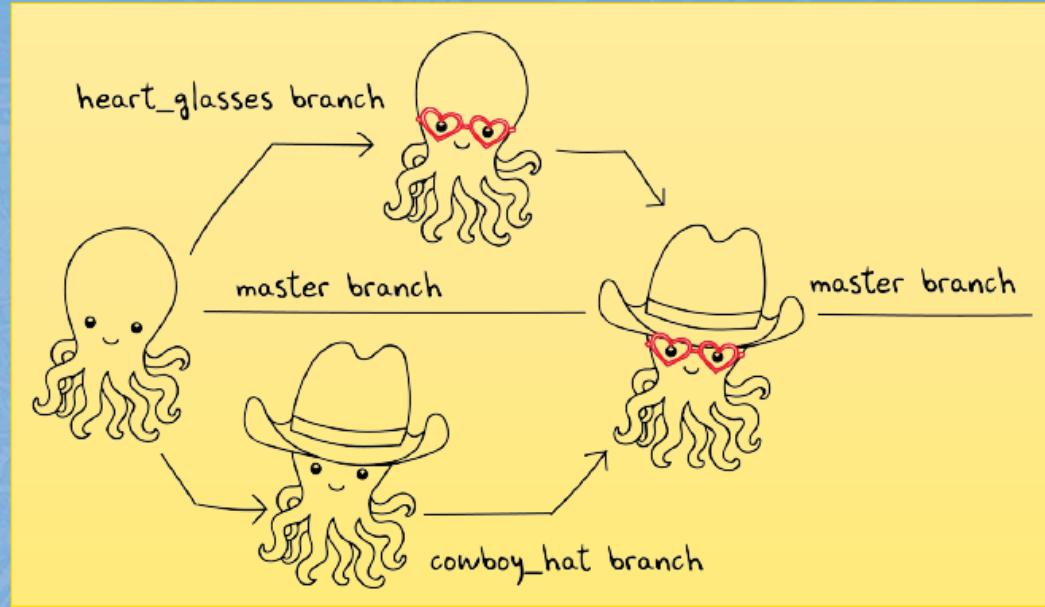
2. Traceability

Git memungkinkan para user mengetahui siapa, kapan, dan alasan kenapa suatu file diubah atau dimodifikasi.

Fitur Git

3. Branching dan Merging

- Fitur branching adalah fitur yang digunakan untuk membuat cabang dari data master yang ada di server.
- Tujuan dari branching adalah supaya data master tidak berubah pada saat user sedang melakukan modifikasi. Jadi, pada saat proses branching, data master di copy ke repository lokal user.
- Merging adalah proses dimana ketika user ingin menggabungkan hasil perubahan yang selesai dimodifikasi di repository lokal ke server/data master.

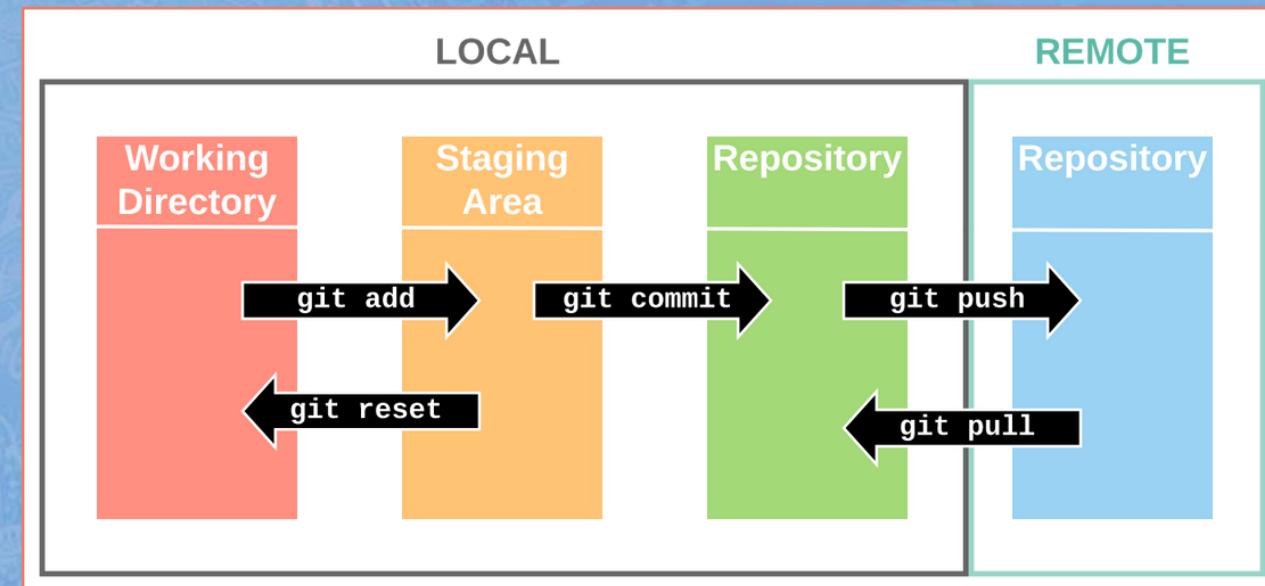


Contoh berdasarkan ilustrasi di atas.

- Data utama kita adalah gambar gurita.
- Branching adalah ketika user A menambahkan gambar "heart glasses" ke gurita dan ketika user B menambahkan gambar "cowboy hat" ke gurita.
- Merging adalah ketika hasil branching user A dan B digabungkan menjadi gurita yang memakai "heart glasses" dan "cowboy hat"

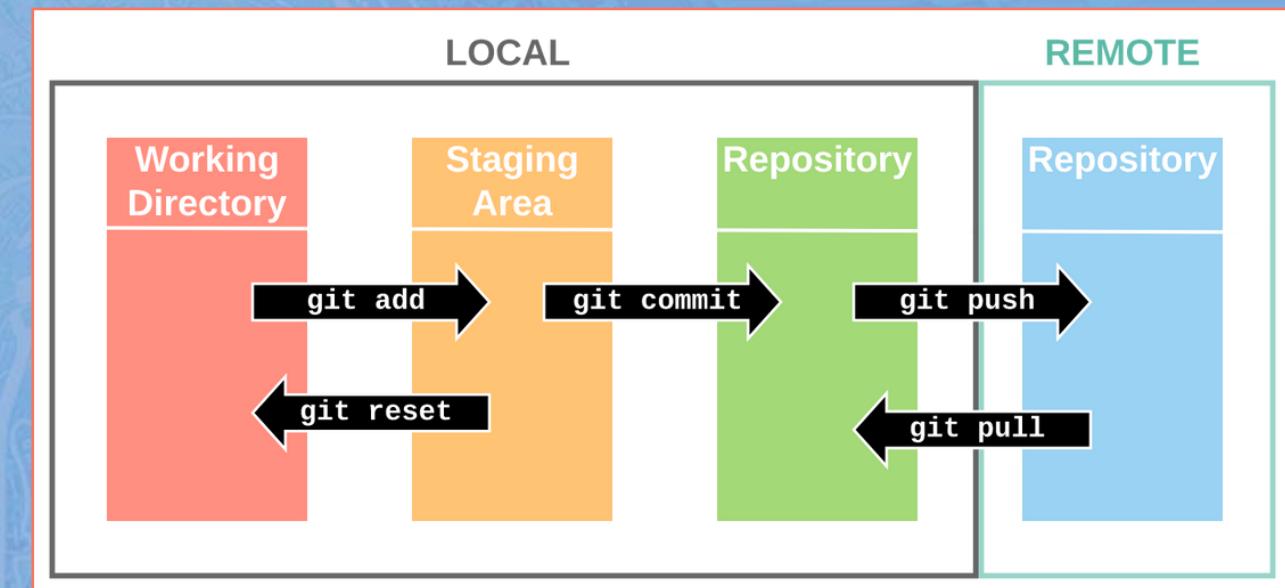
Struktur Repository Git

- Working directory: berisikan file yang ditarik dari database dan disimpan dalam penyimpanan lokal komputer (harddisk komputer). File ini yang akan dimodifikasi.
- Staging area: area penyimpanan file yang menjadi commit selanjutnya.
- Repository lokal: penyimpanan database lokal. File yang berada disini akan di-merge ke repository remote.
- Repository remote: repository cloud yang disediakan oleh developer (seperti GitHub atau GitLab)



Work Flow Git

- Melakukan modifikasi terhadap file yang ada pada working directory.
- Melakukan staging terhadap file yang dimodifikasi, menambahkan snapshots pada staging area.
- Melakukan commit, dimana akan menyimpan data snapshot yang ada pada staging area untuk disimpan secara permanent pada repository local
- Melakukan "git push" ke remote repository



Git Cheat Sheet

Pengaturan Command

Memverifikasi versi Git yang terinstall di komputer

```
git --version
```

Mengkonfigurasi user name dan email

```
git config --global user.name "nama kita bisa pake spasi juga"  
contoh:
```

```
git config --global user.name "Harry Potter"
```

```
git config --global user.email "alamat email kita"
```

```
contoh:
```

```
git config --global user.email "harrypotter_1234@gmail.com"
```

Git Cheat Sheet

Pengaturan Command

Memverifikasi user name dan email

```
git config user.name
```

```
git config user.email
```

Git Cheat Sheet

Repository Command

Membuat repository lokal

```
git init
```

Menyalin repository dari remote ke lokal

```
git add remote origin [url]
```

Contoh:

```
git add remote origin https://github.com/harrypotter/hogwarts.git
```

Menampilkan riwayat seluruh commit

```
git log
```

Git Cheat Sheet

Repository Command

Melakukan cloning dari repository remote ke lokal

```
git clone [url]
```

Contoh:

```
git clone https://github.com/harrypotter/hogwarts.git
```

Git Cheat Sheet

Branching Command

Membuat branch (cabang)

```
git branch "nama cabang"  
Contoh:  
git branch "branch_harry"
```

Menghapus branch (cabang)

```
git branch -d "nama cabang"  
Contoh:  
git branch -d "branch_harry"
```

Masuk ke dalam suatu branch

```
git checkout "nama cabang"  
Contoh:  
git checkout "branch_harry"
```

Mengetahui branch (cabang) yang aktif

```
git branch
```

Git Cheat Sheet

Execution Command

Menambahkan file ke staging area

```
git add.      --> digunakan untuk menambahkan seluruh file yang ada  
| | | | | | | dari working area ke staging area  
git add "nama_file"    --> digunakan untuk menambahkan file yang spesifik  
| | | | | | | dari working area ke staging area
```

Menambahkan file dari staging area ke repository lokal

```
git commit -m "pesan commit"  
Contoh:  
git commit -m "menambahkan value yang baru"
```

Git Cheat Sheet

Execution Command

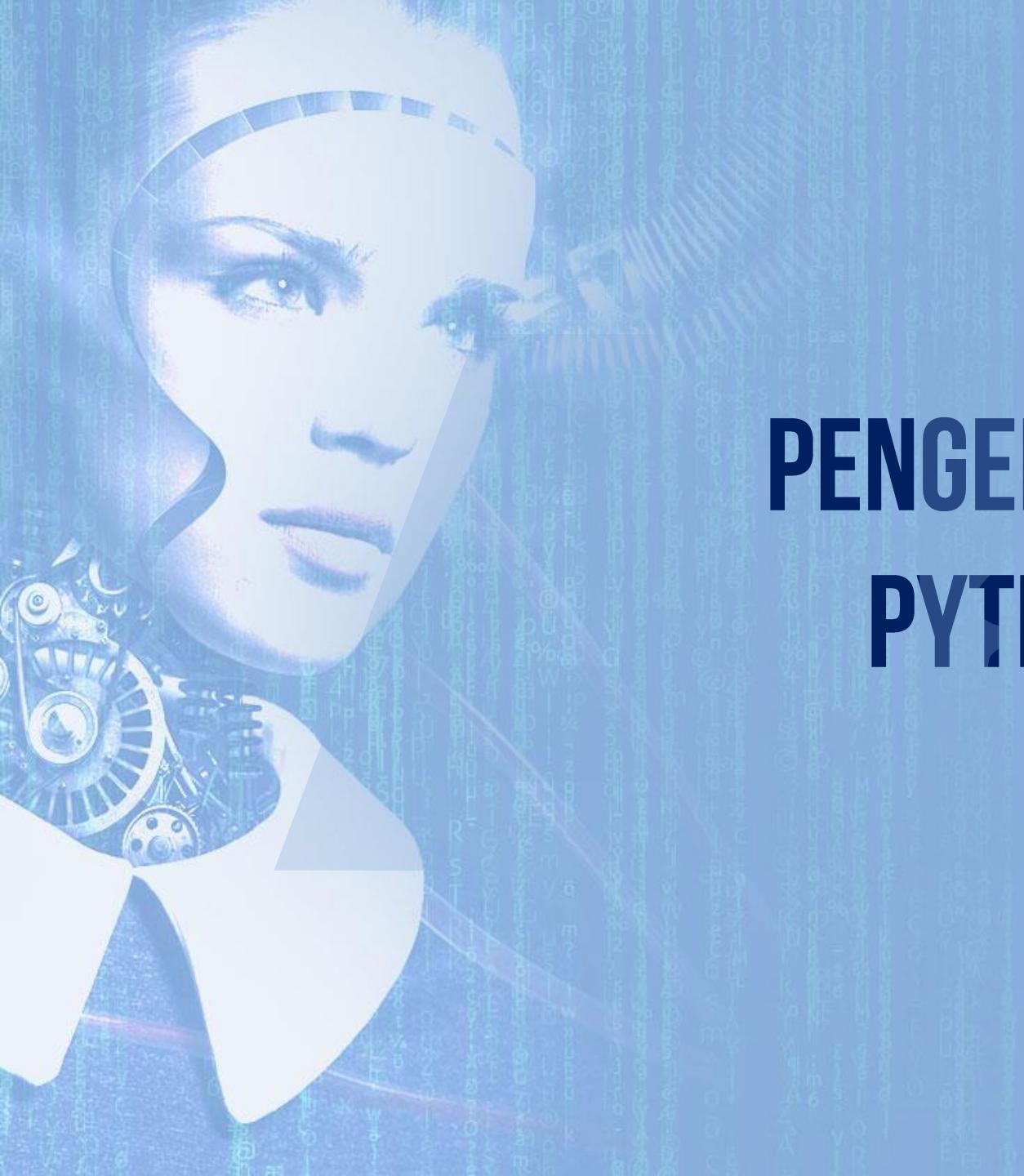
Melakukan push dari repository lokal ke remote

```
git push origin "nama cabang"  
Contoh:  
git push origin "branch_harry"
```

Melakukan pull dari repository remote ke lokal

```
git pull origin "nama cabang"  
Contoh:  
git pull origin "branch_harry"
```

PENGENALAN PYTHON



What is Python?

- Python adalah bahasa pemrograman yang populer. Dibuat oleh Guido van Rossum, dan dirilis pada tahun 1991.
- Python telah menjadi salah satu bahasa pemrograman paling populer di dunia dalam beberapa tahun terakhir. Python dapat digunakan oleh developers dan non-developers.
- Digunakan di berbagai bidang IT:
 - Software Development
 - Web Development
 - Data Science (Data analysis, machine learning, deep learning)
 - Software Testing
 - Automation

Why Python?

- Python bekerja pada macam-macam platform yang berbeda (Windows, Mac, Linux, Raspberry Pi, dll).
- Python memiliki sintaks sederhana yang mirip dengan bahasa Inggris.
- Python memiliki sintaks yang memungkinkan developer untuk menulis program dengan lebih sedikit baris daripada beberapa bahasa pemrograman lainnya.
- Python berjalan pada sistem interpreter, artinya kode dapat dieksekusi segera setelah ditulis. Ini berarti bahwa pembuatan prototipe bisa sangat cepat.
- Python can be treated in a procedural way, an object-oriented way or a functional way.
- Python dapat diperlakukan secara prosedural, object-oriented, atau fungsional

Python Usage in Data Science

Python adalah aplikasi open source, interpreted, high level language dan menyediakan pendekatan object-oriented programming. Merupakan salah satu bahasa pemrograman terbaik yang digunakan oleh ilmuwan data untuk berbagai proyek/aplikasi ilmu data. Python menyediakan fungsionalitas yang hebat untuk menangani matematika, statistik, dan fungsi ilmiah. Dilengkapi pula dengan banyaknya library yang lengkap untuk memudahkan penggunaan.

- Numpy
- Pandas
- Matplotlib
- Scikit-learn

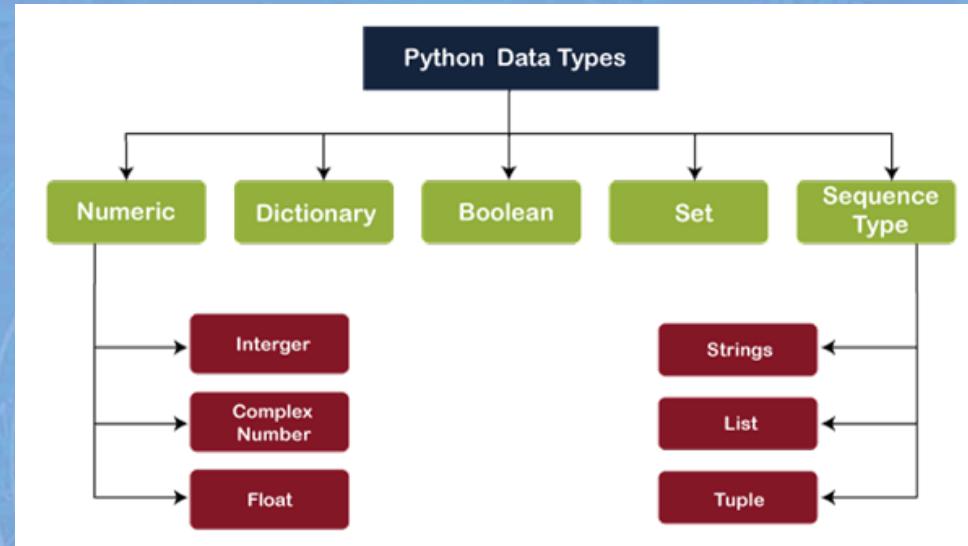
Jupyter Notebook

- Notebook Jupyter adalah aplikasi open source berbasis browser REPLA (Read–Eval–Print–Loop), aplikasi yang dapat kita gunakan untuk membuat dan berbagi dokumen yang berisi code, persamaan, visualisasi, dan teks).
- Nama, Jupyter, berasal dari bahasa pemrograman inti yang didukung yaitu: Julia, Python, dan R. Jupyter berisikan kernel IPython, yang memungkinkan kita untuk menulis program dengan Python

Variables

- Variabel digunakan untuk menyimpan nilai atau data apa pun.
- Python tidak memiliki perintah (command) untuk mendeklarasikan variabel. Variabel dibuat saat kita pertama kali menetapkan atau memasukan nilai pada varieabel tersebut.
- Peraturan untuk Python variables:
 - Nama variabel harus dimulai dengan huruf atau karakter garis bawah ‘_’
 - Nama variabel tidak boleh diawali dengan angka
 - Nama variabel hanya boleh berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan _)
 - Variable names are case-sensitive (age, Age and AGE are three different variables)
 - Nama variabel harus bersifat case-sensitive, penulisan (“age”, “Age”, dan “AGE” adalah tiga variabel berbeda)

Data Type



Name	Data Type	Description
Integer	int	a number that can be written without fractions: 22 10 0 -300
Boolean	bool	logical value that indicates True or False
Floating-point	float	a number that has a decimal component: 3.14 2.73 10.0
String	str	sequence of characters: "hello world" "hey88340" '2018'
List	list	sequence of comma-separated numbers, strings etc.: [10, '2018', "hi"]
Dictionary	dict	collection of key-value pairs: {"key1": "value1", "key2": "value2"}
Tuple	tuple	sequence of comma-separated numbers, strings etc.: (10, 20.0, "world", 5)

Assignment

Kita menggunakan assignment statements Python untuk menetapkan objek ke nama. Target dari assignment statements ditulis di sisi kiri tanda sama dengan (=), dan objek di sebelah kanan bisa menjadi ekspresi arbitrer yang mengkalkulasi objek.

Assignment Operators

Operator	Example	Equivalent Expression
=	$m = 10$	$m = 10$
$+=$	$m += 10$	$m = m + 10$
$-=$	$m -= 10$	$m = m - 10$
$*=$	$m *= 10$	$m = m * 10$
$/=$	$m /=$	$m = m/10$
$\% =$	$m \% = 10$	$m = m \% 10$
$<<=$	$a <<= b$	$a = a << b$
$>>=$	$a >>= b$	$a = a >> b$
$>>>=$	$a >>>= b$	$a = a >>> b$
$\&=$	$a \&= b$	$a = a \& b$
$\^=$	$a \^= b$	$a = a \^ b$
$ =$	$a = b$	$a = a b$

Comment

- Komentar (comment) dapat digunakan untuk menjelaskan kode Python, membuat kode lebih mudah dibaca dan untuk mencegah kode tereksekusi saat menguji kode.
- Comment dimulai dengan simbol '#', dan python akan mengabaikan kodennya

```
#This is a comment  
print("Hello, World!")
```

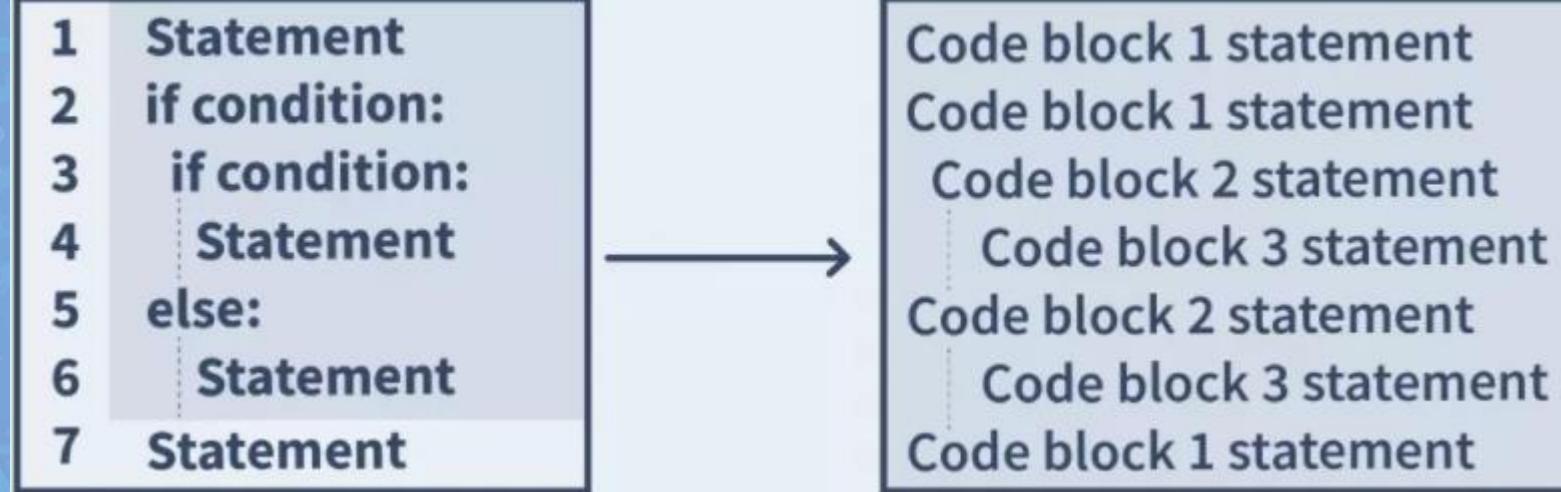
Arithmetic Programming

Kita dapat mengingat urutan kalkulasi aritmatikanya menggunakan PEMDAS: Parentheses, Exponents, Multiplication and Division (from left to right), Addition and Subtraction (from left to right).

Operators	Meaning	Example	Result
+	Addition	$4 + 2$	6
-	Subtraction	$4 - 2$	2
*	Multiplication	$4 * 2$	8
/	Division	$4 / 2$	2
%	Modulus operator to get remainder in integer division	$5 \% 2$	1
**	Exponent	$5^{**2} = 5^2$	25
//	Integer Division/ Floor Division	$5//2$ $-5//2$	2 -3

Indentation

Indentation adalah spasi (spasi dan tab) terdepan sebelum statement apa pun di dalam kode python. Indentation dalam bahasa lain hanya untuk keterbacaan, tetapi dalam python, indentation adalah konsep wajib yang harus diikuti saat menulis kode python, jika tidak, Indentation Error dilemparkan oleh juru bahasa python.



Whitespace

Dalam Python, spasi dan tab dikenali sebagai whitespace. Implementasi whitespace meningkatkan keterbacaan kode.

```
x=1+5 ls
```

equals to

```
x = 1 + 5
```

ls equals to

```
x = 1 + 5
```



THANK YOU