TUGAS INDIVIDU :

KAGGLE, GITHUB, KEDRO

Disusun untuk memenuhi tugas Mata Kuliah Pengantar Sains Data

Dosen Pengampu : **Eko Prasetio Widhi, S.Kom., M.Kom.**

Disusun oleh :

Ghulam Mushthofa 442023611060

PROGAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS DARUSSALAM GONTOR

2025

BAB I:

KAGGLE

1.1 Pengenalan Kaggle

Kaggle adalah platform *online* terbesar di dunia untuk komunitas *data scientist* dan praktisi *machine learning*. Anggap saja Kaggle ini seperti "media sosial" sekaligus "arena pertandingan" bagi para pegiat data. Di sini, orang bisa belajar, berkompetisi, dan berkolaborasi dalam proyek-proyek data.

Tiga fitur utama Kaggle adalah:

1. **Competitions (Kompetisi):** Perusahaan-perusahaan besar (seperti Google, Facebook, bahkan NASA) mempublikasikan masalah nyata beserta datanya, lalu menantang komunitas global untuk membuat model prediktif terbaik. Pemenangnya sering kali mendapatkan hadiah uang tunai yang besar.
2. **Datasets (Kumpulan Data):** Kaggle menyediakan ribuan *dataset* publik yang bisa diakses secara gratis. Ini adalah "harta karun" bagi siapa saja yang ingin berlatih analisis data, mulai dari data film di IMDb, data penyebaran COVID-19, hingga data gambar kucing dan anjing.
3. **Notebooks (Lingkungan Kode):** Kaggle menyediakan lingkungan koding berbasis *web* yang mirip dengan Jupyter Notebook. Pengguna bisa menulis dan menjalankan kode Python atau R langsung di *browser* tanpa perlu instalasi, lengkap dengan akses gratis ke GPU dan TPU yang sangat berguna untuk melatih model *deep learning*.

1.2 Manfaat Kaggle

Bagi seorang mahasiswa, Kaggle adalah "gym" untuk melatih otot-otot *data science*. Berikut adalah contoh cara memanfaatkannya:

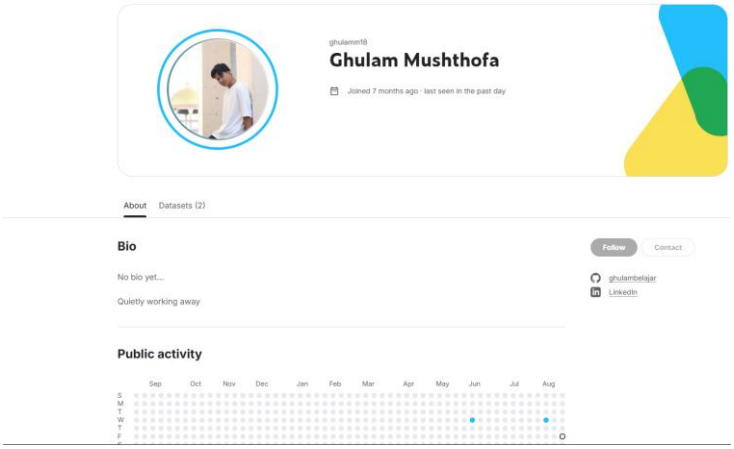
- **Belajar dari Dunia Nyata:** Mahasiswa bisa mengunduh *dataset* dari Kaggle untuk tugas kuliah atau proyek pribadi. Data di Kaggle sering kali lebih kompleks dan "berantakan" daripada data contoh di buku teks, sehingga memberikan pengalaman yang lebih realistis dalam membersihkan dan menganalisis data.
- **Membangun Portofolio:** Mahasiswa dapat mengerjakan analisis pada sebuah *dataset* menarik, menuliskannya dalam sebuah *Notebook* Kaggle, lalu mempublikasikannya.

Notebook yang insightful dan tervualisasi dengan baik ini bisa menjadi portofolio yang kuat untuk ditunjukkan kepada calon perusahaan saat melamar magang atau kerja.

- **Belajar dari Para Ahli:** Dengan mengikuti kompetisi (meskipun tidak harus menjadi juara), mahasiswa bisa melihat *notebooks* dan solusi dari para *Kaggle Grandmaster* (sebutan untuk peringkat tertinggi). Ini adalah cara cepat untuk mempelajari teknik-teknik canggih dan *best practice* dari para praktisi terbaik di dunia.
- **Mencoba Tanpa Batas:** Fitur *Notebooks* dengan GPU gratis memungkinkan mahasiswa untuk bereksperimen dengan model-model *machine learning* yang berat tanpa perlu memiliki komputer berspesifikasi tinggi.

1.3 Membuat Akun Kaggle

Untuk bagian ini, kita perlu membuat akun di kaggle.com jika belum punya. Prosesnya mudah, bisa menggunakan akun Google. Setelah berhasil membuat akun dan masuk, kunjungi halaman profil kita dengan mengklik ikon profil di pojok kanan atas, lalu pilih "Your Profile".



Gambar 1.2 :
(Profile Kaggle)

Salah satu dataset yang sangat menarik bagi saya di Kaggle adalah **"Heart Failure Prediction Dataset"**.

Dataset ini berisi rekam medis dari 299 pasien yang menderita gagal jantung. Yang membuatnya menarik adalah:

1. **Tujuan yang Jelas:** Tujuannya sangat spesifik, yaitu membangun model klasifikasi untuk memprediksi apakah seorang pasien akan meninggal dunia akibat gagal jantung berdasarkan 12 fitur klinis seperti umur, tingkat kreatinin, status diabetes, dan lain-lain.
2. **Relevansi Dunia Nyata:** Kasus ini memiliki dampak langsung pada dunia kesehatan. Model yang akurat dapat membantu dokter mengidentifikasi pasien berisiko tinggi lebih awal untuk memberikan penanganan yang lebih intensif.
3. **Cocok untuk Pemula:** Jumlah datanya tidak terlalu besar dan fiturnya mudah dipahami (campuran antara data numerik dan kategorikal), sehingga sangat cocok sebagai proyek latihan untuk menerapkan berbagai algoritma klasifikasi seperti *Logistic Regression*, *SVM*, atau *Random Forest*.

BAB II:

GITHUB

2.1 Pengenalan GitHub

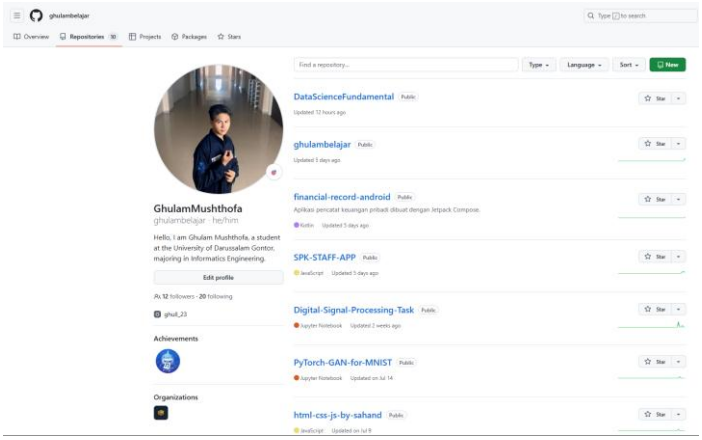
GitHub adalah sebuah platform berbasis *web* yang digunakan untuk *hosting* dan kolaborasi proyek perangkat lunak yang menggunakan sistem kontrol versi **Git**. Sederhananya, GitHub adalah "rumah" bagi kode-kode kamu di internet. Ini memungkinkan programmer, baik individu maupun tim, untuk menyimpan, mengelola, melacak perubahan, dan berkolaborasi dalam pengembangan kode.

Perbedaan antara Repository Publik dan Privat:

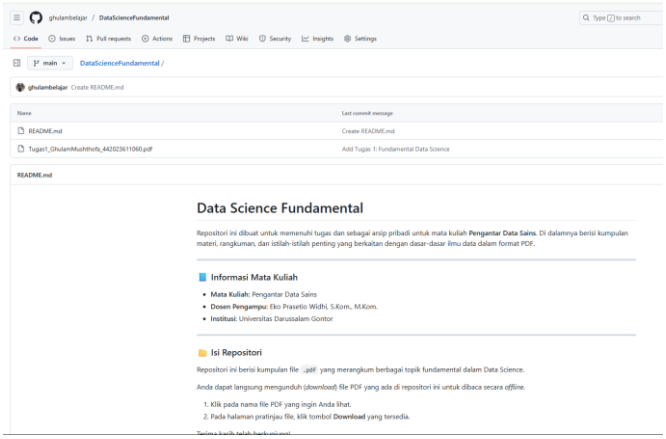
- **Repository Publik (Public):**
    - Dapat dilihat dan diakses oleh siapa saja di internet.
    - Siapa pun bisa menyalin (*fork*) atau mengunduh kodenya.
    - Biasanya digunakan untuk proyek-proyek *open-source*, portofolio pribadi, atau proyek yang memang ingin dibagikan kepada komunitas.
- **Repository Privat (Private):**
    - Hanya dapat diakses oleh pemilik *repository* dan orang-orang yang secara eksplisit diundang sebagai kolaborator.
    - Aksesnya terbatas dan tidak akan muncul di hasil pencarian publik.
    - Biasanya digunakan untuk proyek internal perusahaan, proyek rahasia, atau tugas-tugas pribadi yang belum ingin dipublikasikan.

2.2 Membuat Repository GitHub

Untuk soal ini, kamu perlu *login* ke akun GitHub, lalu buat *repository* baru dengan nama "DataScience-Fundamental". Jangan lupa untuk mencentang opsi "Add a README file" saat membuatnya. Tapi disini gak aku centang.



Gambar 2.1



Gambar 2.2

2.3 Kolaborasi GitHub

GitHub adalah tulang punggung kolaborasi dalam banyak proyek *software*, termasuk *Data Science*. GitHub memungkinkan sebuah tim untuk bekerja pada *file* kode yang sama secara bersamaan tanpa menimbulkan kekacauan atau saling menimpa pekerjaan satu sama lain.

Caranya adalah dengan sistem kontrol versi **Git**. Setiap anggota tim bisa bekerja di "cabang" (*branch*) mereka sendiri. Setelah selesai, perubahan tersebut bisa digabungkan kembali ke cabang utama (*main branch*) melalui proses yang terkontrol.

Contoh fitur yang digunakan untuk kolaborasi:

1. **Pull Request (PR):** Ini adalah fitur inti kolaborasi di GitHub. Ketika seorang anggota tim selesai mengerjakan sebuah fitur di *branch*-nya, ia akan membuat *Pull Request*. Ini adalah permintaan untuk menggabungkan kodenya ke *branch* utama. Sebelum digabungkan, anggota tim lain bisa me-review kode tersebut, memberikan komentar, meminta perbaikan, dan berdiskusi. Ini memastikan kualitas kode tetap terjaga.

2. **Issues:** Fitur ini berfungsi seperti papan tugas atau forum diskusi untuk sebuah proyek. Tim bisa menggunakannya untuk melacak *bug* (kesalahan program), meminta fitur baru, atau mendiskusikan ide-ide. Setiap *issue* bisa di-assign ke anggota tim tertentu, diberi label, dan dilacak progresnya.

3. **Forking:** Jika kamu ingin berkontribusi pada proyek *open-source* milik orang lain, kamu bisa membuat "salinan"-nya ke akunmu sendiri. Proses ini disebut *fork*. Kamu bebas melakukan perubahan pada salinan tersebut, dan jika menurutmu perubahan itu bermanfaat, kamu bisa mengirimkan *Pull Request* kembali ke pemilik proyek asli.

BAB III:

KEDRO

3.1 Pengenalan Kedro

Kedro adalah sebuah *framework* (kerangka kerja) berbasis Python yang *open-source* untuk membuat kode *data science* yang terstruktur, mudah direproduksi (*reproducible*), dan siap untuk produksi (*production-ready*). Kedro dikembangkan oleh QuantumBlack (sebuah perusahaan McKinsey).

Mengapa Kedro digunakan? Proyek *data science* yang dikerjakan di Jupyter Notebook sering kali menjadi sangat berantakan, sulit dibaca oleh orang lain, dan susah untuk dijalankan ulang di lingkungan yang berbeda. Kedro mengatasi masalah ini dengan menyediakan **struktur proyek standar**. Ibarat membangun rumah, Kedro menyediakan cetak biru (*blueprint*)-nya, sehingga setiap bagian (data, kode, konfigurasi) diletakkan di tempat yang seharusnya. Ini membuat proyek menjadi modular, mudah dipelihara, dan gampang untuk dikolaborasikan.

3.2 Struktur Proyek Kedro

Saat kita membuat proyek baru menggunakan Kedro, ia akan secara otomatis menghasilkan struktur folder yang standar. Lima komponen utamanya adalah:

1. **conf/:** Folder ini berisi semua file konfigurasi. Bagian terpenting di sini adalah catalog.yml (untuk mendaftarkan lokasi dan jenis semua sumber data) dan parameters.yml (untuk menyimpan parameter atau variabel yang mungkin ingin diubah-ubah, seperti *learning rate* pada model).

2. **data/:** Ini adalah lokasi standar untuk menyimpan data. Biasanya dibagi lagi menjadi beberapa sub-folder seperti 01_raw (data mentah), 02_intermediate (data hasil olahan sementara), 04_feature (data fitur untuk model), dan 07_model_output (hasil dari model).

3. **src/<nama_proyek>/pipelines/:** Di sinilah "otak" dari alur kerja didefinisikan. Sebuah *pipeline* adalah rangkaian dari beberapa *node* yang saling terhubung untuk mengubah data mentah menjadi hasil akhir.

4. **src/<nama_proyek>/nodes.py (atau di dalam folder pipelines):** File ini berisi fungsi-fungsi Python yang sebenarnya, di mana setiap fungsi melakukan satu tugas

spesifik (misalnya, membersihkan data, membuat fitur, atau melatih model). Setiap fungsi ini disebut *node*.

5. **notebooks/:** Folder khusus untuk melakukan analisis eksplorasi (*exploratory data analysis*) menggunakan Jupyter Notebook. Ini memisahkan kode untuk eksperimen dan eksplorasi dari kode inti *pipeline* yang bersih dan siap produksi.

3.3 Integrasi Kedro dengan GitHub

Mengintegrasikan proyek Kedro dengan GitHub pada dasarnya sama dengan proyek Python lainnya, karena Kedro hanyalah sebuah struktur folder. Langkah-langkah singkatnya adalah sebagai berikut:

1. **Buat Proyek Kedro di Komputer Lokal:** Buka terminal, lalu jalankan perintah kedro new untuk membuat proyek baru.
2. **Masuk ke Folder Proyek:** Pindah ke direktori proyek yang baru saja dibuat menggunakan perintah cd <nama-proyek-kedro>.
3. **Inisialisasi Git:** Jalankan git init di dalam folder tersebut. Template Kedro sudah menyertakan file .gitignore yang berisi daftar file dan folder yang tidak perlu dilacak oleh Git (seperti data mentah atau *virtual environment*).
4. **Buat Repository di GitHub:** Buka situs GitHub, buat sebuah *repository* baru yang **kosong** (tanpa README, license, atau .gitignore).
5. **Hubungkan Lokal ke Remote:** Salin URL dari *repository* GitHub yang baru dibuat, lalu jalankan perintah di terminal lokalmu: git remote add origin <URL_repository_kamu.git>
6. **Unggah Proyek (Push):** Lakukan *commit* pertama dan unggah semua file proyek ke GitHub:

```
git add .
git commit -m "Initial commit of Kedro project"
git push -u origin main
```

Proyek Kedro-mu sudah tersimpan dan bisa dikolaborasikan melalui GitHub.

BAB IV:

ANALISIS PERBANDINGAN

tabel perbandingan yang merangkum fungsi, manfaat, dan peran dari Kaggle, GitHub, dan Kedro dalam alur kerja *Data Science*. Berikut dibawah ini :

Alat	Fungsi Utama	Manfaat Utama	Peran dalam Alur Kerja Data Science
Kaggle	Platform komunitas, kompetisi, dan sumber daya data.	Tempat belajar, berlatih dengan data nyata, membangun portofolio, dan bersaing.	Tahap awal dan eksplorasi. Digunakan untuk mencari ide, menemukan dataset, dan bereksperimen dengan berbagai pendekatan analisis atau model.
GitHub	Platform untuk kontrol versi (Git) dan kolaborasi kode.	Memungkinkan kerja tim yang efisien, melacak setiap perubahan kode, dan menjadi "CV online" bagi developer/data scientist.	Sepanjang siklus hidup proyek. Digunakan dari awal hingga akhir untuk menyimpan, membagikan, dan mengelola semua kode proyek.
Kedro	Framework Python untuk standarisasi struktur proyek data science.	Membuat proyek menjadi terstruktur, modular, mudah direproduksi, dan siap untuk produksi. Mencegah kode menjadi "spaghetti code".	Tahap pengembangan dan produksi. Digunakan untuk membangun alur kerja (pipeline) yang solid dari awal hingga akhir, mulai dari pemrosesan data hingga deployment model.