**LAPORAN PRAKTIKUM BASIS DATA**

**SQL SUB QUERY**



**DISUSUN OLEH :**

Restu Wibisono

2340506061

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**

**JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK**

**UNIVERSITAS TIDAR**

**2024**

**LAPORAN**

**PRAKTIKUM BASIS DATA**



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Diisi Mahsiswa Praktikan** | | | | | | | | |
| Nama Praktikan | Restu Wibisono | | | | | | | |
| NPM | 2340506061 | | | | | | | |
| Rombel | 03 | | | | | | | |
| Judul Praktikum | SQL SUB QUERY | | | | | | | |
| Tanggal Praktikum | 25 April 2024 | | | | | | | |
| **Diisi Asisten Praktikum** | | | | | | | | |
| Tanggal Pengumpulan |  |  |  |  |  |  |  |  |
| Catatan |  | | | | | | | |

|  |  |  |
| --- | --- | --- |
| PENGESAHAN | | NILAI |
| Diperiksa oleh : | Disahkan oleh : |  |
| Asisten Praktikum | Dosen Pengampu |
|  |  |
| Nanda Cahya Septiawan | Imam Adi Nata, S.Kom., M.Kom. |

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS TIDAR**

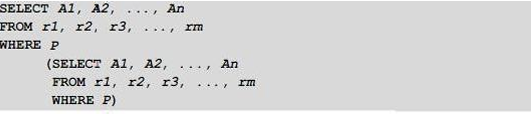
**Genap 2023/2024**

## Tujuan Praktikum

1. Mahasiswa mampu menjelaskan SQL bertingkat dalam basis data
2. Mahasiswa mampu menerapkan SQL bertingkat dalam basis data

## Dasar Teori

Subquery, atau yang juga dikenal sebagai nested query, adalah sebuah query (perintah SQL) yang disisipkan ke dalam query utama sebagai bagian dari klausa WHERE, FROM, atau SELECT. Subquery memungkinkan pengguna untuk mengeksekusi query yang lebih kompleks dengan menggabungkan hasil dari satu query ke dalam query lainnya.

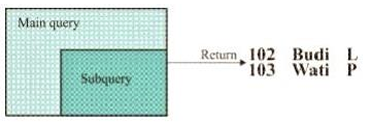


Subquery digunakan untuk mencapai tujuan tertentu dalam pengambilan data atau manipulasi data yang tidak dapat dicapai dengan menggunakan query tunggal. Dengan subquery, pengguna dapat menggabungkan data dari beberapa tabel, melakukan perbandingan antara hasil query, atau membuat pengambilan data yang lebih terperinci.

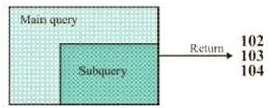
Subquery dapat disisipkan ke dalam klausa WHERE, FROM, atau SELECT dari query utama. Dalam klausa WHERE, subquery digunakan untuk memfilter baris berdasarkan kriteria tertentu. Dalam klausa FROM, subquery digunakan untuk menentukan sumber data untuk query utama. Sedangkan dalam klausa SELECT, subquery dapat menghasilkan nilai tunggal yang akan dimasukkan ke dalam hasil query utama.

Terdapat beberapa tipe subquery yaitu :

1. Multiple column Subquery: Subquery kolom ganda (multiple-column) mengembalikan lebih dari satu baris dan satu kolom data.

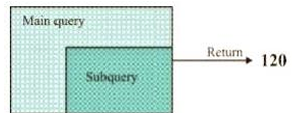


1. Multiple-row Subquery: Subquery ini menghasilkan beberapa baris data.



Subquery baris ganda dapat menggunakan operator komparasi IN, ANY, SOME, atau ALL.

1. Scalar Subquery: Subquery ini menghasilkan nilai tunggal, seperti hasil perhitungan atau pengambilan data tunggal.



Subquery baris tunggal dapat menggunakan operator baris tunggal =, >,>=, <=, atau <>.

Salah satu contoh umum dari subquery adalah penggunaannya dalam klausa WHERE untuk memfilter hasil query berdasarkan kriteria yang berasal dari hasil query lainnya. Misalnya, subquery dapat digunakan untuk mencari pelanggan yang melakukan pembelian diatas rata-rata total pembelian. Contoh lainnya adalah penggunaan subquery dalam klausa SELECT untuk menghasilkan nilai agregat, seperti jumlah total penjualan per kategori produk.

Keuntungan utama dari penggunaan subquery adalah kemampuannya untuk menangani tugas-tugas yang kompleks dengan logika yang lebih rumit dalam manipulasi data. Namun, penggunaan subquery yang berlebihan dapat menyebabkan penurunan kinerja dan kompleksitas yang sulit dipelihara dalam kode SQL.

Dengan pemahaman yang baik tentang subquery, pengguna dapat mengoptimalkan kinerja dan fleksibilitas dalam pengambilan data dan manipulasi data dalam basis data relasional. Subquery adalah alat yang berguna dalam pengembangan aplikasi database yang memungkinkan pengguna untuk mengeksekusi query yang lebih kompleks dan mendapatkan wawasan yang lebih dalam tentang data mereka.

## Metode Praktikum

## Alat dan bahan

Alat :

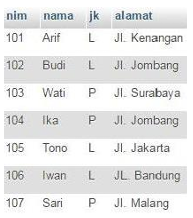
1. PC (Komputer)
2. Keyboard
3. Mouse

Bahan :

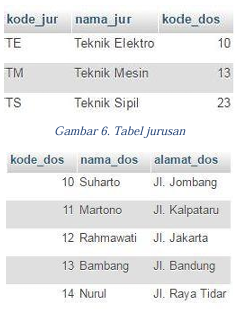
1. Operating System Windows 10
2. File Materi Praktikum
3. Aplikasi Paket Web server XAMPP
4. Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)

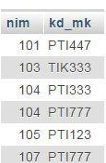
## Langkah kerja

Dalam latihan ini digunakan tiga buah tabel meliputi mahasiswa, ambil\_mk,matakuliah, dosen dan jurusan. Untuk itu, terlebih dahulu buat beberapa tabel dengan struktur sebagai berikut.

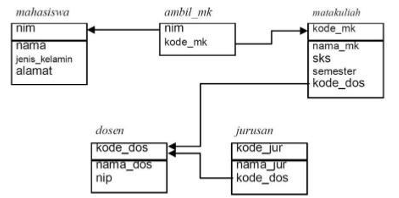






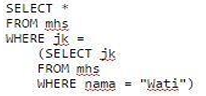


Himpunan entitas di atas dapat direpresentasikan kedalam diagram skema (schema diagram) seperti berikut.

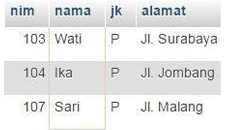


* + 1. SCALAR

Contoh subquery baris tunggal adalah mendapatkan data mahasiswa yang jenis kelaminnya sama dengan mahasiswa dengan nama “Wati”.



Sebagai hasilnya, didapatkan jenis kelamin mahasiswa dengan nama “Wati”, yakni “P” yang selanjutnya digunakan oleh main query sehingga menghasilkan sebagai berikut.

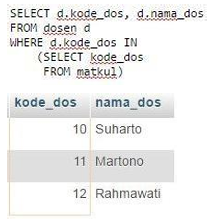


* + 1. Multiple-RowSubquery

Pada subquery ini, kita menggunakan operator komparasi IN, ANY / SOME, atau ALL.

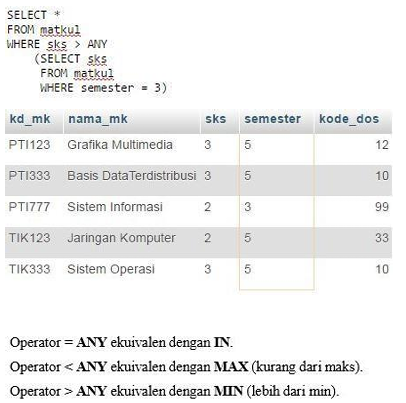
1. Operator IN

Operator IN memiliki arti : sama dengan member di dalam list. Sebagai contoh, kita bisa menggunakan operator ini untuk mendapatkan data dosen yang mengajar matakuliah.



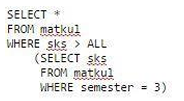
1. Operator ANY/SOME

Operator ANY / SOME memiliki arti : membandingkan suatu nilai dengan setiap nilai yang dikembalikan oleh subquery. Misalkan kita ingin mendapatkan data matakuliah yang memiliki sks lebih besar dari sembarang sks matakuliah di semester 3.



1. Operator ALL

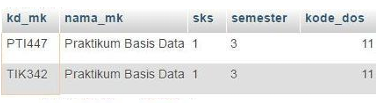
Operator ALL memiliki arti: membandingkan suatu nilai dengan semua nilai yang dikembalikan oleh subquery.Misal, kita ingin mendapatkan data matakuliah yang memiliki sks lebih besar dari semua sks matakuliah di semester 3.



****

* + 1. Multiple-ColumnSubquery

Subquery kolom ganda (atau tabel) juga menggunakan operator komparasi IN, ANY/ SOME, atau ALL. Pada query ini, nilai dari subquery dalam bentuk kolom ganda dikomparasi main query. Sebagai contoh, misalkan kita ingin menampilkan data matakuliah yang semester dan sksnya sesuai dengan semester dan sks matakuliah dengan kode “PTI447”.



* + 1. Operator EXISTS dan NOT EXISTS

Operator EXISTS dan NOT EXISTS digunakan pada correlated subquery untuk memeriksa apakah subquery mengembalikan hasil atau tidak. Apabila subquery mengembalikan hasil, EXIST akan mengembalikan nilai true. Begitu pula sebaliknya, jika tidak mengembalikan hasil.

1. EXIST

Pernyataan berikut akan mendapatkan data matakuliah yang diambil oleh mahasiswa.



1. NOT EXIST

Pernyataan berikut akan mendapatkan data matakuliah yang tidak diambil oleh mahasiswa.



## Soal

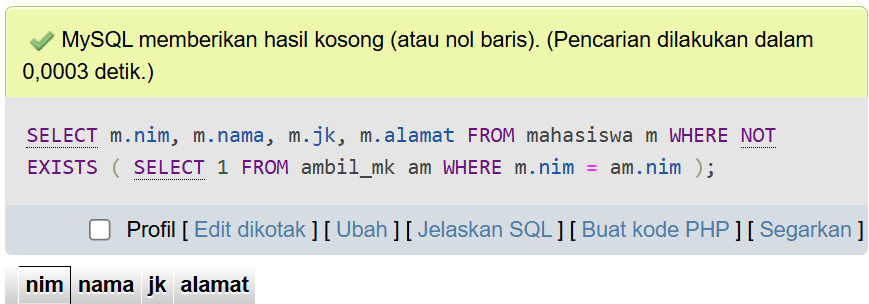
1. Dapatkan kode dan nama matakuliah dosen yang menjadi Ketua Jurusan Teknik Elektro.



(Gambar 4.1.1)

* Memilih kolom `kd\_mk` dan `nama\_mk` dari tabel `mata\_kuliah`.
* Menggabungkan `mata\_kuliah` dengan `dosen` berdasarkan `kode\_dos`.
* Menggabungkan `dosen` dengan `jurusan` berdasarkan `kode\_dos`.
* Menyaring data untuk hanya menampilkan mata kuliah yang diajarkan oleh dosen Ketua Jurusan Teknik Elektro (`j.nama\_jur = 'Teknik Elektro'`).

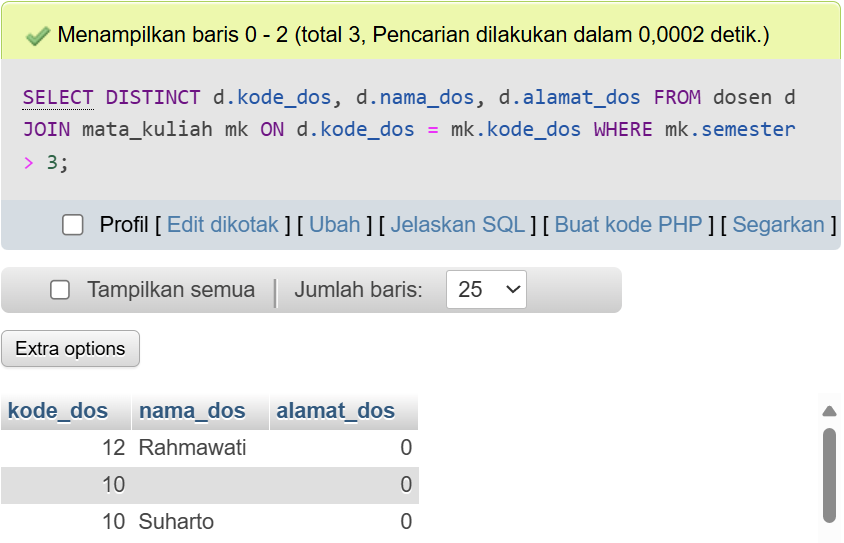
1. Dapatkan data mahasiswa yang tidak mengambil matakuliah.



(Gambar 4.2.1)

* Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.
* Kondisi untuk memeriksa bahwa tidak ada entri di tabel `ambil\_mk` yang memiliki `nim` yang sama dengan `nim` di tabel `mahasiswa`.
* `SELECT 1 FROM ambil\_mk am WHERE m.nim = am.nim` memeriksa apakah ada entri di `ambil\_mk` untuk mahasiswa tertentu.
* Memastikan mahasiswa tersebut tidak memiliki entri di tabel `ambil\_mk`, artinya mahasiswa tersebut tidak mengambil mata kuliah apapun.

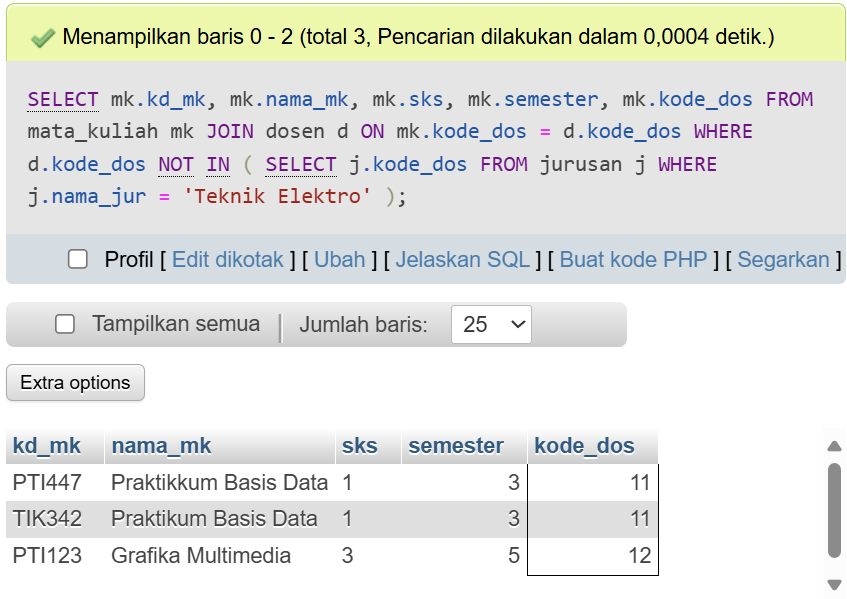
1. Dapatkan data dosen yang mengajar matakuliahdiatas semester 3.



(Gambar 4.3.1)

* Memilih kolom `kode\_dos`, `nama\_dos`, dan `alamat\_dos` dari tabel `dosen`.
* Menggunakan `JOIN` untuk menggabungkan tabel `dosen` dengan tabel `mata\_kuliah` berdasarkan kolom `kode\_dos`.
* Kondisi `WHERE` digunakan untuk memfilter baris yang hanya memiliki nilai semester di atas 3 dari tabel `mata\_kuliah`.

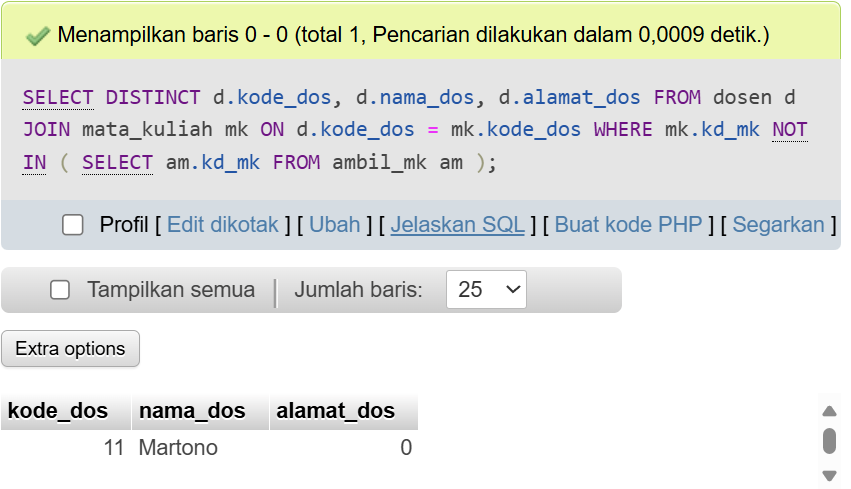
1. Dapatkan data matakuliah dosen yang bukan merupakan Ketua Jurusan Teknik Elektro.



(Gambar 4.4.1)

* Memilih kolom `kd\_mk`, `nama\_mk`, `sks`, `semester`, dan `kode\_dos` dari tabel `mata\_kuliah`.
* Menggunakan `JOIN` untuk menggabungkan tabel `mata\_kuliah` dengan tabel `dosen` berdasarkan kolom `kode\_dos`.
* Menggunakan subquery untuk memfilter baris dengan menggunakan kondisi `NOT IN`, di mana kode dosen yang bukan termasuk dalam daftar dosen yang terkait dengan jurusan 'Teknik Elektro' dari tabel `jurusan`.

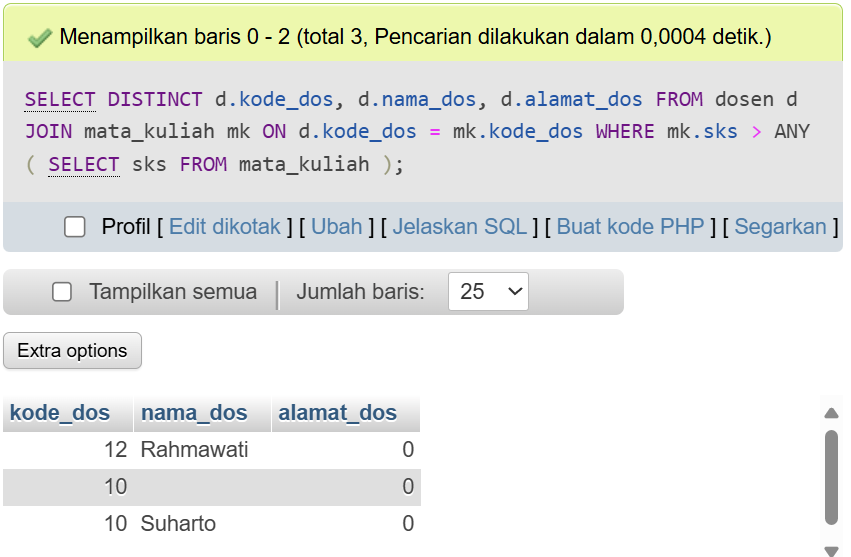
1. Dapatkan data dosen pengajar matakuliah yang tidak diambil oleh mahasiswa.



(Gambar 4.5.1)

* Memilih kolom `kode\_dos`, `nama\_dos`, dan `alamat\_dos` dari tabel `dosen`.
* Menggunakan `JOIN` untuk menggabungkan tabel `dosen` dengan tabel `mata\_kuliah` berdasarkan kolom `kode\_dos`.
* Menggunakan subquery untuk memfilter baris dengan menggunakan kondisi `NOT IN`, di mana kode mata kuliah yang tidak termasuk dalam daftar kode mata kuliah yang diambil oleh mahasiswa dari tabel `ambil\_mk`.

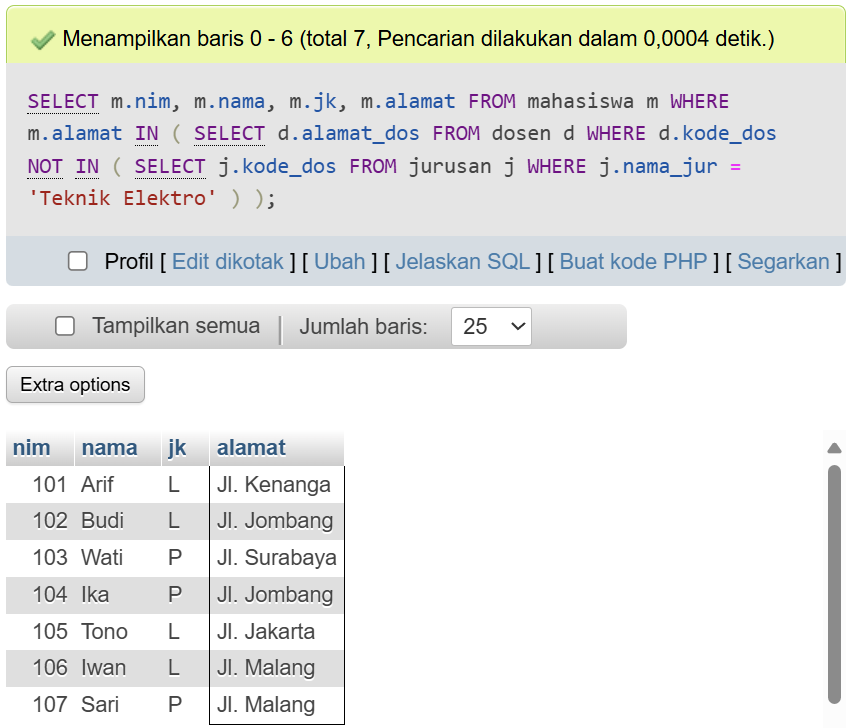
1. Dapatkan data dosen yang mengajar matakuliah dengan sks lebih besar dari sembarang sks.



(Gambar 4.6.1)

* Memilih kolom `kode\_dos`, `nama\_dos`, dan `alamat\_dos` dari tabel `dosen`.
* Menggunakan `JOIN` untuk menggabungkan tabel `dosen` dengan tabel `mata\_kuliah` berdasarkan kolom `kode\_dos`.
* Menggunakan subquery untuk memfilter baris dengan menggunakan kondisi `> ANY`, di mana jumlah SKS mata kuliah yang diajar oleh dosen lebih besar dari salah satu nilai SKS dari semua mata kuliah.

1. Dapatkan data mahasiswa yang tinggal satu wilayah dengan dosen yang bukan merupakan Ketua Jurusan Teknik Elektro.



(Gambar 4.7.1)

* Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.
* Menggunakan kondisi `WHERE` untuk memfilter baris berdasarkan alamat mahasiswa yang ada dalam hasil subquery.
* Subquery digunakan untuk mendapatkan alamat dosen dari tabel `dosen`, yang mana kode dosen tidak termasuk dalam daftar dosen yang terkait dengan jurusan 'Teknik Elektro' dari tabel `jurusan`.

1. Dapatkan data mahasiswa yang diajar oleh Ketua Jurusan Teknik Elektro.



(Gambar 4.8.1)

* Memilih kolom `nim`, `nama`, `jk` (jenis kelamin), dan `alamat` dari tabel `mahasiswa`.
* Menggunakan `JOIN` untuk menggabungkan tabel `mahasiswa` dengan tabel `ambil\_mk` berdasarkan kolom `nim`, kemudian menggabungkan hasilnya dengan tabel `mata\_kuliah` berdasarkan kolom `kd\_mk`, dan menggabungkan lagi dengan tabel `dosen` berdasarkan kolom `kode\_dos`.
* Selanjutnya, melakukan `JOIN` dengan tabel `jurusan` untuk memastikan bahwa dosen yang mengajar mata kuliah tersebut adalah dari jurusan 'Teknik Elektro'.
* Kondisi `WHERE` digunakan untuk memfilter baris di mana nama jurusan yang terhubung dengan dosen adalah 'Teknik Elektro'.

## Kesimpulan

Pemahaman yang baik tentang subquary akan meningkatkan kemampuan Anda untuk mengoptimalkan kinerja dan fleksibilitas pengambilan data dalam database relasional. Subquary adalah alat penting dalam pengembangan aplikasi database, memungkinkan Anda melakukan kueri yang lebih kompleks dan mendapatkan pemahaman yang lebih baik tentang data yang ada. Menguasai subquary bisa sangat berguna saat melakukan berbagai tugas analisis dan manipulasi data yang kompleks di dunia nyata.

## Referensi

Van Rossum, G. (2003). An introduction to Python (p. 115). F. L. Drake (Ed.).

Bristol: Network Theory Ltd.. [*http://atk.fam.free.fr/fichiers/stage/Python/JF/site/pytut.pdf*](http://atk.fam.free.fr/fichiers/stage/Python/JF/site/pytut.pdf)

Kuhlman, D. (2009). A python book: Beginning python, advanced python, and python exercises (pp. 1-227). Lutz: Dave Kuhlman.

[*https://www.davekuhlman.org/python\_book\_01.pdf*](https://www.davekuhlman.org/python_book_01.pdf)

Python, W. (2021). Python. Python Releases for Windows, 24. [*https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97 bfafd514ca2abb7e2c5c86bb*](https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb)