

**LAPORAN TUTORIAL LAB 1
BASIS DATA**



**ALDEN LUTHFI
2206028932
KELAS C**

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS INDONESIA
DEPOK
2024**

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



FACULTY OF
COMPUTER
SCIENCE

Latihan 1

1. [SQL] Jalankan SQL Query pada Contoh 1 hingga Contoh 27 di atas dan cantumkan hasilnya pada laporan

```
alden.luthfi=> SELECT nama
alden.luthfi=> FROM PRODUK
alden.luthfi=> WHERE ID_PRODUK = 'PRO014';
      nama
      -----
vivo V21
(1 row)

alden.luthfi=> SELECT nama, stok, harga
alden.luthfi=> FROM produk
alden.luthfi=> ORDER BY stok DESC;
      nama |   stok | harga
      -----
OPPO Reno |    993 | 5700000
Xiaomi Mi X3 |    937 | 6000000
Samsung Galaxy Note 20 |    817 | 8700000
OPPO Find X3 Pro |    801 | 5800000
Samsung Galaxy Note 10 |    840 | 6100000
OPPO Find X2 Pro |    817 | 1600000
Samsung Galaxy Z Fold 2 |    831 | 12000000
vivo V21 Pro |    501 | 7400000
Samsung Galaxy S20 |    796 | 8300000
Samsung Galaxy M11 |    678 | 4700000
Xiaomi Mi 10 |    658 | 4100000
OPPO Reno 5 |    689 | 6600000
Samsung Galaxy S21 |    580 | 10000000
Xiaomi Poco M3 |    514 | 2300000
vivo V20 |    589 | 2700000
vivo V19 |    445 | 4900000
Xiaomi Poco X3 NFC |    376 | 4600000
Xiaomi Mi 11 |    349 | 10000000
Samsung Galaxy Note 9 |    284 | 7400000
Samsung Galaxy Note 10 |    145 | 3600000
Xiaomi Mi 9 |    88 | 1000000
Xiaomi Mi 11 |    75 | 8800000
```

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



UNIVERSITAS
INDONESIA

Veritas, Praelata, Intellia

FACULTY OF
COMPUTER
SCIENCE

```
Samsung Galaxy M21 | 66 | 5300000
OPPO Reno 4 | 46 | 6200000
Samsung Galaxy Note 20 | 2 | 3800000
(25 rows)
```

```
alden.luthfi=> SELECT nama
alden.luthfi=> FROM produk
alden.luthfi=> WHERE harga >= 1000000 and harga <= 2000000;
nama
```

```
Samsung Galaxy S21
Xiaomi Redmi Note 9
Xiaomi Mi 9
OPPO Find X2 Pro
(4 rows)
```

```
alden.luthfi=> SELECT nama
alden.luthfi=> FROM produk
alden.luthfi=> WHERE nama LIKE 'X%2_';
nama
```

```
vivo V20
vivo V21
(2 rows)
```

```
alden.luthfi=> (SELECT id_kurir
alden.luthfi=> FROM TRANSAKT
alden.luthfi=> WHERE status = 'Diproses')
alden.luthfi=> UNION
alden.luthfi=> (SELECT id_kurir
alden.luthfi=> FROM TRANSAKT
alden.luthfi=> WHERE status = 'Dikirim');
id_kurir
```

```
KUR002
KUR003
KUR004
KUR005
KUR007
KUR008
KUR009
KUR010
KUR012
KUR013
KUR014
KUR015
KUR016
KUR018
(12 rows)
```

```
alden.luthfi=> (SELECT id_pembeli
alden.luthfi=> FROM PEMBAYARAN
alden.luthfi=> WHERE metode_pembayaran = 'Transfer Bank')
alden.luthfi=> INTERSECT
alden.luthfi=> (SELECT id_pembeli
alden.luthfi=> FROM PEMBAYARAN
alden.luthfi=> WHERE metode_pembayaran = 'Kartu Kredit');
id_pembeli
```

```
(0 rows)
```

```
alden.luthfi=> (SELECT id_pembayaran
alden.luthfi=> FROM pembayaran
alden.luthfi=> WHERE nominal > 500000)
alden.luthfi=> EXCEPT
alden.luthfi=> (SELECT id_pembayaran
alden.luthfi=> FROM pembayaran
alden.luthfi=> WHERE metode_pembayaran = 'Kartu Kredit');
id_pembayaran
```

```
PMB001
PMB002
PMB003
PMB004
PMB005
PMB006
PMB007
PMB008
PMB009
PMB010
PMB011
PMB012
PMB013
PMB014
PMB015
PMB016
PMB017
PMB018
(17 rows)
```

```
alden.luthfi=> (SELECT id_pembayaran, id_pembeli
alden.luthfi=> FROM pembayaran
alden.luthfi=> WHERE nominal > 500000)
alden.luthfi=> EXCEPT
alden.luthfi=> (SELECT id_pembeli
alden.luthfi=> FROM pembayaran
alden.luthfi=> WHERE metode_pembayaran = 'Kartu Kredit');
ERROR: each EXCEPT query must have the same number of columns
LINE 5: (SELECT id_pembeli
          ^
```

```
alden.luthfi=> SELECT id_pembayaran
alden.luthfi=> FROM PEMBAYARAN
alden.luthfi=> WHERE EXTRACT(MONTH FROM tanggal_pembayaran) > 6;
id_pembayaran
```

```
PMB001
PMB006
PMB007
PMB009
PMB012
PMB013
PMB014
PMB015
PMB019
(9 rows)
```

```
alden.luthfi=> SELECT COUNT(*)
alden.luthfi=> FROM PRODUK
count
```

```
25
(1 row)
```

```
alden.luthfi=> SELECT COUNT(*)
alden.luthfi=> FROM BRAND;
count
```

```
4
(1 row)
```

```
alden.luthfi=> SELECT COUNT(*)
alden.luthfi=> FROM PRODUK, BRAND;
count
```

```
100
(1 row)
```

```
alden.luthfi=> SELECT PEMBELI.NAMA, NOMINAL
alden.luthfi=> FROM PEMBELI, PEMBAYARAN
alden.luthfi=> WHERE PEMBELI.ID_PEMBELI = PEMBAYARAN.ID_PEMBELI
alden.luthfi=> AND NOMINAL > 5000000;
name | nominal
```

```
Kathleen Gravenell | 5100000
Rochelle Cornwell | 6600000
Cara Sevier | 6800000
Cristina Capasso | 5400000
Noxi D'Emmanuele | 5200000
Trev Domingues | 57100000
Cameron Swann | 5300000
Dix Somers | 6780000
Tina Conchitron | 6500000
```

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



FACULTY OF
**COMPUTER
SCIENCE**

Andras Menezes	7409000													
Lina Guirardin	5800000													
(11 rows)														
alden.luthfi> SELECT *														
alden.luthfi> FROM PRODUK P JOIN TRANSAKSI_PODUK TP ON P.ID_PRODUK = TP.ID_PRODUK														
alden.luthfi> JOIN TRANSAKSI T ON TP.ID_TRANSAKSI = T.ID_TRANSAKSI;														
id_produk id_brand nama stok deskripsi harga id_transaksi id_produkt id_transaksi id_pembeli id_toko id_kurir tanggal_transaksi status no_resi														
PRD022	BNNB03	OPPO Reno 4	46	Toxic effect of contact w/ other venomous plant, undetermined	6280000	TRK001	PRD022	TRK001	PMB017	TOKO003	2823-02-02	Mengunggu Pembayaran		
PRD003	BNNB02	Xiaomi Redmi Note 9	324	Confusion of right middle finger w/ damage to nail, sub-	18000000	TRK002	PRD003	TRK001	PMB027	TOKO006	KUR008	2823-08-23	Diproses	77845555
PRD017	BNNB03	Samsung Galaxy Note 10	18	Disol fiss sexf of ulna, 1 arc, 7cm	61500000	TRK003	PRD017	TRK003	PMB011	TOKO003	KUR012	2823-05-01	Dikirim	2663162869
alden.luthfi>														

alderi_luthfi@SELECT + alderi_luthfi@ FROM PRODUK P JOIN TRANSAKSI_PRODUK TP ON P.ID_PRODUK = TP.ID_PRODUK alderi_luthfi@ JOIN TRANSAKSI T ON TP.ID_TRANSAKSI = T.ID_TRANSAKSI;														
id_produkt	id_brandt	namat	stokt	deskripsi	hargat	id_transaksi	id_produkt	id_transaksi	id_pembeli	id_toko	id_kurir	tanggal_transaksi	status	no_resi
PROD022	BND02	OPPO Reno 4	46	Toxic effect of contact w/oth venomous plant, undetermined	5200000	TP0081	P0022	TP0081	TOKR083	KUR083		2024-07-07	Menunggu Pembayaran	
PROD043	BND02	Samsung Realme Note 9	33	Contusion of right middle finger w/ damage to nail, subs	1800000	TP0083	P0023	TP0083	TOKR086	KUR088		2023-06-23	Diproses	778455528
PROM17	BND01	Samsung Galaxy Note 10	840	Dispil see fx shaft of ulta, l.han, 7thC	6100000	TP0083	P0017	TP0083	TOKR089	KUR102		2023-05-03	Dikirim	2663162869
PROM01	BND01	Samsung Galaxy S21	680	Fracture of first digit mst sites of head, face, and neck	1100000	TP0084	P0001	TP0084	TOKR084	KUR103		2023-04-01	Diterima Pembeli	378357728
PROM84	BND03	OPPO Reno 5	500	TPin pres, unspl num 8/9 amio sacs, first trimester	6600000	TP0085	P0009	TP0085	TOKR085	KUR089		2023-03-29	Selesai	9966768498
PROM11	BND02	Xiaomi Poco X3	97	Open bite of right index finger w/o damage to nail, subs	3300000	TP0087	P0011	TP0087	TOKR087	KUR086		2023-03-29	Selesai	333957728
PROM11	BND02	Xiaomi Poco X3	97	Open bite of right index finger w/o damage to nail, subs	3300000	TP0087	P0011	TP0087	TOKR087	KUR086		2023-03-29	Selesai	625518354
PROM05	BND01	vivo V20	589	Ankylosis, left shoulder	2700000	TP0085	P0007	TP0085	TOKR082	KUR103		2023-03-23	Dikirim	625518354
PROM10	BND02	Xiaomi Mi 9	88	Straight ms/lnd eng ext mst toe at ank/fi lev, l.foot, sola	3000000	TP0088	P0010	TP0088	TOKR088	KUR088		2023-03-19	Diterima Pembeli	384252951
PROM13	BND01	Samsung Galaxy Note 10	145	Other disorders of continuity of bone, right fibula	3000000	TP0089	P0013	TP0089	TOKR089	KUR101		2023-03-14	Selesai	2522212862
PROM13	BND01	Samsung Galaxy Note 10	145	Other disorders of continuity of bone, right fibula	3000000	TP0089	P0013	TP0089	TOKR089	KUR101		2023-03-14	Selesai	2522212862
PROM19	BND01	Samsung Galaxy Z Fold 2	810	Liver & biliary tract disrc in pres, pregn, childrbn and the puerp	3300000	TP0091	P0019	TP0091	TOKR011	KUR084		2023-03-26	Diproses	281556548
PROM11	BND02	Xiaomi Poco X3	97	Open bite of right index finger w/o damage to nail, subs	3300000	TP0092	P0011	TP0092	TOKR012	PMB014		2023-03-24	Dikirim	5136787881
PROD22	BND03	OPPO Reno 4	46	Toxic effect of contact w/oth venomous plant, undetermined	6200000	TP0093	P0002	TP0093	TOKR010	KUR011		2023-03-16	Diterima Pembeli	166176285
PROM88	BND02	Xiaomi Mi 11	75	Severe persistent asthma, uncomplicated	8800000	TP0094	P0018	TP0094	TOKR104	KUR202		2023-02-28	Selesai	1613335719
PROM11	BND02	Samsung Galaxy S20	100	Fracture of right ulna, humerus, and/or radius	5300000	TP0095	P0011	TP0095	TOKR105	KUR202		2023-02-27	Selesai	1613335719
PROM19	BND01	Samsung Galaxy Z Fold 2	810	Liver & biliary tract disrc in pres, pregn, childrbn and the puerp	3300000	TP0095	P0019	TP0095	TOKR105	KUR202		2023-02-27	Selesai	1613335719
PROM07	BND01	Samsung Galaxy M01	66	Type II occipital condyle fracture, right side, 7thC	5300000	TP0096	P0007	TP0096	TOKR106	KUR087		2023-02-18	Diproses	9664347852
PROM13	BND01	Samsung Galaxy Note 10	145	Other disorders of continuity of bone, right fibula	3000000	TP0097	P0013	TP0097	TOKR107	PMB025		2023-02-17	Dikirim	804994616
PROM10	BND02	Xiaomi Mi 9	88	Straight ms/lnd eng ext mst toe at ank/fi lev, l. foot, sola	3000000	TP0098	P0010	TP0098	TOKR108	KUR082		2023-02-17	Dikirim	804994616
PROM11	BND02	Xiaomi Poco X3	97	Open bite of right middle finger w/o damage to nail, subs	3300000	TP0099	P0011	TP0099	TOKR109	KUR081		2023-02-17	Diterima Pembeli	2865673765
PROM11	BND02	Xiaomi Poco X3	97	Open bite of right middle finger w/o damage to nail, subs	3300000	TP0099	P0011	TP0099	TOKR109	KUR081		2023-02-17	Selesai	2865673765
PROM12	BND01	Samsung Galaxy Note 20	917	Subluxation of T4/T5 thoracic vertebrae, subsequent encounter	8700000	TP0100	P0012	TP0100	TOKR110	PMB013		2023-02-11	Diproses	3887855764
PROM88	BND02	Xiaomi Mi 11	75	Severe persistent asthma, uncomplicated	8800000	TP0101	P0002	TP0101	TOKR201	KUR010		2023-02-08	Selesai	710331671
PROD85	BND04	vivo V20	589	Ankylosis, left shoulder	2700000	TP0102	P0005	TP0102	TOKR022	PMB039		2023-02-01	Diterima Pembeli	2077426802
PROM15	BND01	Samsung Galaxy Z Fold 2	810	Inflammation of skin, initial encounter	4700000	TP0103	P0015	TP0103	TOKR023	PMB037		2023-01-31	Selesai	527976848
PROM13	BND01	Samsung Galaxy Note 10	145	Other disorders of continuity of bone, right fibula	3000000	TP0103	P0013	TP0103	TOKR023	PMB039		2023-01-31	Selesai	527976848
PROD24	BND03	OPPO Find X3 Pro	881	Impression of muscle/fasc/tend mthd at ws/hed lv, subs	5800000	TP0104	P0002	TP0104	TOKR024	PMB009		2023-01-26	Diproses	8783677743

```
alden.luthfi:~$ SELECT STATUS, K.id_kurir  
alden.luthfi:~$ FROM TRANSAKSI T LEFT JOIN KURIR K ON T.ID_KURIR = K.ID_KURIR;  
+-----+-----+  
| status | id_kurir |
```

Menunggu Pembayaran	
Diproses	KUR008
Dikirim	KUR012
Diterima Pembeli	KUR011
Selanjutnya	KUR004
Diproses	KUR018
Dikirim	KUR006
Diterima Pembeli	KUR004
Selanjutnya	KUR019
Dikirim	KUR003
Diproses	KUR014
Dikirim	KUR016
Diterima Pembeli	KUR011
Selanjutnya	KUR020
Menunggu Pembayaran	
Diproses	KUR007
Dikirim	KUR002
Diterima Pembeli	KUR017
Selanjutnya	KUR005
Diproses	KUR009
Dikirim	KUR010
Diterima Pembeli	KUR008
Selanjutnya	KUR012
Diproses	KUR013

```
alden.luthfi@SELECT STATUS, K.id_kurir
```

alder.lutnfrd	FROM TRANSAKSI T RIGHT JOIN KURIR K ON T.ID_KURIR = K.ID_KURIR;
status	id_kurir
Diproses	KURR08
Dikirim	KURR12
Diterima Pembeli	KURR13
Selasa	KURR01
Diproses	KURR18
Dikirim	KURR06
Diterima Pembeli	KURR04
Selasa	KURR19
Dikirim	KURR03
Diproses	KURR16
Dikirim	KURR10
Diterima Pembeli	KURR11
Selasa	KURR20
Diproses	KURR07
Dikirim	KURR02
Diterima Pembeli	KURR17
Selasa	KURR15
Diproses	KURR09
Dikirim	KURR10
Diterima Pembeli	KURR08
Selasa	KURR12
Diproses	KURR13
Diproses	KURR05

```
alden.luthfi> SELECT id_transaksi, K_id_kurir  
alden.luthfi> SELECT id_transaksi, K_id_kurir
```

```
alihantiketnya = FROM KURIR_K FULL OUTER JOIN TRANSAKSI_T ON K.ID_KURIR = T.ID_KURIR;
ID_transaksi = ID_kurir;
```

aliden.luthfi@...	SELECT *											
aliden.luthfi@...	FROM PRODUK_P											
aliden.luthfi@...	NATURAL JOIN TRANSAKSI_PRODUK_TP											
aliden.luthfi@...	NATURAL JOIN TRANSAKTI_T;											
id_transaksi	id_produk	id_brand	nama	stok	deskripsi	harga	id_pembeli	id_toko	id_kurir	tanggal_transaksi	status	no_resi
TRK001	PDR022	BND03	Oppo Reno 4	146	Toxic effect of contact w/oth venomous plant, undetermined	6200000	PMB017	TKO003		2024-02-02	Menunggu Pembayaran	
TRK003	PDR003	BND02	Xiaomi Redmi Note 9	320	Contusion of right middle finger w/ damage to nail, subs	1500000	PMB027	TKO006	KUR008	2023-08-23	Diproses	7784555208
TRK003	PDR017	BND01	Samsung Galaxy Note 10	848	Displ seg frx shaft of ulna, l arm, 7HIC	6100000	PMB011	TKO009	KUR012	2023-05-03	Dikirim	2663181854
TRK004	PDR008	BND01	Samsung Galaxy S21	680	Open bite of first molar sites of head, face, and neck	1500000	PMB024	TKO004	KUR013	2023-05-01	Menunggu Pembeli	238377286
TRK005	PDR005	BND04	Vivo V20	580	Open bite of first molar sites of head, face, and neck	1500000	PMB022	TKO005	KUR014	2023-04-03	Salah Pesanan	3079999999
TRK005	PDR005	BND04	Vivo V20	580	Ankylosis, left shoulder	2700000	PMB016	TKO010	KUR018	2023-05-20	Diproses	5379999467
TRK007	PDR011	BND02	Xiaomi Poco X3	937	Open bite of right index finger w/o damage to nail, subs	3300000	PMB022	TKO002	KUR006	2023-07-03	Dikirim	6255181854
TRK007	PDR005	BND04	Vivo V20	580	Ankylosis, left shoulder	2700000	PMB002	TKO002	KUR006	2023-07-03	Dikirim	6255181854
TRK008	PDR010	BND02	Xiaomi Mi 9	88	Strain msmt/straining extn mid toe at ank/fk lev, l foot, sqa	1500000	PMB003	TKO008	KUR004	2023-09-13	Diterima Pembeli	38452352891
TRK009	PDR013	BND01	Samsung Galaxy Note 10	145	Other disorders of continuity of bone, right fibula	3000000	PMB015	TKO001	KUR019	2023-04-19	Salah Pesanan	2522212962
TRK010	PDR004	BND04	Vivo V19	445	Drown due to fall off (nonpowered) inflatable craft, subs	4500000	PMB034	TKO005	KUR003	2023-11-22	Dikirim	2723587738
TRK011	PDR019	BND01	Samsung Galaxy Z Fold 2	618	Liver & biliary tract disord in preg, childbirth and the puer	8300000	PMB008	TKO004	KUR018	2023-06-24	Diproses	8231556848
TRK012	PDR001	BND02	Xiaomi Poco X3	937	Open bite of right index finger w/o damage to nail, subs	3300000	PMB014	TKO002	KUR016	2023-04-09	Dikirim	5136787861

Laporan Tutorial Lab 2

Basis Data

CSGE602070



FACULTY OF
**COMPUTER
SCIENCE**

Semester Genap 2023/2024

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



UNIVERSITAS INDONESIA

Veritas, Praeludes, Intellit

FACULTY OF
COMPUTER
SCIENCE

2. [SQL] Tampilkan nama brand yang memiliki panjang nama antara 3 hingga 7 karakter (inklusif). Anda wajib menggunakan LIKE.

```
alden.luthfi⇒ SELECT nama FROM BRAND WHERE nama LIKE '__%X' AND LENGTH(nama) ≤ 7;  
nama  
Samsung  
Xiaomi  
OPPO  
VIVO  
(4 rows)
```

3. [SQL] Tampilkan seluruh nama pengguna dan jumlah nominal pembayaran yang pernah dilakukannya. Jika tidak pernah melakukan pembayaran, tampilkan nominal 0. Urutkan dari yang paling banyak mengeluarkan uang untuk pembayaran. Apabila jumlah pembayaran sama, urutkan berdasarkan namanya dari A – Z.

```
alden.luthfi⇒ SELECT PL.nama as nama_pengguna, COALESCE(PM.nominal, 0) as jumlah_pembayaran  
alden.luthfi⇒ FROM Pengguna PL JOIN Pembayaran PM  
alden.luthfi⇒ ON PL.id_pembeli = PM.id_pembeli  
alden.luthfi⇒ ORDER BY COALESCE(PM.nominal, 0) DESC, PL.nama;  
nama_pengguna | jumlah_pembayaran  
Trev Duncinges | 17200000  
Jill Addison | 6000000  
Dix Connors | 5700000  
Corey Castagnasso | 5300000  
Andras Menezes | 7400000  
Rochelle Cornwell | 6500000  
Rock D'Amoule | 2000000  
Kathleen Gravewell | 6100000  
Cara Sevier | 6000000  
Lina Guirardin | 5500000  
Cameron Swann | 5300000  
Jill Hartman | 5000000  
Romney Lucas | 4500000  
Maxim Disney | 4000000  
Doloris Beddo | 3300000  
Gonzales Whooley | 3300000  
Karilynn Lamm | 3000000  
Mikel Al Maggi | 1000000  
Selinda Cockett | 2700000  
Darcey Crawshay | 1800000  
Holden Dellow | 1100000  
Katy Seaver | 1000000  
Bridgette Champneys | 0  
Brigitte Robinet | 0  
Camille Ellerton | 0  
Candide Fagg | 0  
Catherine Yardley | 0  
Cleopatra Miyashin | 0  
Davin Cleaves | 0  
Harriette Kippins | 0  
Jed Madigan | 0  
Juice Wimpress | 0  
Kris Pyle | 0  
Lilie Dunkerly | 0  
Lucas Timbridge | 0  
Redd Carnaghan | 0  
Ryan Culterford | 0  
Ryan Homan | 0  
Tabbi Poundesford | 0  
Torrin Legonidec | 0  
(48 rows)
```

4. [SQL] Tampilkan jumlah transaksi yang terjadi setiap bulan. Urutkan yang paling dari yang paling banyak terjadi transaksi.

```
alden.luthfi⇒ SELECT EXTRACT(MONTH FROM T.tanggal_transaksi) as bulan_ke, COUNT(id_transaksi) AS jumlah_transaksi  
alden.luthfi⇒ FROM TRANSAKSI T  
alden.luthfi⇒ GROUP BY EXTRACT(MONTH FROM T.tanggal_transaksi)  
alden.luthfi⇒ ORDER BY COUNT(id_transaksi) DESC;  
bulan_ke | jumlah_transaksi  
4 | 4  
6 | 3  
7 | 3  
5 | 3  
3 | 3  
9 | 2  
8 | 2  
11 | 2  
2 | 1  
1 | 1  
10 | 1  
(11 rows)
```

5. [SQL] Tampilkan jumlah transaksi yang terjadi berdasarkan status transaksi dari toko yang beralamat di wilayah 'Dki Jakarta'.

```
alden.luthfi⇒ SELECT T.status as status_transaksi, COUNT(id_transaksi) as jumlah_transaksi  
alden.luthfi⇒ FROM TRANSAKSI T, TOKO O  
alden.luthfi⇒ WHERE T.id_toko = O.id_toko and O.alamat LIKE 'DKI Jakarta'  
alden.luthfi⇒ GROUP BY T.status;  
status_transaksi | jumlah_transaksi  
Dikirin | 2  
Diproses | 2  
Diterima Pembeli | 2  
Menunggu Pembayaran | 2  
Selesai | 5  
(5 rows)
```

6. [SQL] Tampilkan rata-rata penjualan dari setiap toko.

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



UNIVERSITAS
INDONESIA

Veritas, Praelata, Intellia

FACULTY OF
COMPUTER
SCIENCE

```
alden.luthfi→ SELECT O.id_toko, ROUND(AVG(P.nominal), 2)
alden.luthfi→ FROM TOKO O LEFT JOIN TRANSAKSI T ON O.id_toko = T.id_toko
alden.luthfi→ JOIN PEMBAYARAN P ON T.id_transaksi = P.id_transaksi
alden.luthfi→ GROUP BY O.id_toko
alden.luthfi→ ORDER BY id_toko;
id_toko | round
10 rows
```

7. [SQL] Tampilkan nama pembeli yang sudah menerima produk pada transaksi yang tercatat, yaitu yang memiliki status “Diterima Pembeli” atau “Selesai”.

```
alden.luthfi→ SELECT K.nama_kurir, COUNT(T.id_transaksi) as jumlah_pengiriman
alden.luthfi→ FROM KURIR K LEFT JOIN TRANSAKSI T ON K.id_kurir = T.id_kurir
alden.luthfi→ GROUP BY K.nama_kurir;
nama_kurir | jumlah_pengiriman
Widya Hasanah | 1
Wesita Dangran | 1
Rusman Praksa | 2
Marsudi Prabowo | 1
Akarsana Situmorang | 2
Mustika Ramadhan | 0
Cengkih Mulyadi | 1
Balanga Jallani | 1
Banwi Irawan | 1
Sakura Widjastuti | 1
Dartono Latupone | 1
Sakura Handayani | 1
Farhumina Aryani | 1
Darsirah Wijaya | 1
Jagaraga Prasetyo | 1
Cahyanto Hardiansyah | 1
Ketarya Indradi | 2
Alisira Rahimah | 1
Diana Susanti | 1
Jawi Pranowo | 1
20 rows
```

8. [SQL] Tampilkan nama kurir dan jumlah pengiriman yang dilakukan (termasuk yang tidak pernah melakukan pengiriman).

```
alden.luthfi→ SELECT K.nama_kurir, COUNT(T.id_transaksi) as jumlah_pengiriman
alden.luthfi→ FROM KURIR K LEFT JOIN TRANSAKSI T ON K.id_kurir = T.id_kurir
alden.luthfi→ GROUP BY K.nama_kurir;
nama_kurir | jumlah_pengiriman
Widya Hasanah | 1
Wesita Dangran | 1
Rusman Praksa | 2
Marsudi Prabowo | 1
Akarsana Situmorang | 2
Mustika Ramadhan | 0
Cengkih Mulyadi | 1
Balanga Jallani | 1
Banwi Irawan | 1
Sakura Widjastuti | 1
Dartono Latupone | 1
Sakura Handayani | 1
Farhumina Aryani | 1
Darsirah Wijaya | 1
Jagaraga Prasetyo | 1
Cahyanto Hardiansyah | 1
Ketarya Indradi | 2
Alisira Rahimah | 1
Diana Susanti | 1
Jawi Pranowo | 1
20 rows
```

9. [SQL] Tentukan metode pembayaran yang paling diminati oleh pembeli.

```
alden.luthfi→ SELECT metode_pembayaran, M.banyak
alden.luthfi→ FROM (SELECT metode_pembayaran, count(*) as banyak
alden.luthfi→ GROUP BY metode_pembayaran) AS M
alden.luthfi→ WHERE M.banyak = (SELECT MAX(jumlah_digunakan)
alden.luthfi→ FROM (SELECT metode_pembayaran, count(*) as jumlah_digunakan
alden.luthfi→ FROM PEMBAYARAN
alden.luthfi→ GROUP BY metode_pembayaran) as MP);
metode_pembayaran | banyak
Kartu Kredit | 5
Transfer Bank | 5
2 rows
```

10. [SQL] Tentukan brand produk yang paling diminati oleh pembeli.

```
alden.luthfi→ SELECT BRAND.nama, COUNT(*) AS jumlah
alden.luthfi→ FROM BRAND
alden.luthfi→ JOIN TRANSAKSI_T_PODUKT ON BRAND.id_brand = PRODUK.id_brand
alden.luthfi→ JOIN TRANSAKSI_PODUKT ON PRODUK.id_produk = TRANSAKSI_PODUKT.id_produk
alden.luthfi→ GROUP BY BRAND.nama
alden.luthfi→ ORDER BY jumlah DESC;
alden.luthfi→ LIMIT 1;
nama | jumlah
+-----+
Samsung | 11
1 row
```

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



Latihan 2

- [SQL]** Jalankan SQL query pada Contoh 28 hingga Contoh 36 di atas dan cantumkan hasilnya pada laporan.

```
alden.luthfi→ CREATE VIEW daftar_toko AS
alden.luthfi→   SELECT id_toko, nama
alden.luthfi→   FROM toko;
CREATE VIEW
alden.luthfi→   SELECT * FROM daftar_toko;
id_toko |
TOKOK01 | Fadel, Prabowo and Becker
TOKOK02 | Darmawulan and Schaller
TOKOK03 | Bartoletti-Grant
TOKOK04 | Ankunding-Hoppe
TOKOK05 | Bannister-Crona
TOKOK06 | Lira-Oliver
TOKOK07 | Dach-Murphy
TOKOK08 | Welshi, Goyette and Hyatt
TOKOK09 | Ziemer and Pfeifer
TOKOK10 | Braxton-Saupe
(10 rows)

alden.luthfi→ DROP VIEW daftar_toko;
DROP VIEW
alden.luthfi→ CREATE INDEX index_status_transaksi ON transaksi(status);
CREATE INDEX
alden.luthfi→ CREATE INDEX index_nama_pembeli ON pembeli USING HASH (nama);
CREATE INDEX
alden.luthfi→ CREATE INDEX index_kurir ON kurir (id_kurir, nama_kurir ASC);
CREATE INDEX
alden.luthfi→ CREATE INDEX index_nama_produk ON produk (nama);
CREATE INDEX
alden.luthfi→ CREATE INDEX index_id_brand_produk ON produk (id_brand);
CREATE INDEX
alden.luthfi→ DROP INDEX index_status_transaksi;
DROP INDEX
alden.luthfi→ DROP INDEX index_nama_pembeli;
DROP INDEX
alden.luthfi→ DROP INDEX index_kurir;
DROP INDEX
alden.luthfi→ DROP INDEX index_nama_produk;
DROP INDEX
alden.luthfi→ DROP INDEX index_id_brand_produk;
DROP INDEX
alden.luthfi→ EXPLAIN ANALYZE
alden.luthfi→ SELECT *
alden.luthfi→ FROM kurir
alden.luthfi→ WHERE nama_perusahaan = 'JNE';
QUERY PLAN
Seq Scan on kurir  (cost=0.00..1.25 rows=1 width=452) (actual time=0.022..0.025 rows=1)
  Filter: ((nama_perusahaan)::text = 'JNE'::text)
  Rows Removed by Filter: 1
Planning Time: 0.000 ms
Execution Time: 0.054 ms
(5 rows)

alden.luthfi→
```

2. View

- [TRIVIA]** Apa yang akan terjadi jika kita membuat View menggunakan nama yang sama dengan nama tabel yang ada pada database? Jelaskan!

Akan terjadi error karena view itu ibarat table/relasi baru pada database. sehingga jika kita menggunakan nama yang sama, SQL akan bingung ketika melakukan query.

- [TRIVIA]** Apa fungsi TEMP atau TEMPORARY di View?

View temporary adalah view yang hanya bisa digunakan selama database terhubung, view tersebut akan ada selama sesi koneksi dan akan terhapus saat sesi koneksi tersebut ditutup.

- [SQL]** Buatlah View yang menyimpan **id produk** dan **nama produk** beserta **total transaksi** pada produk tersebut. Jika suatu produk tidak memiliki transaksi, nama produk tetap ditampilkan, namun totalnya ditampilkan 0. Urutkan secara ascending berdasarkan id produk.

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



UNIVERSITAS
INDONESIA

Veritas, Praelatis, Intellita

FACULTY OF
COMPUTER
SCIENCE

```
alden.luthfi=> CREATE VIEW transaksi_produk_virtual AS
alden.luthfi=> SELECT P.id_produk, P.nama, COUNT(T.id_transaksi), 0 FROM PRODUK P
alden.luthfi=> JOIN TRANSAKSI_PODUK T ON P.id_produk = T.id_produk
alden.luthfi=> GROUP BY P.id_produk, P.nama
alden.luthfi=> ORDER BY P.id_produk;
CREATE VIEW
alden.luthfi=> SELECT * FROM transaksi_produk_virtual;
+-----+-----+
| id_produk | nama | coalesce(
+-----+-----+
| PR0001 | Samsung Galaxy S21 | 1
| PR0002 | Samsung Galaxy S20 | 1
| PR0003 | Samsung Galaxy Note 9 | 1
| PR004 | OPPO Reno 5 | 1
| PR005 | vivo V20 | 3
| PR006 | vivo V19 | 1
| PR007 | Samsung Galaxy M21 | 1
| PR008 | Xiaomi Mi 11 | 2
| PR009 | Xiaomi Mi 10 | 1
| PR010 | Xiaomi 9 | 2
| PR011 | Xiaomi Poco X3 | 3
| PR012 | Samsung Galaxy Note 20 | 1
| PR013 | Samsung Galaxy Note 10 | 3
| PR014 | Samsung Galaxy Note 10+ | 1
| PR015 | Samsung Galaxy Note 10 | 1
| PR016 | Samsung Galaxy Z Fold 2 | 2
| PR027 | OPPO Reno 4 | 2
| PR024 | OPPO Find X3 Pro | 1
+-----+-----+
(18 rows)
```

- d. [SQL] Buatlah View yang menyimpan **nama pembeli** (apabila lebih dari 1, pisahkan dengan koma) yang melakukan transaksi pada produk dan dikelompokkan berdasarkan id brand dan **nama brand** dari produk.
(HINT: string_agg)

```
alden.luthfi=> CREATE VIEW transaksi_pembeli AS
alden.luthfi=> SELECT B.id_brand, B.nama_brand, STRING_AGG(PE.nama, ', ') FROM BRAND B
alden.luthfi=> JOIN PRODUK_P ON B.id_brand=PR.id_brand
alden.luthfi=> JOIN TRANSAKSI_PODUK TP ON PR.id_produk = TP.id_produk
alden.luthfi=> JOIN TRANSAKSI T ON TP.id_transaksi = T.id_transaksi
alden.luthfi=> JOIN PEMBELI PE ON T.id_pembeli = PE.id_pembeli
alden.luthfi=> GROUP BY B.id_brand, B.nama
alden.luthfi=> ORDER BY B.id_brand;
CREATE VIEW
alden.luthfi=> SELECT * FROM transaksi_pembeli;
+-----+-----+
| id_brand | nama_brand | string_agg
+-----+-----+
| BN001 | Samsung | Kathleen Gravenell, Holden Dellow, Karyllyn Aspole, Correy Castagnasso, Trev Dominguez, Catherine Yardley, Cameron Swann, Maxim Disney, Dix Somers, Andras Menezes, Malena Di Biaggi
| BN002 | Xiaomi | Darcey Crashaway, Cara Sevier, Kately Seares, Gonzales Wholey, Trev Dominguez, Maxim Disney, Rooney Lucas, Dalaris Beddo, Joe Coddington
| BN003 | OPPO | Reddi Carnaghan, Rochell Cornwell, Roxi D'Emmanuele, Lina Guitard
| BN004 | VIVO | Selinda Cockett, Cara Sevier, Jim Thirwell, Andras Menezes
+-----+-----+
(4 rows)
```

3. Indexing dan Analyze

- a. [SQL] Jalankan perintah EXPLAIN ANALYZE untuk setiap query di atas. Screenshot eksekusinya dan tulis hasilnya pada tabel di bawah, sertakan dalam laporan submisi Anda.

```
alden.luthfi=> EXPLAIN ANALYZE
alden.luthfi=> SELECT *
alden.luthfi=> FROM PEMBELI
alden.luthfi=> ORDER BY tanggal_lahir DESC;
                                         QUERY PLAN
Sort  (cost=2.46..2.56 rows=4 width=464) (actual time=0.122..0.126 rows=4 loops=1)
  Sort Key: tanggal_lahir DESC
  Sort Method: quicksort  Memory: 31kB
    > Seq Scan on pembeli  (cost=0.00..1.40 rows=4 width=464) (actual time=0.053..0.057 rows=4 loops=1)
Planning Time: 1.413 ms
Execution Time: 0.166 ms

alden.luthfi=> EXPLAIN ANALYZE
alden.luthfi=> SELECT *
alden.luthfi=> FROM PRODUK
alden.luthfi=> WHERE harga > 3000000;
                                         QUERY PLAN
Seq Scan on produk  (cost=0.00..1.31 rows=8 width=334) (actual time=0.016..0.021 rows=8 loops=1)
  Filter: (harga > 3000000)
  Rows Removed by Filter: 7
  Planning Time: 0.001 ms
  Execution Time: 0.046 ms
(8 rows)

alden.luthfi=> EXPLAIN ANALYZE
alden.luthfi=> SELECT *
alden.luthfi=> FROM KURIR
alden.luthfi=> WHERE nama_kurir
alden.luthfi=> LIKE 'W%';
                                         QUERY PLAN
Seq Scan on kurir  (cost=0.00..1.25 rows=1 width=452) (actual time=0.118..0.121 rows=2 loops=1)
  Filter: ((nama_kurir)::text ~ 'W%'::text)
  Rows Removed by Filter: 18
  Planning Time: 0.001 ms
  Execution Time: 0.140 ms
(2 rows)

alden.luthfi=> EXPLAIN ANALYZE
alden.luthfi=> SELECT *
alden.luthfi=> FROM TRANSAKSI
alden.luthfi=> ORDER BY tanggal_transaksi
alden.luthfi=> LIMIT 10;
                                         QUERY PLAN
Limit  (cost=1.70..1.78 rows=10 width=256) (actual time=0.072..0.075 rows=10 loops=1)
  > Sort  (cost=1.70..1.82 rows=24 width=256) (actual time=0.071..0.072 rows=10 loops=1)
    Sort Key: tanggal_transaksi
    Sort Method: quicksort  Memory: 27kB
      > Seq Scan on transaksi  (cost=0.00..1.24 rows=24 width=256) (actual time=0.045..0.048 rows=24 loops=1)
Planning Time: 0.345 ms
Execution Time: 0.115 ms
(7 rows)
```

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024

b. [SQL] Buat index berikut (method nya terserah Anda):

- index_tanggal_lahir_pembeli** pada tabel **PEMBELI** kolom **tanggal_lahir**.

```
alden.luthfi=> CREATE INDEX index_tanggal_lahir_pembeli ON PEMBELI(tanggal_lahir);
CREATE INDEX
alden.luthfi=> \d PEMBELI
          Table "simart.pembeli"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 id_pembeli | character varying(10) | | not null |
 nama | character varying(100) | | not null |
 no_telp | character varying(14) | | not null |
 email | character varying(50) | | not null |
 alamat | text | | not null |
 tanggal_lahir | date | | not null |
 gender | character(1) | | not null |
Indexes:
 "pembeli_pkey" PRIMARY KEY, btree (id_pembeli)
 "index_tanggal_lahir_pembeli" btree (tanggal_lahir)
Referenced by:
 TABLE "pembayaran" CONSTRAINT "pembayaran_id_pembeli_fkey" FOREIGN KEY (id_pembeli) REFERENCES pembeli(id_pembeli) ON UPDATE CASCADE ON DELETE CASCADE
 TABLE "transaksi" CONSTRAINT "transaksi_id_pembeli_fkey" FOREIGN KEY (id_pembeli) REFERENCES pembeli(id_pembeli) ON UPDATE CASCADE ON DELETE CASCADE
```

- index_harga_produk** pada tabel **PRODUK** kolom **harga**.

```
alden.luthfi=> CREATE INDEX index_harga_produk ON PRODUK(harga);
CREATE INDEX
alden.luthfi=> \d PRODUK
          Table "simart.produk"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 id_produk | character varying(10) | | not null |
 id_brand | character varying(10) | | not null |
 nama | character varying(100) | | not null |
 stok | integer | | not null |
 deskripsi | text | | not null |
 harga | integer | | not null |
Indexes:
 "produk_pkey" PRIMARY KEY, btree (id_produk)
 "index_harga_produk" btree (harga)
Foreign-key constraints:
 "produk_id_brand_fkey" FOREIGN KEY (id_brand) REFERENCES brand(id_brand) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
 TABLE "toko_produk" CONSTRAINT "toko_produk_id_produk_fkey" FOREIGN KEY (id_produk) REFERENCES produk(id_produk) ON UPDATE CASCADE ON DELETE CASCADE
 TABLE "transaksi_produk" CONSTRAINT "transaksi_produk_id_produk_fkey" FOREIGN KEY (id_produk) REFERENCES produk(id_produk) ON UPDATE CASCADE ON DELETE CASCADE
```

- index_nama_kurir** pada tabel **KURIR** kolom **nama_kurir**.

```
alden.luthfi=> CREATE INDEX index_nama_kurir ON KURIR (nama_kurir);
CREATE INDEX
alden.luthfi=> \d KURIR
          Table "simart.kurir"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 id_kurir | character varying(10) | | not null |
 nama_kurir | character varying(100) | | not null |
 no_telp | character varying(14) | | not null |
 jenis_layanan | text | | not null |
 nama_perusahaan | character varying(50) | | not null |
Indexes:
 "kurir_pkey" PRIMARY KEY, btree (id_kurir)
 "index_nama_kurir" btree (nama_kurir)
Referenced by:
 TABLE "transaksi" CONSTRAINT "transaksi_id_kurir_fkey" FOREIGN KEY (id_kurir) REFERENCES kurir(id_kurir) ON UPDATE CASCADE ON DELETE CASCADE
```

- index_tanggal_transaksi** pada tabel **TRANSAKSI** kolom **tanggal_transaksi**.

```
alden.luthfi=> CREATE INDEX index_tanggal_transaksi ON TRANSAKSI(tanggal_transaksi);
CREATE INDEX
alden.luthfi=> \d TRANSAKSI
          Table "simart.transaksi"
 Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 id_transaksi | character varying(10) | | not null |
 id_pembeli | character varying(10) | | not null |
 id_toko | character varying(10) | | not null |
 id_kurir | character varying(10) | | not null |
 tanggal_transaksi | date | | not null |
 status | character varying(20) | | not null |
 no_resi | character varying(12) | | not null |
Indexes:
 "transaksi_pkey" PRIMARY KEY, btree (id_transaksi)
 "index_tanggal_transaksi" btree (tanggal_transaksi)
Foreign-key constraints:
 "transaksi_id_kurir_fkey" FOREIGN KEY (id_kurir) REFERENCES kurir(id_kurir) ON UPDATE CASCADE ON DELETE CASCADE
 "transaksi_id_pembeli_fkey" FOREIGN KEY (id_pembeli) REFERENCES pembeli(id_pembeli) ON UPDATE CASCADE ON DELETE CASCADE
 "transaksi_id_toko_fkey" FOREIGN KEY (id_toko) REFERENCES toko(id_toko) ON UPDATE CASCADE ON DELETE CASCADE
Referenced by:
 TABLE "pembayaran" CONSTRAINT "pembayaran_id_transaksi_fkey" FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi) ON UPDATE CASCADE ON DELETE CASCADE
 TABLE "transaksi_produk" CONSTRAINT "transaksi_produk_id_transaksi_fkey" FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi) ON UPDATE CASCADE ON DELETE CASCADE
```

Laporan Tutorial Lab 2

Basis Data

CSGE602070

Semester Genap 2023/2024



UNIVERSITAS
INDONESIA

Veritas, Praelata, Intellita

FACULTY OF
COMPUTER
SCIENCE

- c. [SQL] Jalankan kembali **setiap query SELECT** di atas dari pertanyaan nomor 3 menggunakan perintah EXPLAIN ANALYZE. Screenshot eksekusinya dan tulis hasilnya pada tabel di bawah, sertakan dalam laporan submisi Anda.

```
alden.luthfi⇒ EXPLAIN ANALYZE
alden.luthfi⇒ SELECT *
alden.luthfi⇒ FROM PEMBELI
alden.luthfi⇒ ORDER BY tanggal_lahir DESC;
                                         QUERY PLAN
Sort  (cost=2.46..2.56 rows=40 width=464) (actual time=0.032..0.036 rows=40 loops=1)
  Sort Key: tanggal_lahir DESC
  Sort Method: quicksort  Memory: 31kB
  → Seq Scan on pembeli  (cost=0.00..1.40 rows=40 width=464) (actual time=0.010..0.014 rows=40 loops=1)
    Planning Time: 0.469 ms
    Execution Time: 0.059 ms
  (4 rows)

alden.luthfi⇒ EXPLAIN ANALYZE
alden.luthfi⇒ SELECT *
alden.luthfi⇒ FROM PRODUK
alden.luthfi⇒ WHERE harga > 3000000;
                                         QUERY PLAN
Seq Scan on produk  (cost=0.00..1.31 rows=8 width=334) (actual time=0.014..0.019 rows=18 loops=1)
  Filter: (harga > 3000000)
  Rows Removed by Filter: 7
  Planning Time: 0.344 ms
  Execution Time: 0.042 ms
  (5 rows)

alden.luthfi⇒ EXPLAIN ANALYZE
alden.luthfi⇒ SELECT *
alden.luthfi⇒ FROM KURIR
alden.luthfi⇒ WHERE nama_kurir
alden.luthfi⇒ LIKE '%K';
                                         QUERY PLAN
Seq Scan on kurir  (cost=0.00..1.25 rows=1 width=452) (actual time=0.024..0.027 rows=2 loops=1)
  Filter: ((nama_kurir)::text ~ 'WK'::text)
  Rows Removed by Filter: 18
  Planning Time: 0.350 ms
  Execution Time: 0.049 ms
  (5 rows)

alden.luthfi⇒ EXPLAIN ANALYZE
alden.luthfi⇒ SELECT *
alden.luthfi⇒ FROM TRANSAKSI
alden.luthfi⇒ ORDER BY tanggal_transaksi
alden.luthfi⇒ LIMIT 10;
                                         QUERY PLAN
Limit  (cost=1.76..1.78 rows=10 width=256) (actual time=0.029..0.032 rows=10 loops=1)
  → Sort  (cost=1.76..1.82 rows=24 width=256) (actual time=0.027..0.028 rows=10 loops=1)
    Sort Key: tanggal_transaksi
    Sort Method: top-N heapsort  Memory: 27kB
    → Seq Scan on transaksi  (cost=0.00..1.24 rows=24 width=256) (actual time=0.010..0.013 rows=24 loops=1)
      Planning Time: 0.337 ms
      Execution Time: 0.068 ms
  (7 rows)
```

- d. [Trivia] Bandingkan *planning time* dan *execution time* (menggunakan tabel di bawah) dari *query* saat tanpa index dan setelah menggunakan *index*. Mana yang lebih baik? Berikan penjelasan!

Penggunaan indexing mempengaruhi waktu dengan sangat baik, semua query yang dilakukan setelah indexing, walaupun ada sedikit outlier kemampuan optimisasi indexing untuk dataset yang banyak akan lebih terlihat dampaknya.

Query	Planning Time		Execution Time	
	TANPA INDEX	DENGAN INDEX	TANPA INDEX	DENGAN INDEX
1	1.413	0.489	0.166	0.059
2	0.091	0.344	0.046	0.042
3	0.398	0.350	0.140	0.049
4	0.345	0.337	0.115	0.068

Laporan Tutorial Lab 2
Basis Data
CSGE602070
Semester Genap 2023/2024

