

LAPORAN UAS
PROJECT SISTEM KASIR TOKO PLANET
KEMASAN



UNIVERSITAS
DUTA BANGSA
SURAKARTA

DISUSUN OLEH:

Hafidz Fathkurrohman (250119009)

Zalfa' Lutfi Batrisyia (250119025)

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DUTA BANGSA SURAKARTA

2025

DAFTAR ISI

COVER

DAFTAR ISI	1
BAB I	2
PENDAHULUAN	2
1.1 Latar Belakang.....	2
1.2 Tujuan dan Ruang Lingkup	2
1.3 Gambaran Umum Sistem.....	2
BAB II	3
LANDASAN TEORI	3
2.1 Basis Data dan DBMS	3
2.2 Entity Relationship Diagram (ERD) dan Relasi	3
2.3 Normalisasi Data	4
2.4 SQL dan Koneksi Database.....	4
BAB III	5
PERANCANGAN DAN IMPLEMENTASI SISTEM BASIS DATA	5
3.1 Gambaran Umum Sistem.....	5
3.2 Perancangan Basis Data	5
3.3 Entity Relationship Diagram (ERD)	5
3.4 Implementasi Struktur Tabel (DDL).....	6
3.5 Implementasi Data Manipulation Language (DML).....	6
3.6 Implementasi Transaction Control Language (TCL)	6
3.7 Implementasi Query SQL.....	6
BAB IV	7
PENUTUP	7
4.1 Hasil Pengujian Sistem	7
4.2 Kendala dan Solusi	7
4.3 Kesimpulan	7
4.4 Saran Pengembangan	7
LAMPIRAN	8
DAFTAR PUSTAKA	13

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu aspek penting dalam menjalankan kegiatan usaha adalah pengelolaan data penjualan, termasuk bagaimana proses transaksi berlangsung. Hal ini tidak dapat dilakukan sembarangan karena memerlukan ketelitian agar tidak terjadi kesalahan yang dapat menyebabkan masalah bagi usaha. Dengan adanya teknologi informasi yang semakin maju, pemilik usaha dapat memanfaat hal ini untuk mengelola data, terutama data penjualan dan kegiatan transaksi. Hal ini dilakukan agar pengelolaan data menjadi lebih terstruktur dan efisien.

Pencatatan data penjualan yang dilakukan secara manual sangat berpotensi menimbulkan kesalahan pencatatan, redundansi data, dan kesulitan saat membuat laporan. Oleh karena itu, penting untuk menggunakan sistem basis data agar hal tersebut dapat dihindari. Sistem basis data mampu menyimpan, mengelola, dan menampilkan data penjualan dengan baik, terutama jika jumlah datanya banyak. Dengan sistem basis data ini, data dapat disimpan dalam bentuk tabel yang saling berhubungan, sehingga menjaga konsistensi dan integritas data.

Studi kasus pada proyek ini adalah sistem penjualan dari toko “Planet Kemasan”. Dalam laporan ini, Kami akan merancang basis data untuk sistem penjualan toko berdasarkan nota transaksi nyata. Nota tersebut berisi informasi seperti data pelanggan, operator, produk, jumlah pembelian, serta pembayaran. Berdasarkan nota tersebut, akan dirancang basis data dengan baik agar data dapat dikelola secara efisien menggunakan MySQL.

1.2 Tujuan dan Ruang Lingkup

Pembahasan ini berfokus pada perancangan basis data dan implementasi SQL di MySQL dengan tujuan:

1. Melakukan proses normalisasi data
2. Merancang ERD (Entity Relationship Diagram)
3. Mengimplementasikan hasil rancangan menggunakan SQL di dalam MySQL
4. Melakukan pengujian query untuk memastikan query dapat berfungsi

1.3 Gambaran Umum Sistem

Sistem ini dibuat untuk menyimpan data transaksi, data produk, data operator dan customer, serta data pembayaran. Sistem dirancang menggunakan system basis data relasional dengan MySQL sebagai tempat menyimpan data.

BAB II

LANDASAN TEORI

2.1 Basis Data dan DBMS

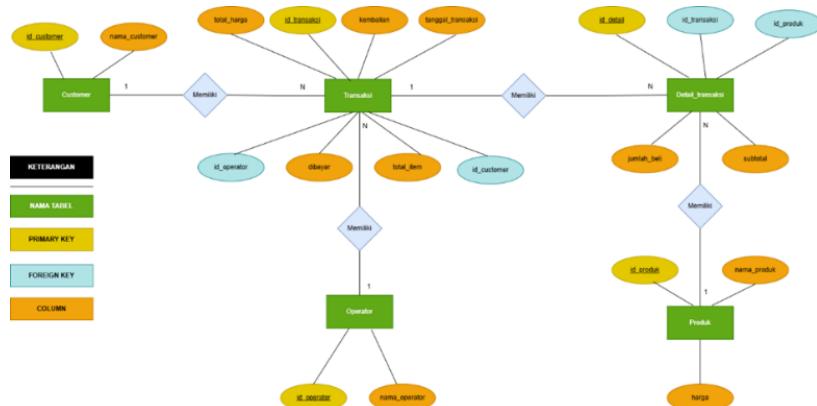
Kumpulan data yang saling berhubungan dan disimpan dengan baik secara bersama-sama tanpa pengulangan (redudansi) yang tidak diperlukan agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah^[1].

Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk mengelola database secara terstruktur sehingga pengguna dapat menyimpan data, mengakses, mengubah, bahkan menghapus dengan mudah^[2].

2.2 Entity Relationship Diagram (ERD) dan Relasi

ERD merupakan diagram untuk memodelkan struktur basis data sebagai tahap awal dalam perancangan basis data, fungsinya untuk menggambarkan entitas, atribut, relasi, serta kardinalitas antar entitas^[3].

Relasi antar tabel ini menjelaskan bagaimana setiap tabel dapat saling terhubung melalui Primary Key (PK) dan Foreign Key (FK). Jenis relasi antar tabel meliputi relasi One-to-One (1:1), One-to-Many (1:N), dan Many-to-Many (N:M) yang digunakan untuk menggambarkan hubungan logis antar entitas dalam basis data^[4].



- ID_Customer berperan sebagai primary key pada tabel Customer dan digunakan sebagai foreign key pada tabel Transaksi untuk menghubungkan data transaksi dengan customer yang melakukan transaksi.
- ID_Operator berperan sebagai primary key pada tabel Operator dan digunakan sebagai foreign key pada tabel Transaksi untuk menunjukkan operator yang memproses transaksi.
- ID_Transaksi berperan sebagai primary key pada tabel Transaksi dan digunakan sebagai foreign key pada tabel Detail_Transaksi untuk menghubungkan transaksi dengan detail item yang dibeli.

- ID_Produk berperan sebagai primary key pada tabel Produk dan digunakan sebagai foreign key pada tabel Detail_Transaksi untuk menunjukkan produk yang terlibat dalam suatu transaksi.
- Tabel Produk ke tabel Detail_Transaksi relasinya adalah One-to-Many (1:N).
- Tabel Customer ke tabel Transaksi relasinya adalah One-to-Many (1:N).
- Tabel Operator ke tabel Transaksi relasinya adalah One-to-Many (1:N).
- Tabel Transaksi ke tabel Detail_Transaksi relasinya adalah One-to-Many (1:N).

2.3 Normalisasi Data

Normalisasi adalah proses mengelompokkan data ke dalam tabel-tabel yang terstruktur untuk mengurangi redundansi dan meningkatkan integritas data. Proses normalisasi dilakukan melalui beberapa tahapan, yaitu First Normal Form (1NF), Second Normal Form (2NF), dan Third Normal Form (3NF)^[4].

- **Tahap 1NF** bertujuan untuk menghilangkan data yang berulang dan memastikan setiap atribut berisi nilai tunggal (atomic value).
- **Tahap 2NF** bertujuan agar setiap atribut bergantung penuh pada primary key, dan tidak hanya pada sebagian key (partial dependency). Untuk mencapainya, tabel dipecah menjadi beberapa tabel baru berdasarkan fungsinya.
- **Tahap 3NF** bertujuan menghilangkan dependensi transitif, yaitu kondisi ketika atribut non-key bergantung pada atribut non-key lain. Setiap tabel hanya berisi data yang langsung berhubungan dengan primary key. Hasilnya terbentuk beberapa tabel baru.

2.4 SQL dan Koneksi Database

Structured Query Language (SQL) merupakan bahasa standar yang digunakan untuk mengelola basis data relasional. Dalam SQL terdapat beberapa kelompok perintah utama, yaitu Data Definition Language (DDL) untuk mendefinisikan struktur basis data, Data Manipulation Language (DML) untuk memanipulasi data, dan Transaction Control Language (TCL) untuk mengatur transaksi basis data^[4].

Koneksi basis data merupakan proses yang memungkinkan aplikasi atau pengguna untuk tersambung ke server database sehingga dapat menjalankan perintah SQL. Koneksi yang berhasil dapat menjalankan query^[5].

BAB III

PERANCANGAN DAN IMPLEMENTASI SISTEM BASIS DATA

3.1 Gambaran Umum Sistem

Sistem basis data yang dikembangkan pada Final Project ini merupakan pengembangan lanjutan dari studi kasus pada Project UTS, yaitu sistem penjualan pada toko Planet Kemasan. Sistem ini digunakan untuk mencatat seluruh aktivitas transaksi jual-beli yang terjadi di toko, mulai dari data pelanggan, operator (kasir), produk yang dijual, hingga detail transaksi dan pembayaran.

Setiap transaksi dicatat oleh seorang operator yang melayani customer. Dalam satu transaksi, customer dapat membeli lebih dari satu produk dengan jumlah yang berbeda. Seluruh data transaksi tersebut disimpan ke dalam basis data agar memudahkan proses pencatatan, pencarian data, serta pembuatan laporan penjualan. Sistem ini dirancang untuk meminimalkan redundansi data dan menjaga konsistensi data melalui penerapan relasi antar tabel.

3.2 Perancangan Basis Data

Perancangan basis data pada proyek ini mengacu pada hasil analisis dan normalisasi data yang telah dilakukan pada Project UTS. Proses perancangan dimulai dari tabel tidak ternormalisasi (unnormalized table) yang merepresentasikan nota transaksi penjualan, kemudian dilakukan normalisasi hingga mencapai bentuk normal ketiga (Third Normal Form / 3NF).

Melalui proses normalisasi, data dipisahkan ke dalam beberapa tabel berdasarkan fungsinya untuk menghindari redundansi dan anomali data. Tabel-tabel yang dihasilkan meliputi tabel customer, operator, produk, transaksi, dan detail_transaksi. Setiap tabel memiliki primary key sebagai identitas unik, serta foreign key yang digunakan untuk membentuk relasi antar tabel sesuai dengan kebutuhan sistem penjualan.

3.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) pada sistem ini disusun berdasarkan hasil normalisasi dan relasi antar tabel yang telah ditentukan pada Project UTS. ERD menggambarkan hubungan antar lima entitas utama, yaitu Customer, Operator, Transaksi, Produk, dan Detail_Transaksi.

Relasi antar entitas sebagian besar bersifat one-to-many (1:N), seperti relasi antara Customer dengan Transaksi, Operator dengan Transaksi, serta Transaksi dengan Detail_Transaksi. Hubungan antara Produk dan Transaksi bersifat many-to-many (M:N) yang direalisasikan melalui tabel Detail_Transaksi sebagai tabel penghubung. ERD ini digunakan sebagai acuan dalam pembuatan struktur tabel dan penerapan foreign key pada database.

3.4 Implementasi Struktur Tabel (DDL)

Implementasi struktur tabel dilakukan menggunakan perintah Data Definition Language (DDL). Pada tahap ini dilakukan pembuatan database, tabel, primary key, foreign key, serta constraint untuk menjaga integritas data. Constraint foreign key digunakan untuk memastikan bahwa data yang dimasukkan ke dalam tabel detail_transaksi dan transaksi harus memiliki data yang valid pada tabel induknya. Dengan adanya constraint ini, sistem dapat mencegah terjadinya data tidak konsisten.

3.5 Implementasi Data Manipulation Language (DML)

Setelah struktur tabel selesai dibuat, tahap selanjutnya adalah pengisian data awal menggunakan Data Manipulation Language (DML). Data awal dimasukkan ke dalam tabel customer, operator, dan produk sebagai data master, kemudian dilanjutkan dengan pengisian data transaksi dan detail_transaksi.

Penggunaan DML ini bertujuan agar database memiliki data yang cukup untuk menjalankan query lanjutan seperti JOIN, agregasi, dan subquery.

3.6 Implementasi Transaction Control Language (TCL)

Transaction Control Language (TCL) digunakan untuk mengatur proses transaksi database agar berjalan secara konsisten. Pada sistem ini, TCL diterapkan untuk memastikan bahwa proses insert transaksi dan detail_transaksi hanya akan disimpan jika seluruh proses berhasil.

Jika terjadi kesalahan pada salah satu proses, maka perubahan data akan dibatalkan menggunakan ROLLBACK. Sebaliknya, jika seluruh proses berjalan dengan baik, data akan disimpan secara permanen menggunakan COMMIT.

3.7 Implementasi Query SQL

Query SQL digunakan untuk menampilkan dan mengolah data sesuai kebutuhan sistem. Query yang digunakan pada proyek ini meliputi:

- Query JOIN untuk menampilkan data transaksi secara lengkap dari beberapa tabel
- Query GROUP BY dan HAVING untuk melakukan agregasi data
- Query DISTINCT untuk menampilkan data unik
- Query LIKE untuk pencarian data berdasarkan pola teks
- Query BETWEEN dan RANGE untuk penyaringan data berdasarkan rentang nilai
- Subquery dan VIEW untuk mempermudah pembuatan laporan

Query-query tersebut diuji dan hasil eksekusinya digunakan sebagai bukti implementasi sistem.

BAB IV

PENUTUP

4.1 Hasil Pengujian Sistem

Pengujian sistem dilakukan dengan menjalankan seluruh query SQL yang telah dibuat pada database. Hasil pengujian menunjukkan bahwa struktur tabel, relasi antar tabel, serta query yang digunakan dapat berjalan dengan baik dan menghasilkan output sesuai dengan tujuan sistem.

Query JOIN berhasil menampilkan data transaksi secara lengkap, sedangkan query agregasi dan GROUP BY mampu menampilkan ringkasan data penjualan. Penggunaan TCL juga berhasil menjaga konsistensi data pada saat proses transaksi.

4.2 Kendala dan Solusi

Selama proses perancangan dan implementasi sistem basis data, terdapat beberapa kendala yang dihadapi, seperti kesalahan relasi foreign key dan error saat menjalankan query JOIN. Kendala tersebut dapat diatasi dengan melakukan pengecekan struktur tabel, kesesuaian tipe data, serta urutan eksekusi query.

Dengan melakukan pengujian secara bertahap, kesalahan dapat diminimalkan dan sistem dapat berjalan dengan baik.

4.3 Kesimpulan

Berdasarkan hasil perancangan dan implementasi, dapat disimpulkan bahwa sistem basis data kasir penjualan telah berhasil dibuat sesuai dengan tujuan proyek. Sistem ini mampu mengelola data penjualan secara terstruktur, konsisten, dan mudah diolah menggunakan query SQL.

Penerapan konsep basis data relasional, normalisasi, serta penggunaan DDL, DML, dan TCL telah membantu dalam menjaga integritas dan konsistensi data.

4.4 Saran Pengembangan

Saran pengembangan yang dapat kami berikan adalah dengan menambahkan fitur laporan penjualan otomatis, sehingga informasi penjualan dapat diperoleh dengan lebih cepat dan efisien. Kemudian Sistem basis data Planet Kemasan dapat dikembangkan menjadi aplikasi berbasis web atau mobile agar proses pengelolaan data transaksi dapat dilakukan secara lebih mudah dan terintegrasi tanpa harus mengetik manual perintah SQL.

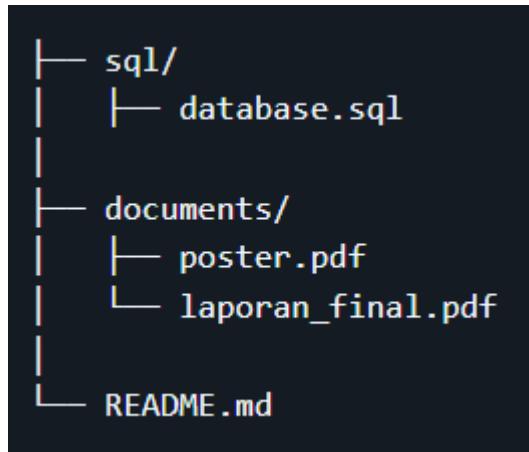
LAMPIRAN

Tautan Repository

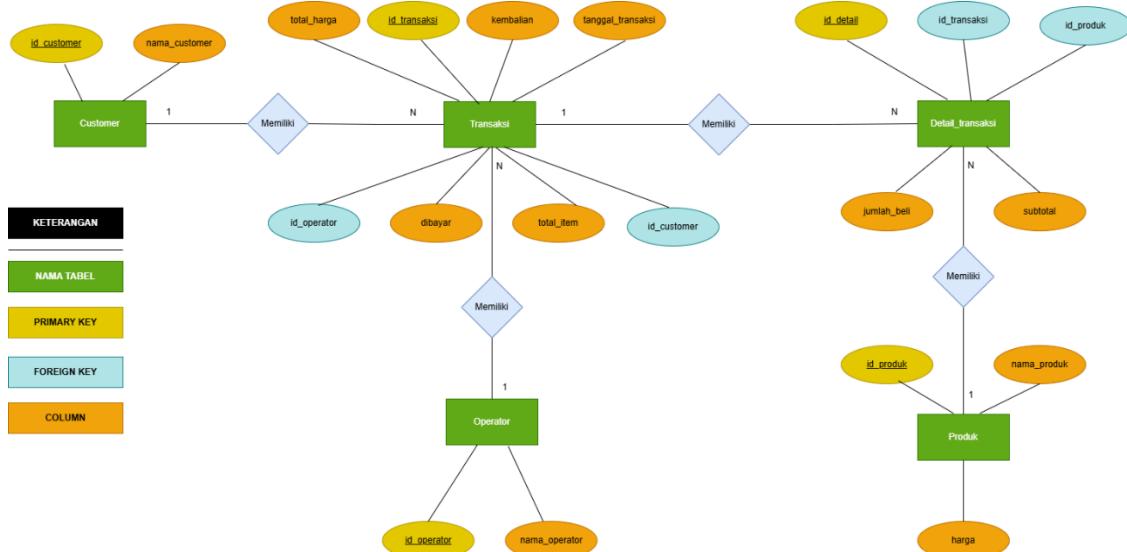
<https://github.com/HafidzFathkurr/final-project-uas-sistem-basis-data.git>

Script SQL lengkap dapat diakses melalui link diatas dan melalui direktori sql/database.sql

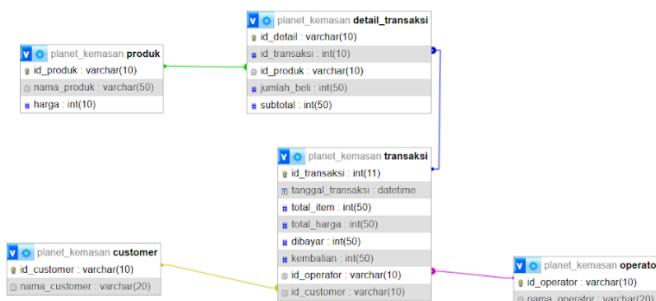
Ringkasan Struktur Folder



ERD



Skema Tabel



Query Agregasi dan Having

```
169      -- HAVING  
170 •   SELECT  
171          id_transaksi,  
172          SUM(subtotal) AS total_belanja  
173      FROM detail_transaksi  
174      GROUP BY id_transaksi  
175      HAVING SUM(subtotal) > 50000;  
176
```

Result Grid | Filter Rows: | Export

	id_transaksi	total_belanja
▶	184630	162000

Query Group By

```
161      -- AGREGASI, GROUP BY  
162 •   SELECT  
163          id_transaksi,  
164          SUM(subtotal) AS total_belanja  
165      FROM detail_transaksi  
166      GROUP BY id_transaksi;  
167
```

Result Grid | Filter Rows: | Export

	id_transaksi	total_belanja
▶	184630	162000
	184631	45000
	184632	9000

Query Inner Join

```
128    -- INNER JOIN
129    SELECT
130        t.id_transaksi,
131        c.nama_customer,
132        o.nama_operator,
133        p.nama_produk,
134        d.jumlah_beli,
135        d.subtotal
136    FROM transaksi t
137    INNER JOIN customer c ON t.id_customer = c.id_customer
138    INNER JOIN operator o ON t.id_operator = o.id_operator
139    INNER JOIN detail_transaksi d ON t.id_transaksi = d.id_transaksi
140    INNER JOIN produk p ON d.id_produk = p.id_produk;
```

Result Grid						
	id_transaksi	nama_customer	nama_operator	nama_produk	jumlah_beli	subtotal
▶	184630	Andi	Fitri	mika gethuk	5	103500
	184630	Andi	Fitri	Mika nampan M	20	54000
	184630	Andi	Fitri	jas hujan ponco armada	1	4500
	184631	Lia	Ahmad	jas hujan ponco armada	10	45000
	184632	Budi	Fitri	jas hujan ponco armada	2	9000

Query Left Join

```
143    -- LEFT JOIN
144    SELECT
145        c.nama_customer,
146        t.id_transaksi,
147        t.tanggal_transaksi
148    FROM customer c
149    LEFT JOIN transaksi t ON c.id_customer = t.id_customer;
```

Result Grid			
	nama_customer	id_transaksi	tanggal_transaksi
▶	Andi	184630	2025-10-17 15:28:00
	Lia	184631	2025-10-29 10:37:00
	Budi	184632	2026-01-17 10:21:36

Query Right Join

```
152      -- RIGHT JOIN
153      SELECT
154          p.nama_produk,
155          d.id_transaksi,
156          d.jumlah_beli
157      FROM detail_transaksi d
158      RIGHT JOIN produk p ON d.id_produk = p.id_produk;
```

Result Grid			
	nama_produk	id_transaksi	jumlah_beli
▶	mika gethuk	184630	5
	Mika nampan M	184630	20
	jas hujan ponco armada	184630	1
	jas hujan ponco armada	184631	10
	jas hujan ponco armada	184632	2
	buku tulis	NULL	NULL

Query TCL berhasil

```
-- TCL BERHASIL
START TRANSACTION;

INSERT INTO transaksi
(id_transaksi, tanggal_transaksi, total_item, total_harga, dibayar, kembalian, id_operator, id_customer)
VALUES
(184632, NOW(), 2, 9000, 10000, 1000, 'PR01', 'CS0002');

INSERT INTO detail_transaksi
(id_detail, id_transaksi, id_produk, jumlah_beli, subtotal)
VALUES
('D0015', 184632, 'P003', 2, 9000);

COMMIT;
```

Query TCL gagal

```
-- TCL GAGAL
START TRANSACTION;

INSERT INTO transaksi
(id_transaksi, tanggal_transaksi, total_item, total_harga, dibayar, kembalian, id_operator, id_customer)
VALUES
(184700, NOW(), 1, 5000, 10000, 5000, 'PR01', 'CS0001');

INSERT INTO detail_transaksi
(id_detail, id_transaksi, id_produk, jumlah_beli, subtotal)
VALUES
('D0999', 184700, 'P999', 1, 5000);

ROLLBACK;
```

DAFTAR PUSTAKA

- [1] Basis Data. Univ Terbuka MSIM4206 Modul 01 [Internet]. Available from:
<https://univterbuka.github.io/MSIM4206/modul-01/kb-01.html>
- [2] Apa Itu DBMS: Pengertian, Jenis, dan Contoh DBMS. BIT Telkom University [Internet]. 2023 [cited 2026 Jan 16]. Available from:
<https://bit.telkomuniversity.ac.id/apa-itu-dbms-pengertian-jenis-dan-contoh-dbms/>
- [3] Materi ERD (Entity Relationship Diagram). Media Pembelajaran Basis Data [Internet]. Available from:
<https://sites.google.com/view/mediapembelajaranbasisdata/materi/materi-5-erd-entity-relationship-diagram>
- [4] Irawan RD. Jurnal Kuliah Pemrograman Basis Data. Surakarta: Universitas Duta Bangsa; 2025.
- [5] How to Connect to MySQL Database Clusters. DigitalOcean Documentation [Internet]. cited 2026 Jan 17. Available from:
<https://docs.digitalocean.com/products/databases/mysql/how-to/connect/>