

Integrasi Data pada Aplikasi Desktop

Agnes Lesmono, Khakim Ghozali, dan Annisaa Sri Indrawanti
Departemen Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: zidan.z2004@gmail.com

Abstrak—Penggunaan basis data sudah semakin luas dan banyak. Banyaknya data tersebut memerlukan suatu jembatan yang dapat menjadi penghubung antar basis data. Basis data yang banyak digunakan tidak hanya berasal dari satu platform, sehingga dibutuhkan integrasi data yang dapat memudahkan basis data dapat saling bertukar informasi. Penelitian ini membuat desain *middleware* integrasi data yang mampu mengintegrasikan data yang berada pada basis data yang heterogen, yaitu MySQL, PostgreSQL, dan SQLite. Pembuatan integrasi ini dibangun pada aplikasi *middleware* yang menjadi penghubung antara pengguna dari sisi client dengan basis data. Dari penelitian yang telah dilakukan didapatkan bahwa dengan bantuan socket yang menghubungkan antara client dan *middleware*, client dapat terhubung dengan ketiga basis data melalui *middleware* dan dapat menjadi backup satu sama lain.

Kata Kunci—Basis Data, Integrasi Data, Middleware.

I. PENDAHULUAN

PENINGKATAN penggunaan basis data turut meningkatkan kebutuhan antar basis data untuk saling bertukar informasi. Pada dasarnya basis data merupakan sekumpulan data yang didesain dan diintegrasikan satu sama lain untuk memenuhi kebutuhan pengguna. Kegunaan basis data dalam menyelesaikan permasalahan pada penyusunan data diantaranya sulitnya mengakses data dan masalah integrasi [1]. Integrasi data pada aplikasi dan proses bisnis dilakukan untuk memenuhi kebutuhan informasi dari sumber data yang berbeda. Penerapan integrasi dapat menambah kemajuan dalam menghadapi persaingan [2]. Integrasi data juga dapat didefinisikan sebagai keterkaitan data antara beberapa sistem yang saling terhubung ke suatu penyimpanan data seperti *data warehouse*. Dengan adanya integrasi data tersebut, dapat memudahkan hal berbagi data dan analisis data untuk mendukung pengelolaan informasi. Dalam penerapan proses integrasi data terdapat beberapa permasalahan untuk diselesaikan. Untuk menyelesaikan masalah tersebut dapat dikelompokkan dalam beberapa cara, namun salah satunya adalah integrasi basis data. Integrasi basis data dilakukan berdasarkan skema basis data yang bertujuan untuk membangun pemetaan serta pencocokan antar skema basis data

Basis data MySQL merupakan basis data *server* gratis dengan lisensi GNU *General Public License* (GPL) yang dapat digunakan untuk keperluan pribadi maupun komersial tanpa harus membayar lisensi. Pada MySQL, sebuah basis data dapat terdiri dari satu atau beberapa Tabel sehingga disebut RDBMS (*Relational Database Management System*). PostgreSQL adalah RDBMS yang bersifat *open source*. PostgreSQL memiliki performa yang stabil, keamanan tinggi, dan fitur yang melimpah. PostgreSQL banyak digunakan pada aplikasi web, aplikasi *mobile* dan aplikasi analitik, sangat cocok untuk aplikasi yang membutuhkan pengolahan data yang lebih kompleks. SQLite merupakan basis data

tanpa *server* dan mandiri, atau sering disebut sebagai basis data tertanam dimana basis data berjalan sebagai bagian dari aplikasi. Basis data SQLite memiliki fitur lengkap dengan banyak Tabel, *index*s, *trigger*, dan tampilan yang tersimpan pada satu *file* tunggal pada hard-disk.

Middleware merupakan perangkat lunak komputer yang menghubungkan bagian-bagian berbeda dari aplikasi dengan sistem operasi. Penggunaan *middleware* banyak digunakan pada pengembangan perangkat lunak untuk komunikasi.

Penelitian oleh Hui-Ru Zhang, dll (2017) membangun *database access middleware* dengan model berbasis teknologi XML dan CORBA bertujuan untuk mengintegrasikan informasi dari berbagai sumber data dan mendukung akses pada lapisan aplikasi [3]. Penelitian lain pada integrasi basis data heterogen adalah dengan memanfaatkan basis data MySQL ODBC dan database link untuk menghubungkan Oracle XE 10g dengan MySQL [4]. Adapula yang mengintegrasikan data kesehatan epidemi yang berasal dari pelaporan puskesmas dan rumah sakit yang menggunakan DBMS yang berbeda, yaitu MySQL dan PostgreSQL [5]. Selain itu, ada pula penelitian integrasi basis data menggunakan data kependudukan nasional sebagai data primer [6].

Pada penelitian ini dilakukan integrasi data yang berada pada tiga basis data berbeda yaitu MySQL, PostgreSQL dan SQLite dimana ketiga basis data dapat saling terhubung melalui *middleware* untuk menemukan solusi integrasi multi platform *database*. Basis data yang dipilih merupakan basis data yang populer penggunaannya berdasarkan *ranking* oleh DB-Engines. Pada Januari dan Februari 2022, basis data MySQL berada pada peringkat dua, basis data PostgreSQL berada pada peringkat empat dan basis data SQLite berada pada peringkat sepuluh.

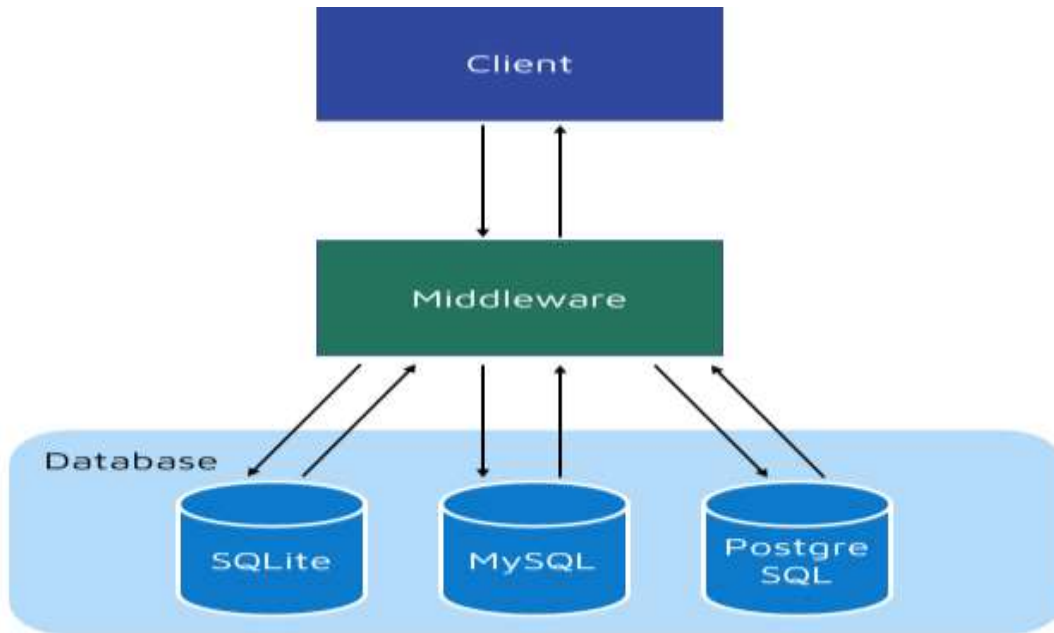
II. METODE PENYELESAIAN

A. Deskripsi Permasalahan

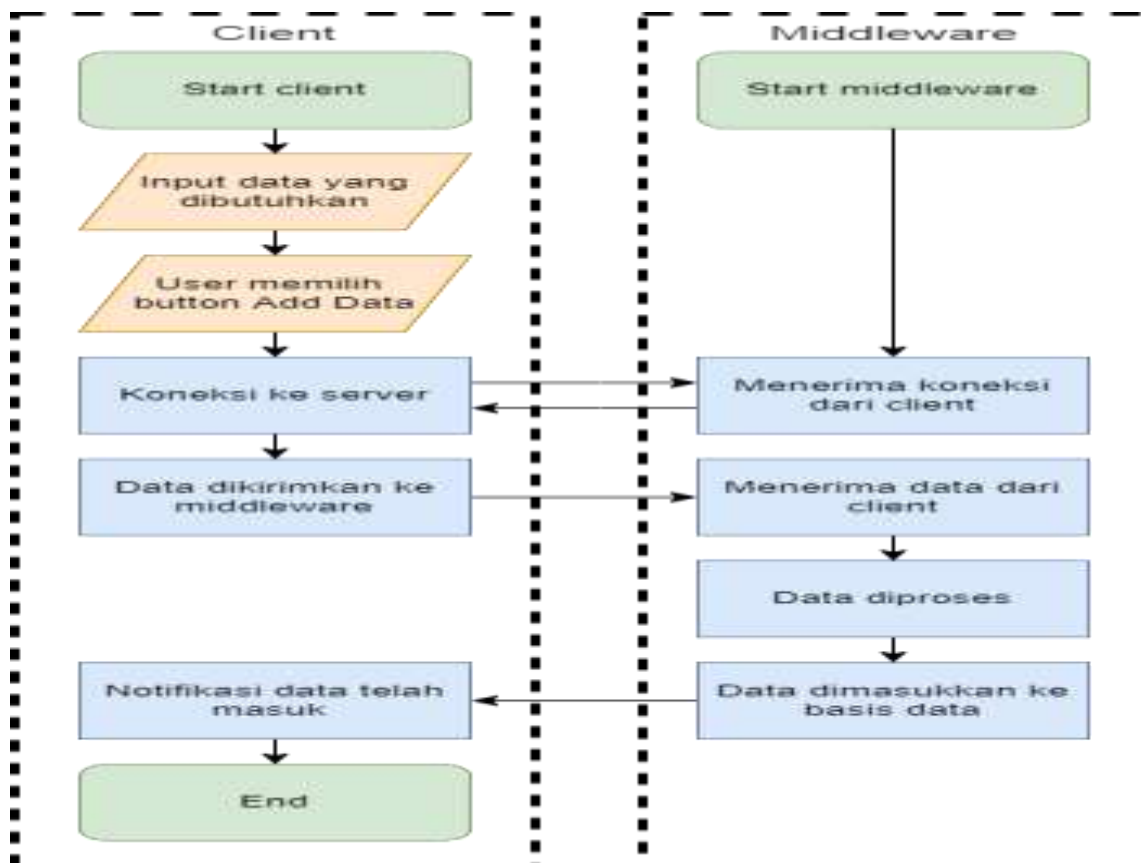
Pada permasalahan ini, terdapat tiga basis data yang berada pada MySQL, PostgreSQL, dan SQLite. Untuk dapat melakukan integrasi data pada basis data yang heterogen, diperlukan suatu program yang dapat terhubung dengan ketiga basis data tersebut. Heterogenitas basis data menjadi hambatan dalam komunikasi antar basis data.

B. Strategi Penyelesaian

Dalam upaya untuk dapat melakukan integrasi data, dilakukan pembangunan *middleware* yang dapat membantu koneksi dari *client* ke basis data. Dalam pembangunan aplikasi digunakan bahasa pemrograman yang dapat mendukung penggunaan basis data yang digunakan, serta membawa pengaruh pada pembangunan aplikasi. Oleh karena itu, digunakan bahasa pemrograman python dengan modul yang mendukung : tkinter untuk graphical user interface (GUI), mysql-connector untuk menghubungkan



Gambar 1. Arsitektur aplikasi.



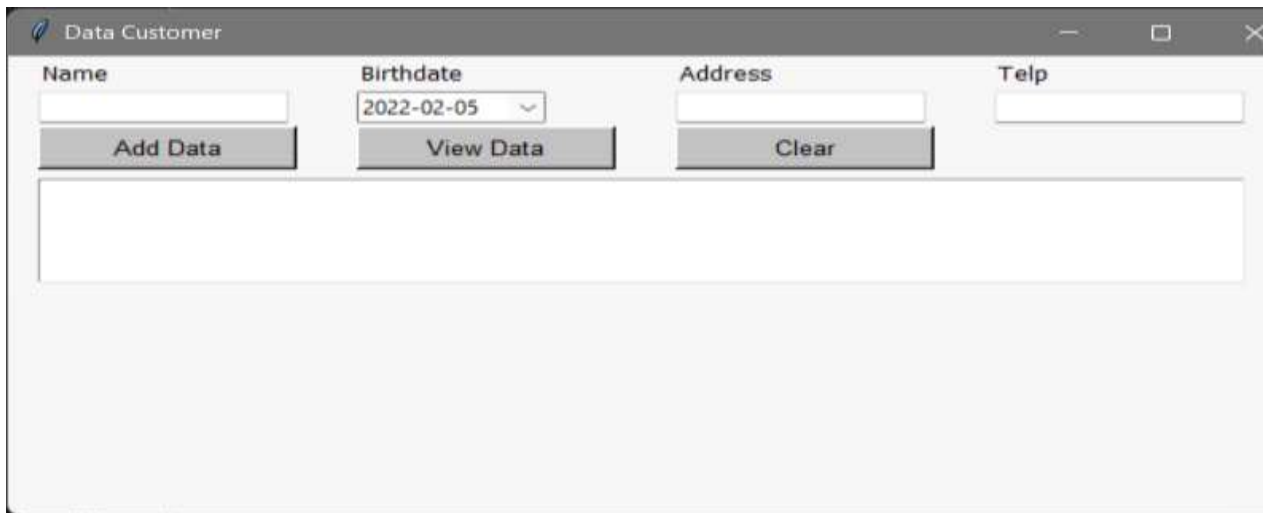
Gambar 2. Alur menambah data customer.

program dengan basis data MySQL, sqlite3 untuk mendukung basis data SQLite, psycopg2 untuk mendukung basis data PostgreSQL, sys, os, dan datetime. Untuk dapat menghubungkan antara *middleware* dan *client*, digunakan *socket* sebagai alat bantu untuk komunikasi. Strategi untuk menyelesaikan masalah ini dilakukan dengan membuat desain perancangan sistem dan pembuatan aplikasi untuk integrasi basis data.

C. Perancangan sistem

Sistem dirancang seperti pada Gambar 1, dimana *middleware* dibangun untuk dapat terhubung ke tiga basis

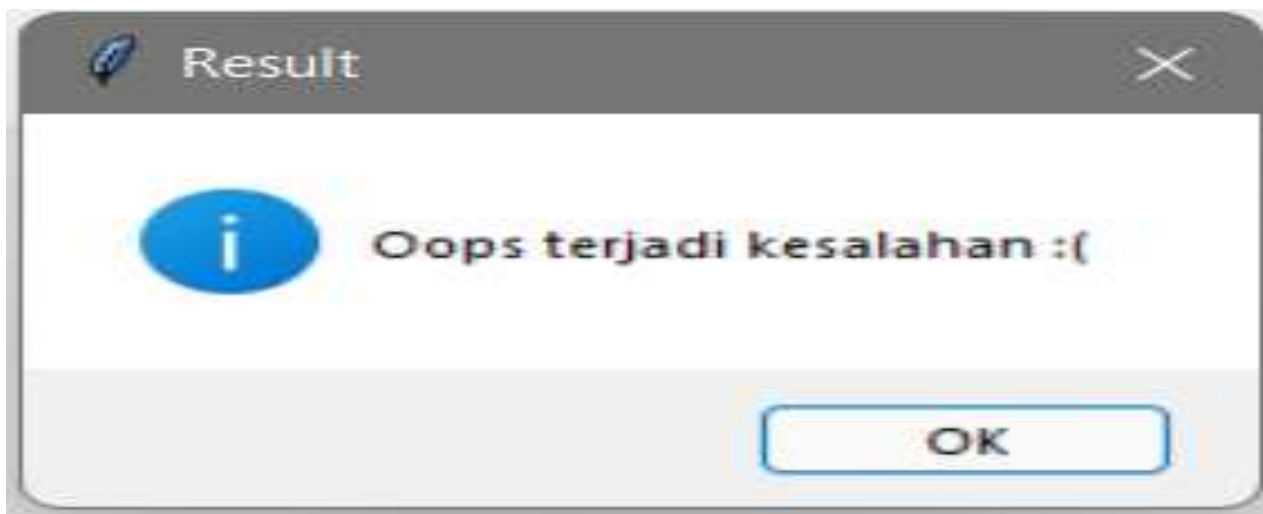
data : SQLite, MySQL, dan PostgreSQL. Pada Gambar 1, *client* mengirimkan request ke *middleware* dan akan diteruskan ke basis data yang sesuai. Setelah respon dari basis data diterima, *middleware* akan meneruskan respon tersebut pada *client*. Karena tujuan dari koneksi ke basis data heterogen ini untuk *back up*, ketika ada pengguna yang memasukkan data melalui *client*, *middleware* akan secara otomatis memasukkan data tersebut pada basis data SQLite, MySQL dan PostgreSQL sehingga ketika ada salah satu *server* basis data *down*, data yang diinputkan oleh pengguna masih ada pada basis data lain tanpa terjadi hilangnya data.



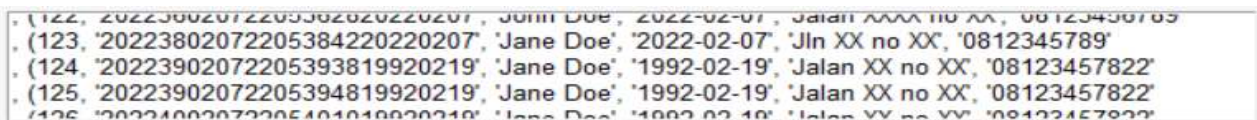
Gambar 3. Tampilan aplikasi dari sisi client.



Gambar 4. Notifikasi ketika data berhasil dimasukkan.



Gambar 5. Notifikasi ketika terjadi kesalahan.



Gambar 6. Tampilan data pada interface pengguna.

Basis data SQLite yang merupakan basis data berbasis *file*, selalu bersifat lokal dan tidak memerlukan *username* maupun *password* untuk dapat mengakses basis data ini. Basis data MySQL dan PostgreSQL dapat bersifat lokal maupun *remote*. Jika menggunakan basis data secara *remote*, *middleware* akan menggunakan *host* alamat IP dan port untuk terhubung ke basis data. Secara *default*, port untuk basis data MySQL adalah 3306 dan port untuk PostgreSQL adalah 5432. Selain dengan *host* dan port, *username*, *password*, dan nama

database yang digunakan turut perlu dimasukkan. *Middleware* menggunakan modul python `mysql.connector` untuk terkoneksi pada basis data MySQL, modul `sqlite3` untuk terhubung pada basis data SQLite dan modul `psycopg2` untuk dapat terhubung pada basis data PostgreSQL.

Aplikasi *middleware* ini memanfaatkan *socket* untuk terhubung dengan *client*. *Middleware* memiliki fungsionalitas: menambah data *customer* dan menampilkan data tersebut. *Middleware* yang digunakan menggunakan port

Tabel 1.
Uji coba performa penambahan data

Uji coba ke - n	Waktu yang digunakan (s)		
	MySQL	PostgreSQL	SQLite
1	0.38876461982727	0.137176990509033	0.007988929748535
2	1.2482011318206787	0.1601274013519287	0.0019838809967041016
3	3.7500922679901123	0.141526460647583	0.009095430374145508
4	1.5067179203033447	0.13791704177856445	0.002285003662109375
5	1.3976123332977295	0.14152026176452637	0.009923696517944336
6	0.8515844345092773	0.14150738716125488	0.003001689910888672
7	0.4660379886627197	0.14558982849121094	0.00316619873046875
8	0.5832157135009766	0.13756346702575684	0.009000301361083984
9	0.5200269222259521	0.15505719184875488	0.0021986961364746094
10	1.1422419548034668	0.1392354965209961	0.008923053741455078
Rerata	1.185449529	0.143722153	0.005756688
Total waktu rerata			1.33492837

Tabel 2.
Uji coba performa menampilkan data

Uji coba ke - n (s)	Waktu yang digunakan (s)		
	MySQL	PostgreSQL	SQLite
1	0.41446781158447266	0.13988614082336426	0.0010001659393310547
2	0.5087981224060059	0.14544177055358887	0.0029997825622558594
3	0.4953579902648926	0.14156031608581543	0.0030417442321777344
4	0.7970757484436035	0.1468982696533203	0.0026073455810546875
5	1.7984793186187744	0.14289379119873047	0.0020020008087158203
6	0.20461583137512207	0.14206886291503906	0.001958608627319336
7	0.7836503982543945	0.14256548881530762	0.0017189979553222656
8	0.19839906692504883	0.14256024360656738	0.0012710094451904297
9	0.20334553718566895	0.1462404727935791	0.0020847320556640625
10	0.5133180618286133	0.1573488712310791	0.0020003318786621094
Rerata	0.591750789	0.144746423	0.002068472

12345 untuk pertukaran informasi antara *middleware* dengan *client*. Uji coba yang dilakukan pada fungsi menambah data *customer* dilakukan sesuai dengan Gambar 2. Diawali dari sisi *client*, pengguna memasukkan data yang dibutuhkan, yaitu nama *customer*, tanggal lahir, alamat, dan nomor telepon. Kemudian pengguna menekan atau memilih *button* Add Data. *Button* Add Data akan *trigger* koneksi ke *middleware*. *Middleware* akan menerima koneksi dari *client*. Data akan diolah dengan membuat kode unik berdasarkan waktu dan tanggal ulang tahun yang dimasukkan oleh pengguna. Program akan *generate* sebuah *string* yang berisi kode tombol, kode unik, dan semua data lain yang dimasukkan oleh pengguna. Kemudian sistem akan mengirimkan data yang dimasukkan pengguna ke *middleware*. Setelah data diterima, data diproses dan dimasukkan ke basis data. Setelah data berhasil dimasukkan notifikasi akan dikirimkan ke pengguna.

Client mengirimkan sebuah *string* yang berisi kode *button* dan data yang akan dimasukkan pada basis data yang dipisahkan dengan tanda titik koma `;`. *Middleware* sebagai *server* pada *socket* akan menerima *string* tersebut dan mengolahnya. *String* akan di-*split* untuk setiap tanda titik koma. Kemudian data akan dimasukkan ke basis data. Setelah data dimasukkan ke basis data, *middleware* akan mengirimkan respon ke *client* untuk menampilkan notifikasi ke pengguna bahwa data telah berhasil dimasukkan.

Middleware akan mati secara otomatis ketika terjadi *error*. Oleh karena itu, dibuatlah sebuah sistem untuk melakukan *looping* untuk memastikan *middleware* tidak mati setelah terjadi *error*.

III. UJI COBA DAN ANALISIS

Uji coba dilakukan pada lingkungan tertentu dengan skenario yang telah ditentukan untuk memastikan

berjalannya aplikasi dan dilanjutkan dengan tahap analisis dari metode yang digunakan untuk menyelesaikan masalah. Lingkungan uji coba dilakukan pada laptop dengan spesifikasi :

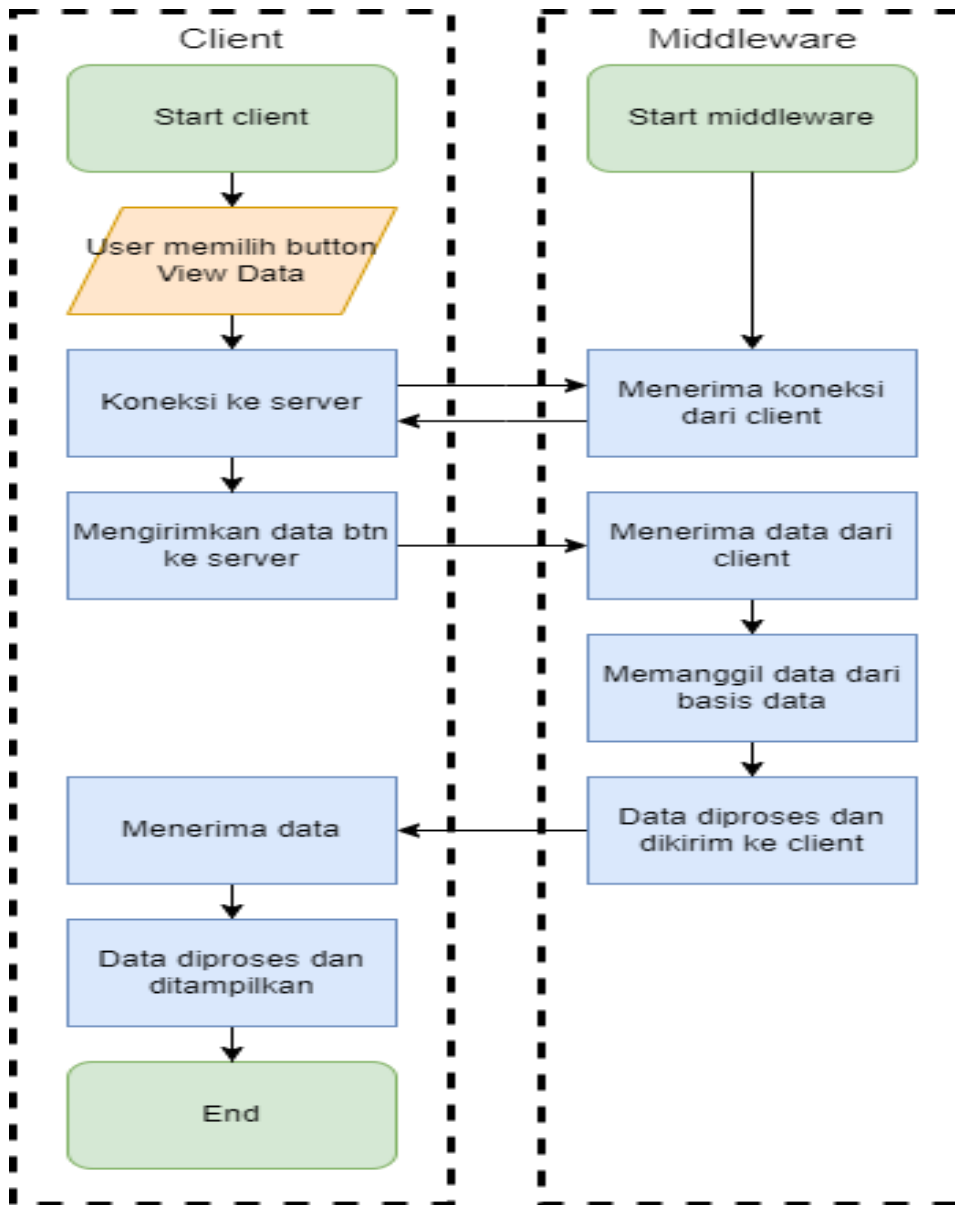
1. Prosesor Intel(R) Core(TM) i7-8550U CPU @1.80GHz 1.99 GHz
2. *Random Access Memory* (RAM) 16.0 GB
3. Sistem Operasi : Windows 11

A. Uji Coba Menambah Data Customer

Pada uji coba menambah data *customer*, pertama dipastikan bahwa *middleware* yang digunakan dalam kondisi *running*. Tampilan awal dari sisi *client* dapat dilihat pada Gambar 3. Kemudian dijalankan program sisi *client*. Pengguna mengisi data yang diperlukan pada *entry*. Dilanjutkan dengan pengguna menekan tombol Add Data. Ketika data berhasil dimasukkan ke tiga basis data, akan ditampilkan notifikasi bahwa data telah berhasil dimasukkan. Jika terjadi kesalahan atau *error* pada koneksi basis data, akan tampil notifikasi yang menginfokan bahwa terjadi kesalahan. Notifikasi akan tampak seperti Gambar 4 jika berjalan lancar dan seperti Gambar 5 ketika terjadi kesalahan. Kesalahan yang terjadi dapat dilihat pada *middleware*. Dengan adanya program untuk *looping*, *middleware* akan diulang kembali ketika terjadi *error*.

B. Uji Coba Menampilkan Data Customer

Setelah dipastikan bahwa *middleware* dalam kondisi *running*, pengguna dari sisi *client* dapat menekan tombol View Data. Tombol tersebut akan *trigger* pengiriman *request* data ke *middleware*. *Middleware* akan memanggil data dari basis data dan memproses data tersebut. Data yang didapat diproses dengan menggunakan modul *pickle* untuk mengirimkan list data pada *client*. Setelah data selesai diproses, data dikirimkan ke sisi *client* dan diterima dalam bentuk list. Data yang baru diterima diolah dengan *pickle*



Gambar 7. Flowchart uji coba menampilkan data customer.

sehingga dapat dibuka. Data list tersebut diubah menjadi *string* dan kemudian ditampilkan seperti pada Gambar 6. Data yang diterima oleh *client* akan ditampilkan pada *box* yang disediakan. Pada uji coba menampilkan data *customer*, dilakukan dengan *flowchart* seperti pada Gambar 7.

C. Uji Coba Performa

Pada uji coba performa dilakukan uji coba untuk mengetahui waktu yang digunakan pada setiap uji coba fungsionalitas. Untuk percobaan, dilakukan pengukuran waktu untuk fungsi menambah data dan menampilkan data. Uji coba dilakukan sebanyak sepuluh kali untuk setiap skenario dengan mencari rata-rata waktu yang digunakan untuk setiap uji coba. Hasil uji coba penambahan data dapat dilihat pada Tabel 1 dan hasil uji coba menampilkan data dapat dilihat pada Tabel 2.

IV. KESIMPULAN

Berdasarkan penjabaran di subbab-subbab sebelumnya dari hasil uji coba yang dilakukan, dapat disimpulkan beberapa poin terkait integrasi basis data berbeda platform.

Integrasi data pada aplikasi dapat dilakukan dengan bahasa pemrograman python dengan modul *mysql.connector*, *sqlite3*, dan *psycopg2* untuk menghubungkan dengan basis data MySQL, SQLite, dan PostgreSQL serta modul *socket* untuk membantu komunikasi antara *client* dan *middleware* sebagai *server*.

Penyimpanan data pada tiga basis data dapat dijadikan *backup* ketika data dimasukkan. Ketika terjadi masalah pada salah satu basis data, *client* masih dapat memasukkan data pada basis data lain yang terhubung sehingga data tersebut tidak hilang atau terlewat.

Untuk pengembangan aplikasi kedepannya, dapat dilakukan optimasi dengan menggunakan struktur data dan/atau metode lain yang lebih efisien, serta melakukan optimasi program untuk mempercepat pertukaran data.

DAFTAR PUSTAKA

[1] T. Setiady and M. B. Rahmad, "Perancangan sistem informasi inventory spare part elektronik berbasis web PHP," *J. Sarj. Tek. Inform.*, vol. 2, no. 2, p. 10, 2014.

[2] A. Wuryanto, "Integrasi sistem informasi dan teknologi informasi melalui metode enterprise architecture planning pada PT. kayu permata bekaasi," *Urnalilmu Pengetah. Dan Teknol. Komput.*, vol. 3, no. 2, pp.

- 1–8, 2018.
- [3] H. Zhang, X. Ji, and Y. Yang, “Research of Hybrid Database Middleware Architecture,” in *International Conference on Communication and Electronic Information Engineering (CEIE 2016)*, 2017, doi: 10.2991/ceie-16.2017.37.
- [4] M. D. R. Wahyudi and A. Aa. Qalban, “Rancang bangun heterogenous distributed database system untuk meningkatkan kapasitas oracle xe 10g pada studi kasus sistem informasi akademik,” *Kaunia*, vol. IX, no. 2, pp. 60–72, 2013.
- [5] F. Budiman, S. S. N, and Muslih, “Desain Integrasi Data Antar Database Epidemiologi Untuk Mendukung Pusat Data Kesehatan Dengan Menggunakan Soa Webservice,” in *Prosiding SNATIF*, Semarang, pp. 95–100, 2015.
- [6] E. Sutanta, “Model integrasi database penduduk indonesia dengan berbagai sistem informasi berbasis komputer,” *J. Inform*, vol. 5, no. 2, pp. 542–553, 2011.