

PERAN DATA STORE DALAM MEMPRESENTASIKAN HUBUNGAN DATA FLOW DIAGRAM SSADM DENGAN ENTITY RELATIONSHIP DIAGRAM

Rizki Ridwan^a, Nunu Kustian^b, Erlin Windia Ambarsari^c

^a Teknik Informatika, rizki8992@gmail.com, Universitas Indraprasta PGRI

^b Teknik Informatika, kustiannunu@gmail.com, Universitas Indraprasta PGRI

^c Teknik Informatika, erlin_windia@barengbikinpaper.com, Universitas Indraprasta PGRI

Abstract

The information system development step often used is the System Development Life Cycle (SDLC). One of the SDLC stages is system design. It uses the Data Flow Diagram version of the Structured System Analysis and Design Methodology (SSADM) because it is a traditional model closely related to the waterfall model. This study aims to prove the role of the data store in the DFD and become an entity in the Entity-Relationship Diagram (ERD) to see the relationship between each entity. Benefit the associative method tests the connection of the data store on the DFD, which is realized to the ERD. The results obtained are the role of the data store in the DFD to become an entity on the ERD with a proven query table to facilitate data retrieval.

Keywords: *Entity Relationship Diagram, Data Flow Diagram, SDLC, SSADM*

Abstrak

Tahapan pengembangan sistem informasi yang sering digunakan adalah *System Development Life Cycle (SDLC)*. Salah satu tahapan SDLC yaitu perancangan sistem. Perancangan sistem yang dibuat pada penelitian ini ditransformasikan ke dalam *Data Flow Diagram* versi *Structured System Analysis and Design Methodology (SSADM)* karena merupakan model tradisional yang erat kaitannya pada metode *waterfall*. Tujuan penelitian ini untuk membuktikan peran data *store* pada DFD dan menjadi entitas didalam *Entity Relationship Diagram (ERD)* untuk melihat hubungan antara masing-masing entitas. Metode yang digunakan adalah metode asosiatif dengan tujuan untuk meneliti kaitan data *store* pada DFD yang direalisasikan ke ERD. Hasil yang didapatkan bahwa terdapat peran data *store* pada DFD menjadi entitas pada ERD dengan dibuktikan tabel *query* untuk memudahkan pencarian data.

Kata Kunci: *Entity Relationship Diagram, Data Flow Diagram, SDLC, SSADM*

1. PENDAHULUAN

Seorang analis sistem membangun perangkat lunak dengan menggunakan beberapa tahapan agar sesuai dengan tujuan dan kebutuhan dari *end user*. Banyak sekali tahapan yang diikuti dalam merancang suatu perangkat lunak, contohnya SDLC (*System Development Life Cycle*) [1], RAD (*Rapid Application Development*) [2], dan *prototype* [3]. SDLC merupakan tahapan pengembangan sistem yang paling klasik yang dimana berbasis pada kinerja *waterfall* [4]

Tahapan *waterfall* diantaranya *Communication, Planning (Estimating, Scheduling, Tracking), Modelling (Analysis and Design), Construction (Code and Test), Deployment (Delivery, Support, Feedback)* [5] [6]. *Modelling* mempunyai dampak yang begitu besar dalam mengembangkan perangkat lunak sebagai kerangka sistem yang beroperasi di dalamnya. Beberapa *modelling* yang sering digunakan pada penelitian, diantaranya *Flowchart, DFD (Data Flow Diagram), dan UML (Unified Modelling Language)*.

DFD pertama kali diperkenalkan pada tahun 1970 oleh Larry Constantine dan Ed Yourdon (ref) yang digunakan untuk pemrograman terstruktur seperti Turbo Pascal, Cobol, dan Quick Basic. Namun, pada tahun 90-an mulai dipopulerkan OOP (*Object Oriented Programming*). Kemudian, *modelling UML* mulai diperkenalkan yang berbasis OOP.

Berdasarkan penelitian-penelitian yang ditemukan, DFD masih digunakan seperti penelitian yang dilakukan oleh [7] membangun DFD dengan tiga entitas yaitu admin, siswa, dan pengajar. Penelitian dari [8]

Received Juni 4, 2022; Revised Juni 12, 2022; Accepted Juni 20, 2022

merancang DFD untuk sistem informasi sekolah pada SMP Plus Terpadu. Penelitian [9] membangun DFD untuk menggambarkan proses *inventory* barang antara entitas departemen, *purchasing*, dan HRD. Selanjutnya, penelitian oleh [10] merancang sistem informasi pemesanan kamar hotel. Penelitian-penelitian tersebut dapat disimpulkan bahwa DFD digunakan untuk merancang sistem informasi.

Sistem informasi berfungsi untuk mengontrol data didalamnya dan kemudian dilanjutkan dengan proses pengolahan untuk mendapatkan suatu informasi yang akan dianalisa dengan tujuan pemberian keputusan. Akan tetapi, pada penelitian-penelitian yang mengkaji DFD masih terdapat kekeliruan dalam mempresentasikan data sehingga hubungan antara DFD dengan ERD (*Entity Relationship Diagram*) menjadi tidak ada. Pemodelan ERD digunakan untuk menggambarkan data berstruktur yang bersifat tetap di dalam *database*. Dengan demikian, pencarian informasi menjadi lebih mudah.

Pengkajian penelitian terdahulu yang sudah disebutkan, perlu dilakukan perbaikan untuk menggambarkan bagaimana aliran data yang terjadi pada sistem informasi dapat dipresentasikan dengan baik di dalam DFD. Konsistensi data antara DFD dan ERD seharusnya dijelaskan pada penelitian terdahulu sehingga hubungan DFD dan ERD dapat lebih jelas. Oleh karena itu, penelitian ini bertujuan untuk mempresentasikan hubungan DFD dan ERD berdasarkan peran data *store* di dalamnya.

2. TINJAUAN PUSTAKA

2.1. Pemodelan Sistem

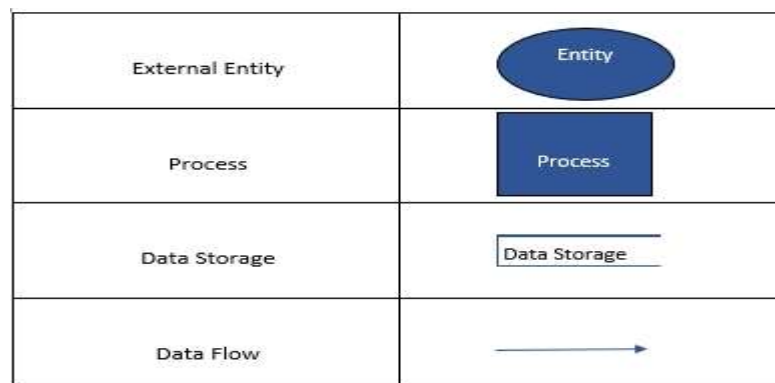
Pemodelan sistem merupakan aktivitas dalam pembuatan model dimana model tersebut sebagai perwakilan atau abstraksi dari sebuah objek atau situasi aktual suatu penyederhanaan dari suatu realitas yang kompleks [11].

2.2. Pemodelan Sistem dengan SDLC

Salah satu pemodelan sistem yang digunakan adalah model SDLC. Siklus hidup pengembangan sistem merupakan keseluruhan proses pengembangan, implementasi, dan mengontrol sistem informasi melalui proses multi langkah dari inisiasi, analisis, desain implementasi, dan pemeliharaan hingga sistem tersebut tidak lagi digunakan. Sesungguhnya SDLC mempunyai berbagai model dan metodologi, tetapi pada umumnya, langkah-langkah pengerjaannya masih sama [12].

2.3. Data Flow Diagram (DFD)

Data merupakan alat yang menggambarkan suatu sistem yang berinteraksi dengan lingkungannya dimana aliran data tersebut dapat masuk dan keluar pada sistem tersebut. DFD dapat digunakan untuk menggali apa yang dibutuhkan pengguna dan dikembangkan yang berfokus kepada struktur dan proses kerjanya [13]. DFD yang sering digunakan yaitu Gane Sarson [14] [15] dan Yourdon De Marco [16] [8]. Namun, terdapat model DFD lainnya seperti Yourdon and Coad [17] dan *Structured System Analysis and Design Methodology* (SSADM) [18] [19]. Beberapa poin penting di dalam DFD adanya suatu entitas, proses, arus data, dan data *store*. Berikut notasi pada DFD SSADM:



Gambar 1. Notasi SSADM

2.4. Entity Relationship Diagram (ERD)

Model *Entity Relationship Diagram* (ERD) adalah konsep umum model data yang digunakan untuk desain *database*. Selain itu, komponen utama dalam model ERD adalah himpunan entitas, himpunan relasi,

dan integritas masalah. Kumpulan entitas menunjukkan objek didunia nyata yang berbeda dengan objek lain [20].

3. METODOLOGI PENELITIAN

Penggambaran DFD dalam sistem informasi mempunyai pengaruh pada data *store* dan entitas dalam *database* [21] [22]. Pendekatan yang digunakan untuk menganalisis data *store* dengan metode asosiatif. Metode tersebut mendeskripsikan keterkaitan data *store* yang ada di dalam DFD diimplementasikan ke dalam bentuk ERD sehingga akan diketahui data *store* tersebut menjadi entitas ERD ke dalam *database*.

Kasus yang digunakan pada penelitian ini adalah kasus tentang *inventory* berdasarkan penelitian [23] melibatkan entitas supplier, devisi, dan gudang dimana terdapat aktivitas sistem mulai dari pemesanan barang, pengiriman barang sampai pelaporan.

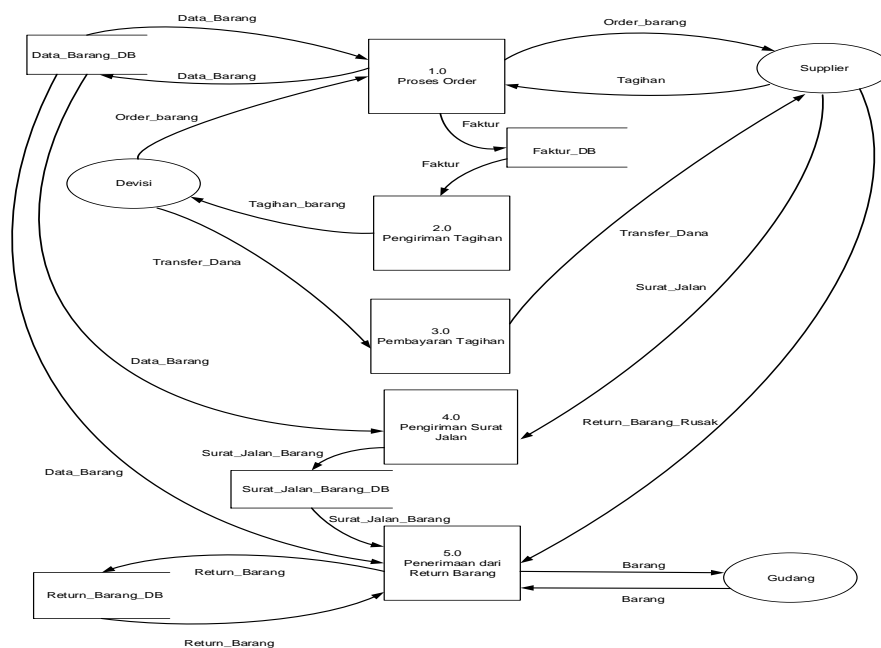
Berikut langkah-langkah pengerjaan penelitian pada gambar 2.

- Menyusun notasi DFD yang terdiri dari entitas devisi, supplier, dan gudang. Kemudian, memasukkan data *store* Data_Barang_DB, Faktur_DB, Surat_Jalan_Barang_DB, Return_DB. Penyusunan tersebut berdasarkan konsep input proses output pada penelitian [24].
- Merancang ERD berdasarkan data *store* DFD. Selanjutnya, menentukan atribut dan *cardinality ratio*.
- Membuat struktur *database* berdasarkan ERD.
- Membuat query berdasarkan primary key yang telah dibuat pada struktur *database* sebagai relasi antar tabel data barang, tabel faktur, tabel surat jalan barang, tabel *return* barang.

4. HASIL DAN PEMBAHASAN

4.1 Diagram Level 1 Model SSADM Untuk Sistem Informasi *Inventory*

Agar mempermudah ilustrasi aliran data dalam DFD maka penelitian ini mengambil contoh penelitian [23]. Hasil DFD Level 1 dapat dilihat pada gambar 1.

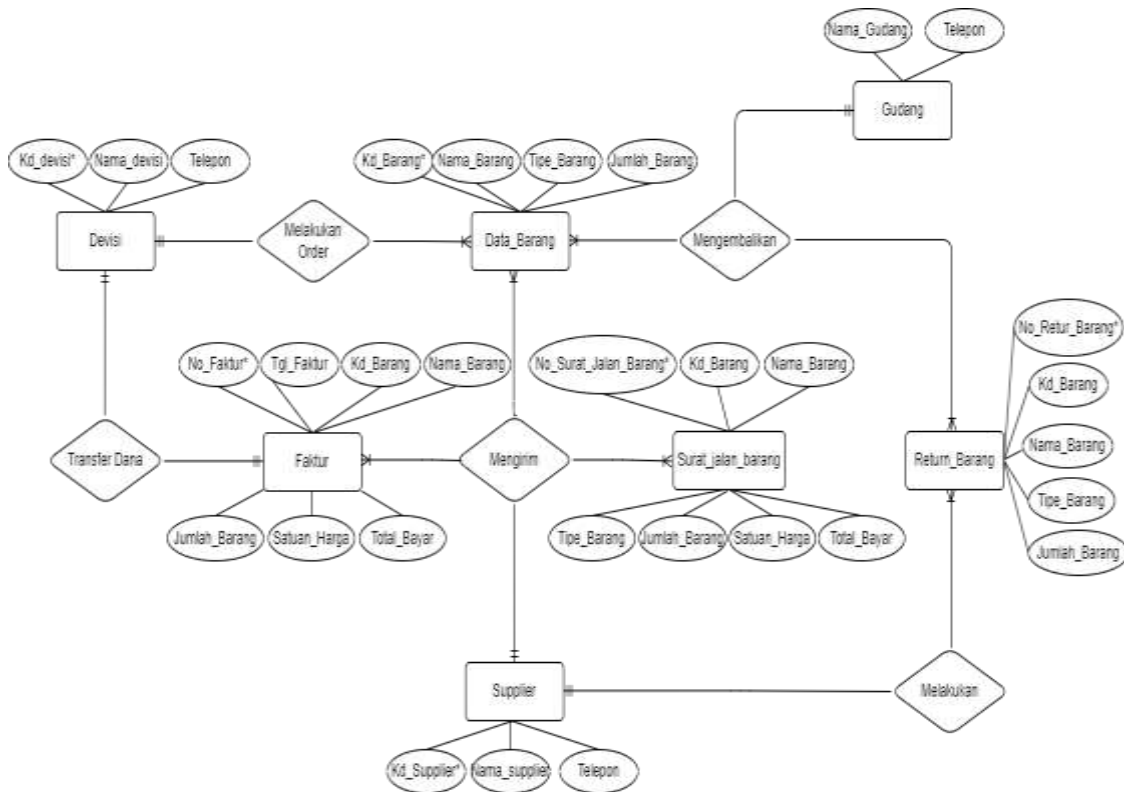


Gambar 2. Diagram Level 1 Model SSADM

Devisi melakukan order barang ke supplier. Selanjutnya, Devisi menginput data barang yang diorder. Kemudian, supplier memberikan tagihan kepada devisi. Devisi mentransfer dana sesuai tagihan yang dikirim oleh supplier. Supplier memberikan dan menginput surat jalan barang. Supplier juga mengembalikan barang yang rusak yang kemudian diterima oleh gudang. Gudang menginput barang yang rusak.

4.2 Perancangan Entity Relationship Diagram (ERD)

ERD dibuat berdasarkan data *store* pada DFD gambar 1, yaitu data barang DB, faktur DB, Surat jalan barang DB, dan *return* barang DB. Hasil ERD dapat dilihat pada gambar 3.



Gambar 3. Entity Relationship Diagram

Devisi melakukan *order* barang dan di input oleh supplier menjadi data barang. Atribut dari devisi adalah Kd_devisi sebagai *primary key*, Nama_devisi, dan Telepon. Atribut dari supplier adalah Kd_Supplier sebagai *primary key*, Nama_Supplier, dan Telepon. Atribut dari data barang adalah Kd_Barang, sebagai *primary key*, Nama_barang, Tipe_Barang, Jumlah_Barang. Kemudian, supplier mengirim faktur dan surat jalan barang. Atribut dari faktur terdiri dari No_Faktur sebagai *primary key*, Tgl_Faktur, Kd_Barang, dan Nama_Barang, Jumlah_Barang, Satuan_Harga, Total_Bayar. Atribut surat jalan barang adalah No_Surat_Jalan_Barang sebagai *primary key*, Kd_Barang, Nama_Barang, Tipe_Barang, Jumlah_Barang, Satuan_harga, Total_Bayar. Setelah itu, devisi mentransfer dana kepada supplier. Setelah itu, devisi mengembalikan data barang yang rusak dan supplier melakukan *return* barang kepada gudang. Atribut *return_barang* terdiri dari No_Retur_Barang sebagai *primary key*, Kd_Barang, Nama_Barang, Tipe_Barang, Jumlah_Barang. Gudang memiliki atribut Nama_Gudang, dan telepon.

4.2.1. Cardinality Ratio

Atas dasar gambar 3, tingkat hubungan *cardinality* yang pertama yaitu devisi melakukan *order* barang sehingga menjadi relasi *one to many* (1→M), artinya 1 ke banyak. Tingkat hubungan kedua, faktur tercipta dari barang yang dipesan, sehingga menjadi relasi *Many to Many* (M→M), artinya banyak ke banyak.

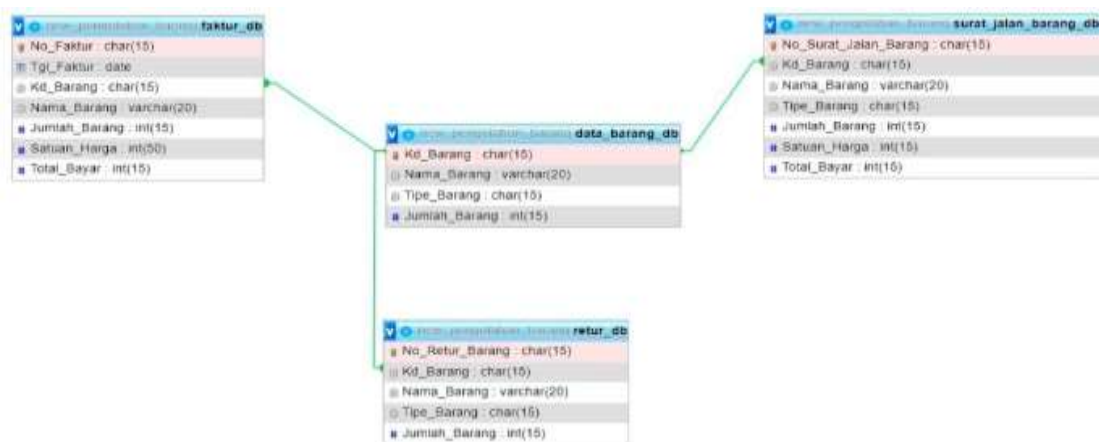
Pada tingkat hubungan ketiga, juga terjadi hubungan M→M dimana surat jalan barang akan ada jika barang dikirim oleh supplier, begitu juga dengan *return* barang jika ada barang yang rusak dikembalikan oleh supplier berdasarkan data barang yang telah dikirim. Devisi melakukan transfer dana berdasarkan faktur sehingga tingkat hubungannya menjadi 1→M, dan 1→M supplier melakukan *return* barang ke Gudang.

Artinya, jika tingkat hubungannya 1→M atau 1→M maka *cardinality*-nya bergabung ke arah *Many* (M). Namun tingkat hubungan *One to One* tidak perlu dilakukan pengelompokkan data.

4.3 Struktur Database

Setelah penyajian DFD dalam proses identifikasi aliran data input dan output, maka gambar 3 menunjukkan penyajian data berupa diagram secara spesifikasi untuk hubungan antar tabel dengan *primary key* dan *foreign key* yang ada pada *database* yaitu ERD [25]. Pendekatan berbasis *record* tidak melihat objek yang terlibat tetapi terlihat data apa saja yang terhubung dan menerapkan sejumlah aturan-aturan sehingga hasilnya berupa tabel-tabel, dan relasinya yang ada pada data *store* pada gambar 2.

Setiap pengelompokkan data menjadi satu kelompok data atau satu tabel yang dipengaruhi oleh *record* dan terbentuk tujuh kelompok data yaitu tabel devisi, tabel faktur, tabel surat jalan barang, tabel *return* barang, tabel supplier, dan tabel gudang yang menunjukkan bahwa pembuatan *database* berdasarkan 1 dokumen yang bernama data barang yang menciptakan tujuh tabel. Struktur *database* ini berfokus pada data barang sebagai pusat relasi antar tabel. Berikut hasil struktur *database* pada gambar 4.



Gambar 4. Struktur Database

4.4 Query SQL Menampilkan Tabel Faktur dan Tabel Data Barang

No_Faktur	Tgl_Faktur	Nama_Barang	Jumlah_Barang	Total_Bayar
12345	2022-04-25	Komputer	5	10000000
56789	2022-04-25	Mouse	4	400000

2 rows in set (0.000 sec)

Gambar 5. Query SQL Menampilkan Atribut Pada Tabel Faktur dan Tabel Data Barang

Untuk menampilkan beberapa atribut yang berada pada tabel faktur yang berelasi dengan tabel data barang, berikut *query*-nya:

```
SELECT faktur_db.No.Faktur, faktur_db.Tgl_Faktur, Faktur_db>Nama_Barang, Faktur.db.Jumlah_Barang, Faktur_db.Total_Bayar
from faktur_db inner join data_barang_db on faktur_db.Kd_Barang=data_barang.Kd.Kd_Barang;
```

4.5 Query SQL Menampilkan Tabel Return Barang dan Tabel Data Barang

No_Retur_Barang	Kd_Barang	Nama_Barang	Tipe_Barang	Jumlah_Barang
56789	BR03	Mouse	Hardware	2

1 row in set (0.000 sec)

Gambar 6. Query SQL Menampilkan Atribut Pada Tabel *Return* Barang dan Tabel Data Barang

Untuk menampilkan beberapa atribut yang berada pada tabel *return* barang yang berelasi dengan tabel data barang, berikut *query*-nya:

```
SELECT return_db.No_Retur_Barang, return_db.Kd_Barang, return_db>Nama_Barang,
return_db.Tipe_Barang, return_db.Jumlah_Barang
from return_db inner join data_barang_db on return_db.Kd_Barang=data_barang_db.Kd_Barang;
```

4.6 Query SQL Menampilkan Tabel Surat Jalan Barang dan Tabel Data Barang

No_Surat_Jalan_Barang	Nama_Barang	Tipe_Barang	Jumlah_Barang	Satuan_Harga	Total_Bayar
01022345	Komputer	Hardware	5	2000000	10000000
01022346	Mouse	Hardware	4	100000	400000

2 rows in set (0.001 sec)

Gambar 7. Query SQL Menampilkan Atribut Pada Tabel Surat Jalan Barang dan Tabel Data Barang

Untuk menampilkan beberapa atribut yang berada pada tabel surat jalan barang yang berelasi dengan tabel data barang, berikut *query*-nya:

```
SELECT surat_jalan_barang_db.No_Surat_Jalan_Barang, surat_jalan_barang_db>Nama_Barang,
surat_jalan_barang_db.Tipe_Barang, surat_jalan_barang_db.Jumlah_Barang,
surat_jalan_barang_db.Satuan_Harga, surat_jalan_barang_db.Total_Bayar from surat_jalan_barang_db
inner join data_barang_db on surat_jalan_barang_db.Kd_Barang=data_barang_db.Kd_Barang;
```

4.7 Pembahasan

Berdasarkan pengamatan yang telah dilakukan bahwa data *store* pada DFD gambar 2 Data_Barang_DB, Faktur_DB, Surat_Jalan_Barang_DB, *Return*_DB mempunyai pengaruh sangat penting dalam aliran DFD. Sebagai contoh, Data_Barang_DB dibuat berdasarkan 1.0 Proses *Order* yang memiliki aliran data barang, dan *order* barang. Data_Barang_DB tersebut dibuat entitas ERD dengan nama Data_Barang. Jika diperhatikan pada 1.0 Proses *Order* dapat disimpulkan bahwa barang tersebut diorder oleh devisi dan pastinya yang diorder adalah nama barang dan jumlah barang. Kemudian, dalam analisa lanjut bahwa nama barang itu sangat banyak. Oleh karena itu, perlu dikategorikan sehingga dibuat tipe barang. Nama barang tergolong identik, misalkan meja kayu dan meja plastik sehingga untuk membedakan diperlukan kode yang unik. Sebab itu, dibuat kode barang dengan alias Kd_Barang.

Serupa dengan Data_Barang_DB, Return_Barang_DB dibuat berdasarkan 5.0 Penerimaan dari Return Barang yang mempunyai aliran data data barang dan return barang rusak. Return_DB dibuat entitas ERD dengan nama Return_DB. Berdasarkan pengamatan pada 5.0 Penerimaan dari Return Barang dapat disimpulkan bahwa return barang dihasilkan dari aliran data barang dan return barang rusak. Artinya, bahwa return barang rusak dilihat dari data barang yang diorder. Oleh karena itu, pada atribut dalam entitas ERD Return_Barang dimasukkan atribut No_Retur_Barang, Kd_Barang, Nama_Barang, Tipe_Barang, Jumlah_Barang. Kd_Barang diperlukan karena sebagai relasi dari entitas Data_Barang dan entitas Return_Barang sehingga dapat disimpulkan bahwa data store Data_Barang_DB dan Return_Barang_DB pada DFD saling berhubungan.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan kasus penelitian yang diambil, sebagai contoh hubungan data store dapat terlihat dari Data_Barang_DB dan Return_Barang_DB pada DFD yaitu barang rusak dikembalikan dari barang yang diorder oleh devisi dan diterima oleh supplier untuk dikembalikan dan diinput oleh gudang. Hal ini menunjukkan bahwa Return_Barang_DB diambil dari Data_Barang_DB. Jika ditransformasikan menjadi entitas Data_Barang dan Return_Barang, hubungan entitas tersebut berdasarkan atribut Kd_Barang.

Dengan demikian, adanya atribut Kd_Barang yang sama pada entitas Data_Barang dan Return_Barang akan memudahkan dalam membuat tabel *query* dengan perintah *select* untuk mem-filter data yang ditunjukkan pada gambar 6.

5.2 Saran

Kendala pada penelitian ini perlu kejelian dalam pengamatan antara DFD dengan ERD karena harus dibandingkan secara terpisah. Oleh karena itu, disarankan perlu adanya diagram yang mempunyai perancangan sistem dan *database* yang terintegrasi sehingga memudahkan pengamatan.

DAFTAR PUSTAKA

- [1] H. Jurnal, P. Dina Mardika, A. Fauzi, U. Indraprasta PGRI Jakarta Jl Nangka Raya No, and J. Selatan, "JURNAL PUBLIKASI TEKNIK INFORMATIKA IMPLEMENTASI METODE SCRUM PADA PERANCANGAN SISTEM INFORMASI TATA USAHA SEKOLAH BERBASIS WEB," *Januari*, vol. 1, no. 1, 2022, Accessed: Jun. 19, 2022. [Online]. Available: <http://ejurnal.stie-trianandra.ac.id/index.php/juhti/article/view/188>
- [2] A. Fauzi and E. Harli, "JURNAL RESTI (Rekayasa Sistem dan Teknologi I nformasi) Peningkatan Kualitas Pelayanan Melalui CRM dengan Metode RAD," 2017, [Online]. Available: <http://jurnal.iaii.or.id>
- [3] E. W. Fridayanthie, H. Haryanto, and T. Tsabitah, "Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web," *Paradigma - Jurnal Komputer dan Informatika*, vol. 23, no. 2, Sep. 2021, doi: 10.31294/p.v23i2.10998.
- [4] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, 2020, Accessed: Jun. 19, 2022. [Online]. Available: https://www.researchgate.net/publication/346397070_Analisis_Metode_Waterfall_Untuk_Pengembangan_Sistem_Informasi
- [5] S. R. Pressman, Roger, S. Pressman, Ph.D. , 2012, *Rekayasa Perangkat Lunak - Google Cendekia*, Ed.7. Yogyakarta: Andi, 2012. Accessed: Mar. 12, 2020. [Online]. Available: https://scholar.google.co.id/scholar?hl=id&as_sdt=2005&scioldt=0%2C5&cites=7972897112769499795&scipsc=&q=+Roger%2C+S.+Pressman%2C+Ph.D.+%2C+2012%2C+Rekayasa+Perangkat+Lunak+%28Pendekatan+Praktisi%29+Edisi+7+%3A+Buku+1+%E2%80%9C%2C+Yogyakarta%3A+Andi&btnG=#d=gs_cit&u=%2Fscholar%3Fq%3Dinfo%3Ay5T8dx2Nox0J%3Ascholar.google.com%2F%26output%3Dcite%26scirp%3D0%26hl%3Did
- [6] N. Kustian, "STRING (Satuan Tulisan Riset dan Inovasi Teknologi) PENGGUNAAN MODEL WATERFALL DALAM PEMBUATAN APLIKASI PEMESANAN TIKET BUS," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 1. No.4, no. 1 Agustus 2019, pp. 94–104, 2019, doi: 10.30998/string.v4i1.3768.

- [7] J. Inovasi Penelitian, O. Bayu Rianto, and M. Giatman, "RANCANG BANGUN APLIKASI WEBSITE E-LEARNING PADA LKP ENGLISH CLUB TEMBILAHAN," vol. 1, no. 9, 2021.
- [8] F. Soufitri, "PERANCANGAN DATA FLOW DIAGRAM UNTUK SISTEM INFORMASI SEKOLAH (STUDI KASUS PADA SMP PLUS TERPADU)".
- [9] N. Sudarsono, "Sistem Informasi Inventory Berbasis Web di PT Autotech Indonesia."
- [10] M. Muliadi, M. Andriani, and H. Irawan, "PERANCANGAN SISTEM INFORMASI PEMESANAN KAMAR HOTEL BERBASIS WEBSITE (WEB) MENGGUNAKAN DATA FLOW DIAGRAM (DFD)," *JISI: Jurnal Integrasi Sistem Industri*, vol. 7, no. 2, p. 111, Sep. 2020, doi: 10.24853/jisi.7.2.111-122.
- [11] M. Arif, *Pemodelan Sistem*, Ed.1 Cet. 2. Yogyakarta: deepublish (Grup Penerbitan CV BUDI UTAMA), 2017. Accessed: Jun. 19, 2022. [Online]. Available: https://books.google.co.id/books?id=LVxDDwAAQBAJ&lpg=PT11&ots=_Ify71kTXs&dq=PEMODELAN%20SISTEM&lr&pg=PA1#v=onepage&q=PEMODELAN%20SISTEM&f=false
- [12] S. Radack, "THE SYSTEM DEVELOPMENT LIFE CYCLE (SDLC) NIST Special Publication (SP) 800-64, Revision 2, Security Considerations in the System Development Life Cycle."
- [13] S. Sarosa, *Metodologi Pengembangan Sistem Informasi*, Cetakan I. Jakarta, 2017. Accessed: Jun. 19, 2022. [Online]. Available: <https://opac.perpusnas.go.id/DetailOpac.aspx?id=1080923#>
- [14] R. Ibrahim and S. Y. Yen, "Formalization of the Data Flow Diagram Rules for Consistency Check," *International Journal of Software Engineering & Applications*, vol. 1, no. 4, pp. 95–111, Oct. 2010, doi: 10.5121/IJSEA.2010.1406.
- [15] "4 Data Flow Diagram."
- [16] M. Woodman, "Yourdon dataflow diagrams: a tool for disciplined requirements analysis," 1988.
- [17] R. G. Fichman and C. F. Kemerer, "Object-Oriented and Conventional Analysis and Design Methodologies Comparison and Critique."
- [18] C. M. Ashworth, "Structured systems analysis and design method (SSADM)."
- [19] M. C. Goodland and R. K. Maji, "The Application of SSADM to the Design of Information Systems in Manufacturing," 1987.
- [20] M. A. Mohammed, J. Abdullah, D. A. Muhammed, D. Abdul Kareem Muhammed, and J. M. Abdullah, "Practical Approaches of Transforming ER Diagram into Tables International Journal of Multidisciplinary and Scientific Emerging Research Practical Approaches of Transforming ER Diagram into Tables," 2015. [Online]. Available: <http://www.ijmser.com/>
- [21] G. T. Mardiani, S. Kom, and M. Kom, "Entity Relationship Model," 2016.
- [22] J. Frantiska, "Entity-Relationship Diagrams," 2018, pp. 21–30. doi: 10.1007/978-3-319-67440-7_4.
- [23] N. Sudarsono and S. Sukardi, "Sistem Informasi Inventory Berbasis Web di PT Autotech Indonesia," *Jurnal Eksplora Informatika*, vol. 5, no. 1, pp. 73–84, 2015.
- [24] H. Herlinda, R. Ramliyana, and E. W. Ambarsari, "Pengejawantahan Pendekatan Storytelling dalam Pemahaman Membangun Diagram Alir Data," *Edumatic: Jurnal Pendidikan Informatika*, vol. 5, no. 2, pp. 223–232, Dec. 2021, doi: 10.29408/edumatic.v5i2.4060.
- [25] S. M. S. Web Dan, "Fathansyah. 2015. Basis Data. Revisi Kedua. Informatika, Bandung."