

## DATA INTEGRATION MODEL DESIGN FOR SUPPORTING DATA CENTER PATIENT SERVICES DISTRIBUTED INSURANCE PURCHASE WITH VIEW BASED DATA INTEGRATION

Sudaryanto<sup>1</sup>, Slamet Sudaryanto Nurhendratno<sup>2</sup>

<sup>1,2</sup>Computer Science Faculty, Dian Nuswantoro University  
Central Java, Indonesia

<sup>1</sup>msdr8047@dsn.dinus.ac.id, <sup>2</sup>slametalica301@dsn.dinus.ac.id

Page | 162

**Abstract** — *Data integration is an important step in integrating information from multiple sources. The problem is how to find and combine data from scattered data sources that are heterogeneous and have semantically informant interconnections optimally. The heterogeneity of data sources is the result of a number of factors, including storing databases in different formats, using different software and hardware for database storage systems, designing in different data semantic models (Katsis & Papakonstantiou, 2009, Ziegler & Dittrich , 2004). Nowadays there are two approaches in doing data integration that is Global as View (GAV) and Local as View (LAV), but both have different advantages and limitations so that proper analysis is needed in its application. Some of the major factors to be considered in making efficient and effective data integration of heterogeneous data sources are the understanding of the type and structure of the source data (source schema). Another factor to consider is also the view type of integration result (target schema). The results of the integration can be displayed into one type of global view or a variety of other views. So in integrating data whose source is structured the approach will be different from the integration of the data if the data source is not structured or semi-structured. Scheme mapping is a specific declaration that describes the relationship between the source schema and the target schema. In the scheme mapping is expressed in in some logical formulas that can help applications in data interoperability, data exchange and data integration. In this paper, in the case of establishing a patient referral center data center, it requires integration of data whose source is derived from a number of different health facilities, it is necessary to design a schema mapping system (to support optimization). Data Center as the target orientation schema (target schema) from various reference service units as a source schema (source schema) has the characterization and nature of data that is structured and independence. So that the source of data can be integrated terstruktur of the data source into an integrated view (as a data center) with an equivalent query rewriting (equivalent). The data center as a global schema serves as a schema target requires a "mediator" that serves "guides" to maintain global schemes and map (mapping) between global and local schemes. Data center as from Global As View (GAV) here tends to be single and unified view so to be effective in its integration process with various sources of schema which is needed integration facilities "integration". The "Pemadu" facility is a declarative mapping language that allows to specifically link each of the various schema sources to the data center. So that type of query rewriting equivalent is suitable to be applied in the context of query optimization and maintenance of physical data independence.*

**Keywords**—Global as View (GAV), Local as View (LAV), source schema, mapping schema

### I. INTRODUCTION

As complete information from an application is often provided by many different and autonomous data sources as a reference in providing an integrated global scheme [1]. Data source formats are often also heterogeneous (text, web pages, XML documents, relational databases). Arrange the data also with different methods and methods such as web forms and database client. This makes the function of combining data from different sources is not easy to do and requires a stronger effort. The first step that must be done in doing data integration is to take separate data from each different source, then memahi each data and linkage relationship. It should also consider the differences in structure and value as well as the potential for inconsistency. In integration and data

exchange requires the transformation of data structures into a schema, ie transformation of the source scheme into a target scheme that has a different structure. The transformation of a local schema into the target scheme is usually called scheme mapping. Mediators and wrappers are needed as a concept of integration architecture between the sources of the schema [7]. The wrapping function is used to wrap the information container and model it into the source of the scheme. While the mediator function is to maintaining global schemes and mapping between global schemes and local schemes. Each user queries all objects related to the global schema then the mediator will use query-reformulation procedure to translate the query into an executable sub query from all the schema sources involved in the query and then reassemble the answer from each - the source of the

schema to be further combined to answer the query. Today various approach methods and categories can be used to integrate federation or multidatabase systems [11]. Today in order to integrate data without realizing the global schema there are three categories of language in expressing the correspondence, namely: Global as View (GAV), Local as View (LAV), Global Local as View (GLAV), Both as View (BAV) together with related theory and system [8], [9]. The query processing algorithm from the literature for LAV describes the data integration system describing and adopting the GAV query integration system and [10] process. In the GAV approach, reforming queries will be able to reduce the application of simple rules (standard execution views on ordinary databases). However, global schemes will require a refission on the application of mapping between global schemes and source schemes in case of changes or additions from an information source. in large-scale applications whereas LAV is easier to maintain than GAV because DBA creates a global schema regardless of source schema If there is a new source schema the DBA simply adjusts the source description that describes the source relationship as the view of the global schema Automatic query reformulation on the LAV has a complex time exponential with respect to the query and definition of the source schema. Therefore, LAV has a low query performance performance when users often request complex queries, as well as data centers.

advantage of this data integration is that the user does not have to search for the relevant data source by query or manually combine data from each data source. As in Figure 2. below, where user requests are submitted to the data integration system, it is first formulated within the framework of a global schema to be executed. The system then translates the request into sub query which is expressed in the local schema of some independent data source.

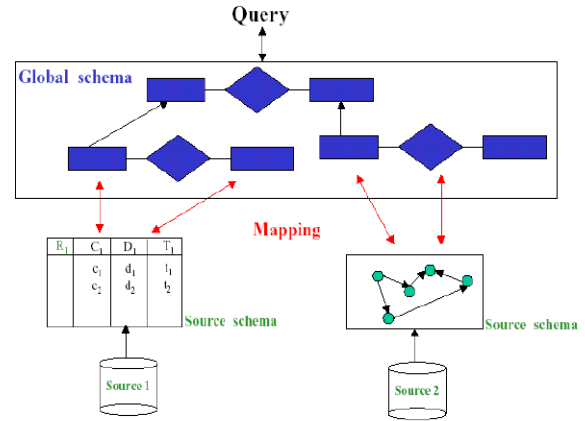


Fig 2. Basic Architecture Data Integration System

There are two main components of data integration that must be considered, namely Schema Integration and Query Processing. The integration of schema is directly related to how various local schema can be combined into one global schema. While query processing relates to how a query can be answered by translating to one or more queries in the source database. There are four main approaches to data integration: Local as View (LAV), Global as View (GV), Global Local as View (GLAV) and Both as View (BAV). All of these approaches are unmaterIALIZED (virtual) in which it uses the definition of a view to determine the mapping between local schema and global schema. A view definition is a query from another construct to determine the extents of a construct. Mapping is used to translate queries that are expressed in the global schema framework to sub queries that are expressed in the local schema.

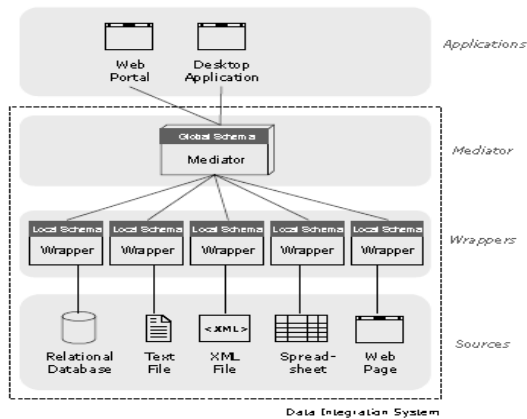


Fig 1. View Based Data Integration System Architecture

## II. METHODOLOGY

Data integration is a triple relationship  $(T, \{S_i\}, \{M_i\})$ , where  $T$  is the target of the scheme,  $\{S_i\}$  is the number of  $n$  source schemes and  $\{M_i\}$  is the number  $n$  source mapping to the target schema. So for every  $S_i$  source scheme is in  $M_i$  from  $S_i$  to  $T$ ,  $1 \leq i \leq n$ . The integration is a process of combining data that is on a different source  $\{S_i\}$  with the associated local schema to form a single virtual database (nonmaterialize) with the associated global schema as the target of the  $T$  [1,2] scheme. This is to provide users a uniform query interface for discrete heterogeneous databases. The

### A. Conjunctive Queries and Datalog Notation

Datalog notation is required to model and express views definitions and queries [11,12]. The data integration system is a triple relationship  $(G, \{S_i\}, \{M_i\})$ , where  $G$  is the schema target or global schema,  $\{S_i\}$  is the number  $n$  source scheme and  $\{M_i\}$  is the number  $n$  source mapping to the target schema. So for every  $S_i$  source scheme is in  $M_i$  from  $S_i$  to  $T$ ,  $1 \leq i \leq n$ . Such integration is the process of combining data residing on different sources  $\{S_i\}$ . A mapping between the source and the global schema is a set of assertions:

1.  ${}^qS \Rightarrow {}^qG$ ,
2.  ${}^qG \Rightarrow {}^qS$ ,

Intuitively, the first statement mentions that the concept is represented by a view (query)  $^Q S$  above the  $S$  source scheme in accordance with the concept defined by  $^G$  about global schemes, and vice versa.

### B. Mapping Schema Construction

As an illustration of the discussion in schema mapping, it uses two databases (local database schema groups as source schemes and databases for targets with global schema). We consider the source database as  $D_s$  which contains local schema (Source Scheme -  $S_s$ ) with some relation (n relation  $R_1 \dots R_n$ ). While the target database (Target-Dt Database) contains a target schema (Schema Target- $St$ ) consisting of one target  $R$  relation. Thus the mapping scheme ( $M$ ) is a join join project that maps  $S_s$  to  $St$ . For each  $R_i, I \in [n]$ , we show the scheme with  $S(R_i)$  and for example with  $I(R_i)$ .  $S(R_i)$  is the set of all attributes in  $R_i$ . Similarly  $R$  (Target Relation) has a  $S(R) = \{A_1, \dots, A_m\}$  scheme, where  $m$  is the target size and  $A_j (j \in [m])$  represents the  $j$ th attribute in  $R$ .  $t[A]$  dari proyeksi tuple pada atribut  $A$ . Jenis pengguna in the input spreadsheet under the target scheme. Each sample  $E$  is a string attribute. We show the first sample baser with  $tE = (E_1, \dots, E_m)$  and call it a tuple sample. Our objective for sample sampling is to encompass all schema mappings that alter the source database of the target object "contains" a sample tuple.

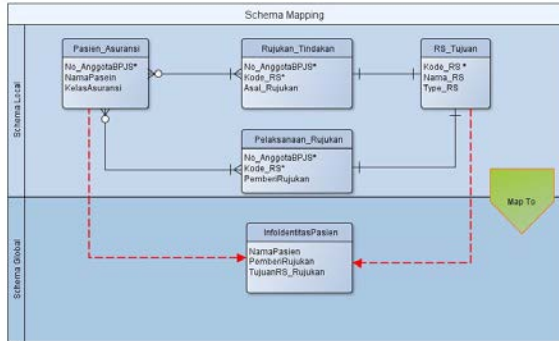


Fig 3. Mapping Local Schemes With Merging to a Global Scheme

Because user inputs are not exactly the same as the source, since we use to generate schema mappings, we use an inaccurate sample by letting it "noisily contain" by some database instances. We formally define the relationship ("noisily contain") by the binary operator  $\succcurlyeq$ , which returns the boolea value based on the desired error model. With this operator we say sample containing sample  $E$  iff  $t[A] \succcurlyeq E$ . Similarly we say that  $t$  contains  $E$  iff  $\exists A$  s.t.  $t[A] \succcurlyeq E$ . Furthermore, given  $tE = (E_1, \dots, E_m)$ , we call  $T$  contains  $tE$ , iff  $\forall i \in [m], t[A_i] \succcurlyeq E_i$ . Finally we can specify the target databases  $D_t$  contains  $t E$  iff  $\exists t \in D_t$  s.t. contains  $tE$ . With this concept, we define sample search as follows: if given the source database  $D_s$  and sample tuples  $tE = (E_1, \dots, E_m)$ , in search to find all  $M$  schema mapping in such a way that the mapping result comes from the

source database  $M$  (schema mapping) of the source database ( $D_s$ ) contains all the tuple database of the  $tE$  source, then each result of the schema mapping is called a valid schema mapping.

### a. Process Query Translation

This section discusses some questions ( $Q$ ), suppose there are two pairs of source databases of different structures as ( $D_s$ ) containing local schema ( $S_s$ ) with some relation (n relation  $R_1 \dots R_n$ ), if there is a question involving all local schemes of pairs  $D_s$  source database is different. It takes several processes from query translation, such as query translation semantic, forward query translation (background tree pattern, translation tree pattern join) and backward query translation. For each set of candidate  $V$  (View) of the selected cover, the common commands used for selection are:

**Select attributes in  $V$**   
**From source relations in the join path for  $V$**   
**Where filter and join conditions from the join path**

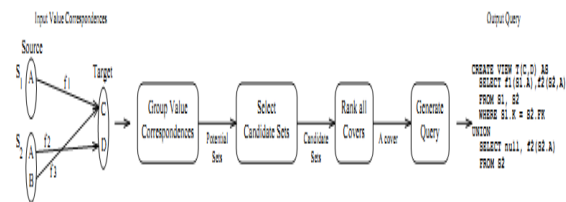


Fig 4. Algoritma Mapping

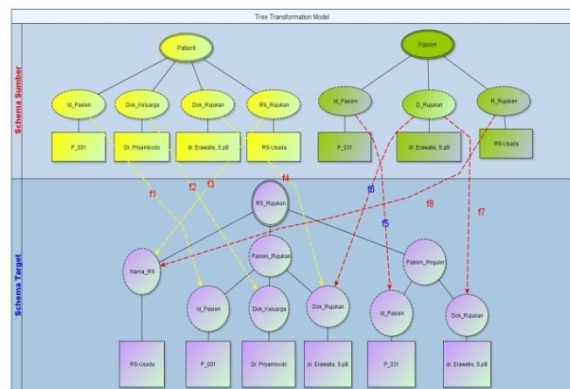


Fig 5. Tree mapping and Translation Correspondences

Note that these principles are restatements of general data principles such as "one fact in one place." Even with this filter, we try to enforce these principles for the values selected by the filter because our goal is schema mapping rather than scheme design, we allow users to change these principles. For example, in publishing information for the "What-if" scenario, the user may sort the cross product until he can evaluate all possibilities. We use these principles to encourage initial mapping, which keeps, as far as possible, the information in the source. The user blocks my target data from this mapping and decides whether to the

district or modify the mapping so that the tree mapping process becomes a reference for performing queries with certain attributes and filters, if applied to the image tree mapping above then the additional correspondence command is:

```
f1 : pasien(id_pasien) → rs_rujuka(id_pasien)
f2 : pasien(dok_keluarga) → rs_rujuka(dok_keluarga)
f3 : pasien(dok_rujukan) → rs_rujuka(do_rujukan)
f4 : pasien(rs_rujukan) → rs_rujuka(rs_rujukan)
f5 : pasien(id_pasien) → rs_rujuka(id_pasien)
f6 : pasien(d_rujukan) → rs_rujuka(dok_rujukan)
f7 : pasien(dok_rujukan) → rs_rujuka(dok_rujuka)
f8 : pasien(r_rujukan) → rs_rujuka(nama_rs)
```

```
select pt.id_pasien, pt.dok_keluarga, pt.dok_rujukan,
ps.id_pasien, ps.nama_rs
from patient pt, pasien ps
where pt.id_pasien=ps.id_pasien
union all
select null as dok_keluarga, ps.id_pasien ,
ps.dok_rujukan*pt.d_rujukan, null as pasien_reguler
from patient pt, pasien ps
where pt.id_pasien=ps.id_pasien
```

```
create view t_rujukan (id_pasien, dok_keluarga,
nama_kasus, nama_rs) as
select f1(s1, id_pasien, dok_keluarga), f2
(s2,dok_rujukan, fast_rujukan, rs_rujukan)
from s1,s2
where s1.id_pasien=s2.id_pasien
union
select f2(s2, id_pasien, null, rs_rujukan)
```

### III. RESULT AND DISCUSSION

#### A. Global As View (GAV) Approach

In the Global as View (GAV) approach as an integration frame work, the global schema is defined as a view of all local schemes involved. To be more precise any constructions or global elements are defined by the view or top views of the corresponding local sources. So the data residing on the source scheme contributes to the construction of a global scheme. The figure below explains the relationship (mapping) between the data source and the global schema obtained by defining each data source as Global as View (GAV). GAV mapping is a series of assertions, one for each element  $g$  of  $G$ , if  $g \Rightarrow qs$  means, mapping specifies  $g$  as query  $qs$  as a data source. This means that the mapping gives us know how the  $g$  elements are calculated. Examples of GAV deliberations:

Source 1	Source 2
Pasien(Id_Pasien, Nama_Pasien, Tgl_Lhr, Dok_Keleluarga,	Patient(Id_Pasien, Nama_Pasien, T_Lahir, D_Rujukan, R_Rujukan, Tgl_Tind,

Dok_Rujukan, RS_Rujukan) Tindakan(NoRM, Kode_Dok, Tgl, Nama_Tind, Qty, Kode_RS, ICD, Total_Bea)	Nama_Tind, Total, ICDX)
<b>Global Schema</b>	
<p><b>Create View Global-Pasien_Rujukan As</b>  Create View Global-Pasien_Rujukan As  Select Id_Pasien As P_Rm, Kode_Dok As Dok_Id, Tindakan.Tgl As Tgl_Tindakan, Nama_Tind, Qty As Jum_tindakan,, Kode_RS, Year(currdate()-Pasien.Year(Tgl_Lhr) AS Umur, ICD, Total_Bea  From source1.Pasien, source1.Tindakan  Where Id_Pasien = No_Rm  Union  Select Id_Pasien As P_Rm, null AS DokId, Tgl_Tind AS Tgl_Tindakan, Nama_Tind, Kode_RS, Year(currdate()-Year(tgl) As Umur, ICDX As ICD , Total As Total_Bea  From source2.Patient</p>	
<b>Query Over The Global Schema</b>	<b>Query Over The Local Schema</b>
Select P_Rm, Kode_Dok, ICD, Total From Global-Pasien_Rujukan Where Kode_RS="Swastal" and Nama_Tind ="Fraktur F"	Select A.Id_Pasien, B.Kode_Dok, B>Nama_Tind,B.Qty From Source1. Pasien A, Source1.Tindakan B Where A.Id_Pasien=B.NoRM And B.Kode_Rs="Swastal" And B>Nama_Tind="Fraktur F" Union Select Id_Pasien, D_Rujukan, Nama_Tind, Total From Source2.Patient Where R_Rujukan="Swastal" and Nama_Tind ="Fraktur F"

Fig 6. Schema Integration Local to Global Schema Based View

#### B. Approach To Query Answering (Using View)

There are several things that pertain to the query process that must be considered and have an effect on the effectiveness of query answering in the data integration scheme. Some things to watch out for are: constraint / integrity constraints in the global schema, permitted classes in the mapping and query classes in the mapping. The treatment algorithm will be different between GAV with constraint and GAV without constraint. For GAV without constraint is the simplest case in answering query. This model is also



considered to be the first order query in the mapping. Its view is exact so it can be proved that there is one global database as a target that is mapped from legal data sources. In GAV without a global database constraint derived from the source scheme where the display's existence is calculated by using the display definition to map it. The answer to the query, the query user is calculated by evaluating the query through the global database. Likewise to modify the query, the user can easily be able to obtain an equivalent query that can be executed based on the source. This can be done easily after the ongoing strategy where each atom is above View (V) where V is the symbol of the relation in the global scheme replaced by a query corresponding to the GAV mapping.

With Q query and view definition V1, ..., Vm rewriting Q by using query view Q 'view which has the purpose of view relationship V1, ..., Vm or predicate comparison. Conceptually rewriting can have two goals: equivalent rewritings and maximally-contained rewritings. In the context of pembatasan in this paper equivalent rewriting is the most suitable choice to be applied, because it has the goal of query optimization and maintenance of the independence of physical data. Query Q1 is contained in query Q2 if for every database, then answering query to Q1 is always a subset of Q2. Queries are equivalent if they are related. Suppose that is the database schema and V is the set view of T. The expansion of the query P uses the view in V, denoted P<sup>exp</sup>, obtained from P by replacing all views in P with the corresponding base relation. By query P on T is called query rewriting Q corresponding to V if P uses only view in V, and P<sup>exp</sup> is contained in Q as query. P is called query rewriting equivalent to Q by using V if P<sup>exp</sup> and Q are equivalent to queries

```
Pasien(Id_Pasien, Nama_Pasien, Tgl_Lhr,
Dok_Keleluarga, Dok_Rujukan, RS_Rujukan)
Tindakan(NoRM, Kode_Dok, Tgl, Nama_Tind, Qty,
Kode_RS, ICD, Total_Bea)
Dokter(Kode_Dok, Nama, Spesialisasi)
```

```
Query Q1,
Select D.Nama, Total_Bea
From Pasien P, Tindakan T, Dokter D
Where P.Id_Pasien=T.NoRM AND
T.Kode_Dok=D.kode_Dok And Dok_Keluarga =" dr.
jennar"
```

The request wanted to ask the doctor's name for the total cost of the patients who had a family doctor dr.jennar. Queries and views are often written in conjunctive, then the query can be rewritten as:

```
Q1(T,G) :-Pasien(P, N, joko), Tindakan(P, D, G),
Dokter(D, T, Q)
```

Small letters (such as "joko") as arguments for constants, capitalized arguments (such as "P") for variables. The ":" symbol is the query body. It has a sub-objective tuga, each of which is a relationship in one body. The "joko" constant on the first subgoal represents the selection condition. The S variable possessed by the first two subregions represents a combination of patient relation and action on the patient-id\_pasien attribute. The T and G variables on the query head (on the left side of the notation ":" symbol, represent the projected final attribute.) The result of the view view defined from the base table is:

```
Views :
V1(I, N, Dk, K, Nt,Q):- Pasien (I, N, Dk),
Tindakan(No, K, Nt, Q)
V2(No, K,Nt,M,Q):- Tindakan(No, K, Nt, Q),
Dokter(K, M,S)
```

The SQL statement for the view definition of V 1 is following

```
CREATE View V1 As
Select P.Id_Pasien, P>Nama_pasien, P.Dok_keluarga,
T.Kode_Dok,T>Nama_Tind, T.Qty
From Pasien P, Tindakan T
Where P.Id_Pasien=T.NoRM
```

This view is a composite of patient relation with action. Similarly what happens to View V2 is a combination of actions and doctors, unless the attribute of value and value is dropped on the final result. Here is the result of query rewriting q1 using 2 View.

```
Answer(N,Nt):-V1(I, N,joko, K, Nt), V2(No, K, N)
```

This view is a natural (integration) combination of relationships between patients and actions. Similarly view V2 is a natural (integration) combination of the relation of action and physician, except that the attribute of value is dropped on the final result. Here is a query rewrite q1 using two views.

```
Answer(M,Nt) :- pasien(I, N, joko, K, Nt),
Tindakan(No,K,Nt),Tindakan(N,K,Nt'), Dokter(K, M,
S')
```

Nt 'and S' are the new variables that are introduced during substitution. This expansion is equivalent to query, so rewriting is equivalent (equivalent) of the query. If there is a new quarter attribute selection condition then the definition is for example in V2 then the view can be defined as follows

```
:
V2'(I,K,M):-Tindakan(S,K,Nt), Dokter(K, M,
internis)
```

If in the definition of V2 there is a selection condition again in the quarter attribute, then the definition of the view becomes:

Answer(M,Nt):-V1(I,N,joko,K, Nt), V2'(I,K,M)

Keterangan Atribut :

(Id\_Pasien =I, Nama\_Pasien=N, Tgl\_Lhr=T, Dok\_Keleluarga=Dk, Dok\_Rujukan=Dr, RS\_Rujukan=R, NoRM=No, Kode\_Dok=K, Tgl=L, Nama\_Tind=Nt, Qty=Q, Kode\_RS=Kr, ICD=I, Total\_Bea=B, Nama=M, Spesialisasi=S )

Mapping the Global View model is particularly suited for relatively stable data sources, but it will be very difficult if there are new and frequently changing data sources. So it requires an algorithm that can pair dynamically anatar scheme in the map. In the GAV Fitting Algorithm the attributes of the source sche attribute to the global schema attribute occur in three stages.

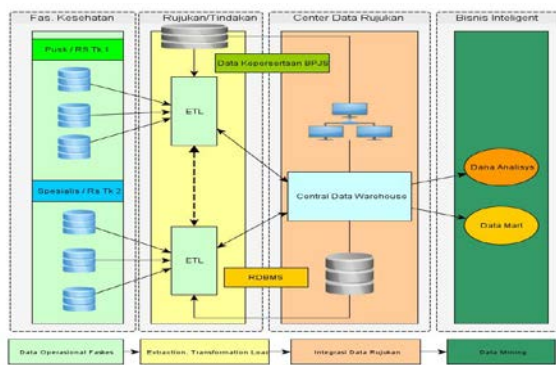


Fig 7. Arsitektur Central Data Warehouse

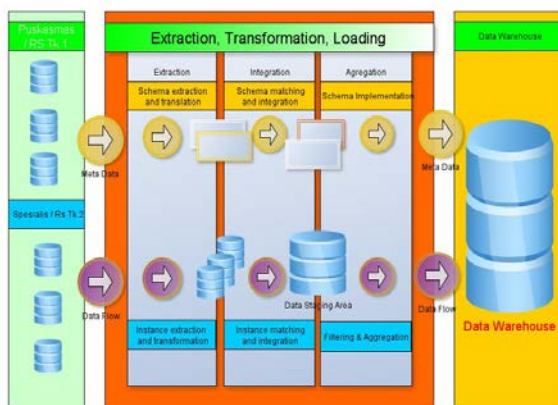


Fig 8. Data Staging Process

#### IV. CONCLUSION

In this paper, we identified a new problem, targeted scheme mapping as a distributed health service data center. Which is very important for some of the increasingly common application classes it will be put as a global schema. This scheme is part of the generalization of some local schemes that usually

occur in basic service units such as puskesmas, family physician practice to basic type hospitals. We distinguish mapping schemes from well-known problem schema integration, and discuss the similarities and differences between the two. By using representative mapping queries, we allow the DBMS to play the role spent as a data transformation engine, as well as data storage. Additionally, we found usage spent on many of the techniques of query optimization, as we applied it to new tasks of query invention or mapping from local schema to global schema.

#### ACKNOWLEDGEMENTS

This research was provided by the Research and Technology Ministry of Higher Education, sponsored under a grant budget of private colleges compete coordinator VI Central Java.

#### REFERENCES

- [1] McBrien, Peter, and Alexandra Poulouvasilis. "Data integration by bi-directional schema transformation rules." *Data Engineering, 2003. Proceedings. 19th International Conference on. IEEE*, 2003.
- [2] Lenzerini, Maurizio. "Data integration: A theoretical perspective." *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM*, 2002.
- [3] Katsis, Yannis, and Yannis Papakonstantinou. "View-based data integration." *Encyclopedia of Database Systems*. Springer US, 2009, 3332-3339.
- [4] Kolaitis, P. G. (2009). Relational Databases, Logic, and Complexity [Powerpoint slides]. Retrieved from <https://users.soe.ucsc.edu/~kolaitis/talks/gii09-final.pdf>
- [5] Hellerstein, Joseph M., Michael Stonebraker, and James Hamilton. "Architecture of a database system." *Foundations and Trends® in Databases* 1.2 (2007): 141-259.
- [6] Miller, Renée J., Laura M. Haas, and Mauricio A. Hernández. "Schema mapping as query discovery." *VLDB*. Vol. 2000. 2000.
- [7] Ullman, Jeffrey D. "Information integration using logical views." *International Conference on Database Theory*. Springer, Berlin, Heidelberg, 1997.
- [8] Levy, Alon, Anand Rajaraman, and Joann Ordille. *Querying heterogeneous information sources using source descriptions*. Stanford InfoLab, 1996.
- [9] Genesereth, Michael R., Arthur M. Keller, and Oliver M. Duschka. "Infomaster: An information integration system." *ACM SIGMOD Record*. Vol. 26. No. 2. ACM, 1997.
- [10] Calvanese, Diego, Domenico Lembo, and Maurizio Lenzerini. "Survey on methods for query rewriting and query answering using views." *Integrazione, Warehousing e Mining di sorgenti eterogenee* 25 (2001).
- [11] Landers, Terry, and Ronni L. Rosenberg. "An overview of Multibase." *Distributed systems, Vol. II: distributed data base systems*. Artech House, Inc., 1986.
- [12] Ullman, Jeffrey D. "Database and Knowledge-Base Systems, Volumes I and II." (1989).
- [13] Abiteboul, Serge. "Foundations of Databases/Serge Abiteboul, Richard B. Hull, and Victor Vianu." (1995).
- [14] Nurhendratno, Slamet Sudaryanto, and Fikri Budiman. "Design Model Integration And Synchronization Between Surveillance Units To Support Data Warehouse Epidemiology." *Journal of Theoretical and Applied Information Technology* 95.3 (2017): 498.