ISSN 1412-2049

An Object Oriented Approach to the Development of Industrial Pollution Information System (Case Study of DKI Jakarta Province)

Aunur Rofiq Mulyarto[1], Kudang Boro Seminar[2] & Suprihatin[3]

[1]MIT Study Program, Bogor Agricultural University, SEAMEO BIOTROP Campus, Jl. Raya Tajur Km. 6 Bogor
[2]Faculty of Agricultural Technology, Bogor Agricultural University, Darmaga Campus, Bogor, Indonesia
[3]Faculty of Agricultural Technology, Bogor Agricultural University, Darmaga Campus, Bogor Indonesia

Abstract

The objective of this research is to develop an industrial pollution information system, particularly for the DKI Jakarta Province by using the object-oriented approach integrated with InPIS and spatial database methodology. The object oriented approach was chosen because of its advantage in modeling the real world problem of industrial pollution. The result of this research showed how the object oriented approach was very useful to describe the static and dynamic view of industrial pollution problem domain. The prototype of Industrial Pollution Information System (InPIS) was built as web application using the three tier architecture which separated presentation, business (problem domain) and data tier. The presentation tier was designed as a web page using the ASP.Net, while the business tier was derived from class design and built using Visual Basic.Net. The data tier was implemented using relational data model completed with distributed class behavior through the stored procedures mechanism. For the industrial pollution monitoring, despite of its accuracy, the Industrial Pollution Projection System (InPIS) can be used to estimate pollution load and pollution impact for several types of pollution parameters. The InPIS can combine it with the spatial data of DKI Jakarta administration boundary to produce the maps of pollution load distribution for the district level

Keywords: Industry, Pollution, Information System, Jakarta Province, Modelling, Remote Sensing, GIS

1. Introduction

1.1. Background

In recent years, issues of industrial pollution become important concern of the local and national community for the possibility of significant environmental degradation and public health hazard. In Indonesia, much of the industrial expansion has taken place without due regard to the environment, and has led to serious environmental degradation, particularly in Java where most of industries is located. Luken *et al.* (2002) stated that industrial emissions of CO_2 in 1997 were 1.25 metric tones per capita, while organic water pollutant emissions were 0.16 kg per day.

Jakarta, as the biggest city in Indonesia, also faces the problem as described above. There are various industries established in this city such as food and beverage, woods, chemicals, metals and textiles. All those industries potentially generate a wide variety of pollutants.

One of the important methods in industrial pollution monitoring, especially in estimating pollution load is the *Industrial Pollution Projection System* (InPIS), that was developed by the Infrastructure and Environment Team of the Development Research Group of the World Bank. According to Hettige *et al.* (1994) the InPIS is a modeling system, which combines data from industrial activity (such as production and employment) with data on pollution emissions to calculate *pollution intensity* factors.

In assessing and monitoring industrial pollution as well as other environmental problems, information system is one of the important factors that need to be considered. Information system casts a shadow over almost every aspect of the environment debate: research, monitoring, management and, decision-making and public involvement in decision-making.

This research will adopt an object-oriented approach because of its advantages. In case of industrial pollution problem, OO approach gives many benefits and possibility to build the system in a better way. By using the OO concepts technology, the object model can be built to imitate the actual condition in the real world (Fayad, 1999). For example, commonly, pollutant can be divided into air pollutant, water pollutant, solid waste pollutant and toxic pollutant. Each of them has their own characteristics, but the same basic characteristics such as, name and industrial source. With the concept of class, object and inheritance, we can build pollutant as a class and the types of pollutant as instance of class, which inherit the basic characteristics of pollutant class. This technique could also be applied to other entities.

1.2. Objective

The objective of this research is to develop the industrial pollution information system, particularly for the DKI Jakarta province by using the object-oriented approach integrated with InPIS and spatial database methodology

2. Material and Method

2.1. Time and Location

This research was conducted from March to July 2003 and covering the DKI Jakarta province.

2.2. Data Sources

Four kinds of data have been used in this research, *i.e.* statistical data of the large and medium industry, demographic data, supporting data for industrial pollution estimation, and DKI Jakarta province spatial data.

2.3. Required Tools

Softwares used in this research were Microsoft Windows 2000 Professional edition with Internet Information System (IIS) installed, Microsoft .NET Framework 1.1, Microsoft ASP.NET Web Matrix and Microsoft Visual Studio .NET, Microsoft SQL Server Desktop Engine (MSDE), Microsoft Visio, ESRI ArcView version 3.x, InovaGIS Geographic COM Objects. Hardware used to develop this system was PC with Pentium III class with minimum 500 MHz and 256 MB RAM.

2.4. Method

2.4.1. System development method

This research follows the Unified Process system development and focused on the three core process workflows namely requirement, analysis and design, and implementation with some adjustment. The scheme of process workflows it shown in Figure 1.

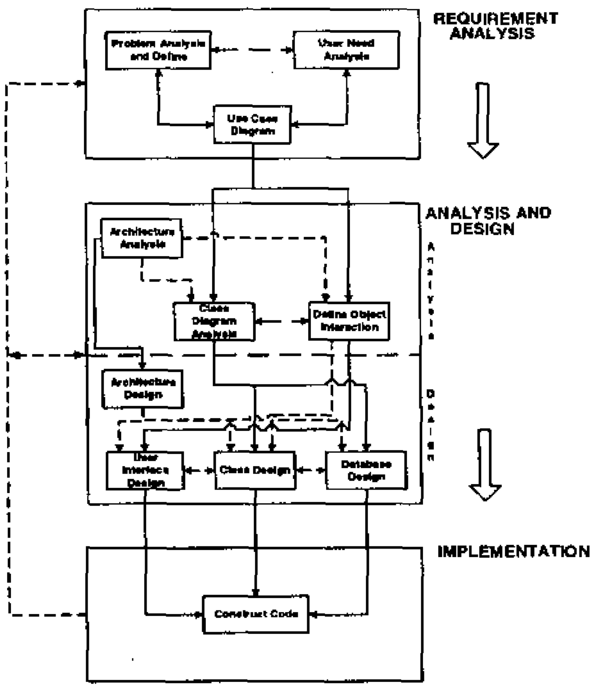

Figure 1. The scheme of process workflows.

The first stage of process work flows was requirement analysis, which has three activities, *i.e. Problem Analysis and Define*, *Users Needs Analysis*, and *Use Case Diagram Analysis*. Those activities were done in order to understand the problem of industrial pollution, to propose high-level solution, identify and analyze the user needs of information and to develop the functional requirement of the proposed system.

The second stage was Analysis and Design. The Analysis was conducted to model the architecture of the Industrial Pollution Information System ***(InPIS)***, identify and develop the class and its interaction. The result of these activities was used as a base for designing the system. There were four activities done in this part, included designing and developing the architecture model, designing the class base on the analysis, designing the user interface, and developing the database scheme.

The final stage was Implementation, which covered two main activities, *i.e.* translating the design stage to the code and testing the prototype of the ***InPIS***.

2.4.2. InPIS Method

The basic idea behind the InPIS is simple - for each sector or sub-sector, determine the appropriate ISIC code and find the corresponding emission factor. The pollution load is then estimated as:

$$PL = \frac{EF \cdot TEM}{10^6}$$

where, *PL* = Pollution load for a sector in tons/year;
EF = Emission factor in kg per thousand employees per year; and,
TEM = Total number of employees in that sector.

3. Result and Discussions

3.1. Requirement Analysis

Most sources of system requirement came from the documents related to industrial pollution problems; only a small portion was collected from interviews. The documents were used include:

- Act No. 23/1997 on Environmental Management
- Government Regulation No. 82/2001 on Water Quality Management and Wastewater Control.
- Government Regulation No. 41/1999 on Air Pollution Control.
- Ministry of Environment Decree No. 3/1998 on Wastewater Standard for Industrial Area.

- Ministry of Environment Decree No. 13/1995 on Emission Standard for Stationary Sources.
- Ministry of Environment Decree No. 51/1995 on Wastewater Standard for Industrial Activity
- Head of Bappedal Decree No. 205/1996 on Technical Guidelines to Control Stationary Sources Air Pollution.

3.1.1. Problem Analysis and Definition

From the initial overview, the proposed system (InPIS) is a system that can store, process and provide the information of industrial pollution particularly in DKI Jakarta province to several users, such as local community, government and industry. It was not intended to replace the whole process in industrial pollution problem, but rather be addressed to equip it. This system uses various data inputs, which come from different institutions. General overview of this system is shown in Figure 2.

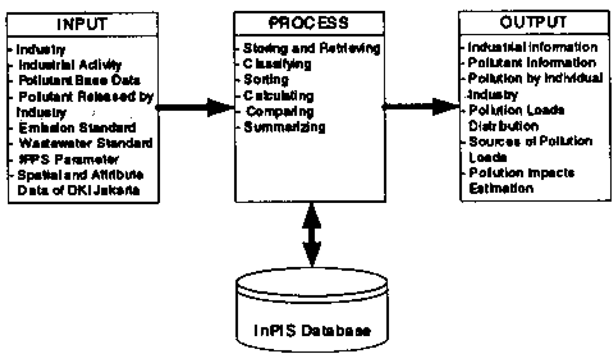

Figure 2. The general overview of the *InPIS*.

3.1.2. User Needs Analysis

Based on problem analysis, users of this proposed system and their needs could be identified. Generally, there were four main users, *i.e.*: government, industry, local community, and database administrator. Government could be divided into several users such as governor, mayor, district governance, government industrial division, municipal industrial division, and BAPEDALDA (The Environmental Impact Management Agency at province level). Table 1 shows the analysis result of user needs.

Table 1. User needs analysis result.

USER	INFORMATION
Governor	The summary report for the following information: - Industrial distribution - Pollution load distribution - Source of pollution loads - Pollutions impacts
Major	The summary or detailed report for the following information: - Industrial distribution - Pollution load distribution - Source of pollution loads - Pollutions impacts
District Governance	The summary or detailed report of their district area for the following information: - Industry and industrial activity - Pollution loads - Source of pollution - Pollution by individual industry - Pollution impact
Government Industrial Division	The detailed report of all area in DKI Jakarta province for the following information: - Industry and industrial activity - Pollution by individual industry
BAPEDALDA	The summary or detailed report of their district area for the following information: - Pollution load distribution - Source of pollution - Pollution by individual industry - Pollution impact
Industry	The detailed report for the following information: - Industry and industrial activity - Pollutant base information - Pollution by individual industry
Local Community	All information
Database Administrator	

3.1.3. Use-Case Diagram Analysis

Use cases are widely used mechanism to discover and record requirements, especially functional (Jacobson *et al.*, 1999). The first step in building use case diagram is to identify actors. An actor is an abstraction for an object outside the system interacting with the system. From user need analysis (**Table 1**), basically all the users have the same role that is using the system to get the information they need. Therefore, the users could be generalized into two groups of actors *i.e.* General User and Database Administrator is shown on Figure 3.

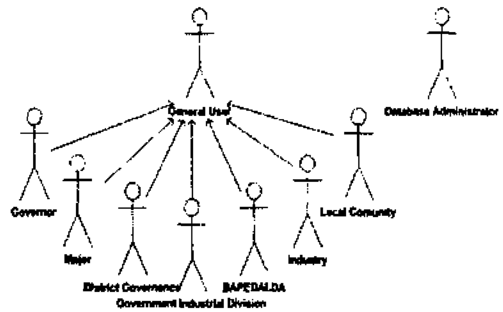

Figure 3. Actors generalization.

By using these actors and problem description several use case can be identified and then the use case diagram can be built. An example of use case diagram for the ***InPIS*** is shown in Figure 4.

Figure 4. The use case diagram.

3.2. Analysis and Design

3.2.1. Analysis

a. Architectural Analysis

To fulfil the user needs, the proposed system will be built as web application. The ***InPIS*** architecture can be divided into Three Tier (or Layer), namely Presentation Logic Tier, Business (Problem Domain) Rules/Logic Tier, and Data Tier. The presentation logic tier should accommodate how to display the map, chart and tabular view in the web page. The business tier included some rules in retrieving data from database and several data

manipulating process to transform data into information. It includes mathematical calculation, grouping, comparing and sorting. The business rules can be built as class library of proposed system. The data tier adopted the relational data model that was commonly used and supported by major database management softwares.

b. Class Diagram Analysis and Refinement

The class diagram is a central modeling technique that runs through nearly all object-oriented methods (Jacobson et al., 1999). This diagram describes the types of objects in the system and various kinds of *static relationships that exist* between them. There were three key steps to be performed including (1) identifying classes, (2) identifying the attributes, and (3) identifying the associations. By distinguishing between association, *aggregation and generalization*, it can be built class diagram close could be constructed to the problem domain. The result of these steps is shown in Figure 5.

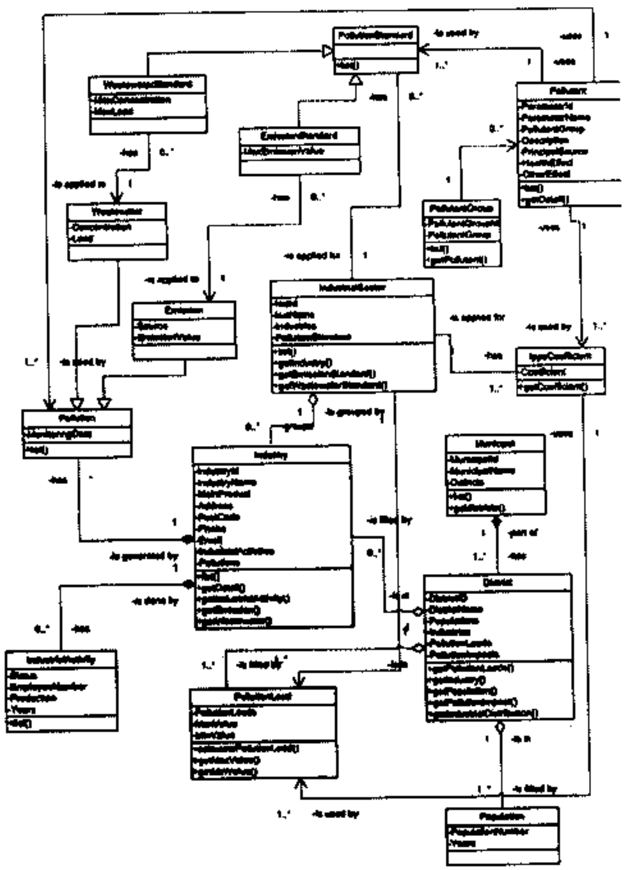

Figure 5. Class Diagram.

c. Define the Interaction among Object

To get the behavior of the system, the interaction among objects or a dynamic model of the system should be defined. Sequence diagram can be used to identify the interaction among objects. It also can be used to find the methods for each class defined in class diagram. Figure 6 shows example of sequence diagrams used in this system. This was built based on the use case of 'Get Pollutant Info'.

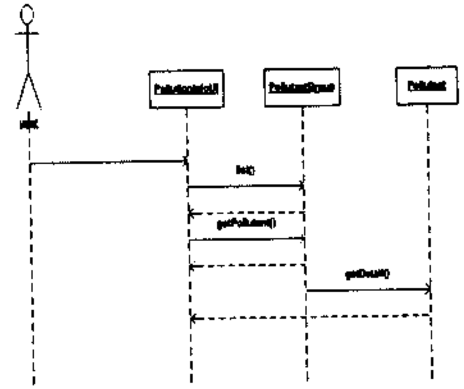

Figure 6. Sequence diagram of get pollutant info.

3.2.2. Design

a. Architecture Design

As mentioned in architecture analysis, this system will be developed as web application using three-tier architecture, which consists of presentation layer, business (problem domain) layer and data layer. In order to fulfill this architecture, the software and hardware needed should be determined first.

Software needed for developing this system includes operating system, development tool, database management system, and browser. Operating system used is Microsoft Windows 2000 Professional or Microsoft Windows XP Professional edition. The Internet Information Service (IIS) must be installed on this operating system. The IIS is a Web Server to simulate this system in intranet environment. Development tool used in this system is the software that has ability to produce ASP.NET web form. Microsoft Visual Studio.Net and Microsoft ASP.Net Web Matrix were used to generate ASP.Net web form. To run both of these softwares, Microsoft .Net Framework 1.1 should be installed properly in Microsoft Windows. Microsoft SQL Server was used to handle the ***InPIS*** database. Browser is used to test the output page of the ASP.Net web form. Internet Explorer version 5 and earlier was used because this browser has capability to display the ASP.Net web form.

Hardware used to develop InPIS web application should be able to support the development process. The minimum specification is: Personal Computer with processor Pentium III 500 MHz, RAM 256 MB, and SVGA screen with minimum resolution 800 x 600.

Based on this configuration and considering the three tier architecture, the detailed architecture of the InPIS was designed. Figure 7 shows the ***InPIS*** architecture.

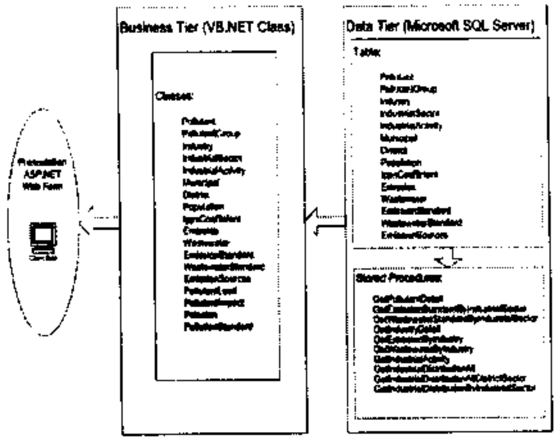

Figure 7. The ***InPIS*** architecture design.

b. Class Design

A class design represents an abstraction of one or several classes in the system's implementation; exactly corresponding to the implementation language. The proper size of the class and its objects depends on the programming language. Classes should map a particular phenomenon in the implementation language, and the classes should be structured so that the mapping results in good code.

There were three activities done in this stage, *i.e.* finding the new classes needed by the system and their relationships, refining the operations, and refining the attributes. The following figure 8 shows an example of class provided with attributes and operations.

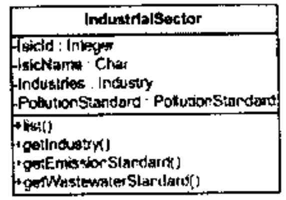

Figure 8. An example of refined class

c. User Interface Design

Designing user interface is the process to define an application visualization and interaction between user and system. Two major activities done in this step were organizing information and defining the elements.

Organizing Information

Based on the requirement analysis, the information produced by the ***InPIS*** can be organized into four groups as presented in Table 2.

Table 2. Group of information produced by the ***InPIS***.

No.	Group	Member
1	Industrial Info	- Information on individual industry - Information on industrial distribution - Information on industrial growth
2	Pollution Based Info	- Information on pollution parameter - Information on emission standard - Information on wastewater standard
3	Pollution Monitoring	- Information on emission monitoring - Information on wastewater monitoring
4	Pollution Estimation	- Information on pollution load distribution - Information on pollution contributor - Information on impact

Defining Elements

The user interface of the ***InPIS*** was designed using the common elements of website as described below.

- *Menus and subsites.* The InPIS user interface uses menu and a number of submenu pages to guide the users to track the information they want. This menu was built based on the information hierarchy presented in Table 2
- *Search features.* Search feature was needed by this system, especially to accommodate searching information for industry, industrial sector and area.
- *Table or Grid.* Many information produced by the ***InPIS*** will be easier to read if presented in tabular form.
- *Image and Chart.* This system used the image to display the map of pollution or industrial distribution. The map image will change depending on the user input. Chart element is used to display information in graph or chart form.

d. Database Design

From the architecture design section, it can be seen that the data tier of the ***InPIS*** was implemented using relational database approach (using Microsoft SQL Server). This becomes the main consideration in database design because relational database approach is not entirely compatible to object oriented system. Thus, in database design; object oriented system should be transformed to relational database management system or called as mapping objects to relational database.

Map persistent design classes

This activity started by identifying the persistent classes (objects) and their attributes that should be stored in relational database. From class design section, all classes can be identified, except PollutionLoad, PollutionImpacts, and persistent Map.

To map persistent classes to relational database, it should be consider how to transform the class relationship including, association, aggregation and generalization. Below is the explanation about the process.

(i) Mapping association and aggregation between persistent classes

Association or aggregation between two persistent classes can be realized as foreign keys to the associate object. The example of mapping association and aggregation is shown in Figure 9 and 10.

Figure 9 shows how to map association between ***IndustrialSector*** and ***IppsCoefficient***. The ***IppsCoefficient*** is a class that needs ***IndustrialSector*** class exist. This association can be represented by the ***IsicId*** attribute in ***IppsCoefficient*** table as a foreign key where this attribute is a primary key in ***IndustrialSector*** table.

Figure 10 shows how to map aggregation between ***Municipal*** and ***District***. Again, the foreign key method was used to map the aggregation. The ***MunicipalId*** is a primary key in ***Municipal*** table and plays a role as foreign key in ***District*** table.

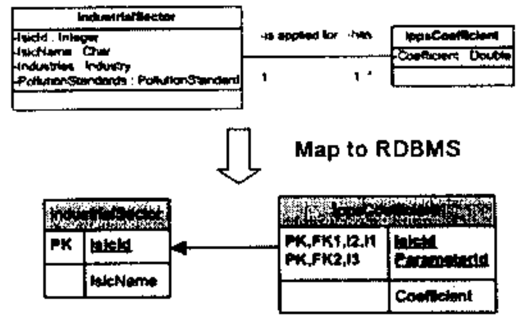

Figure 9. Example of mapping association.

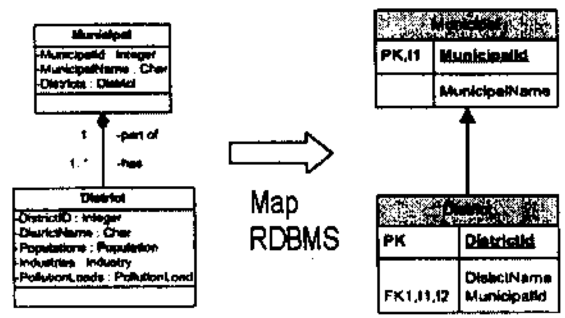

Figure 10. Example of mapping aggregation.

(ii) Mapping generalization

A relational data model does not support modeling inheritance in a direct way. There are two options that can be used: first, using separate tables to model the super class and subclass (*normalized* data), and the second, duplicate all inherited associations and attributes as separate columns in the subclass table (*denormalized* data). In this system the second option was used. The reason is to follow the information hierarchy used in this system (see Table 2).

Figure 11 shows how to map generalization between ***Pollution, Emission*** and ***Wastewater. Pollution*** is a super class, while ***Emission*** and ***Wastewater*** are subclass. The attributes of Pollution that are inherited by subclass are ***IndustryId, Parameter Id*** and ***MonitoringDate.*** These attributes were used by each subclass (Emission and Wastewater).

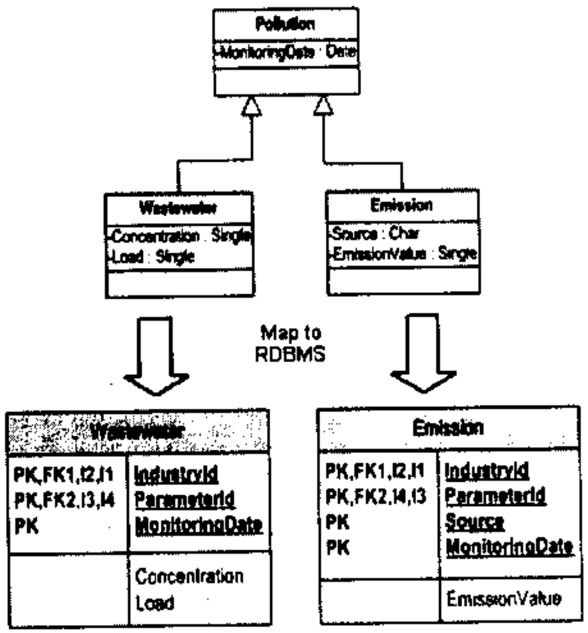

Figure 11. Example of mapping generalization.

By using these mapping processes, the structure of ***InPIS*** database could be constructed.

The ***InPIS*** also used the spatial data of DKI Jakarta province, especially District administration boundary data. To handle this data and link to non-spatial InPIS database, the attribute of DistrictId in spatial data was changed following the rule in District table. Figure 12 shows the relationships between attribute of spatial data and ***District*** table.

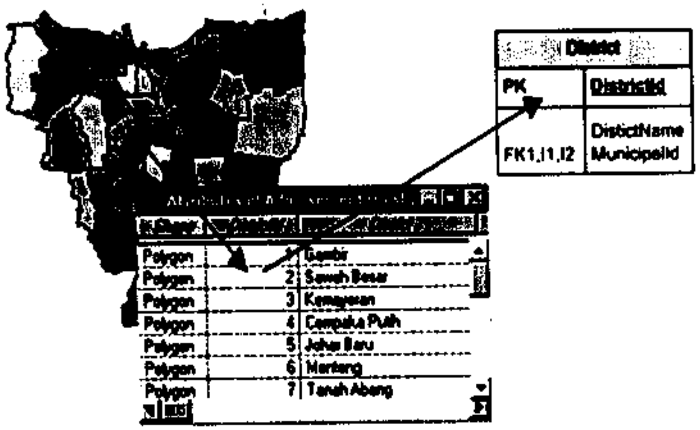

Figure 12. The relationship between spatial and non spatial data.

Distribution class behavior to the database

After the mapping process was done, it should be considered how to distribute behaviors and implement to the database. It can be done by stored procedures mechanism. A stored procedure is executable code, which runs under the RDBMS. Code runs at the server, where the data is, rather than at client with corresponding data exchange overhead. Microsoft SQL Server as RDBMS has the capability to handle the store procedure mechanism.

3.3. Implementation

The final stage of this research deals with activity of coding the elements described in the design stage. Mainly, this stage focuses on how to translate class diagram, use case diagram, and sequence diagram into the programming language code.

3.3.1. Constructing Code

Each class (in design) is implemented by coding it in a programming language, or by using a pre-existing component. Exactly each class (in design) depends on the programming language. This system was developed using ASP.Net, which is built using Visual Basic.Net (VB.Net). VB.Net the new version of Visual Basic has completely different structure than older version of Visual Basic. VB.Net supports major concepts of object-orientation such as, true inheritance, polymorphism (overloading and overriding), and abstraction.

3.3.2. Testing

The prototype of the ***InPIS*** was tested using only one personal computer that was setup both as local server and workstation so that the ***InPIS*** can be accessed. Using Internet Information Service that was installed in operating system can do it. The following figures show the screen shots of the ***InPIS*** prototype.

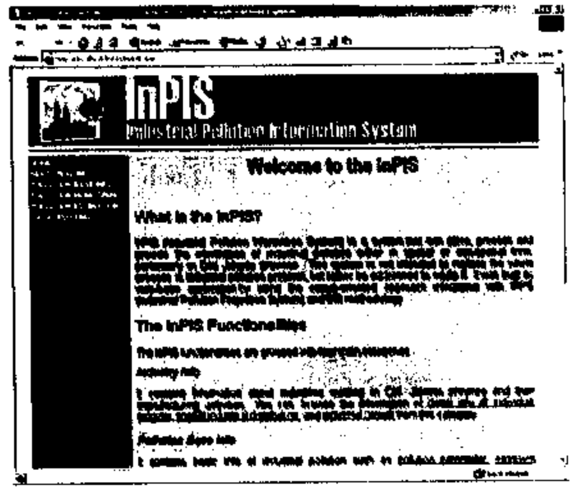

Figure 13. The main page of the InPIS.

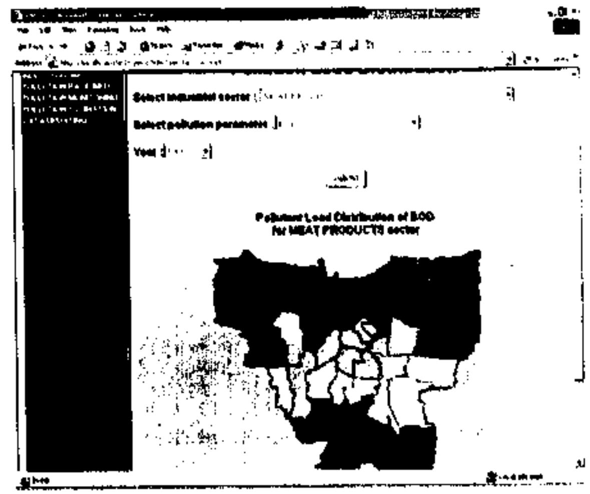

Figure 14. Example of Pollution Load Distribution generated by the ***InPIS*** (The darker colors indicated that the pollution loads are higher).

4. Conclusions and Recommendations

4.1. Conclusions

- This research demonstrates how the object oriented approach can be used in developing a prototype of Industrial Pollution Information System (InPIS). It shows how it is very useful to describe the static and dynamic view of problem domain. Class and object modeling as the core heart of object oriented approach, is very helpful to understand the real world problem of Industrial Pollution Information System.

- The prototype of InPIS was built as web application using the three tier architecture which separates presentation, business (problem domain) and data tier. The presentation tier was designed as a web page using the ASP.Net while the business tier was derived from class design and built using Visual Basic.Net. The data tier was implemented using relational data model completed with distributed class behavior through the stored procedures mechanism.

- Despite of its accuray, the Industrial Pollution Projection System (InPIS) can be used to estimate pollution load and pollution impact for several types of pollution parameters. The InPIS can combine it with the spatial data of DKI Jakarta administration boundary to produce the maps of pollution load distribution for the district level.

4.2. Recommendations

- Due to the time limit and other resources, the activities in gathering the requirement and users' needs could not be fully implemented. Most of the information was collected from literatures and government regulations on environment. It causes some incompatibility between the results of requirement analysis of this research with the real world problem. To refine the InPIS, the requirement analysis should be done more completely so that it can capture all the users' need and the real world problem.

- This research only used the number of employment data to estimate the pollution load for each industrial sector. For better estimation and for the purpose of comparison, it will be more valuable if this system can use other type data sources as the estimation base. The production volume per year is one of data sources that could be considered for being used. The InPIS method also provides the coefficient for this data source type.

- One of the important functions that does not include in the InPIS is how the system can give permission for each industry to update themselves their activities or their pollution directly to the InPIS remote database. For future research in this field this function should be considered.

5. References

Braude, E. 2003. *Software Design: From Programming to Architecture.* John Wiley & Sons, Inc.

Champeaux, D., D. Lea, and P. Faure. 1993. *Object-Oriented System Development.* Addison-Wesley.

Fayad, M. 1999. *Buiding Application Frameworks: Object Oriented Foundations of Framework Design.* John Wiley & Sons, Inc.

Haklay, M. 1999. *From Environmental Information Systems to Environmental Informatics: Evolution and Meaning.* Centre for Advance Spatial Analysis, University College London.

Hettige, H., Martin, P., Singh, M. and Wheeler, D. 1994. *Industrial Pollution Projection System (InPIS).* Working Paper #1431, The World Bank, Washington D.C.

Luken, R., J. Alvarez and P. Hesp. 2002. *Developing Country's Industrial Source Book.* United Nation Industrial Development Organization, Vienna.

O'Brien, J.A. 1999. *Management Information Systems: Managing Information Technology in Internetworked Enterprise.* The McGraw-Hill Companies, Inc.

Jacobson, I., J. Rumbaugh, and G. Booch. 1999. *The Unified Software Development Process.* Addison-Wesley Publishing Co.