APPLICATION FOR MOBILE AGENT TO ACCESS MYSQL

Bambang Sugiantoro[1,2] , Retantyo Wardoyo[1], Sri Hartati[1], Jazi Eko Istiyanto[1]
[1]Dept. Computer Sciences and Electronics, Gadjah Mada University,Yogyakarta, Indonesia.
bambang.s@uin-suka.ac.id
rw@ugm.ac.id
shartati@ugm.ac.id
jazi@ugm.ac.id
[2]Dept. of Informatics Engineering, State Islamic University Sunan Kalijaga, Yogyakarta, Indonesia.

***Abstract*—Availability of information is increasing rapidly, the method for encoding and storage of information also increased. The number of sources of information brings several problems, among them about how to combine the distributed data storage and different. Information on an organization or company is usually stored in separate locations and different formats. When there is an increase in storage capacity and the amount of information search costs, the company faced with an abundance amount of data. System development methodology used is Rapid Application Development (RAD) with a development framework *Guidlines for Rapid Application Engineering* (GRAPPLE). The software used in building this application is Borland Jbuilder 9. Rational Rose 2000 is used for the analysis and design using Unified Modeling Language (UML). Mysql server is a type of database to be accessed on each operating system. Operating system used in this study is Windows XP and Linux Fedora Core. This application was developed with aglets, which is one of software for building mobile agent applications.**

Keywords: Mobile Agent, mysql

I. INTRODUCTION

Availability of information is increasing rapidly, the method for encoding and storage of information also increased. The number of sources of information brings several problems, including how to combine the distributed data storage and different. Information on an organization or company is usually stored in separate locations and different formats. When there is an increase in storage capacity and the amount of information search costs, the company faced with an abundance amount of data[1]

Distributed database is a database where data is placed in several locations, but to implement a specific mechanism to make it into one unified database. A distributed database system may be constructed only in a computer network system. In contrast to the centralized data base for which data is placed in some locations but not interconnected.[2][3][4][5]

Distributed database access is a process to mix and match, query, manipulate, and combine data in a distributed database. Access the database will display the user's desired query results. Access databases do not perform the tracking of changes to the database on each host.[6]

II. THEORY

A. ASDK

IBM aglets Workbench (aglets) was developed by Danny B. Langedan Mitsuru Oshima from IBM Tokyo Research Laboratory in 1996. Aglets are Java objects that can move from one host to another host in a network Aglet who was working in a host can stop the execution, go to another host and then start execution again. When the aglet moves, Aglet bring the program code as well as state of all objects that carry it. A security mechanism that will secure the built-in host from untrusted Aglet [7],[8].

Aglet term is a combination of the words agent and applet. The difference with the applet is aglet also brings with it state, and has itinerary (travel plan). ASDK (aglets Software Development Kit) is a software package that is used for writing mobile agent applications. ASDK using java language and can be obtained for free from the internet in the form of program code or bytecode and binary files that have been compiled.

Tahiti is an application program that works as a server Aglet. Tahiti applications are packaged in an ASDK. Some servers (tahiti) can be executed in a computer by assigning a different port number.Tahiti provides a user interface for monitoring, creation, delivery, and destruction of an aglet and to assign access rights (access privelege) for the agent.

Aglet allows automation of many processes that were previously done manually by the user, this technology could transform the way users interact with the internet.

Aglet underlying structure as follows: Aglet, a java object that can be visited from an Aglet Aglet host to another host, Proxy, a representation of the aglet that protect it from direct access to a public method. Provide location transparency so that the unknown true location of an Aglet, Context, Aglet place is run, the stationary object capable of processing and execution settings Aglet, Message, object exchange between Aglet, Future Reply, asynchronous message delivery, Indetifier, an identity owned by each Aglet is unique and globally.

Aglet can perform basic operations as described: Creation, the creation of an Aglet. Creation happens in context. The new Aglet given an identifier, inserted into the context and initialized. Aglet begin execution immediately after a successful initialization, cloning, copying, aglet Cloning produces a derivative (copy) that is almost identical to the original Aglet in the same context. The difference lies only in the given identifier and the new Aglet execution starts from the initial cloning results (restart).

Note that the thread of execution is not to clone, dispatching, an Aglet transfer from one context to another context. Dispatching will move aglet context of an ongoing, into the context destination and then start the beginning of execution, retraction, a process to "pull" Aglet context of the ongoing context and into a retraction request, Activation, the ability to return to the Aglet context, Deactivation, the ability to temporarily suspend the execution path Aglet Aglet and save state in secondary storage, disposal, a process for stopping the execution of ongoing and Aglet Aglet issue of the ongoing context, Messaging, between Aglet include sending, receiving and handling both synchronous and asynchronous message .

B. JDBC

JDBC (Java Database Connectivity) is a trademark of Javasoft the API (Application Programming Interface) used to execute SQL on the database access. The database can be accessed by JDBC is not limited to any relational database. With a particular development, we can access data stored on Directory Service data.

.

JDBC provides a standard database communication interface that allows access to various databases. This allows the application programmer and tool makers and tool to develop database applications without worrying about the database that is used on the backend. JDBC has received support from a number of database vendors including Oracle, Sybase, Informix, Interbase, DB2, and Microsoft.

JDBC is to keep the code generated by Java does not depend on the specifics of each database vendor. An application can access any data source and can run on any platform that has a JVM (Java Virtual Machine). Someone just wrote a program that uses the JDBC API, and the program can send a SQL statement or other statement to a particular data source.

III. DESIGN

Rapid Application Development methodology which was developed by a development framework Grapple (Guidelines for Rapid Application Engineering) is the planning requirements, analysis and design. The planning stages and needs analysis implicitly addressed in the analysis. In the analysis phase will be used three UML diagrams are class diagrams, use case diagrams and collaboration diagrams, whereas diagram used in the design phase and prototype interface activity. The use of four UML diagrams is considered sufficient to explain the system to be created

The first stage is to identify the problems of distributed database access system to be created. The second stage is to analyze user needs, which meant that the system constructed in accordance with the wishes of the user. The third stage is information access the database as the end result of accessing the database is done. The fourth stage is the determination of system boundaries as a prerequisite for this application to run perfectly.

Distributed database system is one type of database systems that are popular in the world. These systems utilize computer networks to communicate. Supporting applications also have been circulating in the market diataranya, Microsoft SQL Server, MySQL, PostgreSQL, Oracle and others. Each database is controlled (controlled) by a DBMS-independent, which is responsible for maintaining (maintains) the integrity of the database. These databases will work properly if all environments (environtment) running the same DBMS vendor. It would be very difficult to install (installing) DBMS in hardware, different operating systems, or use a DBMS with a different vendor.

Companies and large institutions usually own a few DBMS in the system database. DBMS-DBMS is installed in different operating systems as well. Handling database administrator can do with database access from computer to computer, but it will take a lot of time if used for large-scale network database. Operational costs will increase because the company needed more labor. Seeing such circumstances, it is deemed necessary for building mobile agent applications for distributed database access is the main target is a distributed database system with different operating systems.

Agent for the mobile application is a distributed database access, provide service and convenience for users, diataranya: Some database servers in different locations can be accessed through an application, Allows administrator tasks in

Based on our works presented on ICID2011 [1]
Identify applicable sponsor/s here. *(sponsors)*

a distributed database access. In accessing the database, the administrator does not need to be on the computer where the database is located, able to run on Windows and Linux operating systems pre-installed Java Virtual Machine (JRE / JDK) and Aglet SDK, applications in terms of economic cost, Easy to install, Setting and configuration are not complex, requires only a few resources to access multiple database servers on the network computers.

Access of database information

Access the database information is taken from the hosts after the database manipulation is complete. The report will contain the results of database access query results from each database on the destination host. A class is required to address in order to report the information displayed in accordance with the desired.

Database access is done by sending an agent over the network from one host to the destination host in the network computer. In order for the delivery agent does not spend much time and network bandwidth, the agent must be small.

It has been known IP address or name and identity hostnya can be accessed via the network, has been installed JDK (Java Development Kit) and SDK, or Tahiti Aglet server running on a port that has been specified, the JDBC driver has been installed in accordance with a database that is installed in each host, using the Windows operating system or Linux 9x/NT/2000/XP.

Access to a database sequence of steps can be seen the use case diagram. On each host is running Tahiti server that is ready to accept an incoming aglet on a specified port. Database access steps and making slave : Tahiti Server makes DBApplication then displays the user interface, the Administrator to determine what hosts are to be accessed, only to hosts that are active only do mobile agent delivery, configuration done on the agent to be sent include the delivery mode, JDBC configuration and the type of query is will be held on a database, Slave or agent made according to the configuration and travel plans that have been determined, in each host is visited, the agent will perform a database access query execution on each host in accordance with a predetermined configuration, the Agent took his journey plans (itinerary) and will move to the next host in accordance with the itinerary.

After all the hosts in the itinerary a visit, the agent returned to the sender with a host of objects that have been collected Report.

In this analysis phase of the five artifacts of action, namely: Identification of system requirements, detailed use cases, class diagrams Completing, Defining the interactions between objects.

The results of the two actions and elaborate identification system needs is a use case diagram use case. Complement the action of a class diagram that will produce a class diagram is discussed further. The next action that defines the interactions between objects that will produce a Collaborative diagram .

Benefits to be achieved through modeling in the analysis phase areas follows: Model analysis helps to visualize the system as desired user, Model analysis can reveal the dynamic aspects of the application, the model analysis provides a guide in building an application either via the forward-engineering and reverse-engineering.

Software development can be identified two actors, namely the human form and the Tahiti server software. The actor is someone or something outside the system but interact with the system . Use case diagram padamemiliki the two actors and the Tahiti server administrator. Users can interact with five use cases in software systems in turn. Five of the use case is the use case Slave Configuration Agent, use case-host Specifies the host that is accessible, Define Query, use case Create Aglet slave, Sending agent use case, and use case a report. Actor Tahiti Server is used to create DBApplication with a function to send and receive a pre-configured Slave Aglet. User interaction with the program more made through the main menu on the use case Display User Interface. After the delivery of agents, each agent will perform tasks that previously configured on each host a visit.

In this application used 2 Agent: namely DBApplication and AgletSlave which is derived from the Aglet class. DBApplication served as a stationary agent, which runs on a computer server by the administrator.

DBApplication as the main class in the server user interface DBAgent menginstansiasi. This class is associated with several classes, each of which has a specific function.

AgletSlave served as a mobile agent can perform through the dispatch network to the destination hosts. The information is taken on each network is stored in the object of class Report.

Collaboration diagram analysis on the agency making the use case diagram, class diagram is synchronized with, resulting in the manufacture of an agent collaboration diagram

Once the object of AgletSlave created, the administrator can send it to the network.

Object of class AgletSlave then move from one host to another host in accordance with a predetermined itinerary. Database

access steps are performed by agents with the collaboration diagram
When agents arrived at the host destination, Tahiti Server will call the method run () method of containing the agent doJob () to retrieve the necessary data. The data captured is stored in objects of class Report as a material for making the report.

Unexpected things often happen, for example, be detected when the host is not running, the host will be skipped. If an error occurs or the detection is complete, then the agent will return to host origin with a Report object each host. On the server, calls the method DBApplication buatLaporan () on the object DBAgent. Report object will be converted into an object results. Object stores the results of database access performed on each host. These objects are stored into a vector and can be saved into a text file. The resulting report later in the form of common query execution results generated by each DBMS on each host visited.

Explain about the making of a report by DBApplication. Collaboration Diagram Creating Reports This stage is based on the results of the analysis phase. In this stage there are two actions, namely: Develop and refine the object diagram The resulting product is the object diagram and activity diagram. In this study addressed only the product of any activity diagram. This diagram is a simplification of what is happening during an ongoing process in the system.

Interface design prototypes, this action will result in the form of three prototypes of products that interface prototypes DBAgent form interface, dlgJDBC form, form and form About. In the form DBAgent there are three panels showing the interface with different functions. These panels are implemented with jTabPane consisting of panel results, Message panel and information panel.
Activity diagram is a product of the action to develop and refine the object diagram in the design stage. This diagram describes the activities of users that happen to the system being created. Shipping agent activity diagram

an activity diagram agent delivery process. visible activity begins when the user specify the hosts where the database resides. Then after that followed by the determination of the activity of the JDBC settings. There are two activities that occur after the determination of the setting Setting JDBC JDBC JDBC Settings Fail and Succeed. If the activity is happening is setting JDBC fails then the setting will be repeated until the condition is working. And if the condition is so far it will proceed with the activity determination of command Query. Any activity that occurs will be coupled configuration with the storage configuration information to the class. Then place the activity delivery agents to hosts that have been determined based on the previous configuration.

Software to be constructed consists of three interfaces (user interface) is the interface DBAgent, dlgJDBC, and interfaces About. DBAgent interface is divided into three sections: The panel, the panel Message, and the information panel. The structure of the interface in this software is DBAgent as the main page and then dlgJDBC About the next page or below. Another interface is part of the interface consisting of a panel DBAgent Results, Message panel, and panel information.

This interface is the main page that is used to call all the functions in this application. This page first appeared when the application runs. Form the interface. DBAgent an interface for the administrator to access the interface from the other classes through the main menu. The front includes a query input by the input function as a means of query to be executed on each database. Part of the panel displays the results of the query function in the form of tables of queries that have been implemented in the database.

Panel Message function to display information about the number of lines created from the query execution and error information is obtained in case of errors or failures in accessing the database. Execution button is used to transmit all AgentSlave and start accessing the database. Save button to store the query command that has been configured user in each database on the destination host.

Information panels are used to display status information of the trip AgletSlave. Status shown is the start AgletSlave sent, AgletSlave presence information on the destination host and the error in the event of a failure in the delivery AgletSlave. On this panel there are two buttons that save button to store the resulting information into a file and the delete key to clear the textarea display of information. Forms the interface looks, Prototype interface dlgJDBC.

This interface is used or executed when the user adds a host address. In this case the user can create a new configuration used to access the database. To create a new configuration of the user can select button added to the DBAgent. Then be shown a dialog box that is used to make new connections and to make a connection the user must enter the JDBC configuration.

IV. IMPLEMENTATION

The software used to build this system are: Borland JBuilder 9 is the main software used to compile the program code and user interface making the application. Mysql Server is a type of database server that is accessible applications. Aglets-2.0.2 is a software package used to create this mobile agent software. Mysql-connector-java-3.0.14 JDBC driver that is used as the Java programming language to access the MySQL

Server. Microsoft Windows XP Professional Edition SP1 and Linux Fedora Core 2 as the operating system during application development.

Hardware used in building applications has specs: XP 2400 + Processor Atlon, Memory DDRAM 512 MB PC 3200, VGA ATI Radeon 9200 SE 64 MB, Hard Drive 40 GB 7200 rpm, Telebit 10/100 Mbps. In the test phase using a local computer network with 4 computers connected to the swich a speed of 10 Mbps.

The discussion includes database access mechanism by AgletSlave, handling messages between Aglet and creating reports. Not all the methods discussed here, only the method - a method considered to be essential. The discussion begins with an explanation of the logic followed by implementation in program code.

Database access by AgletSlave, while DBApplication act as a liaison between the administrator and the Slave on the remote host. DBApplication can call existing methods for the DBAgent and vice versa. This is done by referencing the object of DBApplication as input parameter to the constructor DBAgent. Reference of the object is also present in DBAgent DBApplication.
As an object derived from class DBApplication, method is called by the Tahiti Server is created when DBApplication onCreation (Object obj). This method is similar to a constructor in the class - Java class. Making objects from DBAgent done by calling the method buatFrame () which will automatically call the constructor. Program code can be viewed at the following program modules:

```
DBAgent frame;

private void buatFrame(){
  frame=new DBAgent(this);
  info.tampilkanDiTengah(frame);
}

public void onCreation(Object obj) {
     setText("AgletMaster telah diciptakan");
     buatFrame();
   }
```

Create and send to the destination host AgletSlave done by calling the send method (Vector vTujuan) that will automatically call the method buatSlave () on DBApplication. AgletSlave Slave.create made by calling the method with the 6 parameter is the URL from the code, the name of the slave class, where aglet context is created, slave aglet, Vector purposes, and objects to be sent as a result. Each method send is executed, the JDBC configuration automatically each host to be visited are sent to AgletSlave.Code can be seen in the module program.

```
public void buatSlave(Vector tujuan) {
        try{
      AgletSlave.tpass=info.getTpass();
      AgletSlave.tuser=info.getTuser();
      AgletSlave.vdburl=info.getvDBurl();
      AgletSlave.vjenisdriver =
info.getJenisdriver();

AgletSlave.vquerystmt=info.getvQueryStmt()
;

AgletSlave.execute=info.getTipeExecute();
      Slave.create(null, namaSlave,
getAgletContext(),
                              this, tujuan,
new Vector());
      }catch(Exception ex){
      }
  }
    public void kirim(Vector vtujuan) {
      setText("Buat Slave dan kirimkan");
      AgletSlave.vTujuan = vtujuan;
      buatSlave(vtujuan);
    }
```

Any host with a different type of database accessed with different configurations. This configuration parameter in the form of user name, password, address database, and query the database driver type statement. Each configuration is stored in vector form which is dynamic because the value of the database access parameters can be changed at any AgentSlave will be sent. AgletSlave will use this setup to access each database based on a predetermined itinerary. Accessing a database on the destination host starts by calling the method aksesDB () method call arrives automatically Class.forName (). Parameter user name, password, address database and the query statement is used to establish a connection to the database.

Accessing the database can be either changing the structure of the database or displaying the information available in the database server. Changing the structure of the database is done using executeQuery command. The results of this command will be stored in a class Report which will then be sent to DBApplication. Program code can be viewed on the module

```
protected void aksesDB(){
   setText("accessing DB");

 try{

if((_vTujuan.elementAt(0).toString()+"/").equals(lok
asi)){
```

```
         System.err.println("comparing local address
and dest
          address");
      try {
        Class.forName(com.mysql.Driver);
        try {
      conn =

DriverManager.getConnection((String)_vdburl.elementA
t(0),
(String)_tuser.elementAt(0),
(String)_tpass.elementAt(0));
      if(conn!=null){
         Stmt = conn.createStatement();

Stmt.executeQuery((String)_vquerystmt.elementAt(0));

if(_execute.elementAt(0).equals("execute")){

Stmt.execute((String)_vquerystmt.elementAt(0));
report.setErrorMessage("Database changed");
             }
 if(_execute.elementAt(0).equals("executeUpdate")){

Stmt.executeUpdate((String)_vquerystmt.elementAt(0))
;
       report.setErrorMessage("Query OK,0 rows
affected");}

if(_execute.elementAt(0).equals("executeQuery")){

_RS=Stmt.executeQuery((String)_vquerystmt.elementAt(
0));
        try {
               if(_RS != null)
               {
                 rsmd = _RS.getMetaData();
                 if(rsmd == null)
                 {
                   throw (new
Exception("SerializableResultSet:
ResultSetMetaData is null"));
                 }NumberOfColumns  =
rsmd.getColumnCount();
                                }
  else
                                {
  NumberOfColumns = 0;
                 }
```

The process of displaying information from the database is done with executeQuery command execution. Paramenter received by the command is a query statement SELECT and SHOW. Especially for the SELECT query, the result of command execution is stored in the variable Vector of Vector. Use of this variable is intended to keep each table resulting in a vector. The results of the query will be stored in a vector containing the vector column of the table is accessed. Rows of a table stored in the vector rowVector. Mobile Agent on each host did his duty to go, save the results in Report class and then send all the information in the Report to DBApplication class for processing. Data transmission over the network or on the java object can be done if it implements the Serializable interface (). This interface allows data to be sent in a stream object. At this destination host object will be formed again become the object of origin. Objects are not serializable can be sent if declared to be transient.

Configuring JDBC in each class by calling the host tujuandilakukan dlgJDBC. At this dlgJDBC class user to enter data-JDBC configuration data such as username, password, address database, the type of database and database port. Each storage configuration is done by calling the method btSave__actionPerformed (). Configuration data is stored in the class information.

Data exchange mechanism between slave and slave made by sending a message. A message can be sent from the Aglet Aglet A to B if A has AgletProxy of Aglet Aglet B and vice versa. Proxy is a means to communicate with an Aglet. In the proxy, call a method sendMassage (Massage m) to send m to Aglet Massage appointed by proxy. Each receipt of the message, will call the method handleMassage Aglet (massage msg).

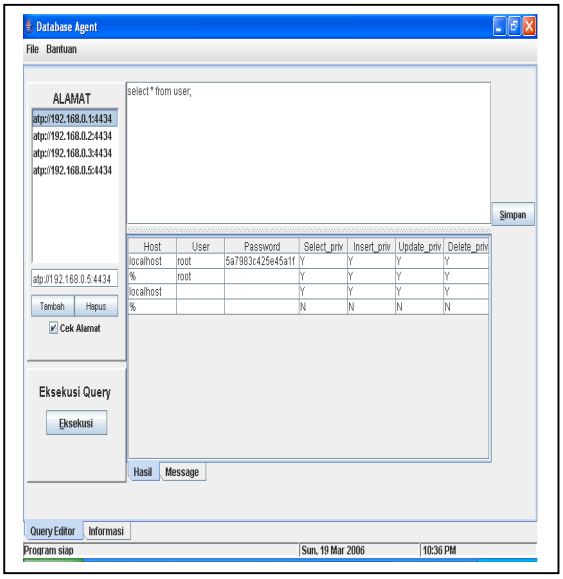

Figure 4.1 Query Exeecution by Mobile agent

Massage is a String object that contains the type of messages to be delivered and the object passed as argument. Interactive message delivery between AgletMaster and AgletCloned through AgletMassager interface. Types of messages retrieved using the method getKind () while the object is passed as an argument made by calling the method getArg (). Objects that are taken should not be directly used, but must be done prior to casting the appropriate data type

The report is based on data taken by the slave on the destination host. AgletSlave will return to host origin after completion of the task with a Vector object containing the Report of the individual - each host. This

vector will be stored in the class information to be processed into reports DBAgent class.

Tests performed on 5 computers with the provisions of a local dijaringan computer as a server and the other four as client computers. The fifth computer is switched on and able to respond to ping commands. The operating system used by the host target is Microsoft Windows XP Professional Service Pack 1, and Linux Fedora Core 2. The database server that is accessible MySQL server is installed on each operating system. The database has been configured and can be accessed via the dos command prompt. Testing is divided into two sections based on the query SELECT and CREATE query.

Slave-making is done through DBAgent. Button added to the display appears as DBAgent. This step is done how many times the adjusted number of databases to be accessed. The configuration of each database tailored to the configuration done on the client server Mysql in general. Besides the destination host can be left in the form localhost to simplify configuring JDBC drivers. This can be done because the slave can access the database locally. Each slave address transmitted in accordance with ATP (Agent Transfer Protocol) and move from and to the purpose of using mobile agent is a special environment. On arrival at the destination of all slave processes will be done in the form of a local process. a view to configure the type of query to be executed on the database server.

Type of query that can be executed in this application may include CREATE, DROP, DELETE, INSERT, ALTER, UPDATE, SELECT and SHOW. Each input query is done, click the save button to save the configuration to the object information. This configuration can be changed each time the slave will be sent. The execution button is used to transmit all the slave to the destination. Display of information on the destination host AgletSlave activity. The information displayed in the form of start time, the presence in the host destination AgletSlave, completion time and error information if something goes wrong in shipping AgletSlave.

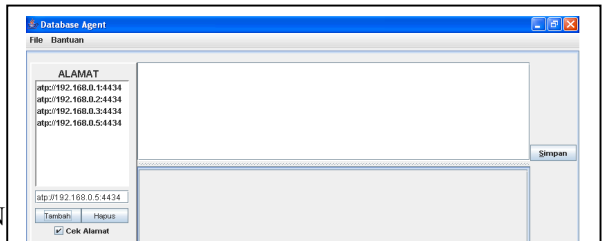

Figure 4.2 Report Mobile Agent

The results of select query execution. The results shown in table form. Display tables will change according to the selected database in the list view. a message display panel tab that displays information about the success of query execution. State error, the information line in the table results, the successful execution of the query are displayed in a tab panel.

CONCLUSION

Conclusions obtained from the analysis, design and implementation in the previous chapters could have been designed and constructed a mobile agent application for distributed database access. This application can access data distributed to multiple database servers on the same type of vendors would do next komputer.penelitian network analysis and design of mobile agent for IDS.

REFERENCES

[1] Sugiantoro, B., Wardoyo, R., Hartati, S., and Istiyanto, J.E., Mobile Agent to Perform Query on Multiple Database Server for Security, Proc. of International Conference on Informatics for Development,(ICID '11) UIN Sunan Kalijaga, Nov. 2011,Yogyakarta, pp. C4102-105.

[2] Sugiantoro, B., Wardoyo, R., Hartati, S., and Istiyanto, J.E., database Access framework Using Distributed Mobile Agent Technology, Proc. of International Conference on Informatics for Development, (ICT4M 2010), Dec 2010, Jakarta, pp. E1-3.

[3] Sugiantoro, B., and Ahmad, A., Agent Pengambil Informasi Menggunakan Software Aglets, Jurnal Teknomatika STMIK J.A. Yani, Vol 3 No 2 Januari 2011, ISSN 1979-7656, Yogyakarta, pp. 37-50.

[4] Sugiantoro, B., and Wardoyo, R, Akses Database Server Menggunakan Teknologi Agent, Proc. Seminar Nasional Akakom, (SRITI 2010), Aug 2010,Yogyakarta, pp. 121-126.

[5] Sugiantoro, B., and Istiyanto, J.E., Analisa sistem Keamanan IDS, Firewall, Database System dan Monitoring Menggunakan Agent Bergerak, Proc. Seminar Nasional UPN, (SEMNASIF 2010), Mei 2010,Yogyakarta, pp. C21-29.

[6] Sugiantoro, B., and Istiyanto, J.E., Analisa Keamanan database Server Menggunakan Teknologi Virtual Private Database dan Notifikasi Database Server Menggunakan Agent Bergerak, Proc. Seminar Nasional UPN, (SEMNASIF 2010), Mei 2010,Yogyakarta, pp. C30-37.

[7] Lange, D.B, 1997, *Java Aglet Application Programming Interface* (J-AAPI) *White Paper – Draft 2,* <http://www.trl.ibm.co.jp/aglets/api/Package-com.ibm.aglets.html>,

[8] Oshima, Mitsuru, 1998, Aglet Spesification 1.1 Draft, <http://www.trl.ibm.co.jp/aglets/spec11.htm>

[9] Bursell , M., 2009, A Aglets Puppies Workshop, online pada www.ansa.co.uk/ANSATech/FollowMe/Puppies/apm/workshop/AGLETS.pdf.

[10] Bace, R., dan Mell, P., 2002, "Intrusion Detection System": NIST Special Publication On IDS, online pada http://www.snort.org/docs/nist-ids.pdf

[11] Balasubramaniyan, J.S., Fernandes, J.O.G., Isacoff, D., Spaffoer, E,. and Zamboni, D., 1998, "An Architecture For Intrusion Detection Using Autonomous Agents" ,online pada https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/98-05.pdf.

[12] Dune, C.R., 2000, "Using Mobile Agents For Network Resource Discovery In PeerToPeerNetworks",onlinepadahttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.98.4772. 27 Februari 2009

[13] Farmer, D., dan Venema, W., 2009, Improving The Security of Your Site by Breaking in to it,online pada http://www.porcupine.org/satan/admin-guide-to-cracking.html.

[14] Gopalakrishna, R., dan Spafford, E., 2000, "A Framework for distributed Intrusion Detection Using Interest Driven Cooperative Agents", online padahttp://www.raidsymposium.org/Raid2001/papers/gopalakrishna_spafford_raid2001.pdf.